WAGO-I/O-SYSTEM 750

Fielbus Independent I/O Modules

Stepper controller 750-672

Manual

Version 1.0.5

WAGO Kontakttechnik GmbH & Co. KG
Hansastraße 27
D-32423 Minden

Phone: +49 (0) 571/8 87 – 0
Fax: +49 (0) 571/8 87 – 1 69

E-Mail: info@wago.com

Web: http://www.wago.com

Technical Support

Phone: +49 (0) 571/8 87 – 5 55
Fax: +49 (0) 571/8 87 – 85 55

E-Mail: support@wago.com

Every conceivable measure has been taken to ensure the correctness and completeness of this documentation. However, as errors can never be fully excluded, we would appreciate any information or ideas at any time.

E-Mail: documentation@wago.com

We wish to point out that the software and hardware terms as well as the trademarks of companies used and/or mentioned in the present manual are generally trademark or patent protected.

Content

1 Important Notes

This section includes an overall summary of the most important safety requirements and notes that are mentioned in each individual section. To protect your health and prevent damage to devices as well, it is imperative to read and carefully follow the safety guidelines.

1.1 Legal Bases

1.1.1 Copyright

This Manual, including all figures and illustrations, is copyright-protected. Any further use of this Manual by third parties that violate pertinent copyright provisions is prohibited. Reproduction, translation, electronic and phototechnical filing/archiving (e.g., photocopying) as well as any amendments require the written consent of WAGO Kontakttechnik GmbH & Co. KG, Minden, Germany. Non-observance will involve the right to assert damage claims.

WAGO Kontakttechnik GmbH & Co. KG reserves the right to provide for any alterations or modifications that serve to increase the efficiency of technical progress. WAGO Kontakttechnik GmbH & Co. KG owns all rights arising from the granting of patents or from the legal protection of utility patents. Third-party products are always mentioned without any reference to patent rights. Thus, the existence of such rights cannot be excluded.

1.1.2 Personnel Qualifications

The use of the product described in this Manual requires special personnel qualifications, as shown in the following table:

Activity	Electrical specialist	Instructed personnel*)	Specialists**) having qualifications in PLC programming
Assembly	X	X	
Commissioning	X		X
Programming			X
Maintenance	X	X	
Troubleshooting	X		
Disassembly	X	X	

*) Instructed persons have been trained by qualified personnel or electrical specialists.

**) A specialist is a person, who – thanks to technical training – has the qualification, knowledge and expertise to meet the required specifications of this work and to identify any potential hazardous situation in the above listed fields of activity.

All responsible persons have to familiarize themselves with the underlying legal standards to be applied. WAGO Kontakttechnik GmbH & Co. KG does not assume any liability whatsoever resulting from improper handling and damage incurred to both WAGO´s own and third-party products by disregarding detailed information in this Manual.

1.1.3 Use of the 750 Series in Compliance with Underlying Provisions

Couplers, controllers and I/O modules found in the modular WAGO-I/O-SYSTEM 750 receive digital and analog signals from sensors and transmit them to the actuators or higher-level control systems. Using programmable controllers, the signals can also be (pre-)processed.

The components have been developed for use in an environment that meets the IP20 protection class criteria. Protection against finger injury and solid impurities up to 12.5 mm diameter is assured; protection against water damage is not ensured. Unless otherwise specified, operation of the components in wet and dusty environments is prohibited.

1.1.4 Technical Condition of Specified Devices

The components to be supplied Ex Works, are equipped with hardware and software configurations, which meet the individual application requirements. Changes in hardware, software and firmware are permitted exclusively within the framework of the various alternatives that are documented in the specific manuals. WAGO Kontakttechnik GmbH & Co. KG will be exempted from any liability in case of changes in hardware or software as well as to non-compliant usage of components.

Please send your request for modified and new hardware or software configurations directly to WAGO Kontakttechnik GmbH & Co. KG.

1.2 Standards and Guidelines for Operating the 750 Series

Please adhere to the standards and guidelines required for the use of your system:

- The data and power lines shall be connected and installed in compliance with the standards required to avoid failures on your system and to substantially minimize any imminently hazardous situations resulting in personal injury.
- For assembly, start-up, maintenance and troubleshooting, adhere to the specific accident prevention provisions which apply to your system (e.g. BGV A 3, "Electrical Installations and Equipment").
- Emergency stop functions and equipment shall not be made ineffective. See relevant standards (e.g. DIN EN 418).
- The equipment of your system shall be conform to EMC guidelines so that any electromagnetic interferences will be eliminated.
- Operating 750 Series components in home applications without further measures is permitted only if they meet the emission limits (emissions of interference) in compliance with EN 61000-6-3. You will find the detailed information in section "WAGO-I/O-SYSTEM 750" → "System Description" → "Technical Data".
- Please observe the safety precautions against electrostatic discharge in accordance with DIN EN 61340-5-1/-3. When handling the modules, please ensure that environmental factors (persons, working place and packaging) are well grounded.
- The valid standards and guidelines applicable for the installation of switch cabinets shall be adhered to.

1.3 Symbols

Danger
Always observe this information to protect persons from injury.

Warning
Always observe this information to prevent damage to the device.

Attention
Marginal conditions that must always be observed to ensure smooth and efficient operation.

ESD (Electrostatic Discharge)
Warning of damage to the components through electrostatic discharge. Observe the precautionary measure for handling components at risk of electrostatic discharge.

Note
Make important notes that are to be complied with so that a trouble-free and efficient device operation can be guaranteed.

Additional Information
References to additional literature, manuals, data sheets and internet pages.

1.4 Safety Information

When connecting the device to your installation and during operation, the following safety notes must be observed:

Danger
The WAGO-I/O-SYSTEM 750 and its components are an open system. It must only be assembled in housings, cabinets or in electrical operation rooms. Access is only permitted via a key or tool to authorized qualified personnel.

Danger
All power sources to the device must always be switched off before carrying out any installation, repair or maintenance work.

Warning
Replace defective or damaged device/module (e.g. in the event of deformed contacts), as the functionality of field bus station in question can no longer be ensured on a long-term basis.

Warning
The components are not resistant against materials having seeping and insulating properties. Belonging to this group of materials is: e.g. aerosols, silicones, triglycerides (found in some hand creams). If it cannot be ruled out that these materials appear in the component environment, then the components must be installed in an enclosure that is resistant against the above mentioned materials. Clean tools and materials are generally required to operate the device/module.

Warning
Soiled contacts must be cleaned using oil-free compressed air or with ethyl alcohol and leather cloths.

Warning
Do not use contact sprays, which could possibly impair the functioning of the contact area.

Warning
Avoid reverse polarity of data and power lines, as this may damage the devices.

ESD (Electrostatic Discharge)
The devices are equipped with electronic components that may be destroyed by electrostatic discharge when touched.

Warning
For components with ETHERNET/RJ-45 connectors:
Only for use in LAN, not for connection to telecommunication circuits.

1.5 Font Conventions

italic Names of paths and data files are marked in italic-type.
e.g.: *C:\Programs\WAGO-IO-CHECK*

italic Menu items are marked in italic-type, bold letters.
e.g.: ***Save***

\ A backslash between two names characterizes the selection of a menu point from a menu.
e.g.: ***File \ New***

END Pushbuttons are marked as bold with small capitals
e.g.: **ENTER**

< > Keys are marked bold within angle brackets
e.g.: **<F5>**

`Courier` The print font for program codes is Courier.
e.g.: `END_VAR`

1.6 Number Notation

Number code	Example	Note
Decimal	100	Normal notation
Hexadecimal	0x64	C notation
Binary	'100' '0110.0100'	In quotation marks, nibble separated with dots (.)

1.7 Scope

This manual describes the Special Module 750-672 Stepper controller of the modular WAGO-I/O-SYSTEM 750.

Handling, assembly and start-up are described in the manual of the Fieldbus Coupler. Therefore this documentation is valid only in the connection with the appropriate manual.

2 I/O Modules

2.1 Special Modules

2.1.1 General Description

2.1.1.1 Safety Information

Observe the following information and notices to prevent injury and damage to persons and equipment.

⚠ **Danger**
Take appropriate measures, such as cordoning off appropriate areas with screens/enclosures, to prevent bodily contact with the system's moving parts.

⚠ **Danger**
Enact and install an EMERGENCY OFF procedure and system that adheres to locally valid regulations and applicable engineering practices.

⚠ **Note**
Install appropriate hardware limit switches that can directly disengage power to the system if a restricted area of movement has been breached.

⚠ **Note**
Install appropriate equipment to protect motors and power electronics, such as motor circuit breakers or fuses.

2.1.1.2 Structure of Positioning Controller

The following figure illustrates the structure of a typical positioning controller, along with the basic elements:

- Control section,
- Power section,
- Drive,
- Mechanical section.

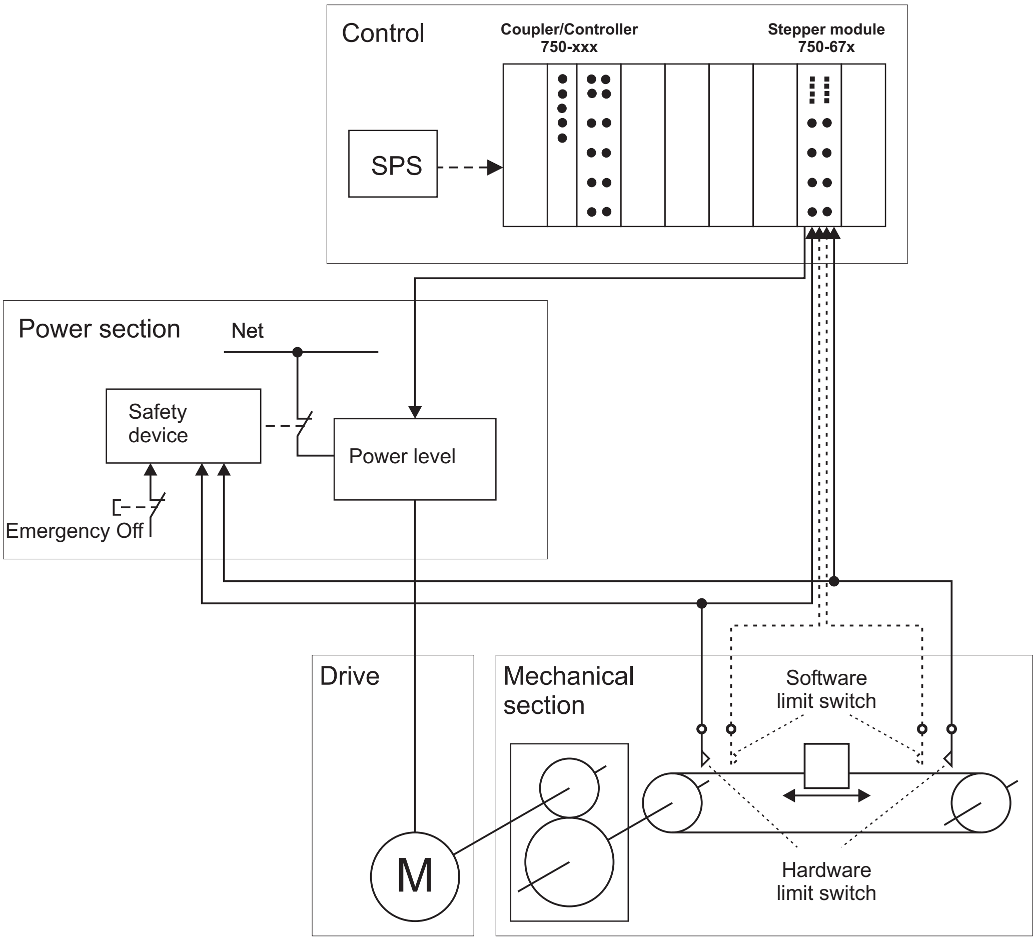


Fig. 2.1.1-1: Structure of position control system g067x00e

2.1.1.2.1 Control Section

The control section consists of the PLC for process control and the stepper module 750-67x for positioning, FM and PWM functions.

2.1.1.2.2 Power Stage

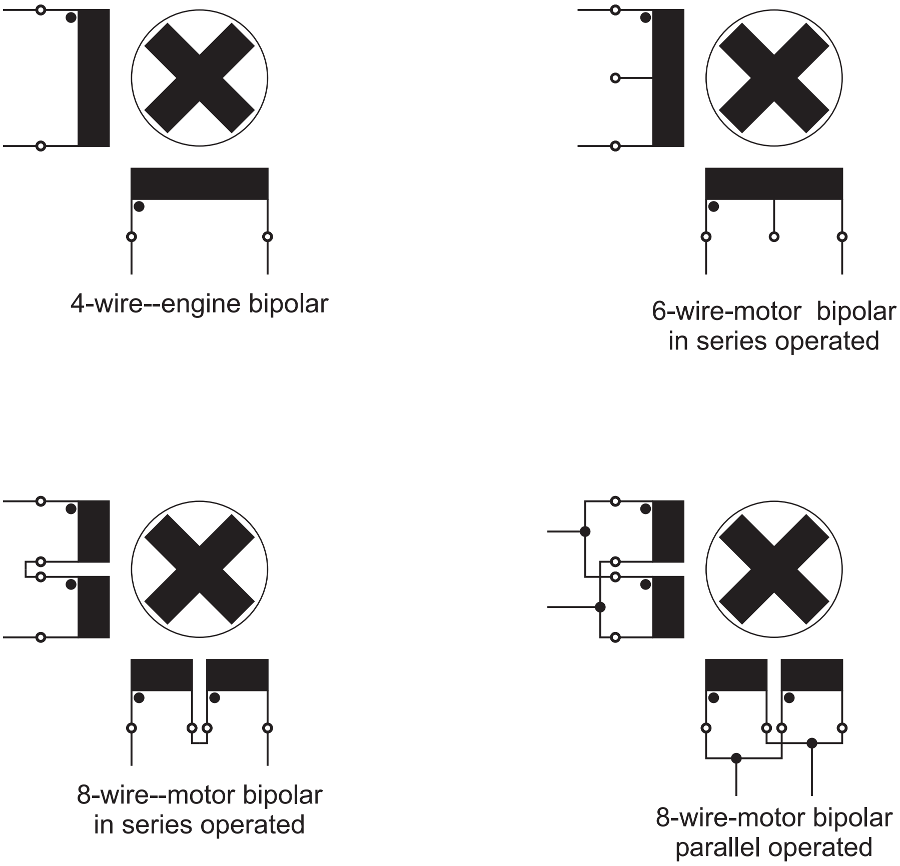
The power stage generates the requisite drive currents from the pulses for the specific motor. Any type of output stage equipped with a pulse direction or incremental encoder interface can be used with the stepper module 750-670. This also allows output stages for 3- or 5-pole stepper motors or DC or AC servo motors to be used. Stepper modules 750-671, -672 and -673 are equipped with an integrated output stage for regulating 2-phase stepper motors.

2.1.1.2.3 Drive system

Stepper motors are simple and economical drives for high-precision tasks for the most varying of applications.

The shaft of a stepper motor rotates by a defined angle at each pulse; a rapid succession of pulses transforms this stepping motion into a continuous turning motion. The natural resonance of stepper motors is greatly suppressed in particular by extremely smooth running produced through microstepping at high resolution, as used in the WAGO modules 750-671, -672 and -673, with 64-fold microstepping.

The figures below illustrate possible types of connections for stepper motors.



4-wire--engine bipolar

6-wire-motor bipolar
in series operated

8-wire--motor bipolar
in series operated

8-wire-motor bipolar
parallel operated

Fig. 2.1.1-2: Types of connections for stepper motors

g067x02e

In addition to the type of connection and the number of phases, the requisite torque progression over speed, along with the motor current necessary for this and the winding resistance and the motor inductance must also be taken into account when selecting an appropriate motor.

The torque progression and speed are dictated by the application to be implemented. Practical experience has shown that a torque margin of approximately 25%, depending on the mechanical system properties, has proven useful. This should be taken into consideration to account for any dynamic effects (resonance in mechanical systems).

The positioning process sequences also determines the average and peak power supplied to the motor; special attention must be given here to the total power loss and temperature of the motor.

Depending on the motor model and design, a corresponding current must be present that must be transferred from the output stage into the motor. The voltage required for this depends on the winding resistance, motor inductance and speed (anti-EMC). It may therefore be necessary to have considerably higher voltage levels for the specific current level, in particular at high speeds, than that specified by the motor data. The motor data provided by the manufacturers is based on motor standstill (ohmic winding resistance). As the power output stages of the stepper modules 750-671, -672 and -673 are equipped with power control systems, it is no problem, for example, to run 12 V motors with 24 V supply systems, as long as the current, power loss and temperature of the motor remain within acceptable limits. Consult the motor manufacturer if you have any doubts or questions.

2.1.1.2.4 Mechanical Section

The requisite motor data can be calculated based on the requirements concerning the load to be moved, and any additional bearings, transmissions, deflection systems, damping elements, etc. that may be required. Important parameters to consider are: moment of inertia, starting torque holding torque torque at the maximum required speed cycle times for positioning requisite acceleration required torque (where applicable) when passing through mechanical resonance fields — particularly when mechanical components such as long drive belts, spring elements or vibration buffers (couplings) are used. There must be no step losses if the requisite mechanical torque does not exceed the torque supplied by the motor (taking into account its own inertia)!

2.1.1.3 Positioning

A distinction is made here between absolute and relative positioning. In addition, a difference is also made between a reference run and the Jog mode.

2.1.1.3.1 Absolute Positioning

Positioning from the absolute position X to absolute position Y.

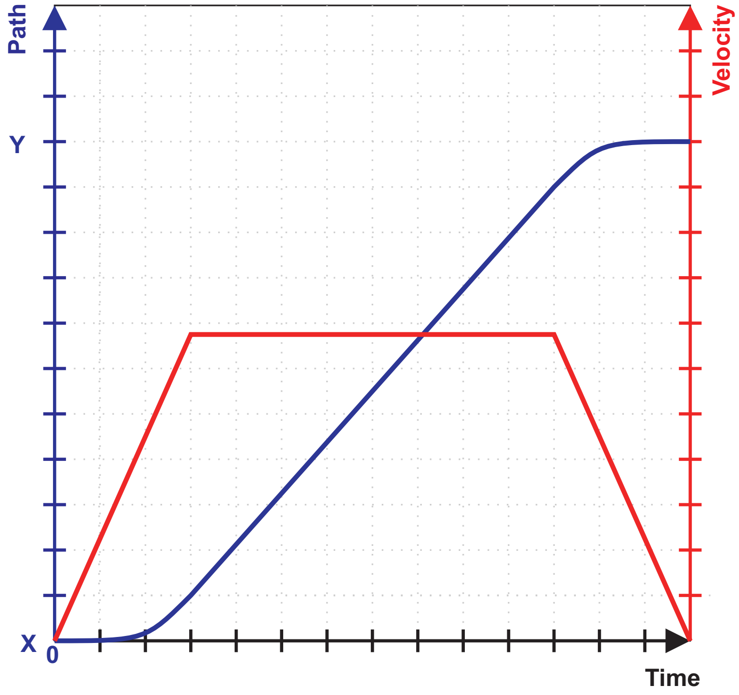


Fig. 2.1.1-3: Absolute positioning g067x03e

Potential applications:

- Positioning shafts
- Transfer carriages
- Pick & Place

2.1.1.3.2 Relative Positioning

Positioning from absolute position X to absolute position Y by the difference x. Also possible as a command during positioning (on the fly).

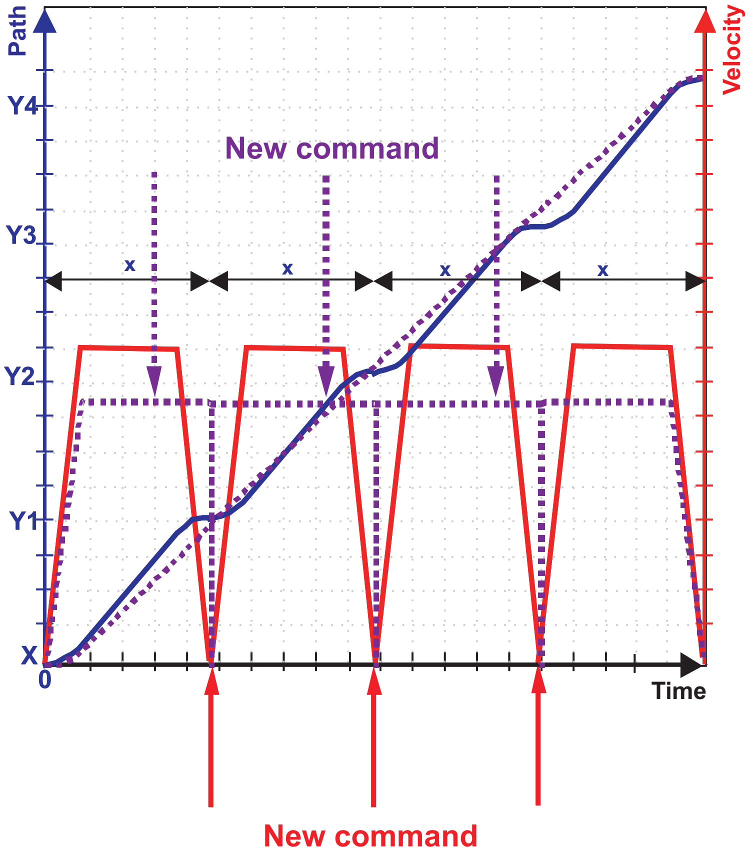


Fig. 2.1.1-4: Relative positioning

g067x02e

Potential applications:

- Incremental dimensions
- Variable reference points

2.1.1.3.3 On-the-Fly Positioning

Termination of ongoing positioning (such as Move to Y) and execution of the new positioning command (Move to Y-n).

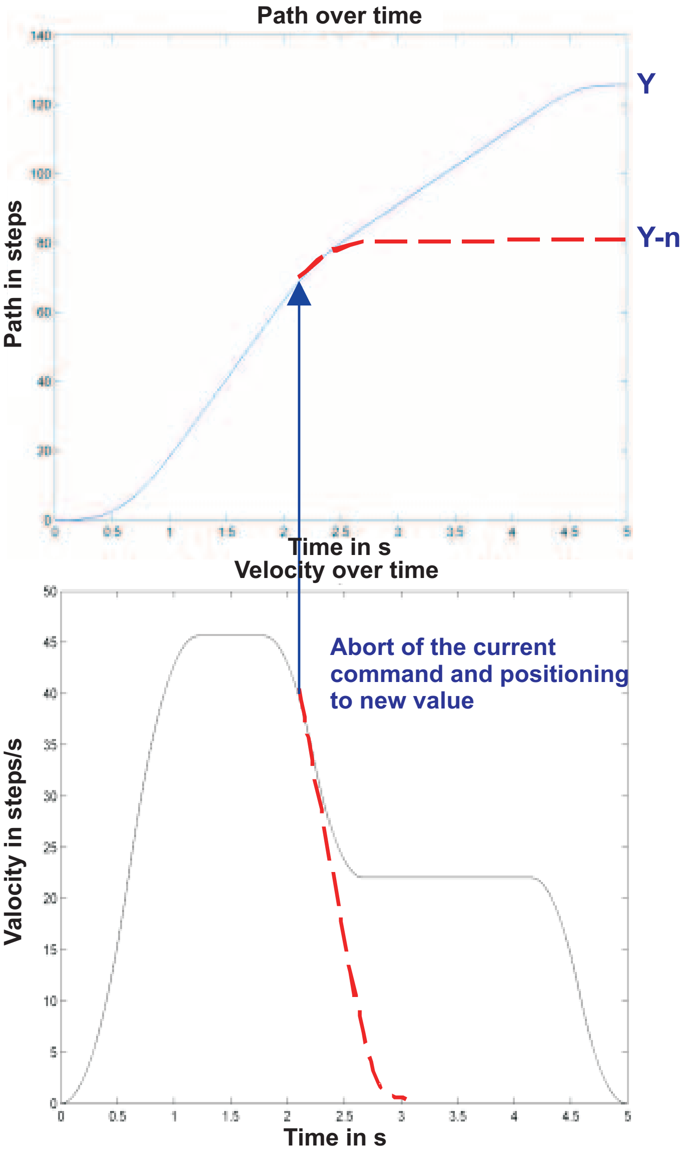


Fig. 2.1.1-5: On-the-fly positioning

g067x05e

Potential applications:

- Event-dependent changing of target position
- Avoidance of collisions
- Process optimization

2.1.1.3.4 Referencing

Referencing is the setup of a measuring system. A distinction is drawn here between referencing to a limit switch and referencing to a special reference switch. A high degree of reproducible accuracy is essential for referencing. Referencing should therefore always be performed from the same end.

Referencing involves searching for the reference switch at the set setup speed and then moving toward that point from the correct end from any position with the movement range.

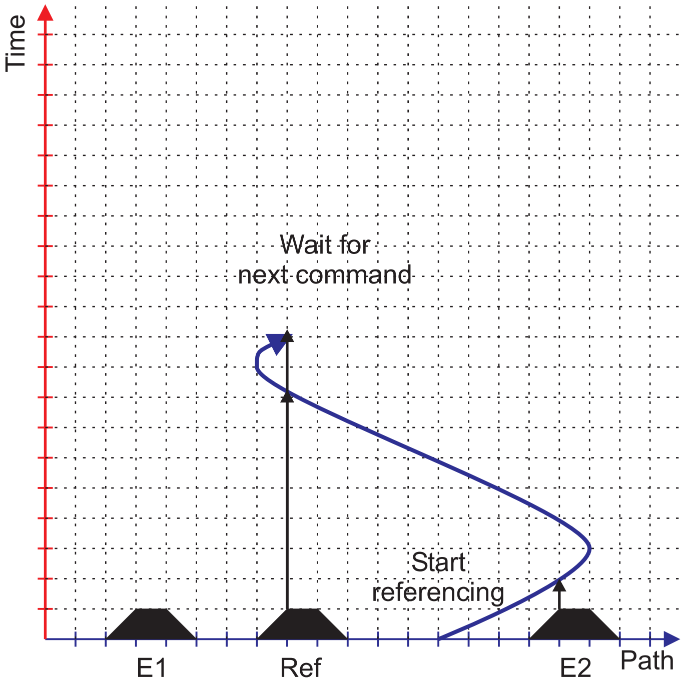


Fig. 2.1.1-6: Referencing

g067x09e

The reference value (usually 0) is accepted at the corresponding edge.

2.1.1.3.5 Jogging Mode

The drive is run at the setup speed via a defined input, or a control bit, as long as the input is active, or as long as the bit is set. A time limit can be activated for the moving process.

2.1.1.3.6 Rotary Axis

The value range (such as -10000 ... +10000) is converted for rotation (360°) around a real or virtual axis for a rotary axis. Overrun is taken into account automatically here, i.e. when 360° is exceeded, counting is restarted at 0°. Based on the example values, after exceeding the +10000 position, the next position would be -10000.

When specifying the parameters for the rotary axis, the following restriction must be observed for the function to be implemented correctly.

$$v \leq \frac{Speed\_Div}{Speed\_Mult} * \frac{Freq\_Div}{80} * 10^3 * p$$

v	Maximum allowed setpoint speed
p	Rotary axis periods (parameter 64, Rotary_Axis_Period)
Speed_Mult	Scaling factor for setpoint speed (parameter 28 from configuration table)
Speed_Div	Scaling factor for setpoint speed (parameter 30 from configuration table)
Freq_Div	Prescaler for maximum speed (parameter 4 from configuration table)

Potential applications:

- Belt control
- Label supply
- Control of rotary tables

2.1.1.3.7 Types of Acceleration

2.1.1.3.7.1 Constant Acceleration

Acceleration has the same value during the entire acceleration phase. Both the onset and completion of acceleration phase jolt the mechanical system; this phenomenon is comparable to the jolt a vehicle experiences when stepping on/stepping off the accelerator.

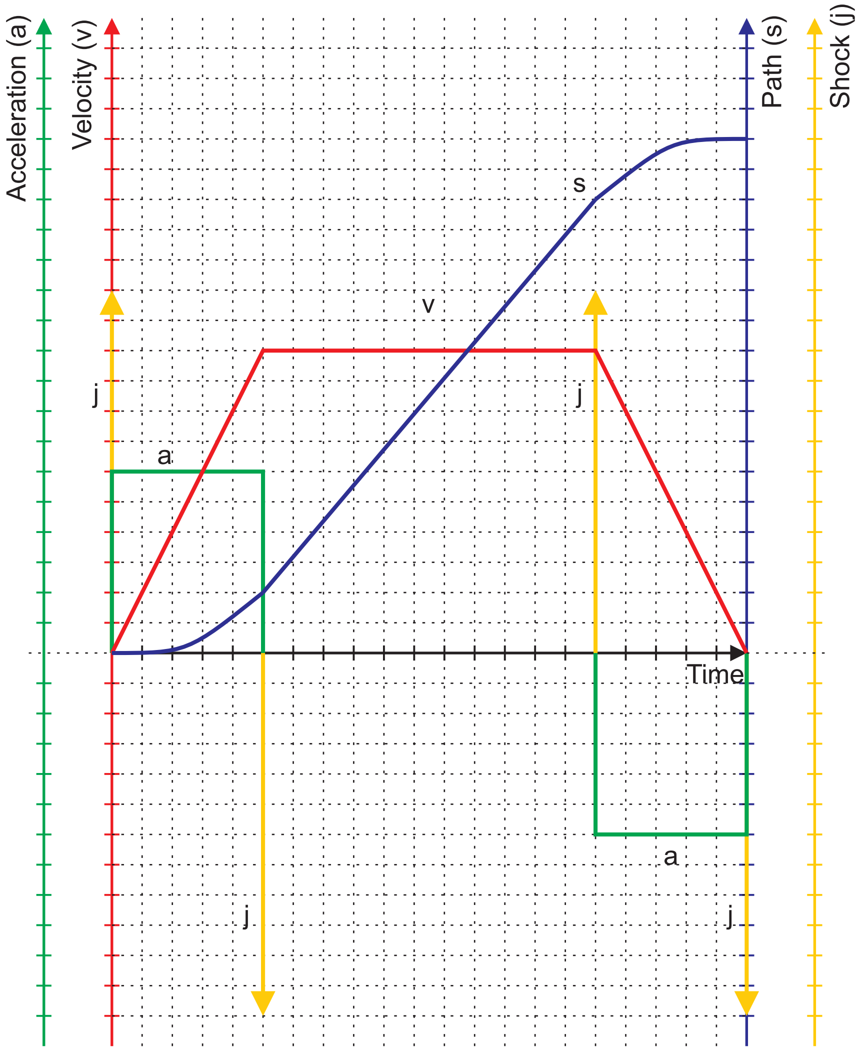


Fig. 2.1.1-7: Constant acceleration

g067x06e

Potential applications:

- Peak acceleration at specified acceleration value,
- Linear path/time response.

2.1.1.3.7.2 Linear Acceleration

Acceleration increases and decreases during the acceleration phase with a linear gradient.
This reduces the jolt experienced by the mechanical system.

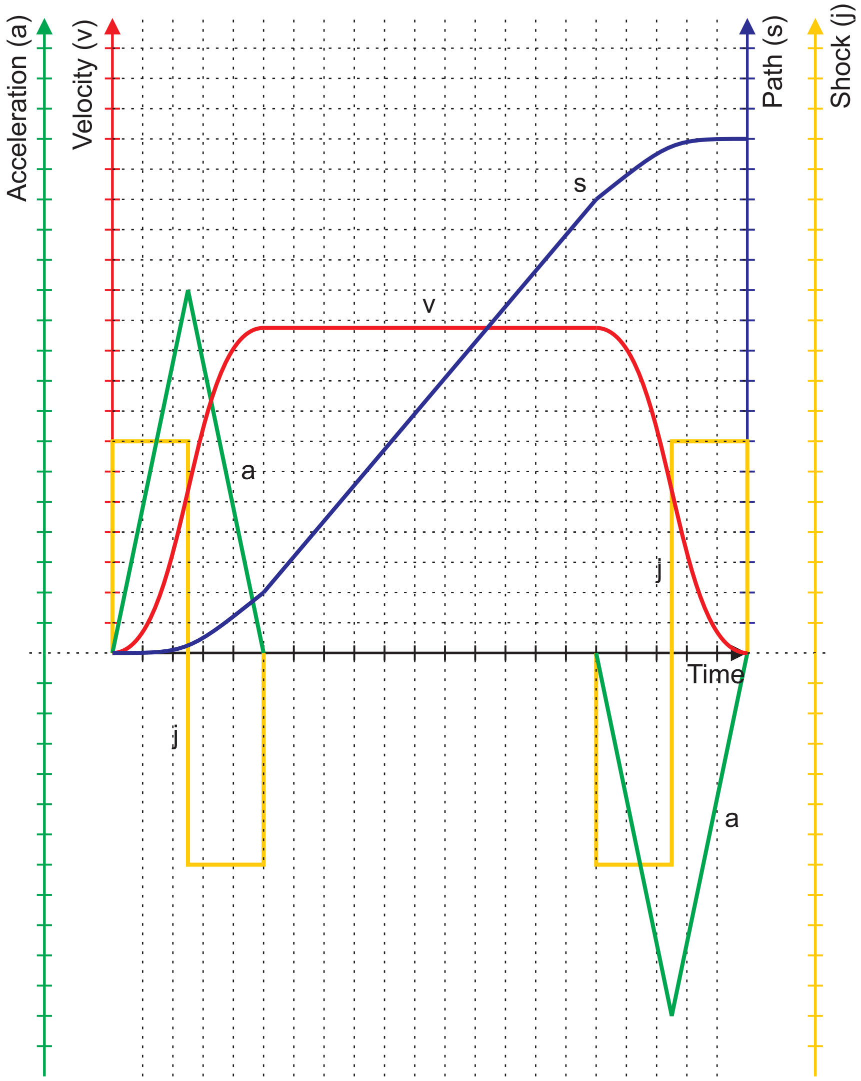


Fig. 2.1.1-8: Linear acceleration

g067x07e

Potential applications:

- Soft start (jolt reduction)
- Less risk of step losses
- Linear (constant) acceleration moment
- Maximum acceleration, in particular with flexible drive systems (belts)

2.1.1.3.7.3 sin²*t Acceleration

The acceleration value progresses in accordance with a sin²*t curve during the acceleration phases.
This minimizes the jolt experienced by the mechanical system. This reduces any remaining harmonic waves still present during linear acceleration.

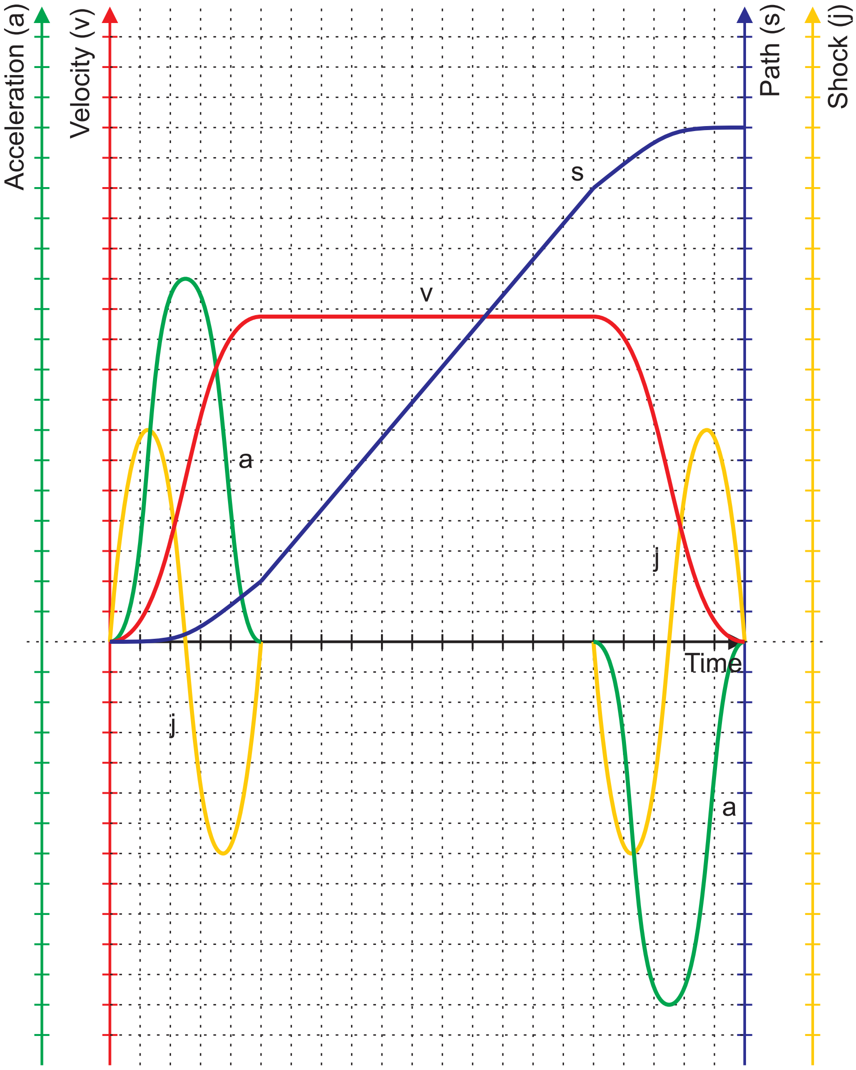


Fig. 2.1.1-9: sin²*t acceleration

g067x08e

Potential applications:

- Soft start (jolt reduction)
- Less risk of step losses
- Maximum acceleration, in particular with flexible drive systems (belts)

2.1.1.3.7.4 Adjustable acceleration

The acceleration and brake ramps can be adjusted individually.

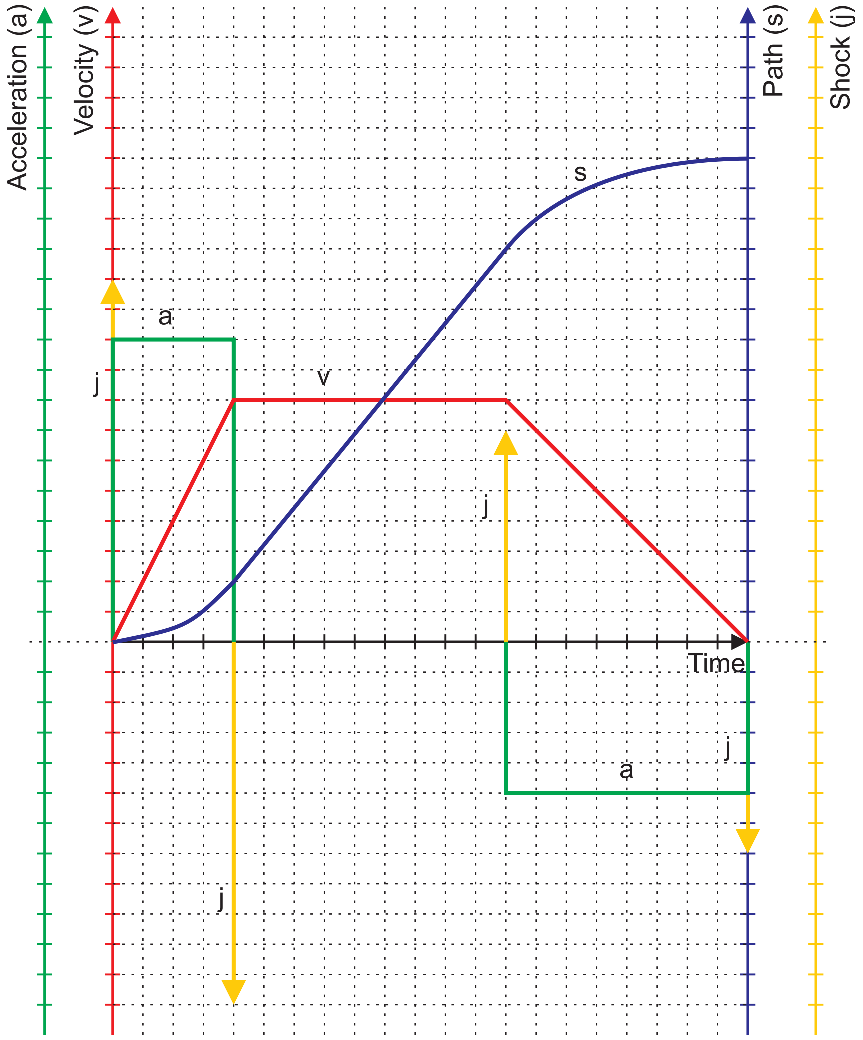


Fig. 2.1.1-10: Adjustable acceleration g067x28e

Possible application, e.g.:

- Defining if the acceleration must be done with or against additional external torques
- Asymmetric retaining forces of toolings (grippers)

The acceleration can be defined as

- Acceleration time
- Acceleration path or
- Acceleration (steps/s^2)

Potential applications:

- Cycle-time dependent applications
- Simple path calculation
- Definition of acceleration torques

2.1.1.4 Current Control

The current is dependent on:

- Acceleration
- Constant speed
- Delay
- Stop (holding torque)

The current can be set as a % of the nominal value.

Values up to 150 % are possible (boost)!

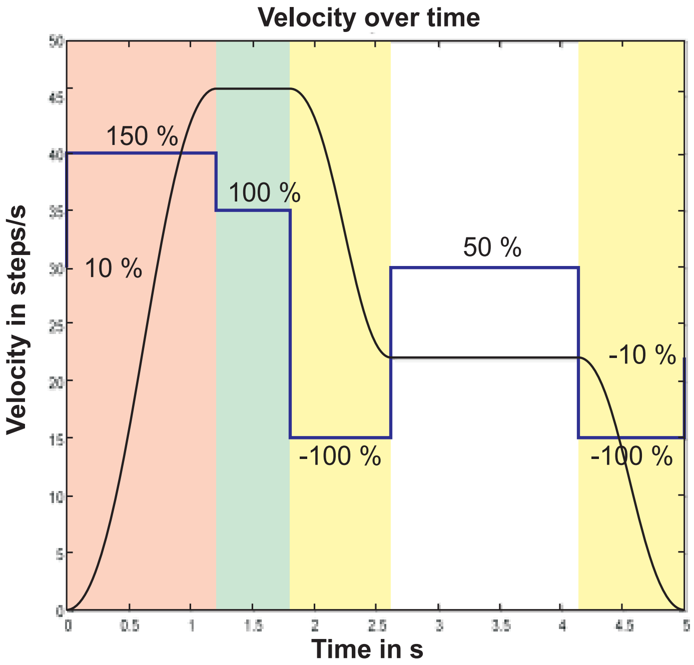


Fig. 2.1.1-11: Current control

g067x31e

Potential applications:

- Power loss limitation
- Torque control

2.1.1.5 Speed Control

The rotational speed is regulated by speed control. Achieving a specified position is not relevant here.

Potential applications:

- Simple interfaces for ready-made application programs
- Belt drives, conveyor systems

2.1.1.6 Camshaft Controller

The camshaft controller allows to set an output or bits in a position window. The position window can be defined absolutely or relatively:

- Set output/bit from Xn to Yn
- Set output/bit from Xn on for ΔYn

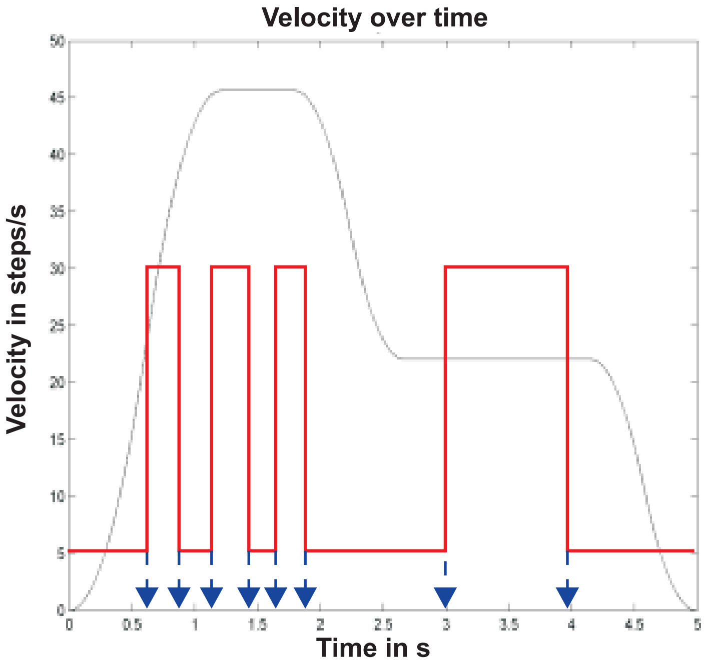


Fig. 2.1.1-12: Camshaft controller g067x32e

Potential applications:

- Setting of glue dots
- Length feeding
- Stamp positions
- Tool operation

2.1.1.7 Brake Control

Brake OFF (Output=1) ΔtOff before start of positioning.

Brake ON (Output=0) ΔtOn before reaching target position.

If the brake was switched on, the execution of the next positioning is delayed by ΔtOff.

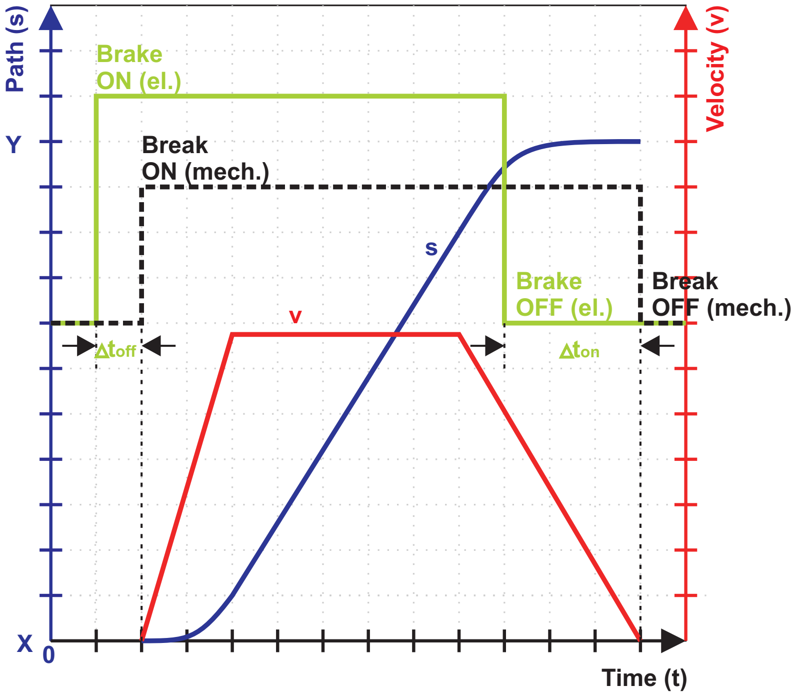


Fig. 2.1.1-13: Brake control

g067x33e

Potential applications:

- Lifting axis
- Parking brakes

2.1.1.8 Command Tables

In the command tables, a complex positioning sequence can be stored and executed independently according to the appropriate sequential list of individual commands.

The command sequence can be changed or stopped depending on external or internal (PLC) events.

The operating mode possibilities are:

- Cyclic (repeats after end of list)
- Event controlled (digital, analog, time)
- Direct addressing
- Skipping to other entries in list

Two tables are available that can alternately be switched over: An offline table (program run) and an online table (program up-/download).

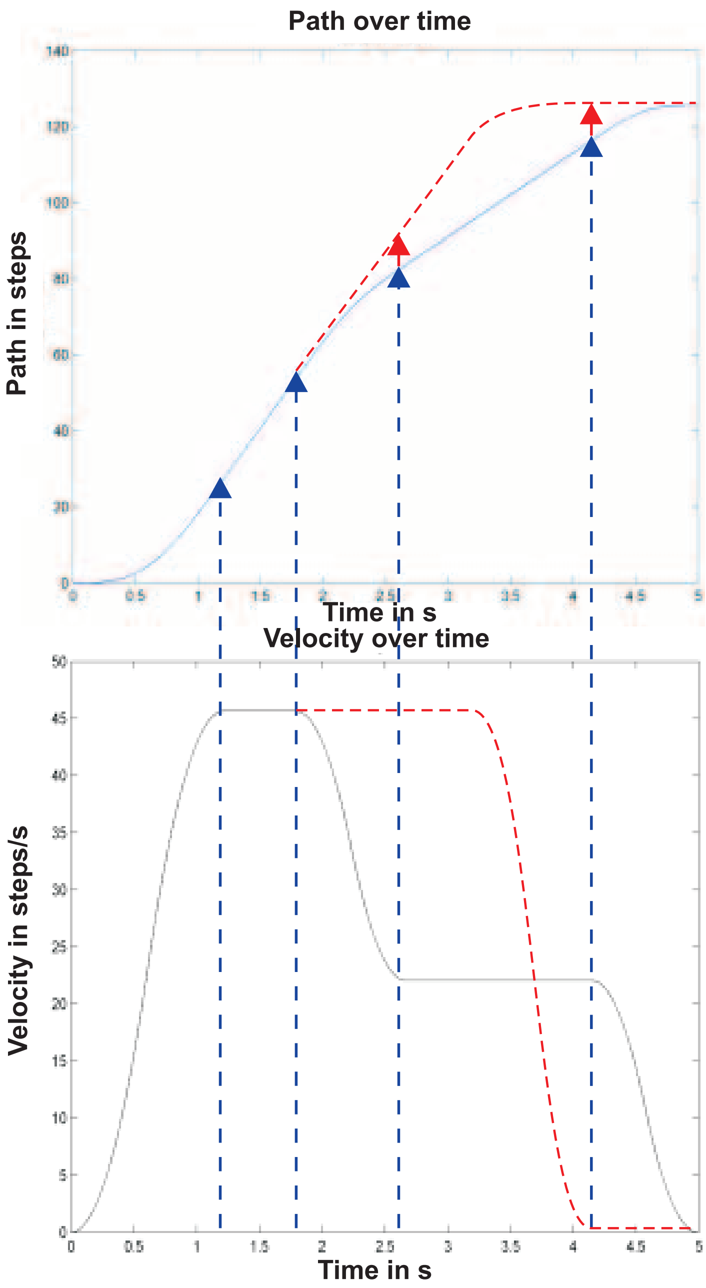


Fig. 2.1.1-14: Command tables g067x29e

Potential applications:

- Relieving the PLC
- Reduction of response times
- Encapsulating the application

3 750-672 [Stepper controller]

The intelligent 750-672 stepper controller has an on-board power driver designed to control 2-phase stepper motors up to 70 V/7.5 A.
The 64 times microstepping prevents step losses due to resonance in the acceleration phases and prevents excessive wear on mechanical parts.
Adjustable current limits for stop, acceleration and constant speed help minimize motor power dissipation.

Six configurable inputs for start/stop, end-stop, reference, jog/tip, etc., can be directly processed by the internal software without delay.

Two outputs can be linked with internal functions or used freely.

Versatile functions, such as positioning with different acceleration slopes, command tables, camshaft controller, auto referencing and other event-dependent properties provide this controller with a wide spectrum of possible uses

The programmer's interface is the same for all WAGO stepper controller modules.

Various applications are implemented in the 750-672 stepper controller.

- Positioning,
- Speed Control

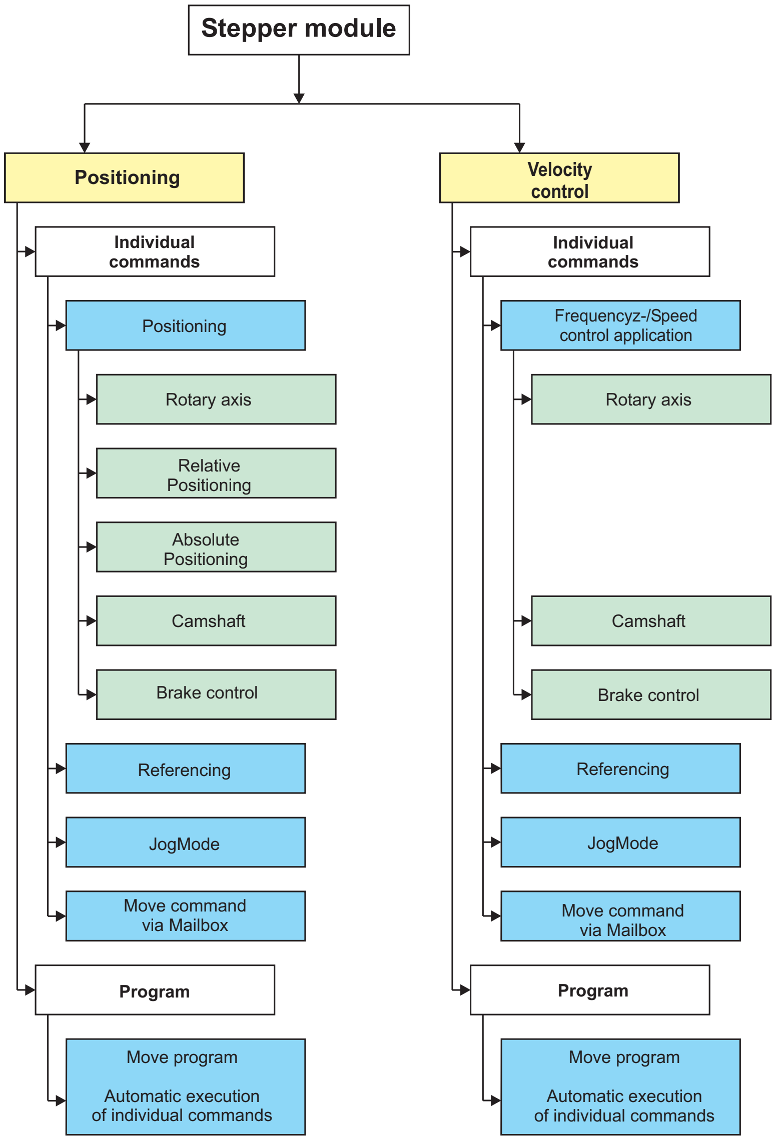


Fig. 2.1.1-1: Stepper controller applications and operating modes g067120e

There are five operating modes available in each of the applications Positioning and Velocity Control:

- Positioning/speed parameter
- Referencing,
- Jog Mode,
- Move command via Mailbox,
- Run program.

The stepper controller function is defined by various tables, with the configuration table and the Bit I/O table playing a particularly important role.

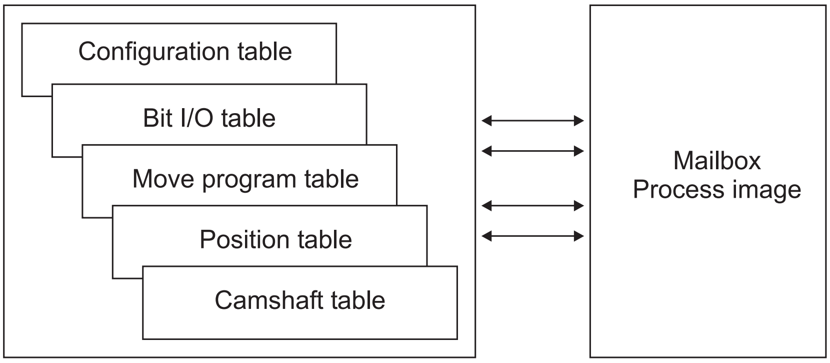


Fig. 2.1.1-2: Tables in the stepper controller

The stepper module has six digital 24 V inputs: DI 1+ and DI 6+.
The inputs are electrically isolated, and each has the reference potential DI 1- through DI 6-.
This allows sensors and switches to be connected in two transmission systems.

In the standard application, the inputs have the following meaning:

DI 1	Enable input
DI 2	Reference input
DI 3	Jogging in positive direction
DI 4	Jogging in negative direction
DI 5	Limit switch in positive direction has been actuated
DI 6	Limit switch in negative direction has been actuated

For particular applications, other function can be assigned to these inputs.

The positive switching outputs DO 1 and DO 2 are short-circuit proof and can be loaded with up to 0.5A.

In the standard application, the outputs have the following meaning:

DO 1	Target position has been reached
DO 2	An error has occurred

For particular applications, other function can be assigned to these outputs.

Corresponding entries in the configuration table refer to the bit table, which contains all internal function-relevant bits.

In addition, the stepper module has connections for motor windings.
The windings are connected to the contacts M_1A–M_1B and M_2A-M_2B.

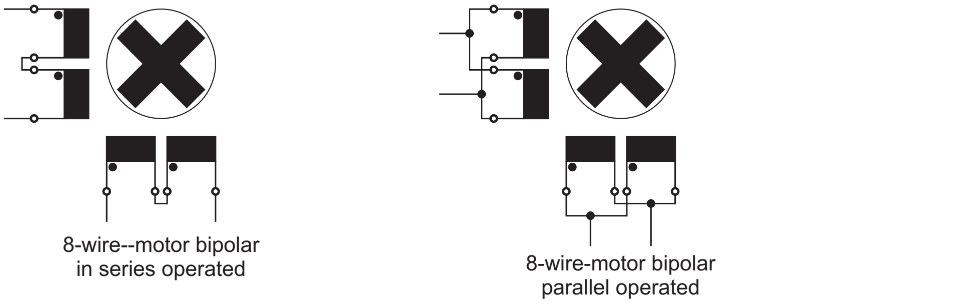


Fig. 2.1.1-3: Stepper controller connections g067205e

Connecting in series is required if the nominal current of a part winding is smaller than the medium operating current.
Otherwise, the motor windings should be switched parallel.

Faulty operating states, such as a short circuit in the motor windings or overheating in the output stage, are detected and signaled.

WAGO shielding clamping brackets must be used to connect the shielding of the motor circuit to the rail.

LEDs signal both the signal state of the digital inputs and outputs and the state of the voltage supply.
Four additional LEDs display the module's operating mode.

Field and system levels are electrically isolated.

Individual I/O modules can be arranged in any combination when configuring the fieldbus node. An arrangement in groups is not necessary.

A 24 V voltage supply is required to supply power to the internal logic. This voltage supply shares the same ground as the UDC motor voltage.

The UDC motor voltage is not protected against reverse polarity.
WAGO Series 281 and 282 fuse terminals are available for fuse protection.

⚠ **Note**
Only operate the module with 24 V DC PELV (Protective Extra Low Voltage) or SELV (Safety Extra Low Voltage) power sources. Using a higher voltage (motor voltages up to 70 V) puts you at risk of electric shock.

The stepper controller can be operated at the following WAGO I/O SYSTEM 750 couplers and controllers:

Bus system	Coupler/Controller	Item No.	Hard-ware vers.	Soft-ware vers.	Max. number of modules
ETHERNET TCP/IP	Fieldbus coupler	750-341	03	06	8
		750-342	04	17	3
	Programmable fieldbus controller	750-841	03	17	16
		750-842	04	12	8
		750-843	12	01	8
		750-871	03	05	16
		750-873	02	02	16
CANopen	Fieldbus coupler	750-337	09	10	8
		750-338	02	16	8
	ECO Fieldbus coupler	750-347	01	04	1
		750-347	01	06	2
		750-348	01	04	1
		750-348	01	06	2
	Programmable Fieldbus controller	750-837	07	12	8
		750-838	02	12	8
DeviceNet	Fieldbus coupler	750-306	12	4J	8
	ECO Fieldbus coupler	750-346	02	07	2
	Programmable fieldbus controller	750-806	04	09	8
LON	Fieldbus coupler	750-319	xx	05	3
	Programmable fieldbus controller	750-819	xx	09	8
PROFIBUS	Fieldbus coupler	750-303	xx	from 08	3
		750-333	12	from 07	8
	ECO Fieldbus coupler	750-343	03	from 06	2
	Programmable fieldbus controller	750-833	16	10	8
Powerlink	Fieldbus coupler	750-350	07	01	8
BACnet	Programmable fieldbus controller	750-830	01	01	8
KNX	Programmable fieldbus controller	750-849	xx	04	16
SERCOS III	Feldbuskoppler	750-351	02	03	08

Bus system	Coupler/Controller	Item No.	Hard-ware vers.	Soft-ware vers.	Max. number of modules
WAGO-IPC	IPC	758-870/ 000-xxx	10	03	16
	IPC	758-874/ 000-xxx	10	03	16
	IPC	758-875/ 000-xxx	10	03	16
	IPC	758-876/ 000-xxx	10	03	16

Other couplers/controllers upon request

Note

The following must be observed when using the stepper module with CANopen bus couplers 750-337, 750-338, 750-837, 750-838, 750-347 and 750-348:

In the default configuration the stepper module data are mapped in consecutive PDOs. Each PDO can take up eight bytes of data. The 12-byte process image for the stepper module contains 2 PDOs, one with 8 and one with 4 bytes.

Problem:

The specified and actual values for the positioning data is distributed among 2 PDOs during positioning using the cycling process image, which could result in the data not being transferred consistently.

Remedy:

- For positioning via Mailbox mode, the mailbox data is consistently transferred in PDO1 and the control bits in PDO2.
- Use of 16-bit specified/actual values

  or
- Omission of "on-the-fly" specified/actual values; i.e., initiation of the function only after setting of the specified values or reading out of the 24-bit actual values has been fully completed and only in the "Standstill" status.

3.1 View

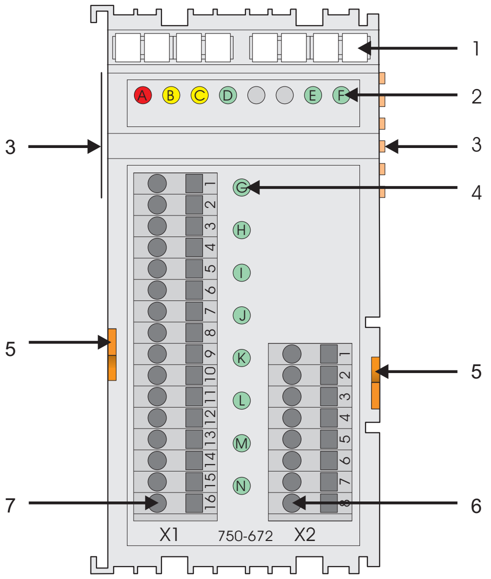


Fig. 2.1.1-1: View

g067200d

No.	Designation	Meaning	For details see chapter
1	---	Marking options using the Mini-WSB	---

2	A ... F	LED Status
		A: Error; B: Operation;
		C: Run program active; D: Enable;
		E: Control voltage; F: Motor voltage
			3.3 „Indicators“

3	---	Data contacts	

4	G ... N	LED Status
		G: Output 1; H: Output 2; I: Input 1;
		J: Input 2; K: Input 3; L: Input 4;
		M: Input 5; N: Input 6
			3.3 „Indicators“

5	---	Releasing strap	

6	1 ... 8	CAGE CLAMP® X2 Connections
		1: Motor winding M_1A,
		2: Motor winding M_1B;
		3: Motor winding M_2A;
		4: Motor winding M_2B;
		5: Motor voltage UDC;
		6: Motor voltage 0 V;
		7: Control voltage 0 V;
		8: Control voltage +24 V
			3.2 „Connecting Elements“

7	1 ...16	CAGE CLAMP® X1 connections 1 ... 16: Output DO 1+ ... input DI 6-	3.2 „Connecting Elements“

3.2 Connecting Elements

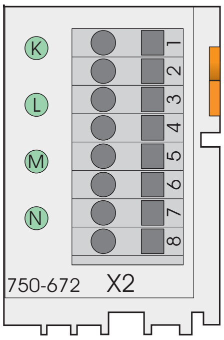


Fig. 2.1.1-1:
Connecting elements
g067203x

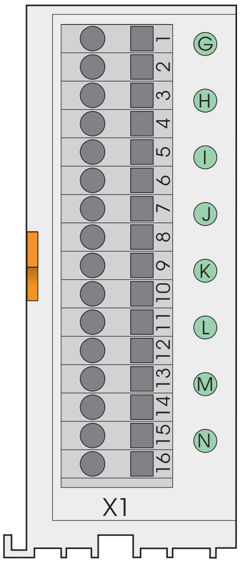


Fig. 2.1.1-2:
Connecting elements
g067204x

Connec-tor	Desig-nation	Name	Standard configuration *)
X2 1	M_1A	Motor winding 1	
X2 2	M_1B	Motor winding 1	
X2 3	M_2A	Motor winding 2	
X2 4	M_2B	Motor winding 2	
X2 5	UDC	Motor supply	Nominal voltage 55 V DC
X2 6	0 V	Motor supply	Ground
X2 7	0 V	Control voltage	Ground
X2 8	24 V	Control voltage	Nominal voltage 24 V DC
X1 1	DO 1+	Digital output 1	Target position has been reached *)
X1 2	DO 0 V	Digital output	
X1 3	DO 2+	Digital output 2	An error has occurred *)
X1 4	DO 0 V	Digital output	
X1 5	DI 1+	Digital input 1+	Enable input Stop1_N *)
X1 7	DI 2+	Digital input 2+	Reference input Set_Reference *)
X1 9	DI 3+	Digital input 3+	Jogging in positive direction Jog_Pos *)
X1 11	DI 4+	Digital input 4+	Jogging in negative direction Jog_Neg *)
X1 13	DI 5+	Digital input 5+	Limit switch in positive direction has been actuated LimitSwitch_Pos *)
X1 15	DI 6+	Digital input 6+	Limit switch in negative direction has been actuated LimitSwitch_Neg *)
X1 6, 8 ... 16	DI 1- ... DI 6-	Digital input 1- ... 6-	0 V for DI 1+ ... DI 6+

*) The given configuration applies only to standard applications. Adaptation for other applications is described in the corresponding sections.

3.3 Indicators

LED	Link	Desig-nation	Status	Function
A	Error	Write access to EEPROM/ Error code Group error	Flashing 10 Hz/ red	Write access to EEPROM
			Blink code/ Red	Group error/warning Error message or warning issued
B	Busy	Busy		The selected operating mode is active and not yet finished. It is possible that the operating mode was cancelled.
		Positioning	Off	Positioning not active, drive motionless
			Yellow	Positioning active, drive in operation
		Run program	Off	Run program not active.
			Yellow	Run program active
		Referencing	Off	Referencing not active, drive motionless
			Yellow	Referencing active, drive in operation
		JogMode	Off	JogMode not active.
			Yellow	JogMode active; motor has been started using Input3 (Jog_Pos) or Input4 (Jog_Neg). LED flashes briefly
		Mailbox mode	Off	Mailbox active, but no command active, drive motionless
			Yellow	Mailbox and command active, drive in operation
		Rotational speed control	Off	The "Busy" LED has no function in this operating mode.
			Yellow	The "Busy" LED has no function in this operating mode
C	M_ Program _ACK	Run program	Off	No Run program being processed
			Yellow	A run program is in process.
D	Stop_N _ACK	Drive stop inverted	Off	The bit Stop1_N or Stop2_N is 0. In addition, the motor is at standstill and frequency output is 0. Startup using Start is not possible.
			Green	The bits Stop1_N and Stop2_N are both set to 1, or the drive is braking the unit.
E		24 V	Off	Field supply logic missing.
			Green	Field supply logic present.
F		Motor supply	Off	Motor supply missing.
			Green	Motor supply present.

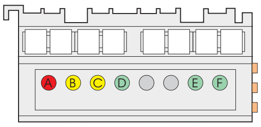


Fig. 2.1.1-1: Indicators

g067202x

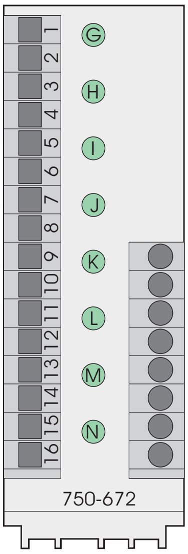


Fig. 2.1.1-2: Indicators
g067205x

LED	Link	Designatio n	Status	Function
G	On_ Target	Status DO 1	Off	Target position has been reached.
			Green	Target position has not been reached.
H	Error	Status DO 2	Off	No error present.
			Green	An error has occurred.
I	Fixed	Status DI 1	Off	Input DI 1: Signal level (0)
			Green	Input DI 1: Signal level (1)
J	Fixed	Status DI 2	Off	Input DI 2: Signal level (0)
			Green	Input DI 2: Signal level (1)
K	Fixed	Status DI 3	Off	Input DI 3: Signal level (0)
			Green	Input DI 3: Signal level (1)
L	Fixed	Status DI 4	Off	Input DI 4: Signal level (0)
			Green	Input DI 4: Signal level (1)
M	Fixed	Status DI 5	Off	Input DI 5: Signal level (0)
			Green	Input DI 5: Signal level (1)
N	Fixed	Status DI 6	Off	Input DI 6: Signal level (0)
			Green	Input DI 6: Signal level (1)

3.4 Panel

The 750-672 stepper controller is not equipped with any operating elements. The configuration and the parameters can be changed via higher-level control or the WAGO-I/O-CHECK configuration tool.

3.5 Schematic Diagram

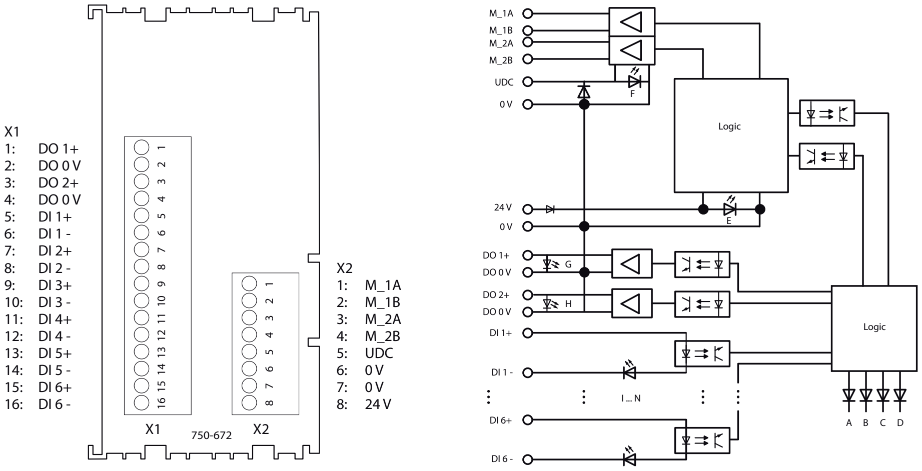


Fig. 2.1.1-1: Schematic circuit diagram

g067201e

3.6 Technical Data

inputs	
Number of inputs	6 (DI 1+ ... DI 6+)
Input voltage	DC -3 V ... 30 V
Signal voltage (0)	DC -3 V ... +5 V
Signal voltage (1)	DC 15 V ... 30 V
Input current typ.	2.8 mA
Input filter	100 µs * (* software filter can be installed)
Input characteristic	Acc. to EN61131-2 Model 1
Function Inputs	DI 1 Enable input DI 2 Reference input DI 3 Jog input, positive direction DI 4 Jog input, negative direction DI 5 Limit switch, positive direction DI 6 Limit switch, negative direction The inputs can be freely reconfigured.
Galvanic isolation	775 V DC among each other and against I/O module 1570 V DC against control voltage and motor voltage.
Outputs	
No. of outputs	2 (DO 1, DO 2)
Output rated current	0.5 A, short-circuit protected
Load types	- Resistive load - Inductive load, max. 2H - Lamps
Switching frequency $_{\text{max.}}$	5 Hz, inductive load acc. to IEC947-5-1, DC13
Function Outputs	Preset: DO 1: Target reached DO 2: Error; the outputs can be freely reconfigured
Galvanic isolation	1570 V DC in countercurrent with I/O module and digital inputs
Module-Specific Data	
Motor Connection	
Number of outputs	1 stepper motor (2 phases)

Output current	2 x 7.5 A temporary (smaller 10 s);
	2 x 5.0 A nominal current;
	Continuous operation: derating 0.1 A/K 50 °C and above

Max. stepper frequency	7812 Hz full step

Diagnostics		Short circuit or ground fault overcurrent Overtemperature monitoring Supply voltage monitoring Motor wire break Wrong rotational direction incremental encoder - motor
Resolution		64 microsteps per full step.
Protective functions		- Short circuit in motor windings is allowed - Short circuit in motor windings approximating 0 V or UZK motor voltage is allowed - Reverse polarity of +24 V control voltage is allowed - Reverse polarity of UZK motor voltage is not allowed
Galvanic isolation		1570 V DC to I/O module and digital inputs
Type of cable		E.g., LAPP UNITRONIC LIYCY 4 * 0.75 mm^2
Cable length		max. 30 m
Shield connector		both sides
Supply Connections		
Power supply	System voltage	via I/O module (5 V DC) Current consumption: 65 mA typ.
	Control voltage output 24 V DC	24 V DC (-25 % ... +30 %) Current consumption 90 mA typ. without ventilator 120 mA typ. with ventilator External fuse protection: Fuse 3.15 A M (medium slow) Fusing integral ≤ 20 A^2s
	Motor voltage 70 V UZK	55 V DC nominal voltage Operating range of 18 ... 70 V DC Current consumption 0 mA for deactivated output stage External fuse protection: Fuse 10 A FF (super quick) Fusing integral ≤ 22 A^2s
Data width		
Data width, internal		12 bytes input/output
Acyclic data		6 bytes mailbox overlapping
Cyclic data		3 bytes control information, 7 bytes data
Mechanical Data		
Dimensions W x H* x D (* from upper edge of rail)		51 mm x 70 mm x 100 mm
Weight		approx. 160 g

Standards and directives (see chapter 2.2 in manual on coupler/controller)		
EMC -Immunity to interference	acc. to EN 61000-6-2 (2005)	
EMC-Emission of interference	acc. to EN 61000-6-3 (2007)	
Approvals (see chapter 2.2 in manual on coupler/controller):		
c UL us	$_C$UL$_{US}$ (UL508C)	Certification in progress
GL	GL – Germanic Lloyd	Certification in progress
CE	Conformity marking	

3.6.1 Derating

In continuous operation, 5 A of motor current up to 50 °C is possible. Derating above 50 °C is 0.1 A per Kelvin.

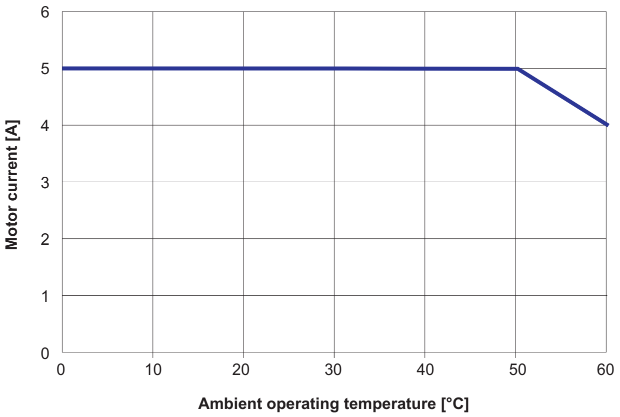


Fig. 3.6.1-1: Derating curve in continuous operation with ventilator g067221e

3.6.2 Specification of Current

3.6.2.1 Setting Motor Nominal Current

The vertex of the motor current can be defined via the configuration parameter Current at a resolution of 0.1 A.

3.6.2.2 Specification of Current Profile

The motor current can be set separately for the various motion phases.

- Standstill
- Acceleration
- Motion at constant speed
- Delay

The motor current is defined relatively to the motor nominal current; values between 0 % and 150 % are possible.

The module can supply a current of 150% of the module's nominal current for a limited time of up to 10 seconds.

To protect the module, a simplified thermal model is calculated for monitoring purposes. For this, the overcurrent is integrated when threshold value 150 % * 10 s is reached, is turned off with an error message.

A lower overcurrent (I_{OV}) can be supplied by the module for a correspondingly longer time. The time until deactivation (t_{off}) results from the following correlation:

$$t_{off} = \frac{5A}{I_{OV} - 5A} * 5s$$

After operation with overcurrent I_{OV} (time t_{OV}), a phase with reduced current I_{re} (time t_{re}) must follow, in order for the module's thermal model to return to its original state. The required cooling time (t_{cool}) is calculated as follows:

$$t_{cool} = 4 * \frac{I_{OV} - 5A}{5A - I_{re}} * t_{OV}$$

The prior overcurrent load can be retrieved in the internal status variables 21. If overcurrent is present, the variable increases to 254; if no overcurrent is present, the variable returns to zero. When 255 is reached, the error message #1413 PARTMODL_CURRENT_TIME is issued.

3.7 Process Image

The 750-672 I/O module provides the fieldbus coupler/controller 12 bytes input and output process image via 1 logical channel. The data to be sent and received are stored in up to 7 output bytes (D0 … D6) and 7 input bytes (D0 … D6), depending on the operating mode. Output byte D0 and input byte D0 are reserved and have no function assigned. 1 I/O module control and status byte (C0, S0) and 3 application control and status bytes (C1 ... C3, S1 ... S3) provide the control of the data flow.

Note
Mapping the process data of some I/O modules (or their variations) into the process image is specific to the fieldbus coupler/controller used. This information, as well as the specific configuration for relevant control/status bytes is located in the section "Fieldbus-Specific Configuration of Process Data." This section describes the process image of the particular coupler/controller.

3.7.1 Overview

A basic distinction is drawn between the cyclic process image and the mailbox process image.

	Mailbox process image (Mailbox activated)		**Cyclic process image (Mailbox deactivated)**	
Off-set	**Input data**	**Output data**	**Output data**	**Input data**
0	Status byte S0	Control byte C0	Status byte S0	Control byte C0
1	Reserved	Reserved	Reserved	Reserved
2	Process data D0 ... D6	Process data D0 ... D6	Mailbox MB0 ... MB5	Mailbox MB0 ... MB5
3				
4				
5				
6				
7				
8			Reserved	Reserved
9	Status byte S3	Control byte C3	Status byte S3	Control byte C3
10	Status byte S2	Control byte C2	Status byte S2	Control byte C2
11	Status byte S1	Control byte C1	Status byte S1	Control byte C1

Switching between the two process images is conducted through bit 5 in the control byte (C0 (C0.5). Activation of the mailbox is acknowledged by bit 5 of the status byte S0 (S0.5).

3.7.2 Control Byte, Status Byte

Control byte C0							
Bit 7	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
0	0	MBX	0	0	0	0	0

MBX	Mailbox operation 0: Mailbox deactivated. 1: Mailbox activated.
0	Reserved

Status byte S0							
Bit 7	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
0	ERR	MBX	X	X	X	X	X

MBX	Mailbox operation 0: Mailbox deactivated. 1: Mailbox activated.
ERR	Signaling errors and warnings ERR (status byte 0, bit 6) follows the general error bit Error (status byte 2, bit 7) and the general warning bit Warning (status byte 3, bit 6). ERR can be enabled via bits in the configuration table. In the default state, the ERR bit is not enabled. This means that errors and warnings will not result in a bit being set. 0: No error and no warning present. 1: An error or a warning is present.
X	Reserved

Note
Configuration of the control and status bytes C1 through C3 and S1 through S3 is described in detail in the corresponding sections.

In the tables of the control bytes and status bytes, the undescribed bits are highlighted gray. The currently described bits are not highlighted gray in the tables.

Control Byte C1							
Bit 7	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
Command					Start	Stop2_N	Enable

Enable Module enable
0: The power output stage is switched off. The current operating mode is ended.
Warning: output stage is immediately stopped; no stop ramp; drive can coast to a stop.
When closing down with a stop ramp, Stop2_N must be used.
1: The module is enabled and can be started when the corresponding return message is also available in the status.

Stop2_N Stopping of drive.
This bit can be used to deactivate the drive from the control system.
The return message is transmitted via Stop_N_ACK bit.
0: Current is being supplied to the motor, but it is at standstill.
If the motor is still turning it is put into standstill by the STOP delay command Acceleration_Stop_Fast.
The motor current can be increased using the parameter Current_Ratio_Stop.
The motor cannot be started-up.
1: The drive may be started.

Start Startup of drive. The drive is started in the selected mode on a positive edge.
If the edge is not accepted (in the Jog or Mailbox mode), an error message is generated.
0→1 The drive is started accordingly on the rising edge.

Single positioning
Speed control
The specified setpoints have been accepted from the process image.
Movement is made directly to the new target, even if the drive is already turning.
A previously calculated movement sequence is started immediately when the PreCalc_ACK bit is set (instant setpoint switch).

Run program
The run program is started at the Start_Instruction_Counter address in the process image.
A run program that is currently running is interrupted, and the program flow is started at the new address.
This allows various program parts to be transferred via the process image.

Referencing
The reference motion is initiated.
If the reference run is still in operation, the (new) setpoints are again accepted and calculated (same procedure as for positioning).
The reference run is newly initiated.

JogMode
No effect.
The drive is started in JogMode by default via Input3 (Jog_Pos) or Input4 (Jog_Neg).
The inputs and outputs can be freely reconfigured.
An error message is generated.

Mailbox operation
No effect in this operating mode.
The command is carried out via the mailbox.
If the bit is set nonetheless, neither an acknowledgment nor an error message is generated.

Command Selecting the operating mode
0: Idle mode
1: Single positioning
For this operating mode, the mailbox must be disabled.
2: Run program
3: Speed control
4: Reference motion
8: JogMode
The drive can be operated manually at the setup speed.
Control is performed via Input3 (Jog_Pos) and Input4 (Jog_Neg)
16: Run commands via mailbox
In this mode, all movement commands are issued directly via mailbox.

Status Byte S1							
Bit 7	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
Command_Ack					Start_ ACK	Stop_N_ ACK	Ready

Ready
Ready for operation
0: The module is not ready for operation. Either a corresponding request is present via Enable, or an error has resulted in cancellation of Ready.
When the bit switches from 1 to 0 the output stage is deactivated, or the output frequency is set to 0.
1: Readiness for operation has been requested via Enable and no error is present.

Stop_N_ACK
Acknowledge request bit Stop2_N.
0: The control system has reset the request bit Stop2_N or the Enable input has a (1) signal or the drive is motionless. The drive can not be started up using the Start control bit in this status.
1: The control system has set the request bit Stop2_N and the Enable input has a (1) signal or the drive is being braked.

Start_ACK
Start sequence in the operating mode.
0: This bit is also set to 0 when the Start request is canceled.
1: The rising edge function is a function of the selected operating mode.

- Single positioning — The specified setpoints have been accepted from the process image.
- Speed control — The drive has been initiated.

Run program — The run program has been initiated.
Referencing — The reference run has been initiated.
JogMode — No effect.
Handshake not performed.
Mailbox operation — No effect.
Handshake not performed.
As soon as the mailbox mode has been enabled, the corresponding commands can be transmitted via the mailbox.

Command_Ack
Confirmation: selection of operating mode
0: Idle mode is selected
1: Positioning mode active
Movement is made to the active setpoint on the next rising edge for Start.
2: Run program mode active
The Run program is started with the first command on the next rising edge for Start.
3: Speed control mode active.
Movement is made to the active setpoint on the next rising edge for Start.
4: Referencing mode active.
The drive is started at the setup speed on the next rising edge for Start.
8: JogMode active
16: Move commands via mailbox mode active.

Control byte C2							
Bit 7	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
Error_ Quit	PreCalc	X	X	Acc_Range_Sel		Freq_Range_Sel	

Freq_Range_Sel
Select frequency prescaler.
The prescaler Frq_Prescaler can be set for frequency using these two bits when the module is to be operated without configuration via the mailbox.
These values are accepted only when Enable is set to 0.

'00': The Freq_Prescaler prescaler is loaded with the parameter Freq_Div from the current configuration data set.
Freq_Div <> 0: Freq_Prescaler = Freq_Div, f_{max} = 2 MHz/Freq_Div
Freq_Div = 0: Freq_Prescaler = 200, f_{max} = 10 kHz

'01':	Freq_Prescaler = 80	f_{max} = 25 kHz
'10':	Freq_Prescaler = 20	f_{max} = 100 kHz
'11':	Freq_Prescaler = 4	f_{max} = 500 kHz

Acc_Range_Sel
Select acceleration factor.
These two bits are used to set the Acc_Multiplier factor for acceleration.
These values are accepted only when Enable is set to 0.

'00': The factor Acc_Multiplier is loaded with the parameter Acc_Fact from the current configuration data set.
Acc_Fact <> 0: Acc_Multiplier = Acc_Fact
Acc_Fact = 0: Acc_Multiplier 8, T = 7600 ms

'01':	Acc_Multiplier = 80	T = 760 ms
'10':	Acc_Multiplier = 800	T = 76 ms
'11':	Acc_Multiplier = 8000	T = 7.6 ms

PreCalc
Precalculation for movement sequence.
The setpoints are taken from the process image and, where required, a movement sequence precalculated.
This bit must only be used in the single positioning and speed control modes. In the other operating modes, the bit must be set to 0.

0: Each setpoint that is transmitted via cyclic telegram traffic must be accepted and processed. A possibly pre-calculated movement process will be discarded. A movement sequence can be calculated and started using Start.

1: The setpoints from the cyclic telegram traffic are ignored and the setpoint saved for the 0→1 edge used instead. If the starting speed is zero, a movement sequence will be calculated in advance using this setpoint; this sequence can then be started with the normal delay using Start.

Error_Quit
Acknowledge errors and warnings. (see chapter 3.13, “Diagnostics”)
All errors and warnings that are present are acknowledged at the rising edge from 0 to 1.
After acknowledgment, the errors and warning bits are set to 0 if no new errors and warnings are present.

X
Reserved

Status byte S2							
Bit 7	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
Error	PreCalc_ ACK	Referenc e_OK	Direction	On_ Speed	StandStill	Busy	On_ Target

On_Target
Target reached.
The significance of this bit depends on the selected operating mode.
0: A new mode will be selected, or a movement made to a new position.
1: Step positioning: / Run program: The specified setpoint within the TargetWindowPosition target window has been reached.
Referencing: The reference point has been moved to and set successfully.
Jog Mode: The bit is not used in this mode and remains at 0.
Mailbox mode: Function of mailbox command.
x Speed control: The bit On_Target has no function in this operating mode

Busy
Run task is executed and setpoint is not yet reached.
The selected mode is active and a task has been started;
the drive is rotating, or frequency output is not equal to 0.
0: No run task executed or setpoint has been reached.
1: Step positioning: Movement being made toward specified position.
Run program: The current Run program is being executed.
Speed control: Specified speed not yet reached.
Referencing: Movement made toward reference point.
Jog Mode: The drive has been started up using the pushbutton and is rotating.
Mailbox mode: Function of mailbox command.

StandStill
Drive standstill
The bit StandStill is set when the setpoint lies within the TargetWindowSpeed or TargetWindowPosition window.
0: Motor is turning.
1: Motor at standstill

On_Speed
Drive speed reached
On_Speed is set by individual commands in mailbox mode and in the run program.
0: The drive has not reached its setpoint speed.
1: The drive has reached its setpoint speed.
Step positioning: Specified speed from the process image has been reached.
Run program: / Speed control: / Mailbox operation The tolerance lies within the TargetWindowSpeed target window.
Referencing: SetupSpeed has been reached.
Jog Mode: The tolerance lies within the TargetWindowSpeed target window.

Direction
Direction of rotation.
This bit is valid only when StandStill is 0.
0: Drive moving in the negative direction.
1: Drive moving in the positive direction.

Reference_OK
Referencing OK.
Set when reference run has been successfully concluded.
0: Reference run initiated or no valid reference.
1: The reference point has been successfully located in the reference run mode.

PreCalc_ACK
Status; precalculation for movement sequence concluded.
This bit acknowledges the request for a precalculation using PreCalc.
0: Precalculation not yet completed, or no request received.
1: Precalculation completed.

Error
Common error for module. (see chapter 3.13, “Diagnostics”)
An error can/must be acknowledged using Error_Quit.
0: No error present for the drive.
1: Error present for the drive.

Control byte C3							
Bit 7	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
Reset_ Quit	0	0	0	Direction _Neg	Direction _Pos	0	SetActual _Pos

SetActual_Pos
At the rising edge, the RefOffsetPos reference position is transferred from the configuration table into the ActualPosition of the process image.
Because no handshake bit is present, the bit must be set for a minimum of 20 ms.
This function is only possible in standstill.
If the bit is set during operation, the actual value remains unchanged and no error is reported.
0: The RefOffsetPos reference position is not transferred
1: The RefOffsetPos reference position is transferred at the positive edge.

Direction_Pos
In the Reference run mode this bit defines that the reference switch be searched for in a positive direction.
0: Drive not to move in a positive direction.
1: Drive should move in a positive direction.
The drive is deactivated when the bit Direction_Neg is set at the same time.

Direction_Neg
In the Reference run mode this bit defines that the reference switch be searched for in a negative direction.
0: Drive not to move in a negative direction.
1: Drive should move in a negative direction.
The drive is deactivated when the bit Direction_Pos is set at the same time.
0: Limiting not active
1: Limiting active

Reset_Quit
Reset acknowledgement.
A Power-on reset or a warm start of the module can be detected by the control system with the Reset status bit; this must be acknowledged using Reset_Quit. This also occurs after saving the user configuration to the EEPROM. Volatile data, parameters and tables for the module may be inconsistent and must be reloaded to ensure proper operation.
0: Function not defined.
1: The Reset signal is reset.

0
Reserved

Status byte S3							
Bit 7	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
Reset	Warning	Input6	Input5	Input4	Input3	Input2	Input1

Input1
Status for Input 1.
In the default setting, input DI1 is linked with the motor shutdown circuit.
Shutdown can be performed via DI1 or through the control system.
0: Current is being supplied to the motor, but it is at standstill.
If the motor is still turning, it is put into standstill by the STOP acceleration command.
The motor cannot be started-up. This is signaled via bit Stop_N_ACK.
1: The drive may be started.

Input2
Status for Input 2.
Input DI2 is used as the reference input in the default settings.
0: The reference switch is not actuated.
1: The reference switch is actuated.

Input3
Status for Input 3. .
Input DI3 is used for JogMode in the default settings.
0: The drive shall not move in positive direction.
1: Drive should move in a positive direction.
If Input4 is set simultaneously, the drive is turned off.

Input4
Status for Input 4.
Input DI4 is used for JogMode in the default settings.
0: The drive shall not move in negative direction.
1: Drive should move in a negative direction.
If Input3 is set simultaneously, the drive is turned off.

Input5
Status for Input 5.
Input DI5 is used as the limit switch input in the default settings.
0: The positive direction limit switch is not actuated
1: The positive direction limit switch is actuated.
The drive is ramped down.

Input6
Status for Input 6..
Input DI6 is used as the limit switch input in the default settings.
0: The negative direction limit switch is not actuated
1: The negative direction limit switch is not actuated.
The drive is ramped down.

Warning
A warning is issued when an adjustable limit is exceeded. The warning can be acknowledged using Error_Quit.
0: No warning present for the drive.
1: Warning present for the drive.

Reset
Module has performed a reset.
With this bit, the control can detect the reset of the terminal. The bit is set after a reset and is confirmed and deleted by Reset_Quit.
0: No reset since last acknowledgment.
1: A reset has been carried out but not yet confirmed with Reset_Quit.
A Power-on reset or a warm start of the module can be detected by the control system with the Reset status bit; this must be acknowledged using Reset_Quit. This also occurs after saving the user configuration to the EEPROM. Volatile data, parameters and tables for the module may be inconsistent and must be reloaded to ensure proper operation.

X
Reserved

3.7.3 Cyclic Process Image

The process image appears as follows when the mailbox is deactivated (C0.5 = 0):

Off-set	Input data		Output data	
0	S0	Status byte S0	C0	Control byte C0
1	Reserved		Reserved	
2	D0	Process Data	D0	Process Data
3	D1	Process Data	D1	Process Data
4	D2	Process Data	D2	Process Data
5	D3	Process Data	D3	Process Data
6	D4	Process Data	D4	Process Data
7	D5	Process Data	D5	Process Data
8	D6	Process Data	D6	Process Data
9	S3	Status byte S3	C3	Control byte C3
10	S2	Status byte S2	C2	Control byte C2
11	S1	Status byte S1	C1	Control byte C1

The configuration of the process data depends on the set operating mode; this is described in the associated sections.

A basic distinction is drawn between the following process images:

- Positioning,
- Jogging,
- Run program,
- Speed setting.

3.7.4 Process Image Mailbox

The process image appears as follows when the mailbox is activated (C0.5 = 1):

Off-set	Input data		Output data	
0	S0	Status byte S0	C0	Control byte C0
1	Reserved		Reserved	
2	MB0	Opcode	MB0	Opcode
3	MB1	Status_Mbx	MB1	Control_Mbx
4	MB2	Reply Parameter byte 1	MB2	Request Parameter byte 1
5	MB3	Reply Parameter byte 2	MB3	Request Parameter byte 2
6	MB4	Reply Parameter byte 3	MB4	Request Parameter byte 3
7	MB5	Reply Parameter byte 4	MB5	Request Parameter byte 4
8	Reserved		Reserved	
9	S3	Status byte S3	C3	Control byte C3
10	S2	Status byte S2	C2	Control byte C2
11	S1	Status byte S1	C1	Control byte C1

The individual applications can be set using opcodes.

The opcodes are assigned to various topical areas. They are described in the following chapters.

The control byte and status byte for the mailbox have the following function:

Control_MBX							
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Toggle flag	0	0	0	0	0	0	0

Toggle flag The status of this bit must be inverted for the mailbox data to be imported. However, data is also accepted if an opcode is specified which differing from the previous one.

0 Reserved

Status_Mbx							
Bit 7	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
Toggle flag	Return-Code						

Return-Code The return code indicates whether the last command was executed without error. If so, a value of 0 is returned. When the return code provides a value other than 0, you must check the corresponding opcode.

Toggle flag If the state of the toggle flag differs in the Control_MBX, the mailbox is read out. Subsequently, the state of the bit is changed.

3.8 Mailbox Operation

The mailbox considerably expands the range of application.

The mailbox is activated when bit 5 of the control byte C0 is set to 1. Activation of the mailbox is acknowledged by bit 5 of the status byte S0.

Note

Mailbox mode is not automatically selected when the mailbox is activated in the coupler/controller! Bit 7 of control byte C1 must be set to 1 for this.

3.9 Table Manager

Access to the tables is handled using the Table Manager. Possible table types are:

- Run programs,
- Positioning of camshaft,
- Target positions,
- Configuration,
- Data recorder.

Several tables of one table type may be present at different memory locations. The storage location is addresses using an index.

Index	Storage location
0:	not available / no table active / Factory Default for configuration / EEPROM
1:	RAM 1
2:	RAM 2

Exceptions are: Configuration and the data recorder use RAM1 exclusively. The data from the data recorder can not be copied to the EEPROM.

One table of each table type may be activated and written when in motion mode.

Of each table type, one table can be saved to the EEPROM. Tables in EEPROM cannot be activated, but only copied to a RAM table. If a table is present in the EEPROM, it will be copied automatically to RAM1 and RAM2 following a reset and is activated such that it can be directly executed in RAM1.

The tables are loaded via a download from the control. The following rules apply for a download:

- A download is always conducted only into one table in the RAM.
- A download is only permitted, when the target table is inactive.

The download is verified for consistency using a checksum. If the checksum is incorrect, the loaded table is marked as invalid.

Tables may be copied. The following rules apply to the copying of tables:

- The target table must be empty. Tables in RAM can be deleted using the command TBL_ERA.

- A copy is only permitted, when the target table is inactive.

The Table Manager detects whether a table is blank, valid or invalid (for example during a download or after a faulted download). This information is saved for each table in a status byte:

Access to a table (except configuration), even from other program modules, can be performed only via the table manager. Depending on the table type, several options are available for access:

- Downloading of a table,
- Copying of a table,
- Deleting one or more tables,
- Activating a table,
- Writing / Teaching of an element (position table only).

The position table saves target positions which can be queried using special commands. This position table enables target positions to be edited and taught, without having to change the Run program.

The table for camshaft saves a bit pattern, which is output depending on the position. The activation of a different table for the camshaft is executed immediately.

The table for the configuration saves a data fields, containing configuration data.

3.9.1 Download

Table download is performed to implement a transport layer for transferring relatively large data volumes via the I/O bus.
The transferred data blocks are fragmented into 4-byte blocks, which are then transferred to the module at each I/O module cycle. These data bytes are embedded in the mailbox and can be transferred simultaneously with the process data, ensuring control over this process while also in this mode.

A download is basically broken down into 3 phases:

1. Preparation for download using the command DLD_START
2. Transfer of data using the command DLD_CONT
3. Completion of download using the command DLD_END.

These commands are elucidated in the appendix.

3.9.2 Control

After downloading of tables types:

- Run program
- Camshaft
- Position table

the tables, and the associated functions, must be enabled.

- The camshaft is always active after this.
- The Run program is active after this and can also be halted again, contrary to the camshaft.

These commands are elucidated in the appendix.

3.10 Configuration

The response of the stepper module is essentially determined by the settings in the configuration table. The configuration table is broken down into several sectors:

Addresses	Meaning
0 ... 127	Direct parameters (basic)
128 ... 223	Indicator table on source bits in the bit I/O table. The indicator table can be used to change the permanent assignment of internal and external function bits. The address of the target bit is consistent with the address of the configuration parameter.
224 ... 376	Direct parameters (basic)
380 ... 508	Direct parameters (extended)

The bit I/O table is broken down into two sectors: Addresses 0 ... 127 describe the data sources; addresses 128 ... 255 described the targets to which the indicator can point.

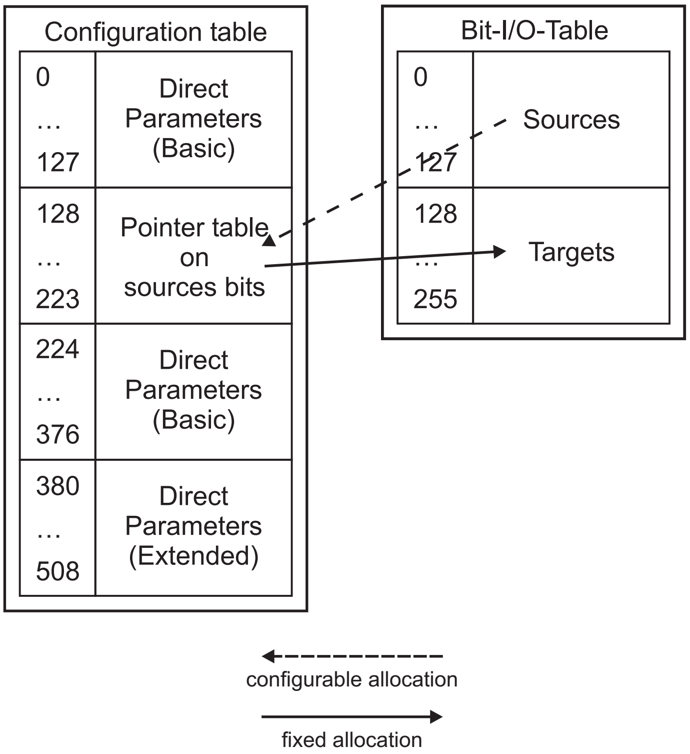


Fig. 3.9.2-1: Configuration and bit I/O tables g067230e

3.10.1 Configuration using Control Byte C2

The following values can be configured using the control byte C2 in the standard configuration.

3.10.1.1 Frequency Prescaler

The values for the frequency prescaler (Freq_Prescaler) are determined by the bits Freq_Range_Sel in control byte 2 (C2.0 and C2.1). If both of these bits are zero (0), the value for the parameter Freq_Div in the configuration table is used.

Freq_Range_Sel	'00'	'01'	'10'	'11'
Freq_Prescaler (frequency prescaler)	Freq_Div <> 0: Parameter Freq_Div from configuration table Freq_Div = 0: 200	80	20	4
$f_{p,max}$	Freq_Div <> 0: 2 MHz/Freq_Div Freq_Div = 0: 10 kHz	25 kHz	100 kHz	500 kHz

Note
If Freq_Range_Sel = 0 and configuration parameter Freq_Div = 0, the variable Freq_Prescaler is set to 200.

The run speed is determined by the pulse frequency (f_p), which is determined by the output data Velocity (D0 and D1) and by the prescaler Freq_Prescaler.

$$fp = \frac{Velocity * 80}{Freq\_\Pr escaler[Hz]}$$

The acceptable velocity range is 1 … 25000. The setting for the pulse frequency in [Hz] is given by selecting Freq_Prescaler = 80.

Note
The bits Freq_Range_Sel may only be modified when the control system is deactivated! These bits are therefore only accepted when Enable is not set.

Correlation between the internal pulse frequency and the mechanical rotational speed at the motor axis.

$$f_m = \frac{1}{p} * \frac{f_p}{256}$$

f_m Rotational speed at the motor axis revolutions per second
p Number of pole pairs
f_p Internal pulse frequency

Correlation between the internal acceleration and the mechanical acceleration

$$a_m = \frac{1}{p} * \frac{a}{256}$$

a_m Acceleration
p Number of pole pairs
a Internal acceleration

3.10.1.2 Acceleration Factor

The value for the acceleration factor (Acc_Multiplier) is determined by the bits Acc_Range_Sel in control byte 2 (C2.2 and C2.3). If both of these bits are zero (0), the value for the parameter Acc_Fact in the configuration table is used.

Acc_Range_Sel	'00'	'01'	'10'	'11'
Acc_Multiplier (acceleration factor)	Acc_Fact <> 0: Parameter Acc_Fact from configuration table Acc_Fact = 0: 8	80	800	8000
Acceleration period T to f_{max} at max. acceleration 32767	Acc_Fact <> 0: Setting from configuration Acc_Fact = 0: 7600 ms	760 ms	76 ms	0.2 ms

Note
If Acc_Range_Sel = 0 and configuration parameter Acc_Fact = 0, the variable Acc_Multiplier is set to 8.

Acceleration is determined by the output data Acceleration (D2 and D3) and the prescaler Freq_Prescaler and by the acceleration factor Acc_Multiplier.

$$a = Acceleration * \frac{Acc\_Multiplier}{Freq\_Prescaler}\left[\frac{Hz}{s}\right]$$

The permissible acceleration range is 1 … 32767.
Acceleration is set in [Hz/s] when the acceleration factor Acc_Multiplier is selected equal to the prescaler Freq_Prescaler.

Note
The bits Acc_Range_Sel may only be modified when the control system is deactivated! These bits are therefore only accepted when Enable is not set.

3.10.2 Configuration via Mailbox

Configuration Table

All parameters for configuring the servo stepper controller are in the configuration table.

The table below shows an excerpt from the configuration table.

Configuration variable	**Offset (Dec.):**	**Bit Offs.:**	**Data type:**	**Default:**	**Range:**	**Description**
User_Conf_Id	0		UINT16	0	0 ...50000	The dataset number can be freely assigned by the user. Numbers above 50000 are reserved.
ConfVersion	2		UINT8	4	0 ...254	Configuration version number
Application_ Selector	3		UINT8	1	0 ...2	Switching of the application The appropriate process image is activated when a new application is selected. 0: Reserved 1: Positioning controller
Freq_Div	4		UINT16	200	4 ...65535	Sets the prescaler for the maximum velocity
Acc_Fact	6		UINT16	80	1 ...65535	Sets factor for maximum acceleration

Access to the table is carried out via the mailbox.

All configuration parameters lie in the configuration data set.

The elements of the configuration data set are referenced via an address. The address results from the table assignment with configuration values. An element can occupy more than one byte, in which case the lowest value byte occupies the given address. For explicit access to an element, the size of the element must be stated.

The table can be found in the appendix to chapter 4.6, „Configuration Variables“.

The complete data set is loaded to the RAM on a download and is then saved to the EEPROM; a module warm start is then executed. The download is executed with the commands DLD_START, DLD_CONT and DLD_END. The commands are described in chapter 3.9, "Table Manager".

The writing procedure to the EEPROM is signaled by LED A.

The download is conducted by the table manager. The configuration dataset is saved to the EEPROM only after a successful download. Independent from the success of the download, a warm restart will be executed in any case.

The Reset status bit is set after the warm start; this must be canceled using the Reset_Quit control bit. Only then is the module operational again.

Alternatively, individual parameters can be modified target-oriented when the terminal is active.

This requires the diagnostics opcodes (see chapter 3.13, „

Diagnostics“).

If a valid configuration data set is available in the EEPROM after power-on, reset or a warm start of the module, this data set is loaded to the RAM; if not, the factory default data set is loaded, i.e. the module is restored to the WAGO as-delivered status.

3.10.2.1 Configuration of Basic Parameters

3.10.2.1.1 Application Selection

Application_Selector, Offset 0, Range [0 ...5]

The Application_Selector determines the basic function:

Value	Application
0	Reserved
1	Positioning controller

3.10.2.1.2 Prescaler for Maximum Velocity

Freq_Div, Offset 4, Range [4 ...65335]

The maximum internal output frequency is derived from an internal 2 MHz cycle by a prescaler. When the smallest possible prescaler (4) is selected, a maximum internal frequency of 500,000 Hz is yielded. The stepper frequency is derived from this fundamental frequency (see chapter 4.1, "Calculation Formulas").

3.10.2.1.3 Factor for Maximum Acceleration

Acc_Fact, Offset 6, Range [1 ...65535]

Acceleration is given in steps/s². The specified value is multiplied by the acceleration factor (Acc_multiplier) and then divided by the frequency prescaler (Freq_Prescaler).

$$a = Beschleunigungswert * \frac{Acc\_multiplier}{Freq\_\mathrm{Pr}escaler}$$

Acceleration value: Setting via process image, or parameter in an opcode.

3.10.2.1.4 Reference Run

Reference_Offset, Offset 108, Range [±8388607]

Position of reference switch.

Reference_Mode, Offset 112

Mode for referencing on start of a reference run using the control bit M_Reference. At the start of a reference run via the mailbox using the Move command START_REFERENCING, the call parameters are used (and NOT the following configuration bits).

Bit 1:	
0:	Reference run to reference switch
1:	Reference run to limit switch
Bit 2:	
0:	Reference run to negative end of a reference switch
1:	Reference run to positive end of a reference switch
Bits 3 ... 7: Reserved	

3.10.2.1.5 JogMode

Acc_Fact, Offset 44, Range [1 ...25000]

Default setup speed.
The current moving speed is used when this parameter is 0.

Acc_Fact, Offset 62, Range [0 ...32767]

Acceleration for JogMode and Referencing.

3.10.2.1.6 Ramps

Acceleration_Stop_Fast, Offset 46, Range [0 ...32767]

Default acceleration for STOP mode; the current acceleration is used when this parameter is 0.

Acceleration_RampUp, Offset 48, Range [0 ...32767]

Default acceleration for acceleration phase.

Acceleration_RampDown, Offset 50, Range [0 ...32767]

Default acceleration for delay phase.

**Acceleration_RampUp_Param, Offset 52,
Range [0–16777216]16777216]**

Default acceleration time or acceleration path

**Acceleration_RampDown_Param, Offset 56,
Range [0–16777216]16777216]**

Default delay period or delay path.

Acceleration_Modes, Offset 60

Bit 0 ... 1: AccType (acceleration type)	
0:	constant acceleration
1:	Linear rise in acceleration; the period for acceleration increase is Acceleration_RampUp_Param
2:	Sin2 rise in acceleration; the period for acceleration increase is Acceleration_RampUp_Param
3:	Reserved
Bit 2 .. Bit 2–3: AccParam (Acceleration parameter)	
0:	no modification
1:	Acceleration_RampUp_Param interpreted as the acceleration period
2:	Acceleration_RampUp_Param interpreted as the acceleration path
3:	Reserved
Bits 4 ... 5: DecType (deceleration type)	
0:	constant acceleration
1:	Linear rise in acceleration; the period for acceleration increase is Acceleration_RampUp_Param
2:	Sin^2 acceleration; the period for acceleration increase is Acceleration_RampUp_Param
3:	Reserved
Bits 6 ... 7: DecParam (decelration parameter)	
0:	no modification
1:	Acceleration_RampUp_Param interpreted as the acceleration period
2:	Acceleration_RampUp_Param interpreted as the acceleration path
3:	Reserved

3.10.2.1.7 Scaling Factors

Pos_Mult, Offset 20, Range [1 ... 65535]
Pos_Div, Offset 22, Range [1 ... 65535]

Scaling factors for position.

Speed_Mult, Offset 28, Range [1 ... 65535]
Speed_Div, Offset 30, Range [1 ... 65535]

Scaling factors for speed.

Acc_Mult, Offset 32, Range [1 ... Acc_Mult, Offset 32, Range [1–65535]
Acc_Div, Offset 34, Range [1–65535]65535]

Scaling factors for acceleration.

3.10.2.1.8 Brake

Braketime_Turn_On, Offset 100, Range [1 ...8388607]

Activation time for brake in [ms].

Braketime_Turn_Off, Offset 104, Range [0 ...8388607]

Switch-off time for brake in [ms].

3.10.2.1.9 Hardware/Software Configuration

HwSwConfig, Offset 19

Bits 0 ... 1: Reserved	
Bit 2: Drive_Direction (Direction of rotation inversion)	
0:	Output signal is processed directly
1:	Output signal: rotary direction is inverted
Bits 3 ... 6: Reserved	
Bit 7: Program_Autostart (Run program Autostart – Normal mode)	
0:	Run program activated only via Run program or Mailbox mode.
1:	Run program activated immediately after startup, see description.

3.10.2.1.10 Current Controller

The current controller works in accordance with the hysteresis procedure, which is distinguished by the following characteristics:

- Relatively easy parameterization
- Regulation characteristics are nearly independent from the motor
- Robust performance without stability problems
- Very quick reaction
- Minimal switching loss

The preset two- or three-point controller regulates the motor current within the hysteresis band that is specified by the parameter Current_Ctrl_Hysteresis.

The hysteresis should not fall below 50 mA, since the measurement noise and the resolution would then become negatively apparent. The upper limit should not exceed 10% of the motor current. The switching frequency falls in the case of an increasing hysteresis and can then lie within the audible range. The upper maximum switching frequency lies at approximately 30 kHz and cannot be exceeded.

Note
Due to the its architecture, readjusting the current controller is typically not necessary. Please contact WAGO Support.

Parameterizing the current controller:

Addr esses	Parameter	Range	Function
384	Currrent_Ctrl_Hysteresis	0 …5000	Current controller hysteresis: The current controller hysteresis has the unit [mA]. The parameter defines the width of the current band.
388	Currrent_Ctrl_TZMin	1…65535	Current three-point controller only: Min. waiting time after switching null pointer & before switching band in [us * 4]
392	Currrent_Ctrl_TZMax	1…65535	Current three-point controller only: Max. time after switching null pointer in [us * 4]. The band will then be forced to switch independently of the current.
394	Curr_Pass2	1…65535	Password for current controller parameters: EXOR link for parameters 384, 388, 392, 396 and 0xCODE. A false password will generate an error message.
396	Current_Ctrl_Type	0. 1	Current controller type: 0: Three-point controller 1: Two-point controller

3.10.3 Digital Signals and their Interconnection

The vital binary signals are addressed by central access functions. This permits easy, external access to all bits and allows linking of the bits to one another to be parameterized. Access to individual bits is performed using an index 0 … 255.

A bit function is defined by the function unit that sets the bit (source) and the function unit that reacts to this bit (target). Only the source is defined for output bits, and only the target for control bits. Only status bits have set links between two function units; they can, however, also be queries as output bits for further processing. Bits for which the source is not fixed are designated as linkable bits.

The universal filter functions FILT1 ... FILT8 possess a special status. The inputs for these filter functions are linkable bits that can be linked to any other bit. A query of these bits, on the other hand, provides information about the status of the filter. As a result, these bits represent a function between inputs and outputs.

The table containing the available bits is given in the appendix in chapter 4.5, „Bit Field for I/O Driver".

The following conventions apply:

- Source bits are assigned numbers 0 to 127 and may not be used as target bits. A source bit may reference several target bits.
- Target bits are assigned numbers 128 to 255 and may also be used as source bits. Target bits have exactly one source.
- References are stored in the configuration table. The names of the table entries correspond to those in the bit table. The prefix Ptr is placed in front of the identifier.
- The standard link between the source and target is entered in the column "Target/Source". This corresponds to the WAGO default settings (FACTORY_DEFAULT).

3.10.4 Linking of Bits

The expanded parameters are set using pointers (indices). The address for these pointers indicates a corresponding address in the bit I/O field. Allocation using names is also provided. "Ptr_nnn" indicates the variable "nnn".

Actual allocation is conducted using the content of the pointer.

The figure below illustrates allocation with an example of motor shutdown using Stop1_N and Stop2_N.

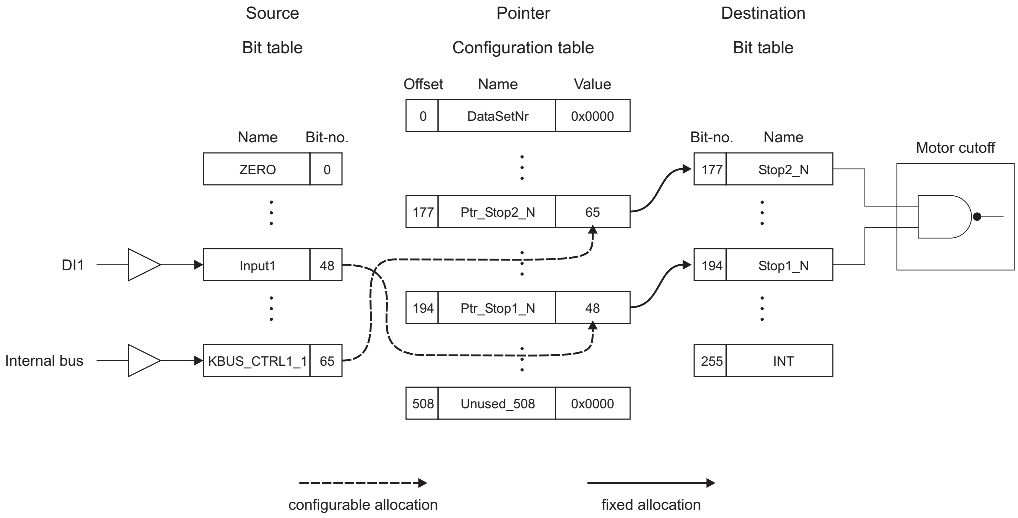


Fig. 3.10.4-1: Linking of bits p067x18e

Index Ptr_Stop1_N has a value of 48 (0x30), thus assigning Input1 to the Stop1_N variable.
Index Ptr_Stop2_N has a value of 65 (0x41), thus assigning control bit C1.1 to the Stop2_N variable.

The table excerpts given below show the corresponding entries for the configuration and bit I/O table.

Configuration Variable	Address		Data type	Default value	Range	Description
	Dec.	Hex.				
…						
Ptr_Stop2_N	177	0xB1	UINT8	0x41	0 ...255	Source for linkable bit 0xB1
…						
Ptr_Stop1_N	194	0xC2	UINT8	0x30	0 ...255	Source for linkable bit 0xC2
…						

Designation	Bit number		Type	Default allocation		Description
	Dec.	Hex.		Target/Source	Bit no.	
…						
Input1	48	0x30	SRC	KBUS_ST3_0 Stop1_N	0x90 0xC2	Input 1
…						
KBUS_CTRL1_1	65	0x41	SRC	Stop2_N	0xB1	Internal bus control byte 1 bit 1
…						
Stop2_N	177	0xB1	DST/ SRC	KBUS_CTRL 1_1	0x41	Drive Stop 2 inverted
…						
Stop1_N	194	0xC2	DST/ SRC	Input1	0x30	Drive Stop 1 inverted
…						

Linkable bits can be programmed for any source. This "linkability" enables flexible configuration and flexible arrangement of module terminal assignments. For example, the Start linkable bit can be set to the Input 1 fixed bit.

Note
A linkable bit can be linked to another linkable bit, but the maximum number of nesting levels is four (4). Too many nesting levels will yield an ambiguous result and the error ERR_LINK_NESTING will be issued.

Note
The nesting levels are not checked until the system run time.

The linking of bits occurs in the device configuration and is modifiable only by a reconfiguration. Exceptions are those bits which are linked to MZERO and MONE. Such bits can be freely set and reset during operation via mailbox commands or the run program.

Please see the following example for clarification:

Input 2 is normally set as the reference input. However, in a certain application is may be more advantageous to use Input 2 for specifying the direction of movement for the JogMode. Moreover, a "1" at the input should signify that the motor is moving in a positive direction.

The address and length of the configuration variables are given in the appendix in chapter 4.6, „Configuration Variables“.

Configuration variable	Address		Data type	Default value	Range	Description
	Dec.	Hex.				
…						
Ptr_FILT1	168	0xA8	UINT8	0x00	0 ...255	Source for linkable bit 0xA8
...						
Ptr_Direction_ Neg	187	0xBB	UINT8	0x53	0 ...255	Source for linkable bit 0xBB
…						
Filter1_Function	224	0xE0	UINT8	0	0 ...11	Function of filter: 1 = inversion
…						

The bits required for this are given in the appendix in chapter 4.5, „Bit Field for I/O Driver“.

Designation	Bit number		Type	Default allocation		Description
	Dec.	Hex.		Target/Source	Bit no.	
…						
Input2	49	0x31	SRC	KBUS_ST3_1 Set_Reference	0x91 0xBC	Input 2
…						
FILT1	168	0xA8	FILT	ZERO	0x00	Timer / Filter 1
…						
Direction_ Neg	187	0xBB	DST/ SRC	KBUS_CTRL 3_3	0x53	Move in negative direction.
…						

A prerequisite for this is that the mailbox has already been activated. This is described in chapter 3.8, "Mailbox Operation".

Using the above described procedures, the bits of control and status byte of the standard assignment can be modified.

Warning
Any change to the standard configuration will invalidate the description given for the changed items.

Unused linkable bits may be linked to the constant bits ZERO or ONE.

3.10.4.1 Special bits ZERO, ONE, MZERO and MONE

The ZERO and ONE bits are fixed.
The ZERO bit is always deleted and has a value of 0; the ONE bit is always set and has a value of 1.
Linkable bits can be set to a fixed value using ZERO and ONE bits.

The MZERO and MONE bits have the same function, but also possess an additional function when they are the source for linkable bits.
The source bit gives the status of linkable bits. As a result, linkable bits cannot be changed by commands or access functions. Exceptions to this rule are the linkable bits that are linked to MZERO and MONE.
A bit that is linked to MZERO is first deleted after a reset, but can be manipulated as required using the mailbox command or the Run program. A bit that is linked to MONE is initially set after a reset and is otherwise treated the same as a bit linked to MZERO.

3.10.4.2 User Bits

The following bits in the bit table are accessible to the user as marker bits:

Designation	Bit number	
	Dec.	Hex.
UserBit_0	152	0x98
UserBit_1	153	0x99
UserBit_2	154	0x9A
UserBit_3	155	0x9B
UserBit_4	156	0x9C
UserBit_5	157	0x9D
UserBit_6	158	0x9E
UserBit_7	159	0x9F

As an example, they can be reasonably implemented in a run program.

Warning
The user bits can only be written when they are properly linked in the configuration table.

Configuration Variable	Address	
		Dec.
Ptr_UserBit_0	X____98	152
Ptr_UserBit_1	X____99	153
Ptr_UserBit_2	X____9A	154
Ptr_UserBit_3	X____9B	155
Ptr_UserBit_4	X____9C	156
Ptr_UserBit_5	X____9D	157
Ptr_UserBit_6	X____9E	158
Ptr_UserBit_7	X____9F	159

The bits are linked with the bit MZERO (2), which enables write access.

3.10.4.3 Filters, Low Pass, Timers and Counters

The filter is configured using the table with the configuration values (see chapter 4.6, „Configuration Variables“).

Eight special bits are defined that can implement the following functions:

1. Inverting
2. Starting edge filter
3. Low pass
4. Pulse extension
5. Monoflop, not retriggerable
6. Pulse delay
7. Math
8. Counter, up
9. Counter, up, stop at overrun
10. Counter, down
11. Counter, down, stop at overrun.

The functions inversion and starting edge filter react to the linkable input bit and trigger the output undelayed.

The functions low pass, impulse extension, monoflop and pulse delay react to the linkable input bit and trigger the output according to the selected function with a definable time constant. This time can be set between 0 … 16777215 ms.

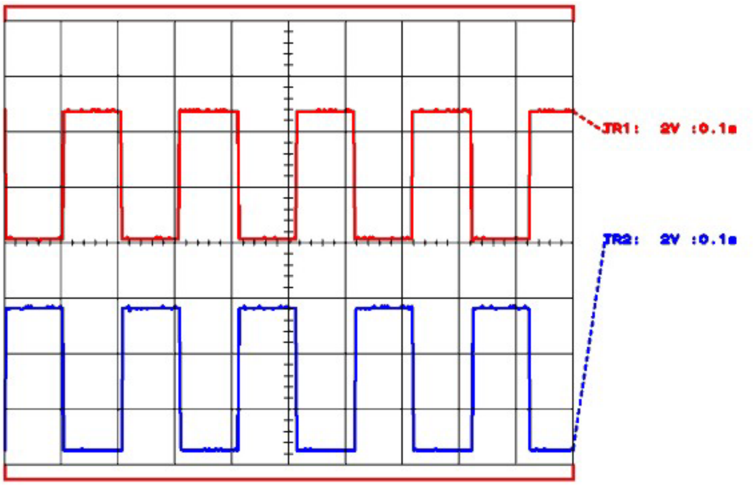


Fig. 3.10.4-2: Inverting p067020x

A starting edge filter recognizes to states. After a reset, the filter is in wait state as long as the input is active. The output supplies 0. As soon as the input signal is Zero for the first time, the filter changes to operating mode and the input signal is passed unchanged to the output. An operational change back to the wait state is not intended.

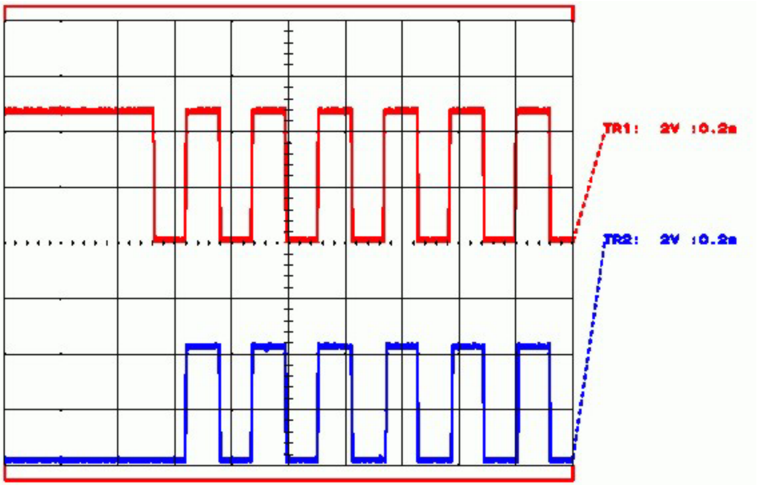


Fig. 3.10.4-3: Starting edge filter p067021x

The Low pass does not accept any change of the input signal until the new status is constant during the runtime.

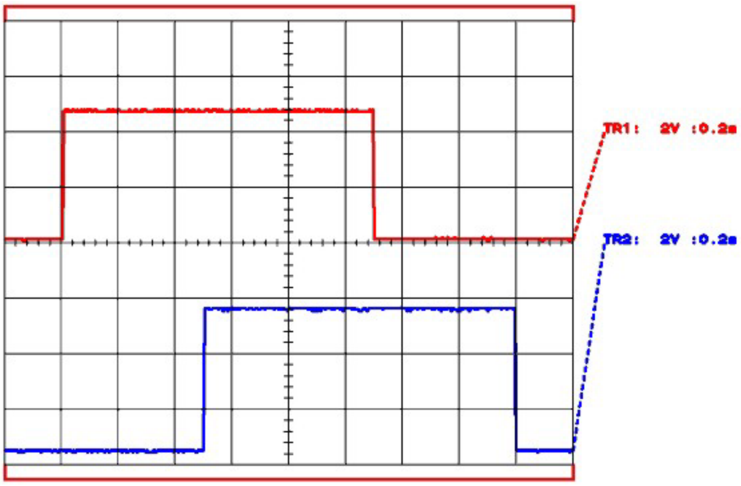


Fig. 3.10.4-4: Low pass 500 ms p067022x

The impulse expansion sets the output at a 0→1 edge of the input.. The output is reset when the time set has passed after the 1→0 edge.. A re-triggering during the runtime is possible.

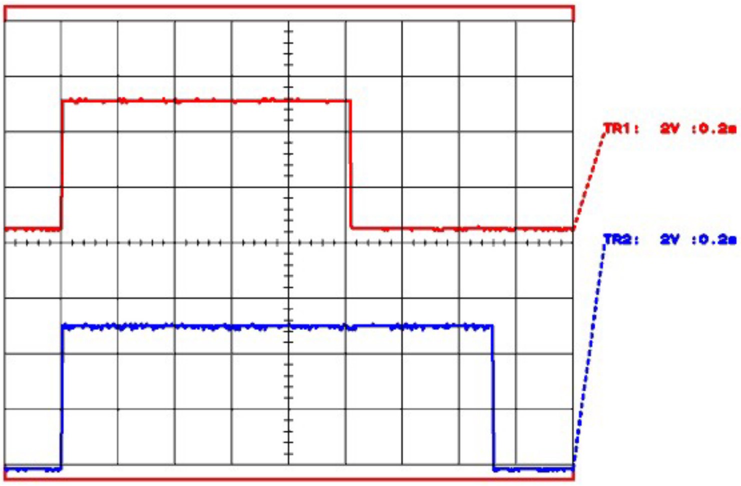


Fig. 3.10.4-5: Pulse extension 500 ms p067023x

The monoflop function sets the output at a 0→1 edge of the input. After the set time has passed, the output is reset.. A re-triggering during the runtime is not possible.

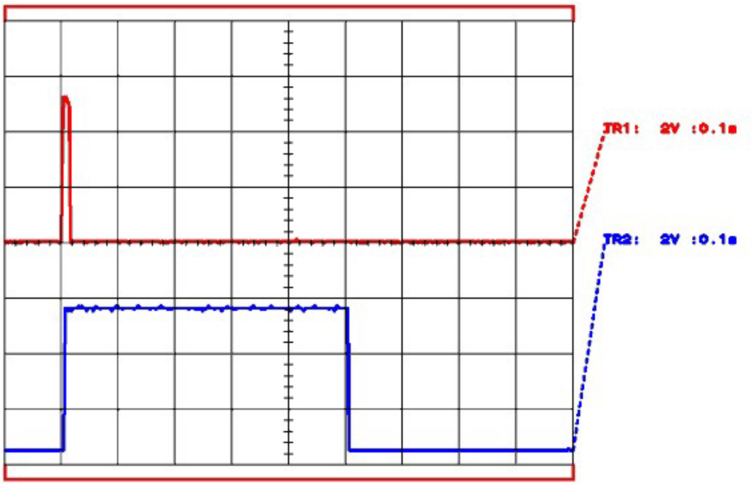


Fig. 3.10.4-6: Monoflop 500 ms cannot be retriggered with short input pulse p067024x

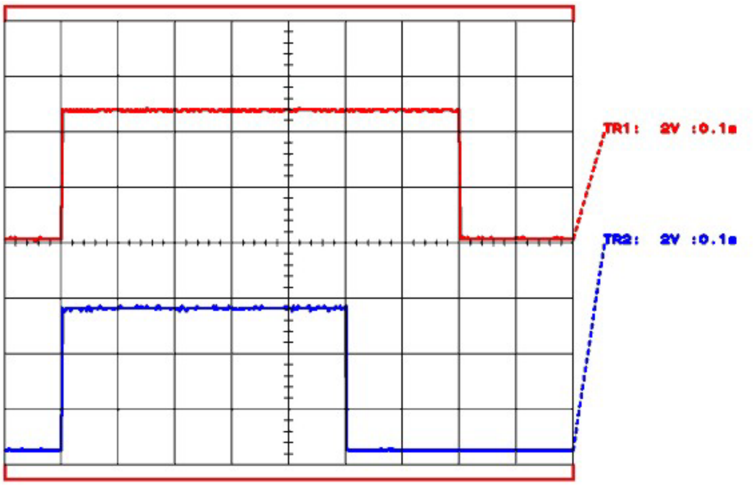


Fig. 3.10.4-7: Monoflop 500 ms cannot be retriggered with long input pulse p067025x

Pulse extension sets the output when the set time has expired after an input 0→1 edge. The output is reset as soon as the input is deactivated. The output is not reset if the input is deactivated before the set time expires.

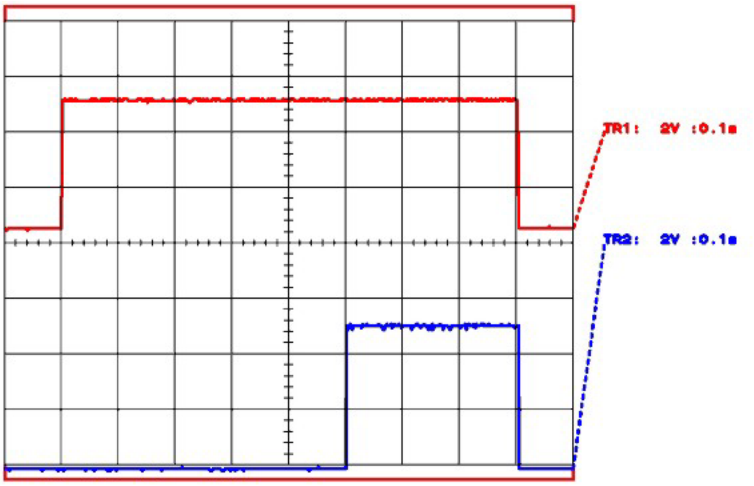


Fig. 3.10.4-8: Pulse delay 500 ms p067026x

Application example: filters connected in series: Monoflop 500 ms, inverting and a second monoflop 100 ms.

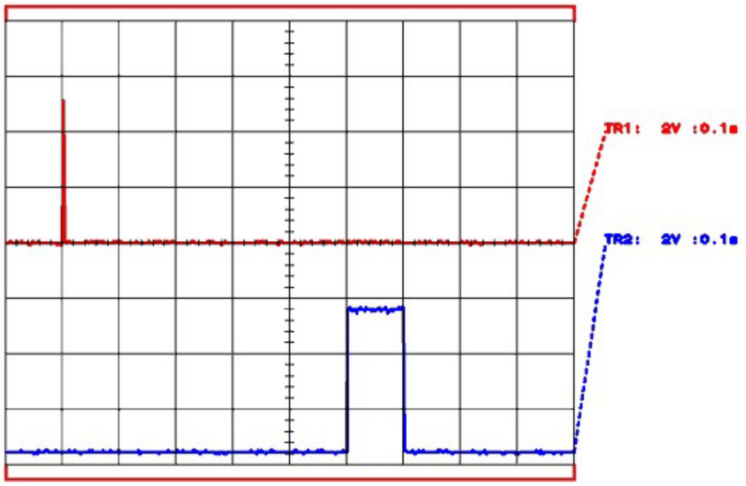


Fig. 3.10.4-9: Filters connected in series: Monoflop 500 ms, Inverting and second monoflop 100 ms p067027x

The math function reacts independently of the input bit. This value can only be set, incremented or decremented using commands. These commands can be transferred in the mode "Mailbox Move command," or processed as a component of the downloaded program in the mode "Program mode." The output is set when the counter has a value other than zero and is reset when the filter reaches zero. The commands for modifying the filter can only be used in the function "Math" or "Counters."

In the mode "Counter, up"/"Counter, down" the filter value is raised or lowered by one for each 0→1 edge of the filter input. The maximum counting frequency for external signals is 1 kHz; internal signals are normally evaluated once per program cycle.

The mode "Counter, up, stop" / "Counter, down, stop" behaves in the same manner as "Counter, up" / "Counter, down", except that counting is halted when the filter value reaches zero. This however means that the filter value must be preloaded with a starting value unequal to Zero, from which it counts towards Zero.

3.10.5 Run Commands

The commands are divided in classes:

- Table commands
- Run commands
- Auxiliary commands

In addition to pure Move commands, auxiliary commands and table commands are also accepted. The Move commands are passed along to the command interpreter. The table commands and auxiliary commands are required exclusively for the Run program mode.

A process is started using Move commands; there is no waiting for the end of the process (such as reaching a target).

Most commands are processed immediately, with the following exceptions.

Motion commands are only started when the previous motion command has arrived at the target. Table processing is interrupted in this case until the target is reached.

An exception here are the direct Move commands _IMM (immediate), which discontinue the Move command currently in progress and are directly processed.

Commands waiting for an event are repeated until the event has occurred.

The Move command manager decides from which source the Move commands are to be accepted. Possible sources may be:

- Status control,
- Limit switches,
- Run commands via mailbox commands,
- Positioning,
- Referencing,
- JogMode,
- Program mode.

Condition control and limit switches are processed in priority; they can only execute commands to brake the drive. The selection of the corresponding operating mode switches between the other setpoint values.

A run program serves in the consecutive execution of individual motions. In addition, some values can be set via the run program.

Two options are available for the processing of a run program:

- Program operation:
  The individual commands are collated and loaded to the terminal via download. The run program can then be executed there through the Program mode (see chapter 3.11.1.7, „Run Program Operating Mode“).

- Run task via mailbox:
  The module can be operated via mailbox using the move commands. Movement can be made directly to different positions. This command is accepted only when the mode "Move task via mailbox"(chapter 3.11.2, “Move Mode via Mailbox”) has been activated.

Each individual step in the Run program has the following format:

Byte	Meaning
1	Command
2	Data 1
3	Data 2
4	Data 3

The individual commands are elucidated in the appendix in chapter 4.3, „Commands for Move Mode“.

3.10.6 Scaling, Number Ranges and Units

Stepper drives rotate at a defined angle on each pulse. The software is oriented to this impulse output; accordingly, the internal unit for the position is a "step". Velocity and acceleration are derived form this. The interface enables adaptation and conversion to application-specific units (path in m, mm, degrees, and velocities in m/s, degrees/s).

3.10.6.1 Internal Measuring Unit

3.10.6.1.1 Time

On account of the periodical processing employed, time is measured in TICKS, with a TICK being the duration of one scan interval. A TICK is equal to one millisecond. Physical units based on time are converted accordingly.

3.10.6.1.2 Path

The travel, or path, is measured in "steps". The number range is 24 bits, including sign.

Position range: -8388608–+8388607,
Presentation in two's complement.

Correlation between the position specification in the process data and the rotor bearing on the motor axis.

$$\varphi = 360° * \frac{s}{64 * N}$$

φ	[°]	Rotor bearing, axis angle
s	[1]	Setpoint position in process image
N	[1]	Number of full steps

3.10.6.1.3 Velocity

Velocity is measured in "steps per unit of time", with the range being ±15 bits.

Velocity range:

Velocity $_{\text{min}}$ ... Velocity $_{\text{max}}$ = -25000 ... +25000

3.10.6.1.4 Rotational Speed

3.10.6.1.4.1 Calculation via Internal Pulse Frequency

Correlation between velocity specification in the process data and internal pulse frequency

$$f_p = \frac{Speed\_Mult}{Speed\_Div} * \frac{80 * v}{Freq\_Div}$$

f_p [1/s]	Internal pulse frequency in increments per second^2
v [1]	Setpoint velocity in process image
Speed_Mult	Scaling factor for setpoint velocity (parameter 28 from configuration table)
Speed_Div	Scaling factor for setpoint velocity (parameter 30 from configuration table)
Freq_Div	Prescaler for maximum velocity (parameter 4 from configuration table)

Correlation between the internal pulse frequency and the mechanical rotational speed at the motor axis

$$f_m = \frac{1}{p} * \frac{f_p}{256}$$

f_p	[1/s]	Internal pulse frequency in increments per second^2
f_m	[1/s]	Rotational speed at motor axis in revolutions per second
p	[1]	Number of pole pairs
v	[1]	Setpoint speed in process image

3.10.6.1.4.2 Direct Calculation

Correlation between velocity specification in the process data and mechanical rotational speed at the motor axis

$$f_m = \frac{1}{p} * \frac{Speed\_Mult}{Speed\_Div} * \frac{v}{3{,}3 * Freq\_Div}$$

$$n_m = 60 * f_m$$

f_p [1/s]	Internal pulse frequency in increments per second^2
f_m [1/s]	Rotational speed at motor axis in revolutions per second
n_m [1/min]	Rotational speed at motor axis in revolutions per minute
p [1]	Number of pole pairs
v [1]	Setpoint velocity in process image
Speed_Mult	Scaling factor in process image (parameter 28 from configuration table)
Speed_Div	Scaling factor for setpoint velocity (parameter 30 from configuration table)
Freq_Div	Prescaler for maximum velocity (parameter 4 from configuration table)

These values can also be specified and defined with user-specific units. For this, the specified data for speed is multiplied by a factor Speed_Mult prior to internal processing and then divided by a factor Speed_Div. If the internal → external or external → internal conversion violates the permissible value range, error message 1513 (UNITS_SPEED_INT_RESULT) or 1514 (UNITS_SPEED_USER_RESULT) will be issued.

3.10.6.1.5 Acceleration

Correlation between internal acceleration and specification in the process data

$$a = \frac{Acc\_Mult}{Acc\_Div} * \frac{Acc\_Fac}{Freq\_Div} * Acceleration$$

a	Internal acceleration
Acc_Mult	Scaling factor for setpoint acceleration (parameter 32 from configuration table)
Acc_Div	Scaling factor for setpoint acceleration (parameter 34 from configuration table)
Acc_Fac	Factor for maximum acceleration (parameter 6 from configuration table)
Freq_Div	Prescaler for maximum velocity (parameter 4 from configuration table)
Acceleration	Setpoint acceleration

Correlation between the internal acceleration and the mechanical acceleration

$$a_m = \frac{1}{p} * \frac{a}{256}$$

a_m [1/s^2]	Acceleration at motor axis in revolutions per second^2
p [1]	Number of pole pairs
a	Internal acceleration

These values can also be specified and defined with user-specific units. For this, the specified data for acceleration is multiplied by a factor Acc_Mult prior to internal processing and then divided by a factor Acc_Div. If the internal → external or external → internal conversion violates the permissible value range, error message 1515 (UNITS_ACC_INT_RESULT) or 1516 (UNITS_ACC_USER_RESULT) will be issued.

3.10.6.2 External Measuring Units

Internal representation can also be converted to application-specific units using conversion factors. Conversion is performed by muliplying by a configurable factor (*_MULT) and then dividing by a configurable factor (*_DIV). This way, fractions can be set with high accuracy within a wide range. These factors are expalined in greater detail in the following sections.

3.10.6.2.1 Path

The setting data for the positions are multiplied by the configuration factor Pos_Mult prior to internal processing and then divided by the configuration factor Pos_Div.

For example, is a step (or microstep) corresponds to travel of 0.12 mm, the setting can be given in µm by selecting Pos_Mult = 1 and Pos_Div = 120, or in mm by selecting Pos_Mult = 25 and Pos_Div = 3.

If the internal → external or external → internal conversion violates the permissible value range, error message 1511 (UNITS_POS_INT_RESULT) or 1516 (UNITS_POS_USER_RESULT) will be issued.

3.10.6.2.2 Velocity

The setting data for velocity are multiplied by the configuration factor Speed_Mult prior to internal processing and then divided by the configuration factor SPEED_DIV and the prescaler Freq_Prescaler.

If the internal → external or external → internal conversion violates the permissible value range, error message
1513 (UNITS_SPEED_INT_RESULT) or
1514 (UNITS_SPEED_USER_RESULT) will be issued.

3.10.6.2.3 Acceleration

The setting data for acceleration are multiplied by the configuration factor Acc_Mult prior to internal processing and then divided by the configuration factor Acc_Div.

If the internal → external or external → internal conversion violates the permissible value range, error message 1515 (UNITS_ACC_INT_RESULT) or 1516 (UNITS_ACC_USER_RESULT) will be issued.

3.11 Operating Modes

3.11.1 Operation via Cyclic Process Image

Various operating modes are available in the stepper controllers. They are described in the following chapters.

An operating mode is selected using the Command[0...5] mode selection bits in the cyclic process image

3.11.1.1 Selecting an Operating Mode

A mode can only be selected when the module is ready for operation and no other mode is active. This is the case when the status bits Ready and Stop_N_ACK are active and Start_ACK and Start are not active. If these conditions is not fulfilled but a Command[0...5] mode selection bit is still set, mode selection is delayed until this condition is fulfilled.
This delay ensures that any tasks in progress are ended properly.

An operating mode is selected by setting the assigned mode selection bit. If an operating mode is already active, setting another mode selection bit has no function, and the old operating mode is maintained.

The selection of an operating mode is acknowledged by the assigned operating mode status bit *_ACK.

3.11.1.2 Exiting an Operating Mode

An operating mode is exited by deleting the assigned mode selection bit. If the drive is still in motion, it is then braked at the acceleration Acceleration_Stop_Fast. The assigned mode status bit will not be canceled until the drive comes to a standstill.

The Standstill status bit is set when the drive is motionless.

A mode is also ended when Stop_N_ACK is reset. If the drive is still in motion, it is then braked at the acceleration Acceleration_Stop_Fast. The assigned mode status bit will not be canceled until the drive comes to a standstill.

A mode is also ended when Ready is reset. Independent from the drive condition, the speed is immediately set to Zero. The assigned operating mode status bit is cleared immediately. This procedure poses loss of control over the drive, allowing the motor to run down in an uncontrolled manner — particularly at loads with high moments of inertia. This is usually combined with step loss, and a reference motion should be initiated subsequently.

3.11.1.3 Sequence Diagram for Selection and Ending of Modes

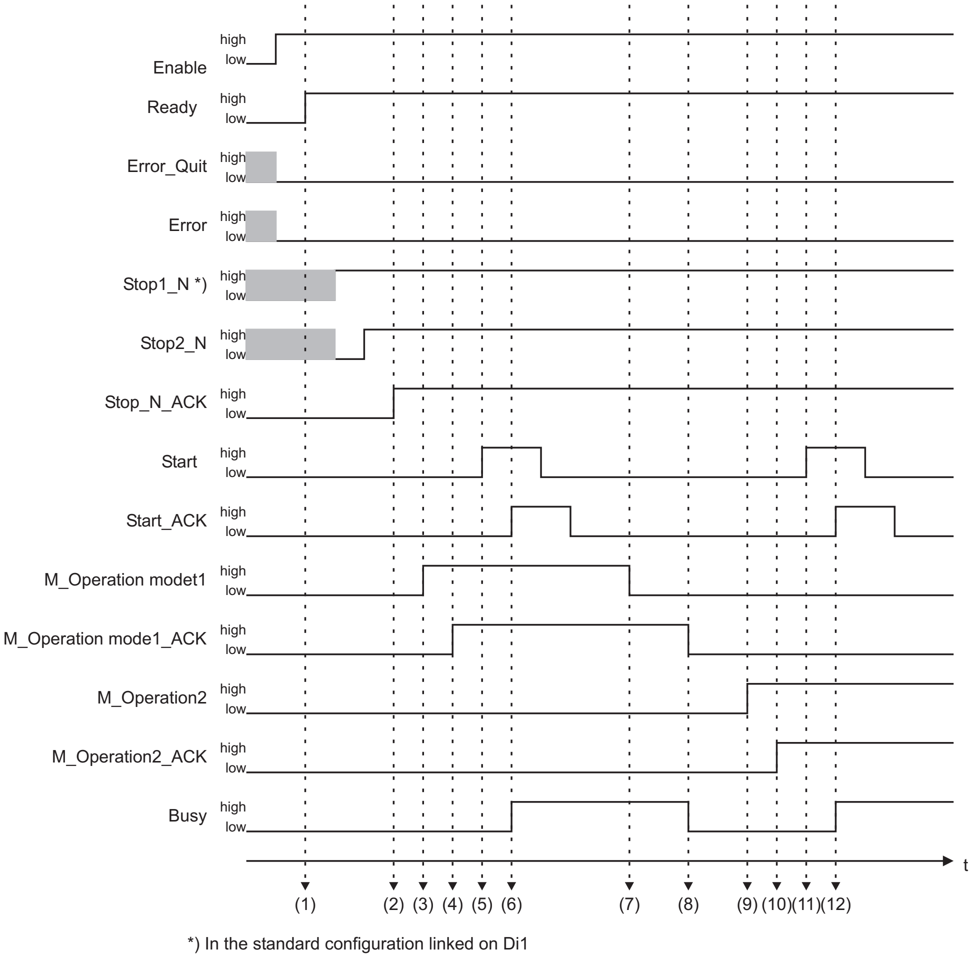


Fig. 3.11.1-1: Sequence diagram for Positioning g067x20e

(1)	The module is ready for operation when no error is present after setting of Enable.
(2)	An operating mode can be selected when Stop_N_ACK has been set.
(3)	Mode 1 is selected.
(4)	Mode 1 has been accepted by the module.
(5)	The drive is started by the Start rising edge in Mode 1.
(6)	Start can be canceled if the Start_ACK bit has been set.
(7)	Mode 1 is ended. The drive is run down to standstill.
(8)	Mode 1 is ended. The drive is motionless.
(9)	Mode 2 is selected.
(10)	Mode 2 has been accepted by the module.
(11)	The drive is started by the Start rising edge in Mode 1.
(12)	Start can be canceled if the Start_ACK bit has been set.

*) Linked to DI 1 in the standard configuration.

3.11.1.4 Single Positioning Operating Mode

The single positioning operating mode is only possible when the mailbox is deactivated.

The mode must first be activated using Command[]=1. When bit Command_ACK[]=1 is set, the singe positioning mode is active. The following setpoints can then be specified:

- Velocity,
- Acceleration,
- Absolute position.

The permissible velocity range is 1 … 25000. A velocity equal to 0, velocities greater than 25000 and negative velocities are not permitted and will result in an error message.

The permissible acceleration range is 1 … 32767. An acceleration equal to 0 and negative acceleration are not permitted and will result in an error message.

The setpoints are not accepted until a rising edge of Start. The drive starts up, or frequency is output, directly after the setpoints are accepted. The bit On_Target is canceled immediately and is not reset until the final position has been reached.

The Busy bit is set from the time when the setpoints are accepted until the target is reached.

New setpoints can be activated during the run by a new rising edge of Start. Movement is made toward the new position immediately at the new velocity and acceleration. Movement toward the old position is not continued (on-the-fly change). This function may also change the velocity or acceleration during an ongoing process.

Note
When using linear or sin^2 acceleration, the rapid setpoint acquisition can only be implemented when the drive is running with v = const. During constant acceleration, the new setpoint is acquired even during periods of increase.

The special features associated with limit switches and the Jog and Referencing modes are described in the corresponding sections.

3.11.1.4.1 Process Image Single Positioning

The single positioning process image represents the standard configuration for stepper positioning control and is shown in the following tables:

Offset	Input data		Output data	
0	S0	Status byte S0	C0	Control byte C0
1	Reserved		Reserved	
2	D0	Actual Velocity L	D0	Velocity L
3	D1	Actual Velocity H	D1	Velocity H
4	D2	Reserved	D2	Acceleration L
5	D3	Reserved	D3	Acceleration H
6	D4	Actual position L	D4	Target position L
7	D5	Actual position M	D5	Target position M
8	D6	Actual position H	D6	Target position H
9	S3	Status byte S3	C3	Control byte C3
10	S2	Status byte S2	C2	Control byte C2
11	S1	Status byte S1	C1	Control byte C1

The meaning of the control and status bytes is given in the following tables:

Control byte C1							
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Command					Start	Stop2_N	Enable

Enable See chapter 3.7.2, "Control Byte, Status Byte".

Stop2_N See chapter 3.7.2, "Control Byte, Status Byte".

Start Startup of drive.
The drive is started in the selected mode on a positive edge.
If the edge is not accepted (in the Jog or Mailbox mode), an error message is generated.
0→1: The drive is started accordingly on the rising edge.
The specified setpoints have been accepted from the process image.
Movement is made directly to the new target, even if the drive is already turning.
A previously calculated movement sequence is started immediately when the PreCalc_ACK bit is set (instant setpoint switch).

Command Selecting the operating mode

0: Idle mode
1: Single positioning
For this operating mode, the mailbox must be disabled.

Status Byte S1							
Bit 7	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
Command_Ack					Start_ ACK	Stop_N_ ACK	Ready

Ready — See chapter 3.7.2, "Control Byte, Status Byte".

Stop_N_ACK — See chapter 3.7.2, "Control Byte, Status Byte".

Start_ACK — Start sequence in the operating mode.
0: This bit is also set to 0 when the Start request is canceled.
1: The rising edge function is a function of the selected operating mode.
Single positioning — The specified setpoints have been accepted from the process image.
The drive has been started.

Command_Ack — Confirmation: selection of operating mode
0: Idle mode is selected
1: Single positioning mode active.
Movement is made to the active setpoint on the next rising edge for Start.

Control Byte C2							
Bit 7	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
Error_ Quit	PreCalc	X	X	Acc_Range_Sel		Freq_Range_Sel	

Freq_Range_Sel — Select frequency prescaler.
The prescaler Frq_Prescaler can be set for frequency using these two bits when the module is to be operated without configuration via the mailbox.
These values are accepted only when Enable is set to 0.
'00': The Freq_Prescaler prescaler is loaded with the parameter Freq_Div from the current configuration data set.
Freq_Div <> 0: Freq_Prescaler = Freq_Div, f_{max} = 2 MHz/Freq_Div
Freq_Div = 0: Freq_Prescaler = 200, f_{max} = 10 kHz
'01': Freq_Prescaler = 80 — f_{max} = 25 kHz
'10': Freq_Prescaler = 20 — f_{max} = 100 kHz
'11': Freq_Prescaler = 4 — f_{max} = 500 kHz

Acc_Range_Sel — Select acceleration factor.
These two bits are used to set the Acc_Multiplier factor for acceleration.
These values are accepted only when Enable is set to 0.
'00': The factor Acc_Multiplier is loaded with the parameter Acc_Fact from the current configuration data set.
Acc_Fact <> 0: Acc_Multiplier = Acc_Fact
Acc_Fact = 0: Acc_Multiplier 8, T = 7600 ms
'01': Acc_Multiplier = 80 — T = 760 ms
'10': Acc_Multiplier = 800 — T = 76 ms
'11': Acc_Multiplier = 8000 — T = 7.6 ms

PreCalc — Precalculation for movement sequence.
The setpoints are taken from the process image and, where required, a movement sequence precalculated.
This bit must only be used in the single positioning and speed control modes. In the other operating modes, the bit must be set to 0.
0: Each setpoint that is transmitted via cyclic telegram traffic must be accepted and processed. A possibly pre-calculated movement process will be discarded. A movement sequence can be calculated and started using Start.
1: The setpoints from the cyclic telegram traffic are ignored and the setpoint saved for the 0→1 edge used instead. If the starting speed is zero, a movement sequence will be calculated in advance using this setpoint; this sequence can then be started with the normal delay using Start.

Error_Quit — See chapter 3.7.2, "Control Byte, Status Byte".

X — Reserved

Status Byte S2							
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Error	PreCalc_ ACK	Referenc e_OK	Direction	On_ Speed	StandStill	Busy	On_ Target

On_Target	Target reached. The significance of this bit depends on the selected operating mode. 0: A new mode will be selected, or a movement made to a new position. 1: Step positioning: The specified setpoint within the TargetWindowPosition target window has been reached.
Busy	Run task is executed and setpoint is not yet reached. The selected mode is active and a task has been started; the drive is rotating, or frequency output is not equal to 0. 0: No run task executed or setpoint has been reached. 1: Movement being made toward specified position.
StandStill	See chapter 3.7.2, "Control Byte, Status Byte".
On_Speed	Drive speed reached. On_Speed is set by individual commands in mailbox mode and in the run program. 0: The drive has not reached its setpoint speed. 1: Specified speed from the process image has been reached. The tolerance lies within the TargetWindowSpeed target window.
Direction	See chapter 3.7.2, "Control Byte, Status Byte".
Reference_OK	See chapter 3.7.2, "Control Byte, Status Byte".
PreCalc_ACK	See chapter 3.7.2, "Control Byte, Status Byte".
Error	See chapter 3.7.2, "Control Byte, Status Byte".

Control Byte C3							
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Reset_ Quit	Setup_ Speed_ Active	0	0	Direction _Neg	Direction _Pos	0	0

Direction_Pos	See chapter 3.7.2, "Control Byte, Status Byte".
Direction_Neg	See chapter 3.7.2, "Control Byte, Status Byte".
SetupSpeed_Active	See chapter 3.7.2, "Control Byte, Status Byte".
Reset_Quit	See chapter 3.7.2, "Control Byte, Status Byte".
0	Reserved

Status Byte S3							
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Reset	Setup_ Speed_ Active_ ACK	Input6	Input5	Input4	Input3	Input2	Input1

Input1	See chapter 3.7.2, "Control Byte, Status Byte".
Input2	See chapter 3.7.2, "Control Byte, Status Byte".
Input3	See chapter 3.7.2, "Control Byte, Status Byte".
Input4	See chapter 3.7.2, "Control Byte, Status Byte".
Input5	See chapter 3.7.2, "Control Byte, Status Byte".
Input6	See chapter 3.7.2, "Control Byte, Status Byte".
SetupSpeed_Active_ ACK	See chapter 3.7.2, "Control Byte, Status Byte".
Reset	See chapter 3.7.2, "Control Byte, Status Byte".

3.11.1.4.2 Single Positioning Sequence Diagram

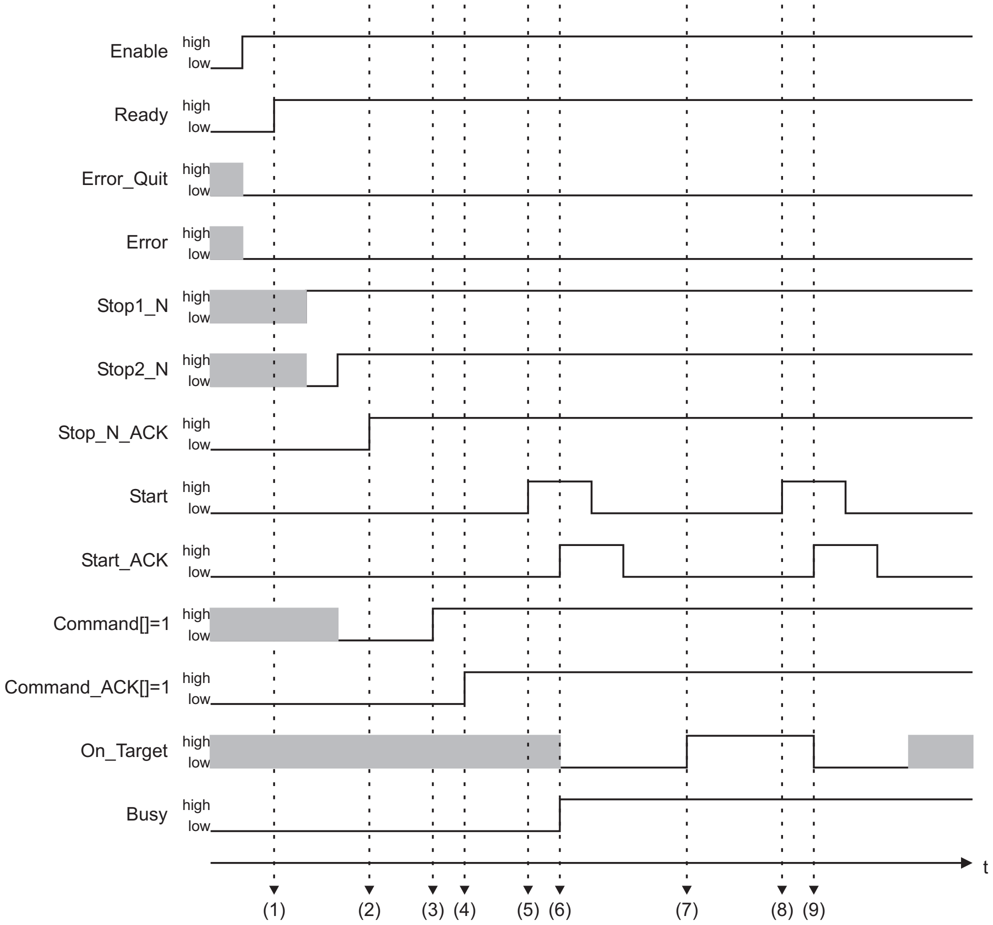


Fig. 3.11.1-2: Sequence diagram for single positioning g067x21x

(1)	The module is ready for operation when no error is present after setting of Enable.
(2)	An operating mode can be selected when Stop_N_ACK has been set.
(3)	Operating mode Single positioning is selected.
(4)	The single positioning mode has been accepted by the module.
(5)	The drive is started by the Start rising edge.
(6)	The setpoint from the process image is accepted and movement made toward the target position. Start can be canceled after Start_ACK has been set.
(7)	The drive has reached its target position.
(8)	The drive is restarted by the Start rising edge.
(9)	The current setpoint from the process image is accepted, the patch recalculated and movement made toward the target position, where applicable, on the fly. Start can be canceled after Start_ACK has been set.

3.11.1.5 Referencing Mode

The mode must first be activated using Command[]=4. The referencing mode is active when the Command_ACK[]=4 bit is set.

Furthermore, the Direction_Neg bit must be set if the reference motion is to be started in negative direction, or correspondingly, the Direction_Pos if the reference motion is to be started in positive direction. The direction for the reference switch and information detailing whether movement is to be made to the reference switch or a limit switch, specified by the Reference_Mode configuration parameter.

If a reference run is started via the mailbox with the move command START_REFERENCING, the referencing velocity, starting direction, moving direction for the reference switch and the information specifying whether movement is to be made toward a reference switch or limit switch, are transferreed as parameters.

The reference run is always performed at the setup speed SpeedSetup and at the setup acceleration SetupAcceleration.

Note
The referencing speed should be low in order to take the mechanical requirements into account (e.g., length of limit switch cams, residual travel after final shutdown, etc).

The reference run is initiated by the rising edge of the start. The drive starts up, or frequency is output, directly after the setpoints are accepted. The On_Target bit is canceled immediately and is not set again.

From the start to the end of the reference motion, the Busy bit is set.

If a limit switch is detected prior to reaching the reference switch, the reference motion is continued in the opposite direction. If prior to the detection of the reference switch a limit switch is detected again, the referencing process is terminated with an error message.

If the drive is at a limit switch it will not move further.
In Mailbox mode, the call of the command START_REFERENCING is acknowledged by error message 23.
No error message is issued in the Referencing mode. In this case, the ERR_RANGE_NEG, or ERR_RANGE_POS bit is set and can then be evaluated.

3.11.1.5.1 Referencing Process Image

The operating mode process image is shown in the following tables:

Control Byte C1							
Bit 7	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
Command					Start	Stop2_N	Enable

Enable See chapter 3.7.2, "Control Byte, Status Byte".

Stop2_N See chapter 3.7.2, "Control Byte, Status Byte".

Start Startup of drive.
The drive is started in the selected mode on a positive edge.
0→1 The drive is started accordingly on the rising edge.
The reference motion is initiated.
If the reference run is still in operation, the (new) setpoints are again accepted and calculated (same procedure as for positioning).
The reference run is newly initiated.

Command Selecting the operating mode

0: Idle mode
4: Reference motion

Status Byte S1							
Bit 7	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
Command_Ack					Start_ ACK	Stop_ N_ACK	Ready

Ready See chapter 3.7.2, "Control Byte, Status Byte".

Stop_N_ACK See chapter 3.7.2, "Control Byte, Status Byte".

Start_ACK Start sequence in the operating mode.
0: This bit is also set to 0 when the Start request is canceled.
1: The rising edge function is a function of the selected operating mode.
The reference run has been initiated.

Command_Ack Confirmation: selection of operating mode
0: Idle mode is selected
4: Referencing mode active.
The drive is started at the setup speed on the next rising edge for Start.

Control Byte C2							
Bit 7	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
Error_ Quit	PreCalc	X	X	Acc_Range_Sel		Freq_Range_Sel	

Freq_Range_Sel See chapter 3.7.2, "Control Byte, Status Byte".

Acc_Range_Sel See chapter 3.7.2, "Control Byte, Status Byte".

PreCalc See chapter 3.7.2, "Control Byte, Status Byte".

Error_Quit See chapter 3.7.2, "Control Byte, Status Byte".

X Reserved

Status Byte S2							
Bit 7	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
Error	PreCalc_ ACK	Referenc e_OK	Direction	On_ Speed	StandStill	Busy	On_ Target

On_Target	Target reached.
The significance of this bit depends on the selected operating mode.
0: A new mode will be selected, or a movement made to a new position.
1: The reference point has been moved to and set successfully.

Busy	Run task is executed and setpoint is not yet reached.
The selected mode is active and a task has been started;
the drive is rotating, or frequency output is not equal to 0.
0: No run task executed or setpoint has been reached.
1: Movement made toward reference point.

StandStill	See chapter 3.7.2, "Control Byte, Status Byte".

On_Speed	Drive speed reached
On_Speed is set by individual commands in mailbox mode and in the run program.
0: The drive has not reached its setpoint speed.
1: The drive has reached its setpoint speed.
SetupSpeed has been reached.
The tolerance lies within the TargetWindowSpeed target window.

Direction	See chapter 3.7.2, "Control Byte, Status Byte".

Reference_OK	Referencing OK.
Set when reference run has been successfully concluded.
0: Reference run initiated or no valid reference.
1: The reference point has been successfully located in the reference run mode.

PreCalc_ACK	See chapter 3.7.2, "Control Byte, Status Byte".

Error	See chapter 3.7.2, "Control Byte, Status Byte".

Control Byte C3							
Bit 7	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
Reset_ Quit	0	0	0	Direction _Neg	Direction _Pos	0	SetActual _Pos

SetActual_Pos	At the rising edge, the RefOffsetPos reference position is transferred from the configuration table into the ActualPosition of the process image.
Because no handshake bit is present, the bit must be set for a minimum of 20 ms.
This function is only possible in standstill.
If the bit is set during operation, the actual value remains unchanged and no error is reported.
0: The RefOffsetPos reference position is not transferred
1: The RefOffsetPos reference position is transferred at the positive edge.

Direction_Pos	In the Reference run mode this bit defines that the reference switch be searched for in a positive direction.
0: Drive not to move in a positive direction.
1: Drive should move in a positive direction.
The drive is deactivated when the bit Direction_Neg is set at the same time.

Direction_Neg	In the Reference run mode this bit defines that the reference switch be searched for in a negative direction.
0: Drive not to move in a negative direction.
1: Drive should move in a negative direction.
The drive is deactivated when the bit Direction_Pos is set at the same time.
0: Limiting not active
1: Limiting active

Reset_Quit	See chapter 3.7.2, "Control Byte, Status Byte".

0	Reserved

3.11.1.5.2 Sequence Diagram for Referencing

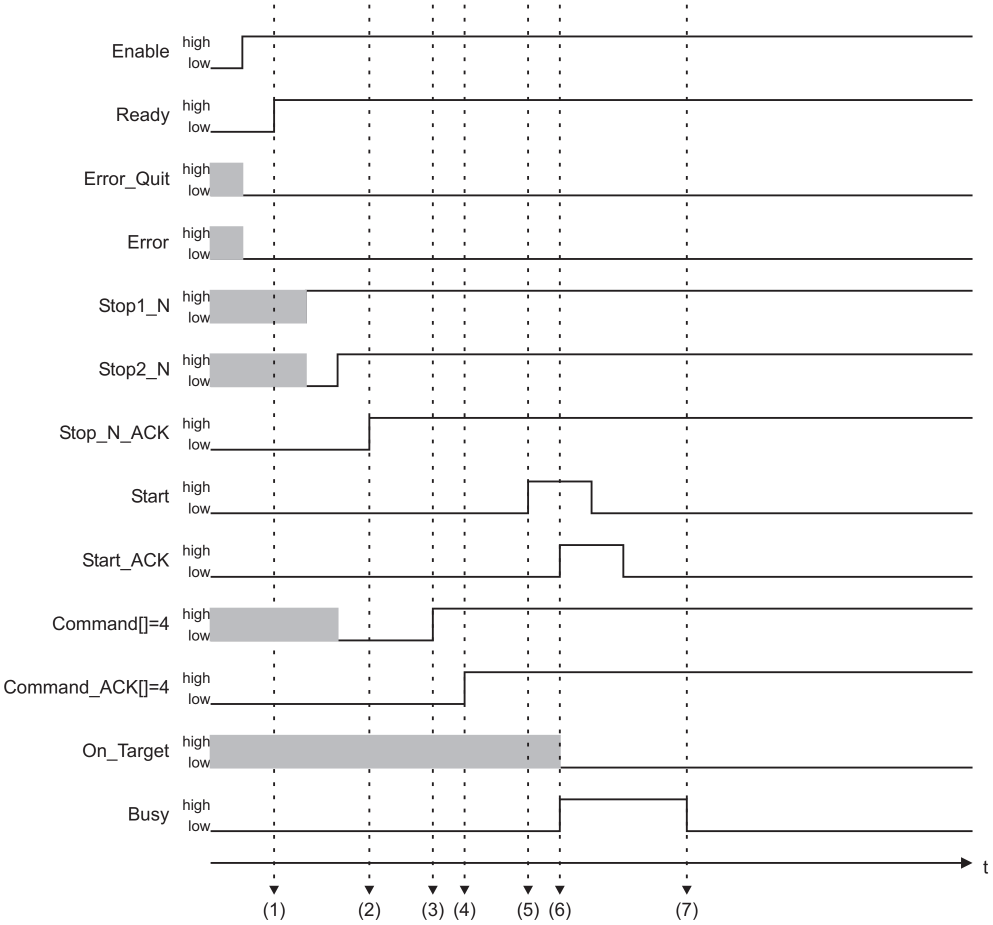


Fig. 3.11.1-3: Sequence diagram for referencing g067x23x

(1)	The module is ready for operation when no error is present after setting of Enable.
(2)	An operating mode can be selected when Stop_N_ACK has been set.
(3)	Referencing mode is selected.
(4)	The Referencing mode has been accepted by the module.
(5)	The drive is started by the Start rising edge.
(6)	The reference motion is initiated. Start can be canceled if Start_ACK has been set.
(7)	The reference point has been moved to and set.

3.11.1.5.3 Start Parameters for Referencing Mode

Referencing to positive end of reference switch, starting in negative direction

Operating mode				**Note**
Referencing, M_Reference = 1		**Mailbox, Command START_ REFERENCING**		
Reference_Mode , Bit 0	0	Parameter 3 Bit 0	0	Referencing to reference switch
Direction_Pos	0	Parameter 3 Bit 1	0	Start in negative direction
Direction_Neg	1			
Reference_Mode , Bit 1	1	Parameter 3 Bit 2	1	Referencing to positive end

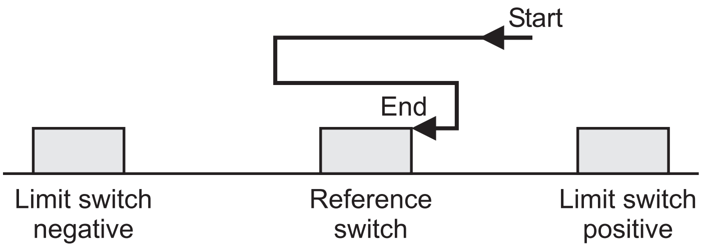


Fig. 3.11.1-4: Referencing to positive end of reference switch, with start in negative direction from positive movement range g067x10e

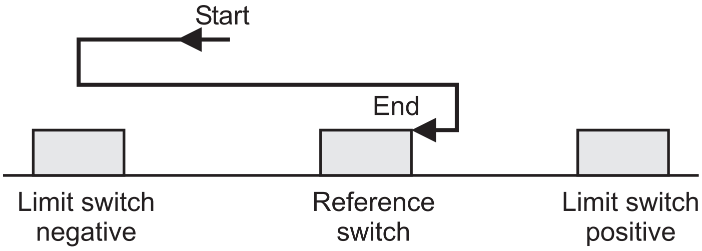


Fig. 3.11.1-5: Referencing to positive end of reference switch, with start in negative direction from negative movement range g067x14e

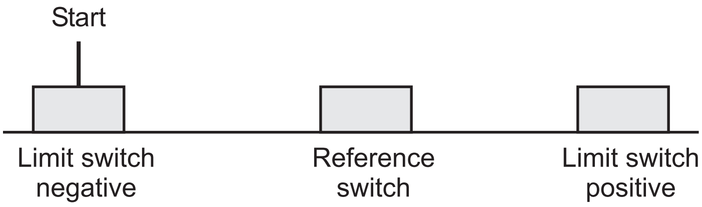


Fig. 3.11.1-6: Referencing to positive end of reference switch with start in negative direction from limit switch g067x13d

Note
The drive does not begin on a start from the limit switch. In the Mailbox mode the error message 23 is generated on calling of command START_REFERENCING. No error message is generated in the Referencing mode. The Bit ERR_RANGE_POS or ERR_RANGE_NEG bit is set!

Referencing to negative end of reference switch, starting in negative direction

Operating mode				
Referencing, M_Reference = 1		**Mailbox, Command START_ REFERENCING**		**Note**
Reference_Mode , Bit 0	0	Parameter 3 Bit 0	0	Referencing to reference switch
Direction_Pos	0	Parameter 3 Bit 1	0	Start in negative direction
Direction_Neg	1			
Reference_Mode , Bit 1	0	Parameter 3 Bit 2	0	Referencing to negative end

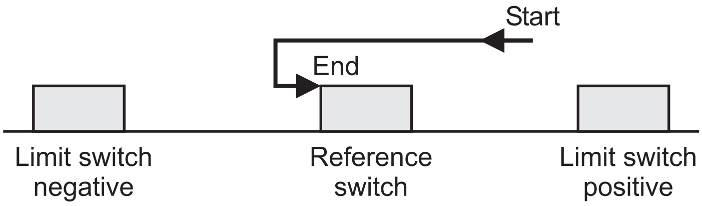


Fig. 3.11.1-7: Referencing at the negative end of reference switch with start in negative direction from positive movement range g067x11e

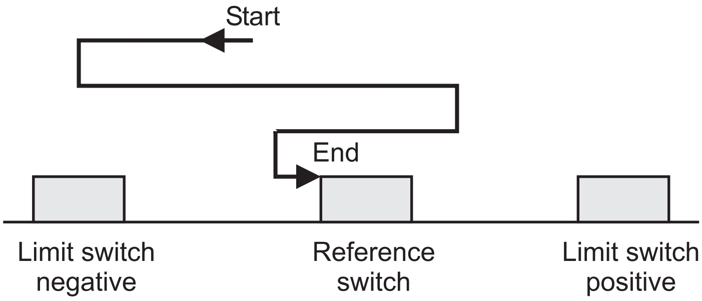


Fig. 3.11.1-8: Referencing at the negative end of reference switch with start in negative direction from negative movement range g067x12e

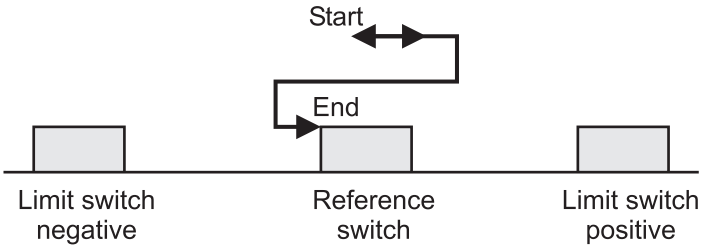


Fig. 3.11.1-9: Referencing at negative end of reference switch with start in negative direction from reference switch
g067x15e

Note
On a start from the reference switch, the module first reverses the starting direction to ensure that the reference switch is free.

Referencing to positive end of reference switch, starting in positive direction

Operating mode				**Note**
Referencing, M_Reference = 1		**Mailbox, Command START_ REFERENCING**		
Reference_Mode , Bit 0	0	Parameter 3 Bit 0	0	Referencing to reference switch
Direction_Pos	1	Parameter 3 Bit 1	1	Start in positive direction
Direction_Neg	0			
Reference_Mode , Bit 1	1	Parameter 3 Bit 2	1	Referencing to positive end

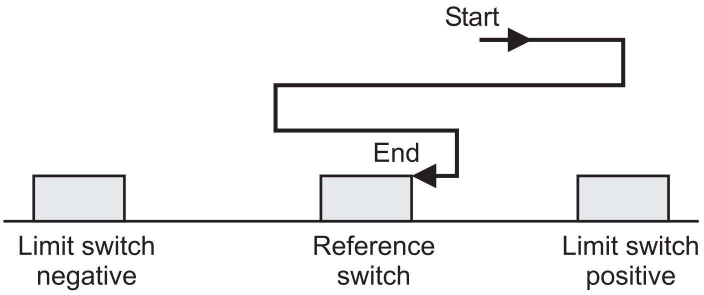


Fig. 3.11.1-10: Referencing at positive end of reference switch with start in positive direction from positive movement range
g067x16e

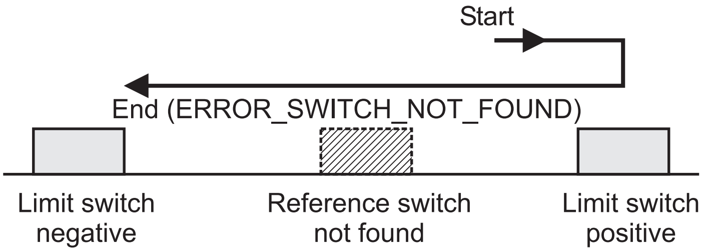


Fig. 3.11.1-11: Referencing at positive end of reference switch with start in positive direction from positive movement range; reference switch not found g067x38e

Referencing to limit switch starting in negative direction

Operating mode				**Note**
Referencing, M_Reference = 1		**Mailbox, Command START_ REFERENCING**		
Reference_Mode , Bit 0	1	Parameter 3 Bit 0	1	Referencing to limit switch
Direction_Pos	0	Parameter 3 Bit 1	0	Start in negative direction
Direction_Neg	1			
Reference_Mode , Bit 1	x			

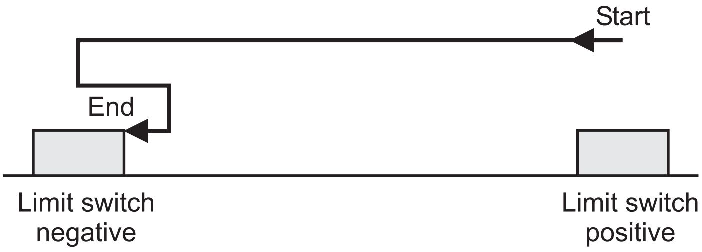


Fig. 3.11.1-12: Referencing to limit switch starting in negative direction from positive limit switch g067x39e

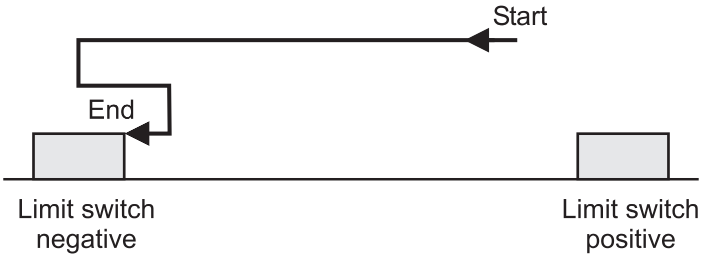


Fig. 3.11.1-13: Referencing to negative limit switch starting in negative direction from positive movement range g067x40e

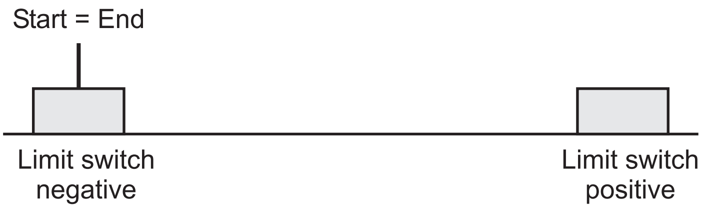


Fig. 3.11.1-14: Referencing to limit switch starting in negative direction from negative limit switch g067x41e

3.11.1.6 JogMode and SteppingMode

The drive can be run manually at the defined setup speed when the Jog mode is active. Control is implemented via Direction_Pos or Direction_Neg. The two control bits are locked against each other. The run is ended when the set timeout period (stepping mode) expires via the process image. A timeout of zero allows unlimited movement, as long as Direction_Pos or Direction_Neg is set (JogMode).

If the setup speed is parameterized as zero, the JogMode is run at speed 1.

If movement is made to a limit switch during the JogMode, the drive will stop. After that, movement can only be made away from the limit switch.

The drive stops when it leaves the movement range defined by the parameter Drive_Range_Neg and Drive_Range_Pos. The drive can then be operated outside the movement range using a repeated JOG command.

3.11.1.6.1 JogMode and SteppingMode Process Image

This process image is different from the standard configuration for stepper positioning control and is shown in the table below.

Off-set	Input data		Output data	
0	S0	Status byte S0	C0	Control byte C0
1	Reserved		Reserved	
2	D0	Current velocity (LSB)	D0	Reserved
3	D1	Current velocity (MSB)	D1	Reserved
4	D2	Reserved	D2	Timeout (LSB)
5	D3	Reserved	D3	Timeout (MSB)
6	D4	Current position (LSB)	D4	Reserved
7	D5	Current position	D5	Reserved
8	D6	Current position (MSB)	D6	Reserved
9	S3	Status byte S3	C3	Control byte C3
10	S2	Status byte S2	C2	Control byte C2
11	S1	Status byte S1	C1	Control byte C1

The meaning of the control and status bytes is given in the following tables:

Control Byte C1							
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Command					Start	Stop2_N	Enable

Enable See chapter 3.7.2, "Control Byte, Status Byte".
Stop2_N See chapter 3.7.2, "Control Byte, Status Byte".
Start Startup of drive.
The drive is started in the selected mode on a positive edge.
An error message is generated if the edge is not accepted.
0→1 The drive is started accordingly on the rising edge.
No effect.
The drive is started in JogMode by default via Input3 (Jog_Pos) or Input4 (Jog_Neg).
The inputs and outputs can be freely reconfigured. 4
An error message is generated.
Command Selecting the operating mode
0: Idle mode
8: JogMode
The drive can be operated manually at the setup speed.
Control is performed via Input3 (Jog_Pos) and Input4 (Jog_Neg)

Status Byte S1							
Bit 7	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
Command_Ack					Start_ ACK	Stop_N_ ACK	Ready

Ready See chapter 3.7.2, "Control Byte, Status Byte".
Stop_N_ACK See chapter 3.7.2, "Control Byte, Status Byte".
Start_ACK Start sequence in the operating mode.
0: This bit is also set to 0 when the Start request is canceled.
1: The rising edge function is a function of the selected operating mode.
No effect.
Handshake not performed.
Command_Ack Confirmation: selection of operating mode
0: Idle mode is selected
8: JogMode active

Control Byte C2							
Bit 7	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
Error_ Quit	PreCalc	X	X	Acc_Range_Sel		Freq_Range_Sel	

Freq_Range_Sel See chapter 3.7.2, "Control Byte, Status Byte".
Acc_Range_Sel See chapter 3.7.2, "Control Byte, Status Byte".
PreCalc Precalculation for movement sequence.
The setpoints are taken from the process image and, where required, a movement sequence precalculated.
This bit must be set to 0.
0: Each setpoint that is transmitted via cyclic telegram traffic must be accepted and processed. A possibly pre-calculated movement process will be discarded. A movement sequence can be calculated and started using Start.
1: The setpoints from the cyclic telegram traffic are ignored and the setpoint saved for the 0→1 edge used instead. If the starting speed is zero, a movement sequence will be calculated in advance using this setpoint; this sequence can then be started with the normal delay using Start.
Error_Quit See chapter 3.7.2, "Control Byte, Status Byte".
X Reserved

Status Byte S2							
Bit 7	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
Error	PreCalc_ ACK	Referenc e_OK	Direction	On_ Speed	StandStill	Busy	On_ Target

On_Target Target reached.
The significance of this bit depends on the selected operating mode.
0: A new mode will be selected, or a movement made to a new position.
The bit is not used in this mode and remains
at 0.

Busy Run task is executed and setpoint is not yet reached.
The selected mode is active and a task has been started;□
the drive is rotating, or frequency output is not equal to 0.
0: No run task executed or setpoint has been reached.
The drive has been started up using the pushbutton and is rotating.

StandStill See chapter 3.7.2, "Control Byte, Status Byte".

On_Speed Drive speed reached
On_Speed is set by individual commands in mailbox mode and in the run program.
0: The drive has not reached its setpoint speed.
1: The drive has reached its setpoint speed.
SetupSpeed has been reached.
The tolerance lies within the TargetWindowSpeed target window.

Direction See chapter 3.7.2, "Control Byte, Status Byte".

Reference_OK See chapter 3.7.2, "Control Byte, Status Byte".

PreCalc_ACK See chapter 3.7.2, "Control Byte, Status Byte".

Error See chapter 3.7.2, "Control Byte, Status Byte".

Control Byte C3							
Bit 7	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
Reset_ Quit	0	0	0	Direction _Neg	Direction _Pos	0	SetActual _Pos

SetActual_Pos See chapter 3.7.2, "Control Byte, Status Byte".

Direction_Pos See chapter 3.7.2, "Control Byte, Status Byte".

Direction_Neg See chapter 3.7.2, "Control Byte, Status Byte".

Reset_Quit See chapter 3.7.2, "Control Byte, Status Byte".

0 Reserved

Status Byte S3							
Bit 7	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
Reset	Warning	Input6	Input5	Input4	Input3	Input2	Input1

Input1 See chapter 3.7.2, "Control Byte, Status Byte".

Input2 See chapter 3.7.2, "Control Byte, Status Byte".

Input3 Status for Input 3. .
Input DI3 is used for JogMode in the default settings.
0: The drive shall not move in positive direction.
1: Drive should move in a positive direction.
If Input4 is set simultaneously, the drive is turned off.

Input4 Status for Input 4.
Input DI4 is used for JogMode in the default settings.
0: The drive shall not move in negative direction.
1: Drive should move in a negative direction.
If Input3 is set simultaneously, the drive is turned off.

Input5 See chapter 3.7.2, "Control Byte, Status Byte".

Input6 See chapter 3.7.2, "Control Byte, Status Byte".

Warning See chapter 3.7.2, "Control Byte, Status Byte".

Reset See chapter 3.7.2, "Control Byte, Status Byte".

X Reserved

3.11.1.6.2 JogMode and SteppingMode Sequence Diagram

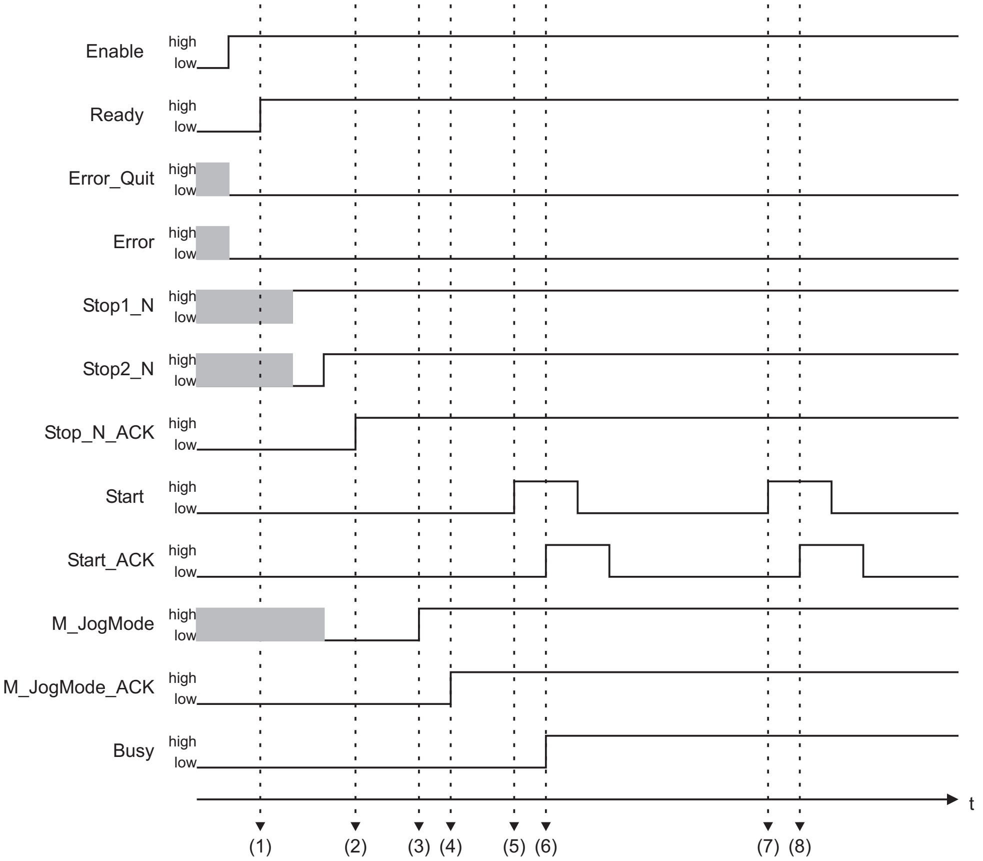


Fig. 3.11.1-15: Sequence diagram, JogMode g067x24x

(1)	The module is ready for operation when no error is present after setting of Enable.
(2)	An operating mode can be selected when Stop_N_ACK has been set.
(3)	The JogMode is selected.
(4)	The JogMode has been accepted by the module.
(5)	The drive is activated by the Start rising edge.
(6)	The Jog mode is activated, the drive can be started using the pushbutton Direction_Pos and Direction_Neg. Start can be canceled if Start_ACK has been set.
(7)	The drive is restarted by the Start rising edge.
(8)	The Jog mode is activated, the drive can be started using the pushbutton Direction_Pos and Direction_Neg. Start can be canceled if Start_ACK has been set.

3.11.1.7 Run Program Operating Mode

A Run program can be downloaded to the I/O module via the mailbox.

The available commands for this are given in the chapter 4.3, „Commands for Move Mode“.

First, the mode must be activated using M_Program. The Run program mode is active when the M_Program_ACK bit is set.

The Run program is started on a rising edge.

The address of the first command to be executed is transferred via the process image.

The Program_Running bit is set from the time of start to the end of the program. This bit can be queried using the mailbox command GET_BIT.

The On_Target and Busy bits are controlled by the individual program commands.

When the bit SetupSpeed_Active_ACK is set at the same time for setup, speed is limited to the defined setup speed.

3.11.1.7.1 Run Program Process Image

This process image is different from the standard configuration for stepper positioning control and is shown in the table below.

Off-set	Input data		Output data	
0	S0	Status byte S0	C0	Control byte C0
1	Reserved		Reserved	
2	D0	Current velocity (LSB)	D0	Reserved
3	D1	Current velocity (MSB)	D1	Reserved
4	D2	Current value for command counter (LSB)	D2	Starting value for command counter (LSB)
5	D3	Current value for command counter (MSB)	D3	Starting value for command counter (MSB)
6	D4	Current position (LSB)	D4	Reserved
7	D5	Current position	D5	Reserved
8	D6	Current position (MSB)	D6	Reserved
9	S3	Status byte S3	C3	Control byte C3
10	S2	Status byte S2	C2	Control byte C2
11	S1	Status byte S1	C1	Control byte C1

The meaning of the control and status bytes is given in the following tables:

Control Byte C1							
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Command					Start	Stop2_N	Enable

Enable — See chapter 3.7.2, "Control Byte, Status Byte".

Stop2_N — See chapter 3.7.2, "Control Byte, Status Byte".

Start — Startup of drive.
The drive is started in the selected mode on a positive edge.
0→1 The drive is started accordingly on the rising edge.
The run program is started at the Start_Instruction_Counter address in the process image.
A run program that is currently running is interrupted, and the program flow is started at the new address.
This allows various program parts to be transferred via the process image.

Command — Selecting the operating mode
0: Idle mode
2: Run program

Status Byte S1

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Command_Ack					Start_ ACK	Stop_N_ ACK	Ready

Ready	See chapter 3.7.2, "Control Byte, Status Byte".
Stop_N_ACK	See chapter 3.7.2, "Control Byte, Status Byte".
Start_ACK	Start sequence in the operating mode. 0: This bit is also set to 0 when the Start request is canceled. 1: The rising edge function is a function of the selected operating mode. Run program The run program has been started.
Command_Ack	Confirmation: selection of operating mode 0: Idle mode is selected 2: Run program mode active The Run program is started with the first command on the next rising edge for Start.

Control Byte C2

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Error_ Quit	PreCalc	X	X	Acc_Range_Sel		Freq_Range_Sel	

Freq_Range_Sel	See chapter 3.7.2, "Control Byte, Status Byte".
Acc_Range_Sel	See chapter 3.7.2, "Control Byte, Status Byte".
PreCalc	See chapter 3.7.2, "Control Byte, Status Byte".
Error_Quit	See chapter 3.7.2, "Control Byte, Status Byte".
X	Reserved

Status Byte S2

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Error	PreCalc_ ACK	Referenc e_OK	Direction	On_ Speed	StandStill	Busy	On_ Target

On_Target	Target reached. The significance of this bit depends on the selected operating mode. 0: A new mode will be selected, or a movement made to a new position. 1: The specified setpoint within the TargetWindowPosition target window has been reached.
Busy	Run task is executed and setpoint is not yet reached. The selected mode is active and a task has been started; the drive is rotating, or frequency output is not equal to 0. 0: No run task executed or setpoint has been reached. The current Run program is being executed.
StandStill	See chapter 3.7.2, "Control Byte, Status Byte".
On_Speed	Drive speed reached On_Speed is set by individual commands in mailbox mode and in the run program. 0: The drive has not reached its setpoint speed. 1: The drive has reached its setpoint speed. Specified speed from the process image has been reached. The tolerance lies within the TargetWindowSpeed target window.
Direction	See chapter 3.7.2, "Control Byte, Status Byte".
Reference_OK	See chapter 3.7.2, "Control Byte, Status Byte".
PreCalc_ACK	See chapter 3.7.2, "Control Byte, Status Byte".
Error	See chapter 3.7.2, "Control Byte, Status Byte".

Control Byte C3							
Bit 7	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
Reset_ Quit	0	0	0	Direction _Neg	Direction _Pos	0	SetActual _Pos

SetActual_Pos	See chapter 3.7.2, "Control Byte, Status Byte".
Direction_Pos	See chapter 3.7.2, "Control Byte, Status Byte".
Direction_Neg	See chapter 3.7.2, "Control Byte, Status Byte".
Reset_Quit	See chapter 3.7.2, "Control Byte, Status Byte".
0	Reserved

Status Byte S3							
Bit 7	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
Reset	Warning	Input6	Input5	Input4	Input3	Input2	Input1

Input1	See chapter 3.7.2, "Control Byte, Status Byte".
Input2	See chapter 3.7.2, "Control Byte, Status Byte".
Input3	See chapter 3.7.2, "Control Byte, Status Byte".
Input4	See chapter 3.7.2, "Control Byte, Status Byte".
Input5	Status for Input 5. Input DI5 is used as the limit switch input in the default settings. 0: The positive direction limit switch is not actuated 1: The positive direction limit switch is actuated. The drive is ramped down.
Input6	Status for Input 6.. Input DI6 is used as the limit switch input in the default settings. 0: The negative direction limit switch is not actuated 1: The negative direction limit switch is not actuated. The drive is ramped down.
Warning	A warning is issued when an adjustable limit is exceeded. The warning can be acknowledged using Error_Quit. 0: No warning present for the drive. 1: Warning present for the drive.
Reset	See chapter 3.7.2, "Control Byte, Status Byte".
X	Reserved

3.11.1.7.2 Sequence Diagram for Run Program

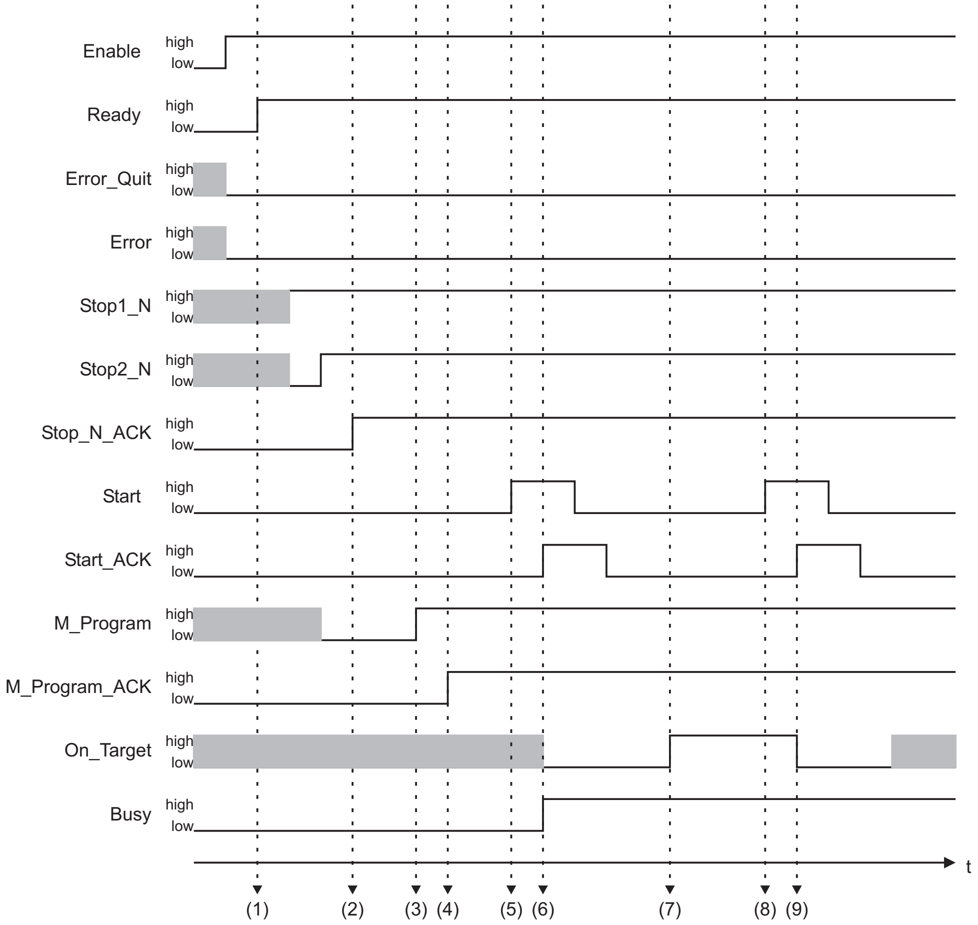


Fig. 3.11.1-16: Sequence diagram, Run program g067x22x

(1)	The module is ready for operation when no error is present after setting of Enable.
(2)	An operating mode can be selected when Stop_N_ACK has been set.
(3)	The Run program mode is selected.
(4)	The Run program mode has been accepted by the module.
(5)	The drive is started by the Start rising edge.
(6)	The Run program is started on the first command. Start can be canceled if Start_ACK has been set.
(7)	The current Run program has reached its last position.
(8)	The drive is restarted by the Start rising edge.
(9)	The Run program in progress will be terminated and the drive set to standstill. The Run program is then restarted on the first command. Start can be canceled if Start_ACK has been set.

3.11.1.7.3 Example of Run Program

In this example of a Move program, first the velocity (20000) and then the acceleration (3000) is set. After this, a "True" signal is anticipated at Input 1 on line 2. If the signal is "True", movement is made to Position 1 (specified position = 65065) and then the system waits for a "False" signal at input 1 on line 4. If the "False" signal is received, movement is made to Position 2 (specified position = 0). The system then returns to line 2 of the Move program, i.e. it waits for a "True" signal at Input 1.

Line in the run table	Opcode	Data (LSB)	Data	Data (MSB)	Meaning
	MB2	MB3	MB4	MB5	
0	0x25	0x20	0x4E	0x00	Set velocity to 20000
1	0x22	0xB8	0x0B	0x00	Set acceleration to 3000
2	0x71	0x30	0x01	0x00	Query, Input 1 = "1"
3	0x02	0x20	0xFE	0x00	Move to position 65065
4	0x71	0x30	0x00	0x00	Query, Input 1 = "0"
5	0x02	0x00	0x00	0x00	Move to position 0
6	0xF5	0x02	0x00	0x00	Go to line 2
7	0x70	0xD0	0x07	0x00	Wait 2000 ms

3.11.1.7.4 Autostart

An existing Run program is started automatically after a module reset when the configuration bit HwSwConfig.Program_AutoStart (Bit 7) is set.

Prerequisites for successful automatic start:

- Valid Run program present in EEPROM,
- The HwSwConfig.Program_AutoStart bit is set in the configuration,
- The Rest status bit is set and has not yet been canceled,
- The Ready status bit is set,
- The Stop_N_ACK status bit is set.

A Run program started by HwSwConfig.Program_AutoStart can be ended by canceling Reset. The bit is reset by Reset_Quit.

A Run program is started only one time after each reset by HwSwConfig.Program_AutoStart.

A Run program started with HwSwConfig.Program_AutoStart is always started at an address of 0.

3.11.1.8 Rotary Axis Operating Mode

The "Rotary axis" function is activated by the parameter Rotary_Axis_Period being written with a value other than zero. If the Rotary_Axis_Period parameter is zero, a linear and limited movement range is assumed.

The position is repeated with a rotary axis every 2π or 360°. The Rotary_Axis_Period parameter indicates how many motor steps correspond to one rotation around the axis by 2π or 360.

The actual value for rotary axis is always within the range 0 … Rotary_Axis_Period. This ensures that no internal overrun occurs with relative motion repeated any number of times.

Parameter	Linear axis	Rotary axis	
Rotary_Axis_Period	0	Micro steps per rotation > 0	
Operating range	Drive_Range_Neg … Drive_Range_Pos	Limited to Drive_Range_Neg … Drive_Range_Pos when Drive_Range_Neg >=0 or Drive_Range_Pos < Rotary_Axis_Period, otherwise unrestricted	
Actual value	Operating range	0 … Rotary_Axis_Period, periodic	
Set value	Operating range	Absolute Positioning: 0 … Rotary_Axis_Period	Relative Positioning: -8388607–8388607. Can be repeated any number of times. No internal overrun

When specifying the parameters for the rotary axis, the following restriction must be observed for the function to be implemented correctly.

$$v \le \frac{Speed\_Div}{Speed\_Mult} * \frac{Freq\_Div}{80} * 10^3 * p$$

v	Maximum allowed setpoint speed
p	Rotary axis periods (parameter 64, Rotary_Axis_Period)
Speed_Mult	Scaling factor for setpoint speed (parameter 28 from configuration table)
Speed_Div	Scaling factor for setpoint speed (parameter 30 from configuration table)
Freq_Div	Prescaler for maximum speed (parameter 4 from configuration table)

The setpoint definition principally differentiates between absolute and relative positioning.

3.11.1.8.1 Relative Positioning

The target position is added to the current position for relative positioning. In the calculation, the operating mode "Round axis" is ignored at first, hence the "virtual target" may be outside the range 0 ... 2π. This allows relative positioning to be performed over several revolutions. The actual value, however, is reported only within the range 0 … 2π; the number of completed revolutions can not be determined.

The motion direction depends on the qualifying symbol of the relative setpoint value.

3.11.1.8.2 Absolute Positioning

The target position is always within the range 0 … 2π for absolute positioning. A setpoint defined outside of this range will result in an error.

On a movement task from standstill, the system determines in what direction the target can be reached in the shortest time.

At a positioning motion with a start speed or a target speed, the motion direction is selected requiring none or the least number of direction changes.

Absolute positioning permits braking at a certain velocity setting to a precisely defined spot (e.g. coil end that is to be stopped exactly at an attitude angle of 0 from full speed, only possible via mailbox!).

3.11.1.9 Camshaft Operating Mode

The camshaft provides pulses as a function of position for nine (9) channels CAM1 … CAM9. Up to 50 switching positions can be freely defined through channels 1 … 8. Channel CAM9, on the other hand, supplies a periodic signal as a function of position.

Parameterization of channels 1 ... 8 is performed using a table containing 50 entries. Each entry consists of a position xp (24-bit) and a bit sample (8-bit). The bit samples each describe the position of the eight output channels CAM1 … CAM8, which is valid starting from the assigned position xp up to the next larger position entry xp+1.

The table entries are sorted in ascending order of the positions.

For positions below the first entry, the bit pattern of the first entry is shown.

The starting position, the cam width and the repeat period are assigned to channel 9 via the configuration.

Except for the reset condition, the activated camshaft is always active independent from the motion operation.

Contrary to other setpoint values, the switching positions of the camshaft are always shown in the unit "micro step". A conversion to or from user-specific units has not been projected.

The camshaft table cannot be edited in the terminal, but must be downloaded as a whole.

Example: Eight (8) entries are to be loaded to the camshaft table.

Step No.	**CAM**								**Position**			**Outputs**
	8	7	6	5	4	3	2	1	Byte 4 MB5	Byte 3 MB4	Byte 2 MB3	Byte 1 MB 3
1	1	0	0	0	0	1	0	0	0x00	0x2e	0xe0	0x84
2	0	1	0	0	0	1	0	0	0x00	0x5d	0xc0	0x44
3	0	0	1	0	0	0	1	0	0x00	0xbb	0x80	0x22
4	0	0	0	1	0	0	1	0	0x01	0x77	0x00	0x12
5	0	0	0	0	1	0	0	0	0x01	0xd4	0xc0	0x08
6	0	1	0	0	0	1	0	0	0x02	0x90	0x40	0x44
7	0	1	0	0	0	0	1	0	0x07	0xb0	0xc0	0x42
8	0	1	0	0	0	0	0	1	0x07	0xdf	0xa0	0x41

The grey-shaded part of the table must be downloaded into the terminal.

In addition, the checksum for the table must be calculated. For this, the sum across all bytes is determined. Here, the checksum is 0xb0d.

Opcodes given in chapter 4.2.4.4, „Table Management Commands“ are required for downloading.

The corresponding Opcodes are adjusted for this case in the table below. The complete scope and significance are described in the chapter indicated above.

3.11.1.10 Position Table Operating Mode

The position table permits a fixed run motion process with variable positions.

The position table stores up to 50 positions which can be invoked by motion commands.. The position table entries can be evaluated by absolute or by relative motion commands.

The position table can be downloaded. As an alternative to this, individual entries can be written or "taught" as absolute or relative positions with the current position.

3.11.1.10.1 Teaching of Positions

Teaching of positions using the mailbox command POS_TABLE_TEACH allows the current actual value to the saved in the position table so that it is available as a target for a Move task.

The current actual value can be saved as the reference point for relative travel measurement.

The soft limit switches Drive_Range_Pos and Drive_Range_Neg can also be taught.

3.11.1.11 Speed Control Operating Mode

The Frequency/Speed Control application represents a variant of stepper control. The basic functions for stepper control can still be utilized. The basic difference is the modified process image, which permits specification of frequency and speed. Speed corresponds directly to the output frequency and the frequency ramps to acceleration.

In this module, the stepper motor is operated with the default rotational speed.

The frequency/speed control system generates a definable motor rotational speed. This rotational speed can be preset via process image, the JOG mode, motion command via mailbox and by program operation, as are single positioning / stepper control.

Alternatively, the Frequency/Speed Control application is also feasible with the function "Rotary axis". If the rotary shaft configuration value does not have a parameter of zero, a value of 100000 is assumed.

The Frequency/Rotational Speed Control application is selected by selecting the Command[]=3 operating image in the cyclic process image.

In Frequency Speed Control, the position recognition operates in the background. To prevent this from triggering a shutdown via the soft limit switches with Drive_Range_Pos or Drive_Range_Neg, the parameters Rotary_Axis_Period can be used and parameters provided or simulated for a rotary axis. If parameters have been provided for the value Rotary_Axis_Period, this value is used for the rotary axis.

The Drive_Range_Neg and Drive_Range_Pos software limit switches are also active in rotary axis mode.

If no movement range limit is desired, these parameters must lie outside the rotary axis range defined by the Rotary_Axis_Period parameter.

Step positioning/Stepper control contains a detailed description of this (selection of mode and accepting of setpoints with Start).

The Frequency/Speed Control application essentially influences speed interpretation in the process image and the evaluation of Rotary_Axis_Period. Otherwise, the complete functions of other modes, such as JogMode, referencing, run task via mailbox and program mode can be utilized.

3.11.1.11.1 Velocity Control Process Image

This process image is different from the standard configuration for stepper positioning control and is shown in the tables below.

Off-set	Input data		Output data	
0	S0	Status byte S0	C0	Control byte C0
1	Reserved		Reserved	
2	D0	Actual Velocity L	D0	Velocity L
3	D1	Actual Velocity H	D1	Velocity H
4	D2	Reserved	D2	Acceleration L
5	D3	Reserved	D3	Acceleration H
6	D4	Actual position L	D4	Reserved
7	D5	Actual position M	D5	Reserved
8	D6	Actual position H	D6	Reserved
9	S3	Status byte S3	C3	Control byte C3
10	S2	Status byte S2	C2	Control byte C2
11	S1	Status byte S1	C1	Control byte C1

The function of the bits in control bytes C1 ... C3 and status bytes S1 ... S3 is determined by the Frequency/Speed Control application. When switchover is made to this application, the linked locations for the old application are retained.

The meaning of the control and status bytes is given in the following tables:

Control Byte C1							
Bit 7	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
Command					Start	Stop2_N	Enable

Enable — See chapter 3.7.2, "Control Byte, Status Byte".

Stop2_N — See chapter 3.7.2, "Control Byte, Status Byte".

Start — Startup of drive.
The drive is started in the selected mode on a positive edge. If the edge is not accepted (in the Jog or Mailbox mode), an error message is generated.
0→1: The drive is started accordingly on the rising edge.
The specified setpoints have been accepted from the process image.
Movement is made directly to the new target, even if the drive is already turning.
A previously calculated movement sequence is started immediately when the PreCalc_ACK bit is set (instant setpoint switch).

Command — Selecting the operating mode.
3: Speed Control

Status Byte S1							
Bit 7	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
Command_Ack					Start_ ACK	Stop_N_ ACK	Ready

Ready See chapter 3.7.2, "Control Byte, Status Byte".
Stop_N_ACK See chapter 3.7.2, "Control Byte, Status Byte".
Start_ACK Start sequence in the operating mode.

0: This bit is also set to 0 when the Start request is canceled.
1: The rising edge function is a function of the selected operating mode. The specified setpoints have been accepted from the process image. The drive has been started.

Command_Ack Confirmation: selection of operating mode

0: Idle mode is selected
3: Speed control mode active. Movement is made to the active setpoint on the next rising edge for Start.

Control Byte C2							
Bit 7	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
Error_ Quit	PreCalc	0	0	Acc_Range_Sel		Freq_Range_Sel	

Freq_Range_Sel See chapter 3.7.2, "Control Byte, Status Byte".
Acc_Range_Sel See chapter 3.7.2, "Control Byte, Status Byte".
PreCalc Precalculation for movement sequence.
The setpoints are taken from the process image and, where required, a movement sequence precalculated. This bit must only be used in the speed control mode.
In the other operating modes, the bit must be set to 0.

0: Each setpoint that is transmitted via cyclic telegram traffic must be accepted and processed. A possibly pre-calculated movement process will be discarded. A movement sequence can be calculated and started using Start.
1: The setpoints from the cyclic telegram traffic are ignored and the setpoint saved for the 0→1 edge used instead. If the starting speed is zero, a movement sequence will be calculated in advance using this setpoint; this sequence can then be started with the normal delay using Start.

Error_Quit See chapter 3.7.2, "Control Byte, Status Byte".
0 Reserved

Status Byte S2							
Bit 7	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
Error	PreCalc_ACK	Reference_OK	Direction	On_Speed	StandStill	Busy	On_Target

On_Target Target reached.
The significance of this bit depends on the selected operating mode.
0: A new mode will be selected, or a movement made to a new position.
X The bit On_Target has no function in this operating mode

Busy Run task is executed and setpoint is not yet reached.
The selected mode is active and a task has been started;□
the drive is rotating, or frequency output is not equal to 0.
0: No run task executed or setpoint has been reached.
1: Specified speed not yet reached.

StandStill See chapter 3.7.2, "Control Byte, Status Byte".

On_Speed Drive speed reached.
On_Speed is set by individual commands in mailbox mode and in the run program.
0: The drive has not reached its setpoint speed.
1: Specified speed from the process image has been reached.
The tolerance lies within the TargetWindowSpeed target window.

Direction See chapter 3.7.2, "Control Byte, Status Byte".
Reference_OK See chapter 3.7.2, "Control Byte, Status Byte".
PreCalc_ACK See chapter 3.7.2, "Control Byte, Status Byte".
Error See chapter 3.7.2, "Control Byte, Status Byte".

Control Byte C3							
Bit 7	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
Reset_Quit	0	0	0	Direction_Neg	Direction_Pos	0	SetActual_Pos

SetActual_Pos See chapter 3.7.2, "Control Byte, Status Byte".
Direction_Pos See chapter 3.7.2, "Control Byte, Status Byte".
Direction_Neg See chapter 3.7.2, "Control Byte, Status Byte".
Reset_Quit See chapter 3.7.2, "Control Byte, Status Byte".
0 Reserved

Status Byte S3							
Bit 7	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
Reset	Warning	Input6	Input5	Input4	Input3	Input2	Input1

Input1 See chapter 3.7.2, "Control Byte, Status Byte".
Input2 See chapter 3.7.2, "Control Byte, Status Byte".
Input3 See chapter 3.7.2, "Control Byte, Status Byte".
Input4 See chapter 3.7.2, "Control Byte, Status Byte".
Input5 See chapter 3.7.2, "Control Byte, Status Byte".
Input6 See chapter 3.7.2, "Control Byte, Status Byte".
Warning See chapter 3.7.2, "Control Byte, Status Byte".
Reset See chapter 3.7.2, "Control Byte, Status Byte".

3.11.2 Move Mode via Mailbox

The mailbox must first be displayed. This is described in chapter 3.8, „Mailbox Operation".

After that, the Move commands via mailbox mode must be activated. This is accomplished by setting bit 7 in the control byte C1.

Only then, the motion commands can be given.

3.11.2.1 Run Commands

The module can be operated with the mailbox by using the move commands. Movement can be made directly to different positions. This command is accepted only when the mode Move mode via mailbox has been activated.

Available commands for run mode are can be found in chapter 4.3, "Commands for Move Mode".

3.11.3 Limitation of Moving Range

3.11.3.1 Hardware, Limit Switch

The hardware limit switches are active in the Positioning and Velocity control applications. These devices limit the movement path.

Any allocation of limits switches to the direction of movement must be maintained. The negative hardware limit switch (LimitSwitch_Neg bit) is linked to Input6 by default and restricts the range to smaller positions; i.e. in the negative direction.
The positive hardware limit switch (LimitSwitch_Pos bit) is linked to Input5 by default and restricts the range to larger positions; i.e. in the positive direction.

Positioning

If movement is made to a limit switch in the Positioning mode, the drive will brake the movement until standstill using the defined deceleration Acceleratio_Stop_Fast.
The drive can only be started in the Jog and Referencing modes when it is located at a limit switch.

JogMode

If movement is made to a limit switch in the JogMode, the drive will brake the movement until standstill using the defined deceleration Acceleratio_Stop_Fast.
The drive can then be moved away from the limit switch by pressing the "Jog" button Direction_Neg or Direction_Pos again, from the positive limit switch in a negative direction and from the negative limit switch in a positive direction.
The drive will again decelerate to a standstill using the defined deceleration Acceleration_Stop_Fast as soon as it moves away from the limit switch. The drive is then no longer located at the limit switch and can be run in any mode without any restrictions.

Referencing

Warning
The software limit switch does not evaluate limiting of the moving range during the reference run. This may result in damage to the system if proper functioning of the hardware limit switch is not ensured!

If the drive is located at a limit switch in the Referencing mode to a reference switch, it can only be started in the Jog or Referencing mode. Only a negative direction of movement is possible from the positive limit switch and vice versa.

If a reference run has been made to a limit switch in the Referencing mode, the drive will end up at the limit switch and a special operating mode will be activated.
In this special mode, the drive can be moved away from the limit switch in any mode, with the positive limit switch only permitting movement in a negative direction and the negative limit switch in a positive direction. The special mode is terminated 100 ms after the drive leaves the limit switch.

3.11.3.2 Software Limit Switch

The permissible movement range of the drive is limited by the hardware limit switch. Options are also available, however, for restricting the permissible movement range using limits that can be parameterized (software limit switches), for example if no hardware limit switches are available.

Note
Evaluation of the hardware limit switches has priority over evaluation of the software limit switches.

The software limit switches are defined by the limits Drive_Range_Neg and Drive_Range_Pos in the Configuration table. The limit Drive_Range_Neg restricts the range to smaller positions; i.e., in the negative direction. Drive_Range_Pos restricts the range to larger positions; i.e., in the positive direction.

The software limit switches are only active in the Positioning and Run program via mailbox modes, as well as some subfunctions being available in the JogMode. The switches are not evaluated in other modes.

Warning
The software limit switch does not evaluate limiting of the moving range during a reference run. This may result in damage to the system if proper functioning of the hardware limit switch is not ensured!

The limits Drive_Range_Neg and Drive_Range_Pos define the permissible range of movement. If one of these defined limits is violated, the associated bit is set.
The default setting is Drive_Range_Neg = 0x800001 and Drive_Range_Pos = 0x7FFFFF.

If movement is made beyond a defined movement range, the drive is brought to standstill using the defined deceleration Acceleration_Stop_Fast and, after that, only those directions of movement accepted that move the unit back into the permissible range; the exception here is the JogMode.

In the JogMode the drive is brought to a standstill each time it attempts to move out of the permissible range. In this mode the drive can also be operated outside the movement range defined by the software limit switches with repeated JOG commands. The software limit switches are not active again until the drive is back within the defined range.

3.11.4 Control of a Motor Brake

Control of the motor brake is conducted using the Brake bit (see chapter 4.5, „Bit Field for I/O Driver“).

Control of this bit is performed from two sources that are connected with an OR link.
On the one hand, the brake bit is set automatically as soon as the drive is running and is canceled as soon as the drive is at standstill. Automatic control is configured using the configuration parameters Braketime_Turn_On and Braketime_Turn_Off. The Brake bit is then activated directly after the start of a run command. If the bit has not been set, execution of the move command will, however, be delayed by the Braketime_Turn_On time. The configuration parameter Braketime_Turn_Off defines the deactivation time for the Brake bit. This bit is deactivated before the target is reached by the Braketime_Turn_Off time. The brake can be controlled directly with this bit. The brake is released when the bit is set, and is applied when the bit is canceled.

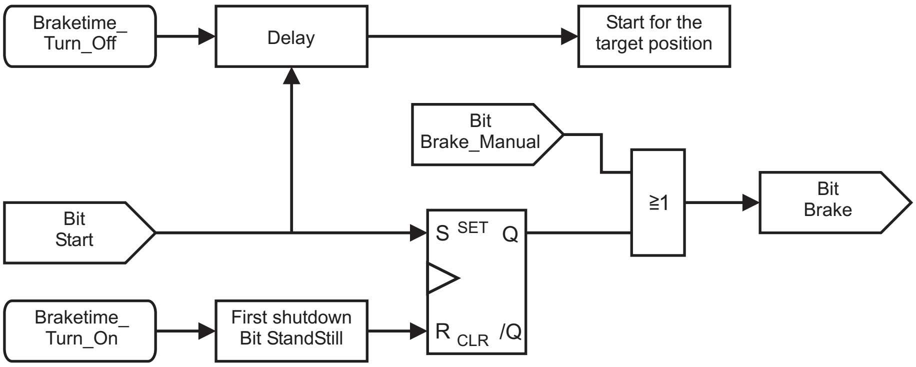


Fig. 3.11.4-1: Control of motor brake g067x17e

As an alternative, the brake can also be operated independently using the Brake_Manual bit (see chapter 4.5, „Bit Field for I/O Driver“). This bit can be set and canceled externally and can also, for example, be linked to a camshaft channel so that it is switched as a function of position.

3.12 Diagnostics and Errors

The diagnostics commands allow internal module information to be accessed. This includes:

- Error status of the device,
- Variables and status bits,
- Password,
- Configuration table and
- Position table

3.12.1 Error Messages and Evaluation

Depending on the configuration in synchronous process image, errors and warnings are indicated by the following common error bits:

-	Error (S2.7)	(see chapter 3.7.2,"Control Byte, Status Byte")
-	Error_Quit (C2.7)	(see chapter 3.7.2,"Control Byte, Status Byte")
-	ERR (S0.6)	(see chapter 3.7.2,"Control Byte, Status Byte")

The module has an error memory that can record exactly one error code. The simultaneous occurrence of multiple errors is not supported.

To determine the cause of the error, the error code can be queried via the mailbox. The following commands are available:

Function	**Opcode**	**Meaning**	**Page**
Diagnostics Commands			
DIAG_RD_ERROR	0x49	Error information is recovered from the error memory.	169
DIAG_QUIT_ERROR	0x4A	Terminates a device error condition	170

The commands are described in the appendix in chapter 4.2.4.5, "Diagnostics Commands".

3.12.2 Error Acknowledgement

The errors and warnings of all subsystems are analyzed at a central location.

The occurrence of errors and warnings is recorded and leads the device into an error condition.

In the case of error messages, the output stage is terminated immediately. The motor has no torque in this state and is at risk of uncontrolled movements, which can be prevented with an external brake.

The error or warning condition continues until the cause is remedied.

Note
The bit Error (S2.7) follows statically bit Error_Quit (C2.7). If the cause of the error is not eliminated, the bit Error_Quit is set again!

If no more errors or warnings are reported by the hardware, the error or warning condition can be acknowledged with the positive edge by the control bit Error_Quit or with a mailbox command DIAG_RD_ERROR.

3.12.3 Internal Data Bus Parameterization: Accumulative Diagnostics

Errors/warnings can also be reported in status byte S0, bit 6 (ERR) and thus initiate an acyclic diagnostics message (e.g., via Profibus).
This bit is addressed continuously as long as the error is present and Error_Quite is not set.
Enabling for the indication of errors is set using the configuration parameter ErrorNotificationMode.SystemFlagEnable (parameter 113, bit 0). This bit has the following meaning:

0: Errors are not reported via status byte S0, bit 6 (ERR).

1: Errors are reported via status byte S0, bit 6 (ERR).

Warning displays are enabled using the configuration parameter WarningNotificationMode.SystemFlagEnable (parameter 116, bit 0).

This bit has the following meaning:

0: Warnings are not reported via status byte S0, bit 6 (ERR).
1: Warnings are reported via status byte S0, bit 6 (ERR).

3.12.4 Sequence Diagram for Troubleshooting

3.12.4.1 Cause of error was eliminated before acknowledgement

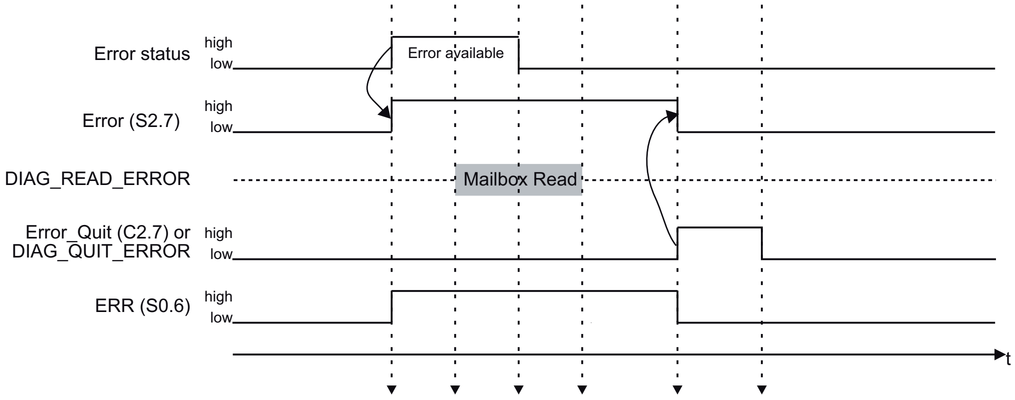


Fig. 3.12.4-1: Sequence diagram for troubleshooting 1 g067225e

3.12.4.2 Cause of error was not eliminated before acknowledgement

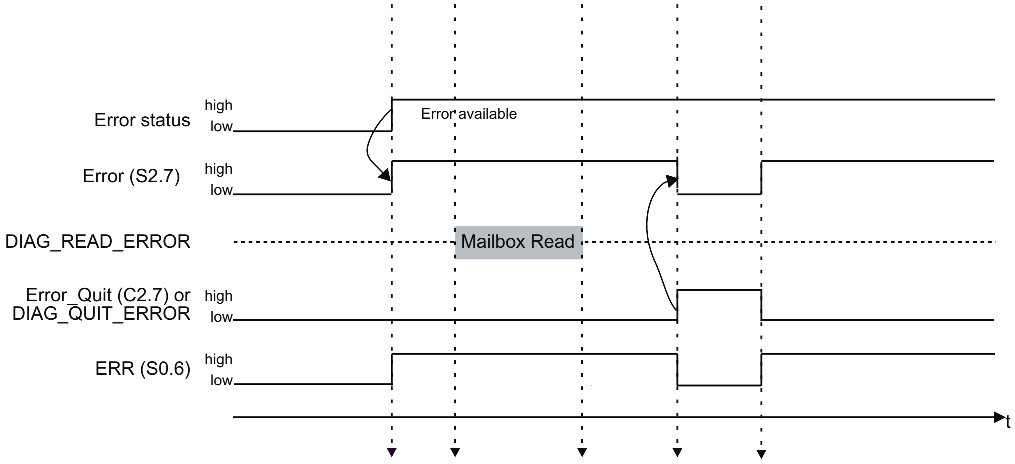


Fig. 3.12.4-2: Sequence diagram for troubleshooting 2 g067226e

3.13 Diagnostics

The diagnostics commands allow internal module information to be accessed. This includes:

- Error status of the device,
- Variables and status bits,
- Password,
- Configuration table and
- Position table

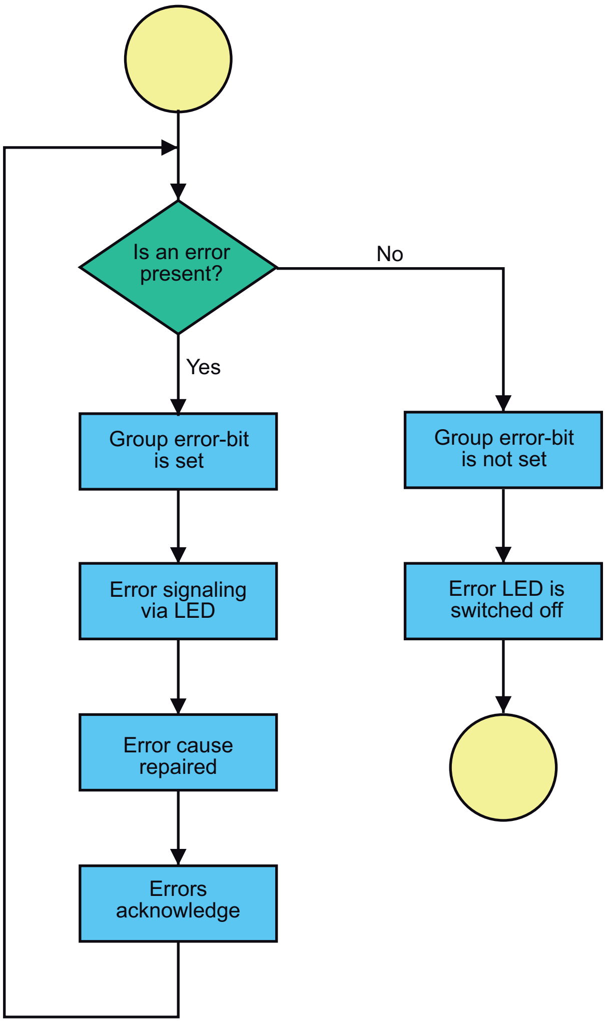


Fig. 3.12.4-1: Advanced diagnostics g067211e

The commands are elucidated in the appendix in section Diagnostics Commands, „Diagnostics Commands“.

The procedures for error correction are described in the appendix to chapter 4.4, "Error Blink Codes".

3.13.1 Internal Status Variables

The module is provided with internal status variable that can be read out using the mailbox command DIAG_RD_VAR. These variables can also be acquired automatically using a data recorder.

The number of the variable determines which source is read:

Variable number	Source
0–0x1000	predefined variables are read (see chapter 4.7, „Internal State Variables“)
0x1000…0x1100:	predefined bits 0 ….. 0x100 are read (see chapter 4.5, "Bit Field for I/O Driver")

The status variables are elucidated in the appendix in chapter 4.7, „Internal State Variables“.

3.13.2 Data Recorder

The data recorder facilitates the recording of two internal variables in a selectable time grid for a later analysis. 500 values are recorded each time.

The configuration values Trace_Var1 and Trace_Var2 contain the index for the variables to be recorded (see chapter 4.7, "Internal State Variables"). The configuration value Trace_MsecCycleTime denotes the scan (cycle) time in ms.

Configuration values	Meaning
Trace_Var1	Index of the first variable to be recorded
Trace_Var2	Index of the second variable to be recorded
Trace_MsecCycleTime	Scan time in ms

The Trace_Stored bit indicates that a complete data set has been recorded.

A 0→1 edge of Trace_Trigger initiates recording when the Trace_Armed bit is set.

A traced (recorded) data set can be read out using an upload command from the table manager (see chapter 3.9, "Table Manager").

Bits	**deleted**	**set**
Trace_Stored	No dataset present	Dataset is recorded
TRACE_TRIGGER	0→1 Edge starts the recording	
Trace_Armed	Triggering is disabled, an existing dataset is not overwritten.	Triggering is active, the next trigger event initiates the recording.

The internal bits (see chapter 4.5, „Bit Field for I/O Driver“) can be used as triggering sources. This is accomplished by entering the corresonponding link in the configuration table. The control system can also initiate recording using the mailbox commands GET_BIT and SET_BIT. The trigger bits must be linked to the MONE internal bit for this.

3.14 Installation Instructions

The following representation describes the components that are required for a structure that complies with standards.

Additional measures may be necessary, according to the table, depending on whether installation is made in a residential, industrial, or marine area.

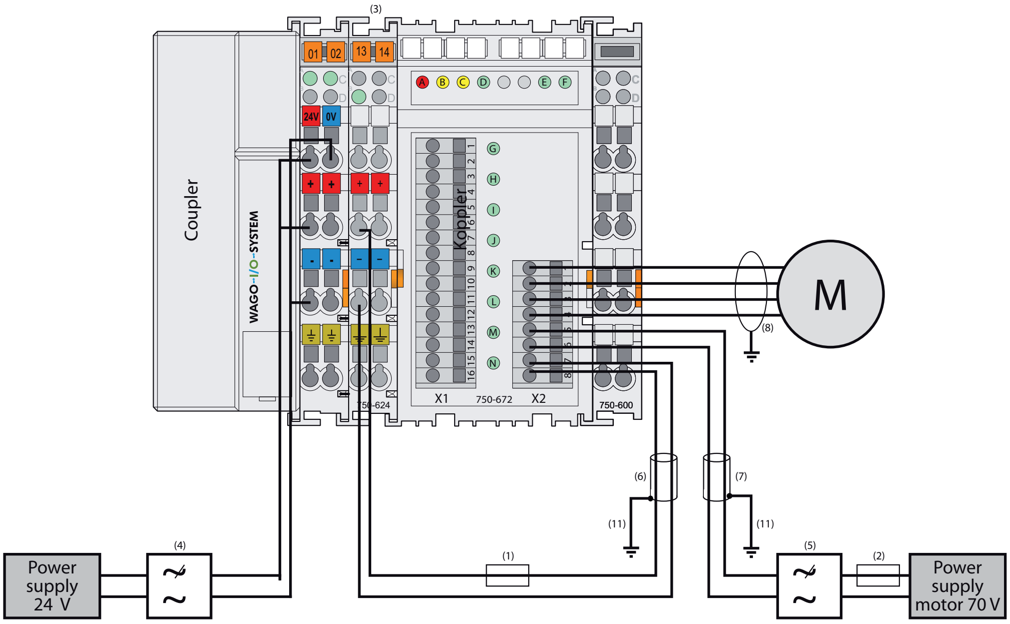


Fig. 3.13.2-1: Overview of installation g067206e

Ref.	Designation	Manufacturer/Item No.	Comments
		Fuses	
1	Field supply fuse	E.g., Siba model 172100, 3.15 A medium slow	A fuse must be connected upstream of the 24 V field voltage in feed as an energy limit in case of error.
2	Motor supply fuse	E.g., Wickmann/Littlefuse No. 230 10 A: 2302100000 Manufacturer: Püschel No. 105000 Manufacturer: Siba Model 7000140	To protect the motor voltage supply against reverse polarity, a 5x20 mm micro fuse with the "Super Quick (FF)" shut-down behavior must be connected upstream. The fusing integral i2 t shall not exceed $22A^2s$

Ref.	Designation	Manufacturer/Item No.	Comments
Filtering measures to adapt to certain application areas			
3	24VDC filter module	WAGO 750-624	GL 2003 EN61131-2 (1) EN61000-6-2 (2) EN61000-6-3 (2) EN61800-3 (1) If cable length > 30 m (2) If cable length > 3 m
4	Radio interference filter on 24VDC Field supply	Schaffner Company FN2060-6-06	GL 2003 EN61000-6-3 EN61800-3 (for implementation in residential area)
5	Radio interference filter on 70VDC Motor voltage	Schaffner Company FN2080-6-06	GL 2003 EN61131-2 EN61000-6-3 EN61800-3 (for implementation in residential area)
6	Ferrite on 24VDC Field supply	Würth-Elektronik 742 711 12	GL 2003 EN61000-6-3 EN61800-3 (for implementation in residential area)
7	Ferrite on 70VDC Motor voltage	Würth-Elektronik 742 711 12	
EA cables			
8	shielded cable Motor	E.g., LAPP UNITRONIC LIYCY 4 * 0.5mm^2	Depending on software filter, the SURGE pulse is detected as an event.
10	shielded cable digital outputs	E.g., Weydemeyer 4 * AWG20 C UL/CSA Style 2464	Depending on software filter, the SURGE pulse is detected as an event.
11	Shielded module	WAGO	The WAGO Shield Connection System consists of shielding clamping brackets, bus bars and diverse mounting bases to carry out a variety of constructional systems.

3.15 Connection Example

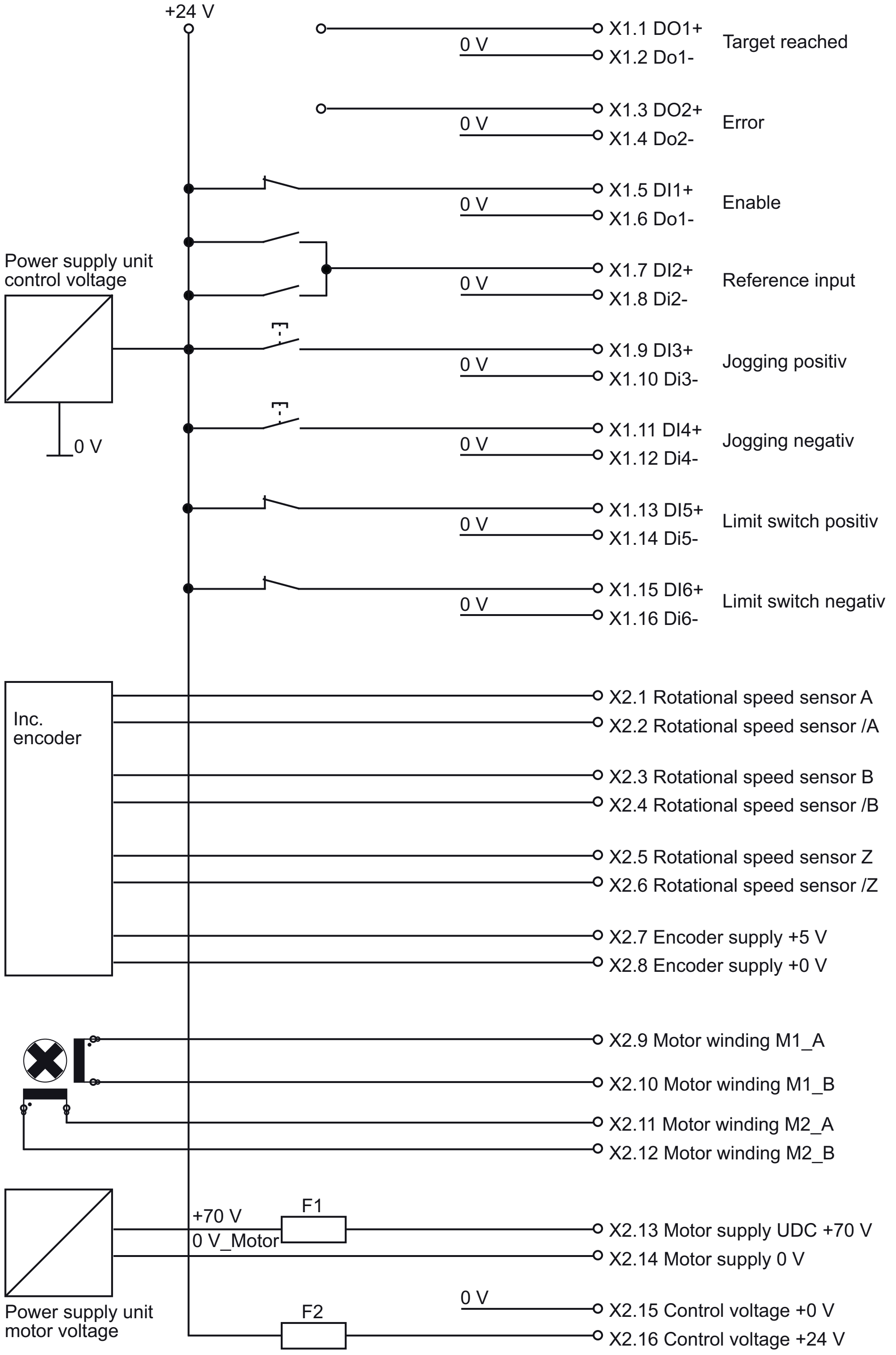


Fig. 3.13.2-1: Connection example g067207e

Note
Observe the star wiring in the 0 V and 0 V_Motor masses!

3.16 Recommendations to Avoid Overvoltage when Decelerating or Sinking a Vertical Load

During quick deceleration or reversal of the axis, or during sinking of a vertical load, the braking energy is fed back into the motor's current voltage. This can increase the intermediate circuit voltage.

However, both the drive itself and the feeding device, as well as other users of this voltage increase, can still be endangered and should be protected!
When the voltage reaches the definable limit value of U_{max} = DC_Link_U_{max}, the overvoltage error is issued and the motor control is switched off.

A larger power supply unit, additional users or an external condenser must be able to accommodate the energy that has been fed back in. The external condenser must be designed for a voltage of at least 100 V.

$$C = \frac{2 * W_{mech}}{U_{\max}{}^2 - U_{DC}{}^2} - C_{\text{int}}$$

W_{mech}	Mechanical energy fed back in
U_{max}	Parameter DC_Link_U_{max}
U_{DC}	Motor supply
C_{int}	400 µF

3.17 Information on Motor Selection

3.17.1 Nominal Current

The specifications in the manual for the motor current are peak values.
The data sheet specifications for motors must be interpreted as effective values.
In full step operation, there is no difference between effective values and peak values.

However, due to the microsteppings, the stepper controller 750-672 has an approximately sinusoidal current flow.

Example:

A motor should be implemented for the stepper controller 750-672.
The nominal current for continuous operation results in:

$$I_n = \frac{5A}{\sqrt{2}} = 3{,}5A$$

This value is the nominal current in the motor's data sheet.
The motor can be implemented in continuous operation without overheating.

The 7.5 A current only applies to the acceleration phases (150%).

Warning: this current is only delivered for a maximum of 10 s!

The current controller is implemented as a two- or three-point controller. This has the advantage that the controller is always stable and the comparison is simplified. The controller frequency adjusts automatically and changes with the current. The controller frequency is prevented from becoming too high. A controller frequency that is too small leads to a whistle and can be changed as needed with the parameter Current_Ctrl_TZMax.
The motor inductance should lie in the range of a few mH.
The winding resistance should lie in the range of 0.2 Ω to 0.8 Ω.

3.17.2 Measurement of Motor Inductance

The following equations are tailored quantity equations.

The formula symbols have the following units:

I	[A]	Motor current (parameter 14, current)
U	[V]	UDC motor supply
L	[mH]	Inductance of a cable's motor winding
f_m	[Hz]	Mechanical rotational speed
p	[1]	Number of pole pairs (typically 50)

1. Requirement for minimum inductance:

The current ripple should not exceed 10% of the motor current.

$$L \geq \frac{0{,}16 * U}{I}$$

2. Requirement for maximum inductance:

a) For a good dynamic, the inductance should not become too large.

$$L \leq \frac{U}{I}$$

b) The self-induction voltage should not exceed 1/3 of the motor voltage.

$$L \leq \frac{53}{p * f_m} * \frac{U}{I}$$

The ranges of acceptable motor inductance are shown in the following graphics for several combinations.

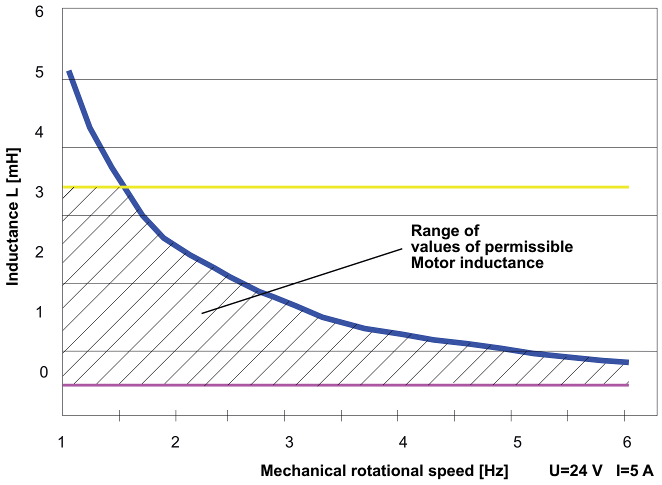


Fig. 3.17.2-1: Range of acceptable motor inductance for I=5 A

g067222e

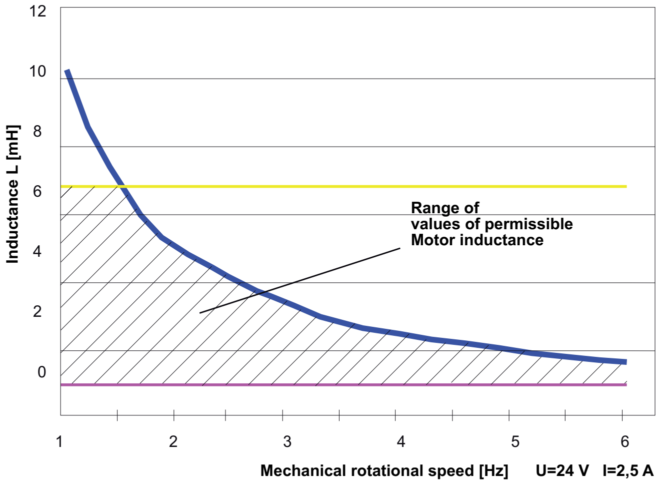


Fig. 3.17.2-2: Range of acceptable motor inductance for I=2.5 A

g067223e

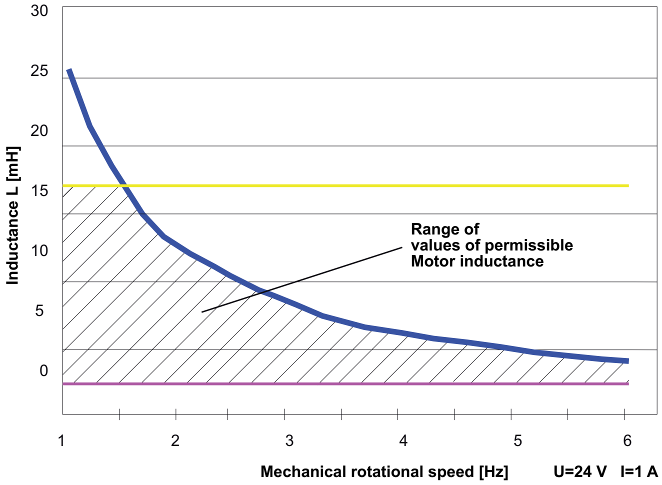


Fig. 3.17.2-3: Range of acceptable motor inductance for I=1 A

g067224e

3.18 Process Priorities

The task of process control is to coordinate timely semi-parallel running processes in a reasonable manner and to divide up the computing time of the microprocessor according to the necessary priorities.
Foreground computing processes have a high priority.

The following table provides an overview of the task system.

Prio-rity	Interrupt	Type	Fre-quency Type	Exe-cution Time Type	Incorporated Functionality
high	Exception	Exception	-	-	Error message, unknown system status
	Watchdog	Exception	-	-	Error message, watchdog
	Timer 62 µs	periodic	16 kHz	10 ... 20 µs	- Time measurement - Frequency ramps - Position recognition
	Timer 62 µs	periodic	4 kHz	20 µs	- Set DOs - Filter functions: counter
	Timer 1 ms	periodic	1 kHz	70 ... 320 µs	- Position recognition - Analyze DI´s - Path segment calculation - Data recorder - Error LED - Current monitoring
low	Slow_Inte rrupt	Back-ground I	-	80 µs ... 7 ms	- I/O module processing - Control/status bytes - I/O module-mailbox - Status control - Filter functions - Data recorder - Auxiliary functions
none	Backgrou nd	Back-ground II	-	-	- Configuration update

4 Appendix

4.1 Calculation Formulas

4.1.1 Position

4.1.1.1 Full Step Angle

Correlation between the number of full steps and the step angle

$$N = \frac{360°}{\alpha}$$

N	[1]	Number of full steps
α	[°]	Full step angle

Correlation between the number of pole pairs and the full step angle

$$\alpha = \frac{180°}{(p * s)}$$

$$\alpha = \frac{90°}{p}$$

α	[°]	Full step angle
p	[1]	Number of pole pairs
s	[1]	Line number s = 2

A typical industrial motor has a step angle of 1.8° corresponding to 200 full steps and a pole pair number of p = 50.

Correlation between electric and mechanical sizes

$$f_e = p * f_m \qquad \text{or} \qquad \omega_e = p * \omega_m \quad \text{with} \quad \omega = 2 * \pi * f$$

f	[1/s]	Frequency
p	[1]	Number of pole pairs
ω_e	[1/s]	Angular frequency of motor current of winding
ω_m	[1/s]	Angular frequency of axis

4.1.1.2 Rotor Bearing

Correlation between the position specification in the process data and the rotor bearing on the motor axis

$$\varphi = 360° * \frac{s}{64 * N}$$

φ	[°]	Rotor bearing, axis angle
s	[1]	Setpoint position in process image
N	[1]	Number of full steps

4.1.1.3 Rotary Axis

When specifying the parameters for the rotary axis, the following restriction must be observed for the function to be implemented correctly.

$$v \leq \frac{Speed\_Div}{Speed\_Mult} * \frac{Freq\_Div}{80} * 10^3 * p$$

v	Maximum allowed setpoint speed
p	Rotary axis periods (parameter 64, Rotary_Axis_Period)
Speed_Mult	Scaling factor for setpoint speed (parameter 28 from configuration table)
Speed_Div	Scaling factor for setpoint speed (parameter 30 from configuration table)
Freq_Div	Prescaler for maximum speed (parameter 4 from configuration table)

4.1.2 Rotational Speed

4.1.2.1 Frequency

Correlation between the internal pulse frequency and the mechanical rotational speed at the motor axis

$$f_m = \frac{1}{p} * \frac{f_p}{256}$$

f_m	Rotational speed at the motor axis revolutions per second
p	Number of pole pairs
f_p	Internal pulse frequency

4.1.2.2 Frequency Prescaler

The drive speed is identified by the pulse frequency f_p

$$fp = \frac{Velocity * 80}{Freq\_Prescaler[Hz]}$$

f_p Pulse frequency

The acceptable velocity range is 1 … 25000. The setting for the pulse frequency in [Hz] is given by selecting Freq_Prescaler = 80.

4.1.2.3 Calculation via Internal Pulse Frequency

Correlation between velocity specification in the process data and internal pulse frequency

$$f_p = \frac{Speed\_Mult}{Speed\_Div} * \frac{80 * v}{Freq\_Div}$$

f_p [1/s]	Internal pulse frequency in increments per second^2
v [1]	Setpoint velocity in process image
Speed_Mult	Scaling factor for setpoint velocity (parameter 28 from configuration table)
Speed_Div	Scaling factor for setpoint velocity (parameter 30 from configuration table)
Freq_Div	Prescaler for maximum velocity (parameter 4 from configuration table)

Correlation between the internal pulse frequency and the mechanical rotational speed at the motor axis

$$f_m = \frac{1}{p} * \frac{f_p}{256}$$

f_m	[1/s]	Rotational speed at the motor axis in revolutions per second
p	[1]	Number of pole pairs
f_p	[1/s]	Internal pulse frequency in increments per second^2

4.1.2.4 Direct Calculation

Correlation between velocity specification in the process data and mechanical rotational speed at the motor axis

$$f_m = \frac{1}{p} * \frac{Speed\_Mult}{Speed\_Div} * \frac{v}{3{,}2 * Freq\_Div}$$

$$n_m = 60 * f_m$$

f_m [1/s]	Rotational speed at the motor axis in revolutions per second
p [1]	Number of pole pairs
v [1]	Setpoint velocity in process image
Speed_Mult	Scaling factor for setpoint velocity (parameter 28 from configuration table)
Speed_Div	Scaling factor for setpoint velocity (parameter 30 from configuration table)
Freq_Div	Prescaler for maximum velocity (parameter 4 from configuration table)

4.1.3 Acceleration

Acceleration Factor

$$a = Acceleration * \frac{Acc\_Multiplier}{Freq\_\Pr escaler\left[\frac{Hz}{s}\right]}$$

a	Acceleration
Acceleration	Setpoint acceleration

The permissible acceleration range is 1 … 32767.
Acceleration is set in [Hz/s] when the acceleration factor Acc_Multiplier is selected equal to the prescaler Freq_Prescaler.

Correlation between internal acceleration and specification in the process data

$$a = \frac{Acc\_Mult}{Acc\_Div} * \frac{Acc\_Fac}{Freq\_Div} * Accleration$$

a	Acceleration
Acceleration	Setpoint acceleration
Acc_Mult	Scaling factor for setpoint acceleration (parameter 32 from configuration table)
Acc_Div	Scaling factor for setpoint acceleration (parameter 34 from configuration table)
Acc_Fac	Factor for maximum acceleration (parameter 6 from configuration table)
Freq_Div	Prescaler for maximum velocity (parameter 4 from configuration table)

Correlation between the internal acceleration and the mechanical acceleration

$$a_m = \frac{1}{p} * \frac{a}{256}$$

a_m [1/s^2]	Acceleration at the motor axis in revolutions per second^2
p [1]	Number of pole pairs
v [1]	Target velocity in process image

4.1.4 Electric Parameters

4.1.4.1 Motor Constant

The motor constant K_m can be calculated from the data sheet specifications as follows:

$$M_h = K_m * I_n$$

M_h [N]	Holding torque
K_m	Motor constant
I_n [A]	Motor current vertex

or, alternatively, from the measurement of open-circuit voltage U_{emf}:

$$U_{emf} = K_m * \omega_m$$

U_{emf}	[V]	Open-circuit voltage vertex of a motor winding
K_m	Motor constant	
ω_m	[1/s]	Angular frequency, mechanical

4.1.4.2 Supply Voltage

The following formulas describe the correlation between rotational speed, torque and the minimum required supply voltage U$_{DC}$.

The correlation between the voltage vectors can be depicted by the following vector diagram:

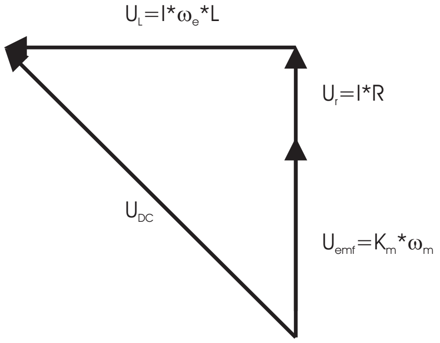


Fig. 4.1.4-1: Vector diagram g067208d

Minimum Required Motor Voltage

$$U_{DC} = \sqrt{U^{2}{}_{L} + \left(U_{r} + U_{emf}\right)^{2}}$$

U_{DC}	[V]	Minimum required motor voltage
U_L	[V]	Voltage via inductance of motor winding
Ur	[V]	Voltage via ohmic resistance of motor winding
U_{emf}	[V]	Open-circuit voltage vertex of a motor winding

Voltage via ohmic resistance of motor winding

$$U_r = I * R$$

U_r	[V]	Voltage via ohmic resistance of motor winding
I	[A]	Motor current vertex
R	[W]	Winding resistance of a motor winding, incl. supply line resistance

Voltage via inductance of motor winding

$$U_L = I * \omega_m * L$$

U_L	[V]	Voltage via inductance of motor winding
I	[A]	Motor current vertex
ω_e	[1/s]	Angular frequency, electric
L	[H]	Inductance of a motor winding

Open-circuit voltage vertex of a motor winding

$$U_{emf} = K_m * \omega_m$$

U_{emf}	[V]	Open-circuit vertex of a motor winding
K_m		Motor constant
ω_m	[1/s]	Angular frequency, mechanical

4.1.4.3 Current Profile Setting

A lower overcurrent can be supplied by the module for a correspondingly longer time.

The time until deactivation results from the following correlation:

$$t_{off} = \frac{5A}{I_{OV} - 5A} * 5s$$

t_{off}	Time until shutdown
I_{0V}	Overcurrent

After operation with overcurrent, a phase with reduced current must follow, in order for the module's thermal model to return to its original state. The required cooling time (t_{cool}) is calculated as follows:

$$t_{cool} = 4 * \frac{I_{OV} - 5A}{5A - I_{re}} * t_{OV}$$

t_{cool}	Cooling time
I_{0V}	Overcurrent
I_{re}	Reduced current
t_{0V}	Duration

4.1.4.4 Buffer Condenser

$$C = \frac{2 * W_{mech}}{U_{\max}{}^2 - U_{DC}{}^2} - C_{\text{int}}$$

W_{mech}	Mechanical energy fed back in
U_{max}	Parameter DC_Link_U$_{max}$
U_{DC}	Motor supply
C_{int}	400 µF

4.2 Mailbox Commands

4.2.1 Overview of Mailbox Commands

Function	Opcode	Meaning	Page
General commands			
IDLE	0x00	No task	152
Drive commands			
DRIVE_COMMAND	0x40	Command for Move mode	153
Download command			
DLD_START	0x41	Download Start	154
DLD_CONT	0x42	Download Continue	157
DLD_END	0x43	Completion of download	160
Table management commands			
TABLE_ERASE	0x44	Tables will be deleted.	161
TABLE_COPY	0x45	Tables will be copied.	163
TABLE_START	0x46	Table is activated	166
TABLE_STOP	0x48	Ends table processing	167
TABLE_GET_ACTIVE	0x4F	Determine active table	168
Diagnostics commands			
DIAG_RD_ERROR	0x49	Information about error retrieved from error memory	169
DIAG_QUIT_ERROR	0x4A	Terminates a device error condition	170
DIAG_RD_VAR	0x4C	Read out internal variable	171
DIAG_RD_BIT	0x4D	Read out internal bit	172
DIAG_QUERY_ STORAGE	0x4E	Read out storage process status bit	173
Configuration table commands			
CONFIG_SET_PTR	0x50	Set address for data access to the configuration	174
CONFIG_WR	0x51	Write access to configuration value	175
CONFIG_RD	0x52	Read access to configuration value	176
CONFIG_SAVE	0x53	Saves the current RAM configuration	177
CONFIG_RESTORE	0x54	Restores the configuration	178

Function	Opcode	Meaning	Page
Position table commands			
POS_TABLE_CREATE	0x5C	Generates a position table in the RAM.	180
POS_TABLE_SET_PTR	0x5D	Sets an index for the subsequent entry to be written with POS_TABLE_WR in the position table	181
POS_TABLE_WR	0x5E	Writes an entry to the active position table	182
POS_TABLE_TEACH	0x5F	Writes the current position to the active position table	183

4.2.2 Overview of Mailbox Commands, Sorted by Opcodes

Function	Opcode	Meaning	Page
IDLE	0x00	No task	152
DRIVE_COMMAND	0x40	Command for Move mode	153
DLD_START	0x41	Download Start	154
DLD_CONT	0x42	Download Continue	157
DLD_END	0x43	Completion of download ...	160
TABLE_ERASE	0x44	Tables being deleted	161
TABLE_COPY	0x45	Tables being copied ...	163
TABLE_START	0x46	Activates a table	166
TABLE_STOP	0x48	Ends table processing	167
DIAG_RD_ERROR	0x49	Information about error retrieved from error memory	169
DIAG_QUIT_ERROR	0x4A	Terminates a device error condition	170
DIAG_RD_VAR	0x4C	Read out internal variable	171
DIAG_RD_BIT	0x4D	Read out internal bit	172
DIAG_QUERY_ STORAGE	0x4E	Read out storage process status bit	173
TABLE_GET_ACTIVE	0x4F	Determine active table	168
CONFIG_SET_PTR	0x50	Set address for data access to the configuration	174
CONFIG_WR	0x51	Write access to configuration value	175
CONFIG_RD	0x52	Read access to configuration value	176
CONFIG_SAVE	0x53	Saves the current RAM configuration	177
CONFIG_RESTORE	0x54	Restores the configuration	178
POS_TABLE_CREATE	0x5C	Generates a position table in the RAM.	180
POS_TABLE_SET_PTR	0x5D	Sets an index for the subsequent entry to be written with POS_TABLE_WR in the position table	181
POS_TABLE_WR	0x5E	Writes an entry to the active position table	182
POS_TABLE_TEACH	0x5F	Writes the current position to the active position table	183

4.2.3 Overview of Mailbox Commands, Sorted by Functions

Function	Opcode	Meaning	Page
CONFIG_RD	0x52	Read access to configuration value	176
CONFIG_RESTORE	0x54	Restores the configuration	178
CONFIG_SAVE	0x53	Saves the current RAM configuration	177
CONFIG_SET_PTR	0x50	Set address for data access to the configuration	174
CONFIG_WR	0x51	Write access to configuration value	175
DIAG_QUERY_ STORAGE	0x4E	Read out storage process status bit	173
DIAG_QUIT_ERROR	0x4A	Terminates a device error condition	170
DIAG_RD_BIT	0x4D	Read out internal bit	172
DIAG_RD_ERROR	0x49	Information about error retrieved from error memory	169
DIAG_RD_VAR	0x4C	Read out internal variable	171
DLD_CONT	0x42	Download Continue	157
DLD_END	0x43	Completion of download ...	160
DLD_START	0x41	Download Start	154
DRIVE_COMMAND	0x40	Command for Move mode	153
IDLE	0x00	No task	152
POS_TABLE_CREATE	0x5C	Generates a position table in the RAM.	180
POS_TABLE_SET_PTR	0x5D	Sets an index for the subsequent entry to be written with POS_TABLE_WR in the position table	181
POS_TABLE_TEACH	0x5F	Writes the current position to the active position table	183
POS_TABLE_WR	0x5E	Writes an entry to the active position table	182
TABLE_COPY	0x45	Tables being copied ...	163
TABLE_ERASE	0x44	Tables being deleted	161
TABLE_GET_ACTIVE	0x4F	Determine active table	168
TABLE_START	0x46	Activates a table	166
TABLE_STOP	0x48	Ends table processing	167

4.2.4 Reference Commands – Mailbox Commands

4.2.4.1 General commands

4.2.4.1.1 IDLE (0x00)

No task is performed if the value for "Opcode" is 0.

Request								
Byte	**2^7**	**2^6**	**2^5**	**2^4**	**2^3**	**2^2**	**2^1**	**2^0**
MB0	**0x00**							
MB1	T	-						
MB2	Reserved							
MB3	Reserved							
MB4	Reserved							
MB5	Reserved							

Response								
Byte	**2^7**	**2^6**	**2^5**	**2^4**	**2^3**	**2^2**	**2^1**	**2^0**
MB0	**0x00**							
MB1	T	Return Code						
MB2	Reserved							
MB3	Reserved							
MB4	Reserved							
MB5	Reserved							

Return Code	0x00:	OK
	0x01:	General error

4.2.4.2 Move Commands

4.2.4.2.1 DRIVE_COMMAND (0x40)

The module can be operated via the mailbox using the move commands. Movement can be made directly to different positions. This command is accepted only when the mode "Move task via mailbox"(chapter **Fehler! Verweisquelle konnte nicht gefunden werden.**, „**Fehler! Verweisquelle konnte nicht gefunden werden.**“) has been activated.

The commands available for the Move mode are described in chapter 4.3, “4.3”.

Request								
Byte	**2^7**	**2^6**	**2^5**	**2^4**	**2^3**	**2^2**	**2^1**	**2^0**
MB0	**0x40**							
MB1	T	-						
MB2	Command							
MB3	Data 1							
MB4	Data 2							
MB5	Data 3							

Response								
Byte	**2^7**	**2^6**	**2^5**	**2^4**	**2^3**	**2^2**	**2^1**	**2^0**
MB0	**0x40**							
MB1	T	Return Code						
MB2	Command							
MB3	Data 1							
MB4	Data 2							
MB5	Data 3							

Return Code	0x00:	OK
	0x01:	General error
	0x11:	The last command is still being executed
	0x12:	Command not accepted, for example, when a Move command has not yet been completed.
	0x13:	Unknown command
	0x23:	Access denied

4.2.4.3 Download Commands

4.2.4.3.1 DLD_START (0x41)

Download Start

Tables are always loaded into RAM (1 or 2) first. The cursor is first placed on the first entry. Only one table can be loaded at any one time; any previous, incomplete download is canceled and becomes invalid. Direct transfer to / from the EEPROM is not possible (see also TABLE_COPY). Download to the same RAM sector is rejected with an error message when a move program table is still active. Camshaft and position tables can also be overwritten when they are active.

Default assignment

The default assignment for moving curve tables is PROG_END (0x00). Camshaft tables have the default assignment 0x80000000 (invalid position). The default assignment for position tables is 0.

Configuration tables

The EEPROM version number is expected in byte 5 during download of a configuration table. A complete table, with 128 data values, 32 bit each, is always expected.

Download formats (see also Request Data 4)

Expanded 32-bit down-/upload

A DLD_CONT must be used for an 8-bit command / data sample and for a 32-bit data entry / position entry in the table when downloading a Move program / a camshaft table. MB 4…6 are ignored for an 8-bit command / data sample.

Compressed 24-bit down-/upload

When downloading a Move program / a camshaft table, those items are transferred with a DLD_CONT command for an 8-bit command / data sample and a 24-bit date entry / position entry.

Maximum number of data sets (see also Request, Byte 5 and 6)

Table	**Type**	**Max. number of data sets**
Move program:	1	400
Camshaft	2	50
Position table	3	50
Configuration	4	128
Trace	5	1000

Request								
Byte	**2^7**	**2^6**	**2^5**	**2^4**	**2^3**	**2^2**	**2^1**	**2^0**
MB0	**0x41**							
MB1	T	-						
MB2	Storage location							
MB3	Table type							
MB4	Number of data (LSB)							
MB5	Transfer		Number of data (MSB)					

Storage location	Table type 1, 2, 3	0:	Reserved
		1:	RAM Table 1
		2:	RAM Table 2
		3 ... 255:	Reserved
	Table type 4, 5	0:	Reserved
		1:	RAM Table 1
		2 ... 255:	Reserved
Table type	0:	Reserved	
	1:	Move program:	
	2:	Camshaft	
	3:	Position table	
	4:	Configuration data set, User configuration	
	5:	Trace	
Transfer	0:	24-bit data download	
	1:	24-bit data upload	
	2:	32-bit data download	
	3:	32-bit data upload	
	4 ... 255:	Reserved	

Response								
Byte	**2^7**	**2^6**	**2^5**	**2^4**	**2^3**	**2^2**	**2^1**	**2^0**
MB0	**0x41**							
MB1	T	Return Code						
MB2	Status							
MB3	Reserved							
MB4	Number of data sets							
MB5	EEPROM version number							

Return Code	0x00:	OK
	0x30:	Table being used
	0x31:	General error
Status	0:	Download/Upload can be started
	1:	Error; Download/Upload not possible

4.2.4.3.2 DLD_CONT (0x42)

Download Continue

An entry is written to the selected table. The cursor is then moved to the next element. The request data are ignored for an upload. An error is returned when it is detected during a download that a transmitted Move table command is invalid, or if camshaft entries are transmitted NOT in ascending order. The data that has been transmitted will not be corrected. The table can not be valid with DLD_END however.

Download formats:

Expanded 32-bit down-/upload

A DLD_CONT must be used for an 8-bit command / data sample and for a 32-bit data entry / position entry in the table when downloading a Move program / a camshaft table. MB 4 … 6 are ignored for an 8-bit command / data sample.

Move program table (Type 01)

Step	MB2	MB3	MB4	MB5
1.1	Command 1	Reserved	Reserved	Reserved
1.2	Data 1 (LSB)	Data 1	Data 1	Data 1 (MSB)
2.1	Command 2	Reserved	Reserved	Reserved
2.2	Data 2 (LSB)	Data 2	Data 2	Data 2 (MSB)
…				

Camshaft Table (Type 02)

Step	MB2	MB3	MB4	MB5
1.1	Bit sample 1	Reserved	Reserved	Reserved
1.2	Position 1 (LSB)	Position 1	Position 1	Position 1 (MSB)
2.1	Bit sample 2	Reserved	Reserved	Reserved
2.2	Position 2 (LSB)	Position 2	Position 2	Position 2 (MSB)
…				

Compressed 24-bit down-/upload

When downloading a Move program / a camshaft table, those items are transferred with a DLD_CONT command for an 8-bit command / data sample and a 24-bit date entry / position entry:

Move program table (Type 01)

Step	MB2	MB3	MB4	MB5
1	Command 1	Data 1 (LSB)	Data 1	Data 1 (MSB)
2	Command 2	Data 2 (LSB)	Data 2	Data 2 (MSB)
…				

Camshaft Table (Type 02)

Step	MB2	MB3	MB4	MB5
1	Bit sample 1	Position 1 (LSB)	Position 1	Position 1 (MSB)
2	Bit sample 2	Position 2 (LSB)	Position 2	Position 2 (MSB)
…				

Only 32-bit data exists for the position tables (Type 03) and the configuration data set (Type 04). Therefore, only the 32-bit down-/upload are given for both types of tables.

Request								
Byte	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
MB0	**0x42**							
MB1	T	-						
MB2	Data							
MB3	Data							
MB4	Data							
MB5	Data							

Response								
Byte	**2^7**	**2^6**	**2^5**	**2^4**	**2^3**	**2^2**	**2^1**	**2^0**
MB0	**0x42**							
MB1	T	Return Code						
MB2	Data							
MB3	Data							
MB4	Data							
MB5	Data							

Return Code	0x00:	OK
	0x31:	Upload/Download not started, or all data have already been transferred
	0x38:	Transferred data set corrupt

4.2.4.3.3 DLD_END (0x43)

End of Download

The download is completed and the stepper module checks the checksum. If the checksum is not OK, the table is invalid and can not be activated. The checksum is the sum of all data transferred with DLD_CONT. Summation is performed at 8 bits, with the 4 bytes that were transferred with DLD_CONT each being taken as 8-bit values. The difference between the sum of all transferred data and the checksum must therefore be zero. The request data are ignored for an upload. If a configuration table is transferred, saving to EEPROM is performed automatically (but only when saving has been completed successfully), with a subsequent warm start (even if the transfer was faulted) that re-initializes all software modules.
The Reset status bit is set after the warm start; this must be canceled using Reset_Quit.
Only then is the module operational again.
After a successful download of a Move program to RAM Table 1, that table is automatically activated. (only when no other table is active however, see also TABLE_START)

Request								
Byte	**2^7**	**2^6**	**2^5**	**2^4**	**2^3**	**2^2**	**2^1**	**2^0**
MB0	**0x43**							
MB1	T	-						
MB2	Checksum for transferred data (LSB)							
MB3	Checksum for transferred data							
MB4	Checksum for transferred data							
MB5	Checksum for transferred data (MSB)							

Response								
Byte	**2^7**	**2^6**	**2^5**	**2^4**	**2^3**	**2^2**	**2^1**	**2^0**
MB0	**0x43**							
MB1	T	Return Code						
MB2	Checksum for stored data (LSB)							
MB3	Checksum for stored data							
MB4	Checksum for stored data							
MB5	Checksum for stored data (MSB)							

Return Code	0x00:	OK
	0x31:	General error

4.2.4.4 Table Management Commands

4.2.4.4.1 TABLE_ERASE (0x44)

Tables are deleted by setting their status to invalid. An active table can not be deleted. A table can not be deleted during ongoing transfer using DLD_START, DLD_CONT or DLD_END.

Deleting of an EEPROM table is performed in the background, independently of processing of the table command (see also DIAG_QUERY_STORAGE). The "FACTORY_DEFAULT" configuration contained in the EEPROM can not be deleted (not even when using 255 as byte 2). "FACTORY_DEFAULT" may only be overwritten.

Request								
Byte	**2^7**	**2^6**	**2^5**	**2^4**	**2^3**	**2^2**	**2^1**	**2^0**
MB0	**0x44**							
MB1	T	-						
MB2	Storage location							
MB3	Table type							
MB4	Reserved							
MB5	Reserved							

Storage location	0:	EEPROM table
	1:	RAM Table 1
	2:	RAM Table 2
	3 ... 255:	Reserved
Table type	0:	Reserved
	1:	Move program:
	2:	Camshaft
	3:	Position table
	4:	Configuration data set, User configuration
	5 ... 255:	Reserved

Response								
Byte	**2^7**	**2^6**	**2^5**	**2^4**	**2^3**	**2^2**	**2^1**	**2^0**
MB0	**0x44**							
MB1	T	Return Code						
MB2	Status							
MB3	Reserved							
MB4	Reserved							
MB5	Reserved							

Return Code	0x00:	OK
	0x30:	Table active
	0x31:	General error
Status	0:	Successfully deleted
	1:	Deleting aborted

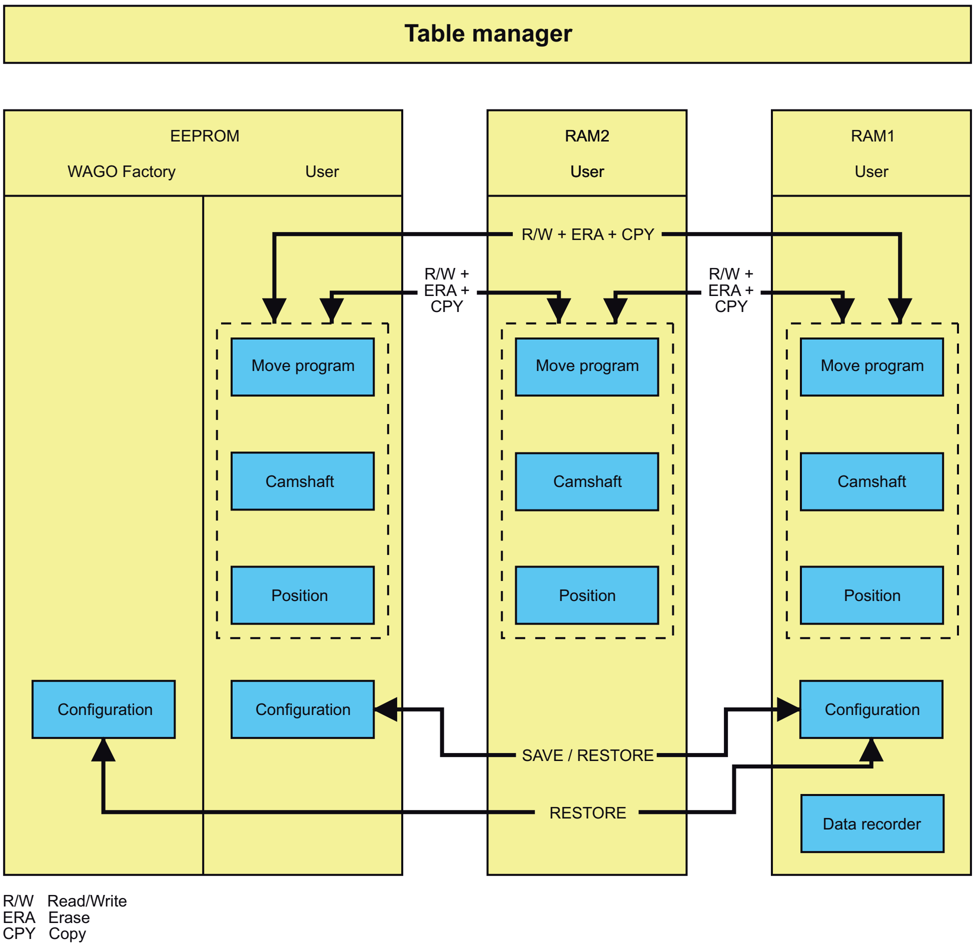


Fig. 4.2.4-1: Table manager

g067120e

4.2.4.4.2 TABLE_COPY (0x45)

Tables will be copied.

The target may not be identical to the source.

A table can not be specified either as the target nor as the source of the copying command when transfer using DLD_START, DLD_CONT or DLD_END has not been completed. Writing of the EEPROM is performed in the background, independent of processing of the table command (see also DIAG_QUERY_STORAGE).

The tables located in the EEPROM are always copied to RAM 1 when the system is started up. This command can not be used for copying out of the EEPROM.

A configuration table can not be copied with this command (see also CFG_SAVE, CONFIG_RESTORE).

Copying options:

1. RAM → RAM

2. RAM → EERPOM

Request								
Byte	**2^7**	**2^6**	**2^5**	**2^4**	**2^3**	**2^2**	**2^1**	**2^0**
MB0	**0x44**							
MB1	T	-						
MB2	Table type							
MB3	Data source							
MB4	Storage target							
MB5	Reserved							

Table type	0:	Reserved
	1:	Move program:
	2:	Camshaft
	3:	Position table
	4:	Reserved
	5:	Status query for a previous copying process (see also DIAG_QUERY_STORAGE)
	6 ... 255:	Reserved
Data source	0:	Reserved
	1:	RAM Table 1
	2:	RAM Table 2
	3 ... 255:	Reserved
Storage target	0:	EEPROM table
	1:	RAM Table 1
	2:	RAM Table 2
	3 ... 255:	Reserved

Response								
Byte	**2^7**	**2^6**	**2^5**	**2^4**	**2^3**	**2^2**	**2^1**	**2^0**
MB0	**0x44**							
MB1	T	Return Code						
MB2	Status							
MB3	Reserved							
MB4	Reserved							
MB5	Reserved							

Return Code	0x00:	OK
	0x31:	General error
	0x33:	Copying process still active
	0x34:	EEPROM copying process aborted
	0x35:	Target table not empty
Status	0:	Successfully copied
	1:	Copying aborted

4.2.4.4.3 TABLE_START (0x46)

Activates a table Only a valid table can be activated (transfer using DLD_START, DLD_CONT and DLD_END completed successfully and checksum valid). This command can only be used after the Move program has been stopped.

Request								
Byte	**2^7**	**2^6**	**2^5**	**2^4**	**2^3**	**2^2**	**2^1**	**2^0**
MB0	**0x46**							
MB1	T	-						
MB2	Storage location							
MB3	Table type							
MB4	Reserved							
MB5	Reserved							

Storage location	0:	No table
	1:	RAM Table 1
	2:	RAM Table 2
	3 ... 255:	Reserved
Table type	0:	Reserved
	1:	Move program:
	2:	Camshaft
	3:	Position table
	4 ... 255:	Reserved

Response								
Byte	**2^7**	**2^6**	**2^5**	**2^4**	**2^3**	**2^2**	**2^1**	**2^0**
MB0	**0x46**							
MB1	T	Return Code						
MB2	Status							
MB3	Reserved							
MB4	Reserved							
MB5	Reserved							

Return Code	0x00:	OK
	0x31:	General error
Status	0:	Successfully activated
	1:	Activation aborted

4.2.4.4.4 TABLE_STOP (0x48)

Ends table processing; after this, the STOP_FAST command is executed internally in the system.

Request								
Byte	**2^7**	**2^6**	**2^5**	**2^4**	**2^3**	**2^2**	**2^1**	**2^0**
MB0	**0x48**							
MB1	T	-						
MB2	Reserved							
MB3	Reserved							
MB4	Reserved							
MB5	Reserved							

Response								
Byte	**2^7**	**2^6**	**2^5**	**2^4**	**2^3**	**2^2**	**2^1**	**2^0**
MB0	**0x48**							
MB1	T	Return Code						
MB2	Status							
MB3	Reserved							
MB4	Reserved							
MB5	Reserved							

Return Code (error code for previous command SPEED_ STOP_ IMM)	0x00:	OK
	0x01:	General error
	0x11:	The last command is still being executed
	0x12:	Command not accepted, for example, when a Move command has not yet been completed.
	0x23:	Access denied
Status	exact error code, when return code <> 0	

4.2.4.4.5 TABLE_GET_ACTIVE (0x4F)

Determine the active table.

Request								
Byte	**2^7**	**2^6**	**2^5**	**2^4**	**2^3**	**2^2**	**2^1**	**2^0**
MB0	**0x4F**							
MB1	T	-						
MB2	Table type							
MB3	Reserved							
MB4	Reserved							
MB5	Reserved							

Table type	0:	Reserved
	1:	Move program:
	2:	Camshaft
	3:	Position table
	4:	Configuration data set, User configuration
	5 ... 255:	Reserved

Response								
Byte	**2^7**	**2^6**	**2^5**	**2^4**	**2^3**	**2^2**	**2^1**	**2^0**
MB0	**0x4F**							
MB1	T	Return Code						
MB2	Status							
MB3	Reserved							
MB4	Reserved							
MB5	Reserved							

Return Code	0x00:	OK	
	0x31:	Invalid table type	
Status	Table type 1, 2, 3	0:	No table active (even when 0x31 returned)
		1:	RAM Table 1 active
		2:	RAM Table 2 active
		3 … 255:	Reserved
	Table type 4	0:	Reserved (even when 0x31 returned)
		1:	User data set active
		2:	Factory default active
		3 … 255:	Reserved

4.2.4.5 Diagnostics Commands

4.2.4.5.1 DIAG_RD_ERROR (0x49)

Information about error is retrieved from the error memory.

Request								
Byte	**2^7**	**2^6**	**2^5**	**2^4**	**2^3**	**2^2**	**2^1**	**2^0**
MB0	**0x49**							
MB1	T	-						
MB2	Reserved							
MB3	Reserved							
MB4	Reserved							
MB5	Reserved							

Response								
Byte	**2^7**	**2^6**	**2^5**	**2^4**	**2^3**	**2^2**	**2^1**	**2^0**
MB0	**0x49**							
MB1	T	Return Code						
MB2	Error code (LSB)							
MB3	Error code (MSB)							
MB4	Extra information (LSB)							
MB5	Extra information (MSB)							

Return Code	0x00:	OK

4.2.4.5.2 DIAG_QUIT_ERROR (0x4A)

Terminates a device error condition.

Request								
Byte	**2^7**	**2^6**	**2^5**	**2^4**	**2^3**	**2^2**	**2^1**	**2^0**
MB0	**0x4A**							
MB1	T	-						
MB2	Reserved							
MB3	Reserved							
MB4	Reserved							
MB5	Reserved							

Response								
Byte	**2^7**	**2^6**	**2^5**	**2^4**	**2^3**	**2^2**	**2^1**	**2^0**
MB0	**0x4A**							
MB1	T	Return Code						
MB2	Reserved							
MB3	Reserved							
MB4	Reserved							
MB5	Reserved							

Return Code	0x00:	OK

4.2.4.5.3 DIAG_RD_VAR (0x4C)

Read out status variable. The variable number determines the source to be read from.

Request								
Byte	**2^7**	**2^6**	**2^5**	**2^4**	**2^3**	**2^2**	**2^1**	**2^0**
MB0	**0x4C**							
MB1	T	-						
MB2	Variable number							
MB3	Reserved							
MB4	Reserved							
MB5	Reserved							

Variable number	0 … 0x1000:	predefined variables are read (see chapter 4.7, “Internal State Variables”)
	0x1000 … 0x1100:	predefined bits 0 ….. 0x100 are read (see chapter 4.5, “Bit Field for I/O Driver “)
	0x40000000 … 0x40004000:	Direct reading out of RAM
	0xE0000000 … 0xE0200000:	Direct reading out of controller periphery
	0xFFE00000 … 0xFFFFFFFF:	Direct reading out of controller periphery

Response								
Byte	**2^7**	**2^6**	**2^5**	**2^4**	**2^3**	**2^2**	**2^1**	**2^0**
MB0	**0x4C**							
MB1	T	Return Code						
MB2	Variable (LSB)							
MB3	Variable							
MB4	Variable							
MB5	Variable (MSB)							

Return Code	0x00:	OK

4.2.4.5.4 DIAG_RD_BIT (0x4D)

Read out status bit (see also chapter 4.5, “Bit Field for I/O Driver”).

Request								
Byte	**2^7**	**2^6**	**2^5**	**2^4**	**2^3**	**2^2**	**2^1**	**2^0**
MB0	**0x4D**							
MB1	T	-						
MB2	Bit number							
MB3	Reserved							
MB4	Reserved							
MB5	Reserved							

Bit number	0 ... 255:	Specifies which predefined bit is being requested.

Response								
Byte	**2^7**	**2^6**	**2^5**	**2^4**	**2^3**	**2^2**	**2^1**	**2^0**
MB0	**0x4D**							
MB1	T	Return Code						
MB2	Status							
MB3	Reserved							
MB4	Reserved							
MB5	Reserved							

Return Code	0x00:	OK
Status	0:	Bit deleted
	1:	Bit set

4.2.4.5.5 DIAG_QUERY_STORAGE (0x4E)

Read out storage process status bit

Request								
Byte	**2^7**	**2^6**	**2^5**	**2^4**	**2^3**	**2^2**	**2^1**	**2^0**
MB0	**0x4E**							
MB1	T	-						
MB2	Reserved							
MB3	Reserved							
MB4	Reserved							
MB5	Reserved							

Response								
Byte	**2^7**	**2^6**	**2^5**	**2^4**	**2^3**	**2^2**	**2^1**	**2^0**
MB0	**0x4E**							
MB1	T	Return Code						
MB2	Status							
MB3	Reserved							
MB4	Reserved							
MB5	Reserved							

Return Code	0x00:	OK
Status	0:	Storing completed
	1 ... 255:	Storing in progress

4.2.4.6 Configuration Table Commands

4.2.4.6.1 CONFIG_SET_PTR (0x50)

Set address for data access to the configuration, see chapter **Fehler! Verweisquelle konnte nicht gefunden werden.**, "**Fehler! Verweisquelle konnte nicht gefunden werden.**". The specified address is the same as the byte address.

Request								
Byte	**2^7**	**2^6**	**2^5**	**2^4**	**2^3**	**2^2**	**2^1**	**2^0**
MB0	**0x50**							
MB1	T	-						
MB2	Address (LSB)							
MB3	Address (MSB)							
MB4	Number of bytes							
MB5	Reserved							

Number of bytes	0:	Reserved
	1 ... 4:	Number of bytes that are written for access with Config_WR.
	5 ... 255:	Reserved

Response								
Byte	**2^7**	**2^6**	**2^5**	**2^4**	**2^3**	**2^2**	**2^1**	**2^0**
MB0	**0x50**							
MB1	T	Return Code						
MB2	Reserved							
MB3	Reserved							
MB4	Reserved							
MB5	Reserved							

Return Code	0x00:	OK
	0x23:	Access denied; invalid number of bytes or invalid index

4.2.4.6.2 CONFIG_WR (0x51)

Write access to configuration value.

Request								
Byte	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
MB0	**0x51**							
MB1	T	-						
MB2	Data (LSB)							
MB3	Data							
MB4	Data							
MB5	Data (MSB)							

Response								
Byte	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
MB0	**0x51**							
MB1	T	Return Code						
MB2	Reserved							
MB3	Reserved							
MB4	Reserved							
MB5	Reserved							

Return Code	0x00:	OK
	0x23:	Access denied

4.2.4.6.3 CONFIG_RD (0x52)

Read access to configuration value. The value 0 is returned when invalid access size specified.

Request								
Byte	**2^7**	**2^6**	**2^5**	**2^4**	**2^3**	**2^2**	**2^1**	**2^0**
MB0	**0x52**							
MB1	T	-						
MB2	Address (LSB)							
MB3	Address (MSB)							
MB4	Number of bytes							
MB5	Reserved							

Number of bytes	0:	Reserved
	1 ... 4:	Number of bytes that are written for access with Config_RD.
	5 ... 255:	Reserved

Response								
Byte	**2^7**	**2^6**	**2^5**	**2^4**	**2^3**	**2^2**	**2^1**	**2^0**
MB0	**0x52**							
MB1	T	Return Code						
MB2	Data (LSB)							
MB3	Data							
MB4	Data							
MB5	Data (MSB)							

Return Code	0x00:	OK

4.2.4.6.4 CONFIG_SAVE (0x53)

Saves the current RAM configuration in the EEPROM. The configuration is saved as a user data set in the EEPROM with password 0x0001. The configuration is saved as RACTORY_DEFAULT in the EEPROM with password 0xE17E. At the same time, EEPROM sectors of the module registry set are also saved. A FACTORY_DEFAULT data set that has been saved can never be deleted again, only overwritten. This function does not wait for the saving process to be completed. This can be determined with DIAG_QUERY_STORAGE. Complete activation of the saved data set is conducted only after a (manual) restart of the module.
The Reset status bit is set after the warm start; this must be canceled using Reset_Quit.
Only then is the module operational again.

Request								
Byte	**2^7**	**2^6**	**2^5**	**2^4**	**2^3**	**2^2**	**2^1**	**2^0**
MB0	**0x53**							
MB1	T	-						
MB2	Password (LSB)							
MB3	Password							
MB4	Password							
MB5	Password (MSB)							

Response								
Byte	**2^7**	**2^6**	**2^5**	**2^4**	**2^3**	**2^2**	**2^1**	**2^0**
MB0	**0x53**							
MB1	T	Return Code						
MB2	Reserved							
MB3	Reserved							
MB4	Reserved							
MB5	Reserved							

Return Code	0x00:	OK
	0x31:	Fault

4.2.4.6.5 CONFIG_RESTORE (0x54)

The configuration is restored and the user data set overwritten. A warm start is carried out after the command has been successfully executed to ensure that all data is accepted.

Warning
During warm start the mailbox data are undefined. They may be evaluated only again if the status bit Reset signals the end of warm start.

Request								
Byte	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
MB0	**0x54**							
MB1	T	-						
MB2	Restore							
MB3	Warm start							
MB4	Reserved							
MB5	Reserved							

Restore	0:	Reserved
	1:	Last saved user data set loaded from EEPROM
	2:	User data set overwritten with FACTORY_DEFAULT
	3 ... 255:	Reserved
Warm start	0:	User data set overwritten and warm start carried out
	1:	The required data set is only loaded to the RAM, without a warm start being performed. Error 2821 = CFG_FACTORY_LOAD is reported. A warm start is carried out only when this error is acknowledged, with the original configuration being restored.
	2 ... 255:	Reserved

Response								
Byte	**2^7**	**2^6**	**2^5**	**2^4**	**2^3**	**2^2**	**2^1**	**2^0**
MB0	**0x54**							
MB1	T	Return Code						
MB2	Reserved							
MB3	Reserved							
MB4	Reserved							
MB5	Reserved							

Return Code	0x00:	OK
	0x31:	Fault

4.2.4.7 Position table commands

4.2.4.7.1 POS_TABLE_CREATE (0x5C)

Generates a position table in the RAM. The table status is set to "valid".

Request								
Byte	**2^7**	**2^6**	**2^5**	**2^4**	**2^3**	**2^2**	**2^1**	**2^0**
MB0	**0x5C**							
MB1	T	-						
MB2	Storage location							
MB3	Number of elements							
MB4	Initialization							
MB5	Reserved							

Storage location	0:	Reserved
	1:	RAM Table 1
	2:	RAM Table 2
	3 ... 255:	Reserved
Number of elements	1 … 50:	Number of elements
	51 … 255:	Reserved
Initializatio n	0:	Install table completely with 0x80000000
	1:	Expand existing table (all existing entries are retained). Is executed only when the new size is larger than the existing table!

Response								
Byte	**2^7**	**2^6**	**2^5**	**2^4**	**2^3**	**2^2**	**2^1**	**2^0**
MB0	**0x5C**							
MB1	T	Return Code						
MB2	Status							
MB3	Reserved							
MB4	Reserved							
MB5	Reserved							

Return Code	0x00:	OK
	0x32:	Invalid table specified
	0x3A:	Invalid number of elements
Status	0:	Initialization successful
	1:	Initialization aborted

4.2.4.7.2 POS_TABLE_SET_PTR (0x5D)

Sets an index for the subsequent entry to be written with POS_TABLE_WR in the active position table.

Request								
Byte	**2^7**	**2^6**	**2^5**	**2^4**	**2^3**	**2^2**	**2^1**	**2^0**
MB0	**0x5D**							
MB1	T	-						
MB2	Index							
MB3	Reserved							
MB4	Reserved							
MB5	Reserved							

Index	0 ... 49:	Index
	50 ... 255:	Reserved

Response								
Byte	**2^7**	**2^6**	**2^5**	**2^4**	**2^3**	**2^2**	**2^1**	**2^0**
MB0	**0x5D**							
MB1	T	Return Code						
MB2	Status							
MB3	Reserved							
MB4	Reserved							
MB5	Reserved							

Return Code	0x00:	OK
	0x37:	Table does not exist, or index not assigned
Status	0:	Successfully indexed
	1:	Indexing aborted

4.2.4.7.3 POS_TABLE_WR (0x5E)

Writes an entry to the active position table. The table index that was last set using POS_TABLE_SET_PTR is always overwritten.

Request								
Byte	**2^7**	**2^6**	**2^5**	**2^4**	**2^3**	**2^2**	**2^1**	**2^0**
MB0	**0x5E**							
MB1	T	-						
MB2	Save value (LSB)							
MB3	Save value							
MB4	Save value							
MB5	Save value (MSB)							

Response								
Byte	**2^7**	**2^6**	**2^5**	**2^4**	**2^3**	**2^2**	**2^1**	**2^0**
MB0	**0x5E**							
MB1	T	Return Code						
MB2	Status							
MB3	Reserved							
MB4	Reserved							
MB5	Reserved							

Return Code	0x00:	OK
	0x37:	Table does not exist, or index not set
Status	0:	Writing completed successfully
	1:	Writing aborted

4.2.4.7.4 POS_TABLE_TEACH (0x5F)

Writes the current position to the active position table.

Request								
Byte	**2^7**	**2^6**	**2^5**	**2^4**	**2^3**	**2^2**	**2^1**	**2^0**
MB0	**0x5F**							
MB1	T	-						
MB2	Target for measured value							
MB3	Measurement							
MB4	Reserved							
MB5	Reserved							

Target for measured value	0 ... 49:	Index at which the current position in the currently active position table is to be filed (see also TABLE_START)
	-1 (0xFF):	Save current position as negative limit Drive_Range_Neg (see configuration table)
	-2 (0xFE):	Save current position as positive limit Drive_Range_Pos (see configuration table)
	-3 (0xFD):	The current position is the zero point for a relevant measurement
	50 ... 252:	Reserved
Measurement	0:	Absolute measurement: Save current position
	1:	Relative measurement: Zero point for relative measurement – save current position
	2 ... 255:	Reserved

Response								
Byte	**2^7**	**2^6**	**2^5**	**2^4**	**2^3**	**2^2**	**2^1**	**2^0**
MB0	**0x5F**							
MB1	T	Return Code						
MB2	Status							
MB3	Reserved							
MB4	Reserved							
MB5	Reserved							

Return Code	0x00:	OK
	0x37:	Table or specified index does not exist
Status	0:	Writing completed successfully
	1:	Writing aborted

4.3 Commands for Move Mode

4.3.1 Overview of Commands for Move Mode

Function	Opcode	Meaning	Page
Setpoint commands			
MOVE	0x02	Each MOVE command initiates a positioning process.	193
MOVE_IMMEDIATE	0x03	Each MOVE command initiates a positioning process.	194
MOVE_TABLE	0x04	Each MOVE command initiates a positioning process.	195
MOVE_TABLE_ IMMEDIATE	0x05	Each MOVE command initiates a positioning process.	196
MOVE_REL	0x06	Each MOVE command initiates a positioning process.	197
MOVE_TABLE_REL	0x08	Each MOVE command initiates a positioning process.	198
SPEED	0x10	SPEED commands run the drive to a speed..	199
SPEED_IMMEDIATE	0x11	SPEED commands run the drive to a speed..	200
STOP_FAST	0x18	SPEED commands run the drive to a speed..	201
STOP_NO_RAMP	0x19	SPEED commands run the drive to a speed..	202
TORQ	0x1C	Each TORQ command begins a current setting (applies only to 750-673)	203
TORQ_IMM	0x1D	Each TORQ command begins a current setting (applies only to 750-673)	204
START_REFERENCING	0x20	Starts a reference run.	205
SET_ACC_MODE	0x21	Sets the acceleration and decceleration mode.	207
SET_ACC	0x22	Sets acceleration and/or delay; valid as of the next positioning process.	209
SET_ACC_PARAM_UP	0x23	Set the Acc_ParamUp parameter for acceleration; valid as of the next positioning process.	210
SET_ACC_PARAM_ DOWN	0x24	Set the Acc_ParamDown parameter for delay (deceleration); valid as of the next positioning process.	211
SET_VELOCITY	0x25	Sets the positioning speed; valid as of the next positioning process.	212

Function	Opcode	Meaning	Page
SET_VELOCITY_ TARGET	0x2B	Velocity to target position. Valid only for the next positioning process and is then reset automatically to zero after the next positioning process.	213
SET_ACTUALPOSITON	0x2E	The current position is applied to the transferred value.	214
SET_ACTUALPOSITION_ ZERO	0x2F	The current position is set to zero	215
SET_CURRENT	0x39	Sets the motor current for drive movement.	216
Mathematic commands			
VAR_SET	0x50	Sets one of the variables FILT1 ... FILT8 to the specified value	217
VAR_INC	0x51	Adds the specified value to one of the variables FILT1 … FILT8.	218
VAR_DEC	0x52	Subtracts the specified value from one of the variables FILT1 … FILT8.	219
VAR_ADD	0x53	Adds two variables and writes the result to a third variable	220
VAR_SUB	0x54	Subtracts two variables and writes the results to a third variable	221
VAR_MUL	0x55	Multiplies one variable by another one and writes the results to a third variable	222
VAR_COPY	0x56	Copes one variable to another variable.	223
VAR_DIV	0x57	Divides one variable by another one and writes the results to a third variable.	224
WAIT_TIME	0x70	Waits a defined time period before processing the next command.	225
WAIT_TEST_BIT	0x71	Before processing the next command waits until the specified bit has the status 0 or 1.	226
Auxiliary commands			
WR_BIT	0x78	Sets a bit to 0 or 1.	227
NOP	0xF0	No function	228
PROG_STOP	0xF1	Ends table processing.	229
PROG_END	0x00 or 0xFF	End of table.	230
GOTO	0xF5	Continues table process at the addressed entry.	231
GOTO_IF	0xF6	If a bit has been set, table processing is continued at the addressed entry; otherwise from the next entry.	232

Function	Opcode	Meaning	Page
GOTO_IF_NOT	0xF7	If a bit has been deleted, table processing is continued at the addressed entry; otherwise from the next entry.	233
GOTO_LABEL	0xF8	Continues table process from a defined label.	234
GOTO_LABEL_IF	0xF9	Continues table processing for a defined label if a bit has been set.	235
GOTO_LABEL_IF_NOT	0xFA	Continues table processing for a defined label if a bit has been deleted.	236
LABEL	0xFB	Defines a label for a step target	237

4.3.2 Overview of Move Mode Commands, Sorted by Opcodes

Function	Opcode	Meaning	Page
PROG_END	0x00 or 0xFF	End of table.	230
MOVE	0x02	Each MOVE command initiates a positioning process.	193
MOVE_IMMEDIATE	0x03	Each MOVE command initiates a positioning process.	194
MOVE_TABLE	0x04	Each MOVE command initiates a positioning process.	195
MOVE_TABLE_ IMMEDIATE	0x05	Each MOVE command initiates a positioning process.	196
MOVE_REL	0x06	Each MOVE command initiates a positioning process.	197
MOVE_TABLE_REL	0x08	Each MOVE command initiates a positioning process.	198
SPEED	0x10	SPEED commands run the drive to a speed..	199
SPEED_IMMEDIATE	0x11	SPEED commands run the drive to a speed..	200
STOP_FAST	0x18	SPEED commands run the drive to a speed..	201
STOP_NO_RAMP	0x19	SPEED commands run the drive to a speed..	202
TORQ	0x1C	Each TORQ command begins a current setting (applies only to 750-673)	203
TORQ_IMM	0x1D	Each TORQ command begins a current setting (applies only to 750-673)	204
START_REFERENCING	0x20	Starts a reference run.	205
SET_ACC_MODE	0x21	Sets the acceleration and decceleration mode.	207
SET_ACC	0x22	Sets acceleration and/or delay; valid as of the next positioning process.	209
SET_ACC_PARAM_UP	0x23	Set the Acc_ParamUp parameter for acceleration; valid as of the next positioning process.	210
SET_ACC_PARAM_ DOWN	0x24	Set the Acc_ParamDown parameter for delay (deceleration); valid as of the next positioning process.	211
SET_VELOCITY	0x25	Sets the positioning speed; valid as of the next positioning process.	212
SET_VELOCITY_ TARGET	0x2B	Velocity to target position. Valid only for the next positioning process and is then reset automatically to zero after the next positioning process.	213

Function	Opcode	Meaning	Page
SET_ACTUALPOSITON	0x2E	The current position is applied to the transferred value.	214
SET_ACTUALPOSITION_ ZERO	0x2F	The current position is set to zero	215
SET_CURRENT	0x39	Sets the motor current for drive movement.	216
VAR_SET	0x50	Sets one of the variables FILT1 ... FILT8 to the specified value	217
VAR_INC	0x51	Adds the specified value to one of the variables FILT1 … FILT8.	218
VAR_DEC	0x52	Subtracts the specified value from one of the variables FILT1 … FILT8.	219
VAR_ADD	0x53	Adds two variables and writes the result to a third variable	220
VAR_SUB	0x54	Subtracts two variables and writes the results to a third variable.	221
VAR_MUL	0x55	Multiplies one variable by another one and writes the results to a third variable.	222
VAR_COPY	0x56	Copes one variable to another variable.	223
VAR_DIV	0x57	Divides one variable by another one and writes the results to a third variable.	224
WAIT_TIME	0x70	Waits a defined time period before processing the next command.	225
WAIT_TEST_BIT	0x71	Before processing the next command waits until the specified bit has the status 0 or 1.	226
WR_BIT	0x78	Sets a bit to 0 or 1.	227
NOP	0xF0	No function	228
PROG_STOP	0xF1	Ends table processing.	229
GOTO	0xF5	Continues table process at the addressed entry.	231
GOTO_IF	0xF6	If a bit has been set, table processing is continued at the addressed entry; otherwise from the next entry.	232
GOTO_IF_NOT	0xF7	If a bit has been deleted, table processing is continued at the addressed entry; otherwise from the next entry.	233
GOTO_LABEL	0xF8	Continues table process from a defined label.	234
GOTO_LABEL_IF	0xF9	Continues table processing for a defined label if a bit has been set.	235
GOTO_LABEL_IF_NOT	0xFA	Continues table processing for a defined label if a bit has been deleted.	236
LABEL	0xFB	Defines a label for a step target	237

4.3.3 Overview of Move Mode Commands, Sorted by Function

Function	Opcode	Meaning	Page
GOTO	0xF5	Continues table process at the addressed entry.	231
GOTO_IF	0xF6	If a bit has been set, table processing is continued at the addressed entry; otherwise from the next entry.	232
GOTO_IF_NOT	0xF7	If a bit has been deleted, table processing is continued at the addressed entry; otherwise from the next entry.	233
GOTO_LABEL	0xF8	Continues table process from a defined label.	234
GOTO_LABEL_IF	0xF9	Continues table processing for a defined label if a bit has been set.	235
GOTO_LABEL_IF_NOT	0xFA	Continues table processing for a defined label if a bit has been deleted.	236
LABEL	0xFB	Defines a label for a step target	237
MOVE	0x02	Each MOVE command initiates a positioning process.	193
MOVE_IMMEDIATE	0x03	Each MOVE command initiates a positioning process.	194
MOVE_REL	0x06	Each MOVE command initiates a positioning process.	197
MOVE_TABLE	0x04	Each MOVE command initiates a positioning process.	195
MOVE_TABLE_ IMMEDIATE	0x05	Each MOVE command initiates a positioning process.	196
MOVE_TABLE_REL	0x08	Each MOVE command initiates a positioning process.	198
NOP	0xF0	No function	228
PROG_END	0x00 or 0xFF	End of table.	230
PROG_STOP	0xF1	Ends table processing.	229
SET_ACC	0x22	Sets acceleration and/or delay; valid as of the next positioning process.	209
SET_ACC_MODE	0x21	Sets the acceleration and decceleration mode.	207
SET_ACC_PARAM_ DOWN	0x24	Set the Acc_ParamDown parameter for delay (deceleration); valid as of the next positioning process.	211
SET_ACC_PARAM_UP	0x23	Set the Acc_ParamUp parameter for acceleration; valid as of the next positioning process.	210

Function	Opcode	Meaning	Page
SET_ACTUALPOSITION_ ZERO	0x2F	The current position is set to zero	215
SET_ACTUALPOSITON	0x2E	The current position is applied to the transferred value.	214
SET_CURRENT	0x39	Sets the motor current for drive movement.	216
SET_VELOCITY	0x25	Sets the positioning speed; valid as of the next positioning process.	212
SET_VELOCITY_ TARGET	0x2B	Velocity to target position. Valid only for the next positioning process and is then reset automatically to zero after the next positioning process.	213
SPEED	0x10	SPEED commands run the drive to a speed..	199
SPEED_IMMEDIATE	0x11	SPEED commands run the drive to a speed..	200
START_REFERENCING	0x20	Starts a reference run.	205
STOP_FAST	0x18	SPEED commands run the drive to a speed..	201
STOP_NO_RAMP	0x19	SPEED commands run the drive to a speed..	202
TORQ	0x1C	Each TORQ command begins a current setting (applies only to 750-673)	203
TORQ_IMM	0x1D	Each TORQ command begins a current setting (applies only to 750-673)	204
VAR_ADD	0x53	adds two variables and writes the result to a third variable	220
VAR_COPY	0x56	Copes one variable to another variable.	223
VAR_DEC	0x52	Subtracts the specified value from one of the variables FILT1 … FILT8.	219
VAR_DIV	0x57	Divides one variable by another one and writes the results to a third variable.	224
VAR_INC	0x51	Adds the specified value to one of the variables FILT1 … FILT8.	218
VAR_MUL	0x55	Multiplies one variable by another one and writes the results to a third variable.	222
VAR_SET	0x50	Sets one of the variables FILT1 ... FILT8 to the specified value	217
VAR_SUB	0x54	Subtracts two variables and writes the results to a third variable.	221
WAIT_TEST_BIT	0x71	Before processing the next command waits until the specified bit has the status 0 or 1.	226

Function	Opcode	Meaning	Page
WAIT_TIME	0x70	Waits a defined time period before processing the next command.	225
WR_BIT	0x78	Sets a bit to 0 or 1.	227

4.3.4 Reference Commands for Move Mode

4.3.4.1 Set point commands

4.3.4.1.1 MOVE (0x02)

Starts a positioning process. This command immediately deletes the "On_Target" bit and sets that bit when the target position is reached. This command is accepted only when the last positioning process has been completed and the "On_Target" bit has been set. The acceleration and velocity specified by SET_ACC (0x22) and SET_VELOCITY (0x25) are used for positioning commands.

Request								
Byte	**2^7**	**2^6**	**2^5**	**2^4**	**2^3**	**2^2**	**2^1**	**2^0**
MB0	**0x40**							
MB1	T	-						
MB2	**0x02**							
MB3	Position (LSB)							
MB4	Position							
MB5	Position (MSB)							

Response								
Byte	**2^7**	**2^6**	**2^5**	**2^4**	**2^3**	**2^2**	**2^1**	**2^0**
MB0	**0x40**							
MB1	T	Return Code						
MB2	**0x02**							
MB3	Reserved							
MB4	Reserved							
MB5	Reserved							

4.3.4.1.2 MOVE_IMMEDIATE (0x03)

Starts a positioning process. This command immediately deletes the "On_Target" bit and sets that bit when the target position is reached. This command will interrupt any positioning process that may already be in progress and immediately starts a new positioning process. The acceleration and velocity specified by SET_ACC (0x22) and SET_VELOCITY (0x25) are used for positioning commands.

Request								
Byte	**2^7**	**2^6**	**2^5**	**2^4**	**2^3**	**2^2**	**2^1**	**2^0**
MB0	**0x40**							
MB1	T	-						
MB2	**0x03**							
MB3	Position (LSB)							
MB4	Position							
MB5	Position (MSB)							

Response								
Byte	**2^7**	**2^6**	**2^5**	**2^4**	**2^3**	**2^2**	**2^1**	**2^0**
MB0	**0x40**							
MB1	T	Return Code						
MB2	**0x03**							
MB3	Reserved							
MB4	Reserved							
MB5	Reserved							

4.3.4.1.3 MOVE_TABLE (0x04)

Starts a positioning process. This command immediately deletes the "On_Target" bit and sets that bit when the target position is reached. This command is accepted only when the last positioning process has been completed and the "On_Target" bit has been set. This command reads the target position from the specified location in a separate position table. The acceleration and velocity specified by SET_ACC (0x22) and SET_VELOCITY (0x25) are used for positioning commands.

Request								
Byte	**2^7**	**2^6**	**2^5**	**2^4**	**2^3**	**2^2**	**2^1**	**2^0**
MB0	**0x40**							
MB1	T	-						
MB2	**0x04**							
MB3	No. of table entry with target position							
MB4	Reserved							SRC
MB5	Reserved							

SRC 0: Read out from position table.

1: Read out from variables FILT1 ... FILT8.

Response								
Byte	**2^7**	**2^6**	**2^5**	**2^4**	**2^3**	**2^2**	**2^1**	**2^0**
MB0	**0x40**							
MB1	T	Return Code						
MB2	**0x04**							
MB3	Reserved							
MB4	Reserved							
MB5	Reserved							

4.3.4.1.4 MOVE_TABLE_IMMEDIATE (0x05)

Starts a positioning process. This command immediately deletes the "On_Target" bit and sets that bit when the target position is reached. This command will interrupt any positioning process that may already be in progress and immediately starts a new positioning process. This command reads the target position from the specified location in a separate position table. The acceleration and velocity specified by SET_ACC (0x22) and SET_VELOCITY (0x25) are used for positioning commands.

Request								
Byte	**2^7**	**2^6**	**2^5**	**2^4**	**2^3**	**2^2**	**2^1**	**2^0**
MB0	**0x40**							
MB1	T	-						
MB2	**0x05**							
MB3	No. of table entry with target position							
MB4	Reserved							SRC
MB5	Reserved							

SRC	0:	Read out from position table.
	1:	Read out from variables FILT1 ... FILT8.

Response								
Byte	**2^7**	**2^6**	**2^5**	**2^4**	**2^3**	**2^2**	**2^1**	**2^0**
MB0	**0x40**							
MB1	T	Return Code						
MB2	**0x05**							
MB3	Reserved							
MB4	Reserved							
MB5	Reserved							

4.3.4.1.5 MOVE_REL (0x06)

Starts a positioning process. This command immediately deletes the "On_Target" bit and sets that bit when the target position is reached. This command is accepted only when the last positioning process has been completed and the "On_Target" bit has been set. This command calculates the target position relative to the last accepted target (if available), or to the current position. The acceleration and velocity specified by SET_ACC (0x22) and SET_VELOCITY (0x25) are used for positioning commands.

Request								
Byte	**2^7**	**2^6**	**2^5**	**2^4**	**2^3**	**2^2**	**2^1**	**2^0**
MB0	**0x40**							
MB1	T	-						
MB2	**0x06**							
MB3	Position (LSB)							
MB4	Position							
MB5	Position (MSB)							

Response								
Byte	**2^7**	**2^6**	**2^5**	**2^4**	**2^3**	**2^2**	**2^1**	**2^0**
MB0	**0x40**							
MB1	T	Return Code						
MB2	**0x06**							
MB3	Reserved							
MB4	Reserved							
MB5	Reserved							

4.3.4.1.6 MOVE_TABLE_REL (0x08)

Starts a positioning process. This command immediately deletes the "On_Target" bit and sets that bit when the target position is reached. This command is accepted only when the last positioning process has been completed and the "On_Target" bit has been set. This command reads the target position from the specified location in a separate position table. This command calculates the target position relative to the last accepted target (if available), or to the current position. The acceleration and velocity specified by SET_ACC (0x22) and SET_VELOCITY (0x25) are used for positioning commands.

Request								
Byte	**2^7**	**2^6**	**2^5**	**2^4**	**2^3**	**2^2**	**2^1**	**2^0**
MB0	**0x40**							
MB1	T	-						
MB2	**0x08**							
MB3	No. of table entry with target position							
MB4	Reserved							SRC
MB5	Reserved							

SRC	0:	Read out from position table.
	1:	Read out from variables FILT1 ... FILT8.

Response								
Byte	**2^7**	**2^6**	**2^5**	**2^4**	**2^3**	**2^2**	**2^1**	**2^0**
MB0	**0x40**							
MB1	T	Return Code						
MB2	**0x08**							
MB3	Reserved							
MB4	Reserved							
MB5	Reserved							

4.3.4.1.7 SPEED (0x10)

Runs the drive up to a defined speed. On execution of this command, the "On_Target" bit is deleted immediately and then set when the target speed is reached. This command does NOT set the positioning speed! SET_VELOCITY (0x25) must be used for that. The SPEED command is accepted only when the last process has been completed and the "On_Target" bit has been set.

Velocity range: -25000 … 25000.

Request								
Byte	**2^7**	**2^6**	**2^5**	**2^4**	**2^3**	**2^2**	**2^1**	**2^0**
MB0	**0x40**							
MB1	T	-						
MB2	**0x10**							
MB3	Velocity (LSB)							
MB4	Velocity (MSB)							
MB5	Reserved							

Response								
Byte	**2^7**	**2^6**	**2^5**	**2^4**	**2^3**	**2^2**	**2^1**	**2^0**
MB0	**0x40**							
MB1	T	Return Code						
MB2	**0x10**							
MB3	Reserved							
MB4	Reserved							
MB5	Reserved							

4.3.4.1.8 SPEED_IMMEDIATE (0x11)

Runs the drive up to a defined speed. On execution of this command, the "On_Target" bit is deleted immediately and then set when the target speed is reached. This command does NOT set the positioning speed! SET_VELOCITY (0x25) must be used for that. This command will interrupt any process that may already be in operation and immediately starts speed control.

Velocity range: -25000 … 25000.

Request								
Byte	**2^7**	**2^6**	**2^5**	**2^4**	**2^3**	**2^2**	**2^1**	**2^0**
MB0	**0x40**							
MB1	T	-						
MB2	**0x11**							
MB3	Velocity (LSB)							
MB4	Velocity (MSB)							
MB5	Reserved							

Response								
Byte	**2^7**	**2^6**	**2^5**	**2^4**	**2^3**	**2^2**	**2^1**	**2^0**
MB0	**0x40**							
MB1	T	Return Code						
MB2	**0x11**							
MB3	Reserved							
MB4	Reserved							
MB5	Reserved							

4.3.4.1.9 STOP_FAST (0x18)

Brakes the drive directly with the acceleration command SET_ACC_STOP down to standstill. The internal processing has been designed for this command to be processed in priority. Especially in mailbox mode, every other command is cancelled immediately. This command is also initiated internally when a stop condition is present, such as limit switch or stop input.

Request								
Byte	**2^7**	**2^6**	**2^5**	**2^4**	**2^3**	**2^2**	**2^1**	**2^0**
MB0	**0x40**							
MB1	T	-						
MB2	**0x18**							
MB3	Reserved							
MB4	Reserved							
MB5	Reserved							

Response								
Byte	**2^7**	**2^6**	**2^5**	**2^4**	**2^3**	**2^2**	**2^1**	**2^0**
MB0	**0x40**							
MB1	T	Return Code						
MB2	**0x18**							
MB3	Reserved							
MB4	Reserved							
MB5	Reserved							

4.3.4.1.10 STOP_NO_RAMP (0x19)

Sets the output frequency immediately to zero. The internal processing has been designed for this command to be processed in priority. Especially in mailbox mode, every other command is cancelled immediately. This command has priority over STOP_FAST. This command is also triggered internally, if release is not set.

Request								
Byte	**2^7**	**2^6**	**2^5**	**2^4**	**2^3**	**2^2**	**2^1**	**2^0**
MB0	**0x40**							
MB1	T	-						
MB2	**0x19**							
MB3	Reserved							
MB4	Reserved							
MB5	Reserved							

Response								
Byte	**2^7**	**2^6**	**2^5**	**2^4**	**2^3**	**2^2**	**2^1**	**2^0**
MB0	**0x40**							
MB1	T	Return Code						
MB2	**0x19**							
MB3	Reserved							
MB4	Reserved							
MB5	Reserved							

4.3.4.1.11 TORQ (0x1C)

This function exists at servo stepper controller 750-673.

Begins a current setting. The speed regulator is given a target velocity. As long as the target velocity is not achieved, the current setpoint generates a torque whose sign depends on the speed error.
If the target velocity is achieved, the speed regulator will automatically reduce the current. Because the current is regulated immediately, the bit "On_Target" is set immediately.
This command is accepted only when the last positioning process has been completed and the "On_Target" bit has been set.

Velocity range: -25000 … 25000.

Request								
Byte	**2^7**	**2^6**	**2^5**	**2^4**	**2^3**	**2^2**	**2^1**	**2^0**
MB0	**0x40**							
MB1	T	-						
MB2	**0x1C**							
MB3	Velocity (LSB)							
MB4	Velocity (MSB)							
MB5	Current setting (0 … 150 %)							

Response								
Byte	**2^7**	**2^6**	**2^5**	**2^4**	**2^3**	**2^2**	**2^1**	**2^0**
MB0	**0x40**							
MB1	T	Return Code						
MB2	**0x1C**							
MB3	Reserved							
MB4	Reserved							
MB5	Reserved							

4.3.4.1.12 TORQ_IMM (0x1D)

This function exists at servo stepper controller 750-673.

Begins a current setting. The speed regulator is given a target velocity. As long as the target velocity is not achieved, the current setpoint generates a torque whose sign depends on the speed error.
If the target velocity is achieved, the speed regulator will automatically reduce the current. Because the current is regulated immediately, the bit "On_Target" is set immediately. This command will interrupt any positioning process that may already be in progress and immediately begins a new current setting.

Velocity range: -25000 ... 25000.

Request								
Byte	**2^7**	**2^6**	**2^5**	**2^4**	**2^3**	**2^2**	**2^1**	**2^0**
MB0	**0x40**							
MB1	T	-						
MB2	**0x1D**							
MB3	Velocity (LSB)							
MB4	Velocity (MSB)							
MB5	Current setting (0 … 150 %)							

Response								
Byte	**2^7**	**2^6**	**2^5**	**2^4**	**2^3**	**2^2**	**2^1**	**2^0**
MB0	**0x40**							
MB1	T	Return Code						
MB2	**0x1D**							
MB3	Reserved							
MB4	Reserved							
MB5	Reserved							

4.3.4.1.13 START_REFERENCING (0x20)

This function exists at servo stepper controller 750-673.

Starts a reference run. This command immediately deletes the bits On_Target and Reference_OK. When the reference point is reached, Reference_OK is set. The On_Target bit is not set. The Busy bit is set during a reference run.

Request								
Byte	**2^7**	**2^6**	**2^5**	**2^4**	**2^3**	**2^2**	**2^1**	**2^0**
MB0	**0x40**							
MB1	T	-						
MB2	**0x20**							
MB3	Reserved							
MB4	Reserved		EMS	EMF	EMR		BAM	
MB5	Reserved					DIR	STD	SWT

BAM Selection motor alignment

0: Normal reference run

1: Reserved

10: Motor alignment followed of reference run

11: Only motor alignment

EMR If Bit 0 ... 1 not 0, then Bit 2 ... 7 configured the motor alignment

0: Motor alignment; the motor moves up to ± 1 full step

1: Motor alignment based on stored measured value and Z-impuls. In addition the motor makes a movement up to ±360°.

10: Reserved

11: Reserved

EMF 0: Automatic: Motor alignment only if yet does not take place.

1: Motor alignment always.

EMS 0: The measured value is not stored after an motor alignment (normal operation).

1: The measured value is stored after an motor alignment (should take place only during start-up).

SWT 0: Reference run to reference switch:

1: Reference run to limit switch:

STD If SWT = 0, then STD indicates the starting direction, if SWT = 1, then STD specifies the limit switch.

0: Starting direction negative / negative limit switch:

1: Starting direction positive / positive limit switch:

DIR DIR is evaluated only when SWT = 1.

0: Reference run started from negative end.

1: Reference run started from positive end.

Response								
Byte	**2^7**	**2^6**	**2^5**	**2^4**	**2^3**	**2^2**	**2^1**	**2^0**
MB0	**0x40**							
MB1	T	Return Code						
MB2	**0x20**							
MB3	Reserved							
MB4	Reserved							
MB5	Reserved							

4.3.4.1.14 SET_ACC_MODE (0x21)

Sets acceleration and delay type, valid from next positioning process

Request								
Byte	**2^7**	**2^6**	**2^5**	**2^4**	**2^3**	**2^2**	**2^1**	**2^0**
MB0	**0x40**							
MB1	T	-						
MB2	**0x21**							
MB3	DEC_M		DEC_T		ACC_M		ACC_T	
MB4	Reserved							
MB5	Reserved							

ACC_T 0: constant acceleration

1: Linear increase in acceleration
With ACC_M = 0, the period for acceleration increase is Acc_ParamUp, the increase in acceleration is calculated at ACC_M <> 0.

2: Sin^2 increase in acceleration
With ACC_M = 0, the period for acceleration increase is Acc_ParamUp, the increase in acceleration is calculated at ACC_M <> 0.

3: Reserved

ACC_M 0: no modification

1: Acc_ParamUp interpreted as the acceleration period

2: Acc_ParamUp interpreted as the acceleration path

3: Reserved

DEC_T 0: constant deceleration

1: Linear deceleration
With DEC_M = 0, the period for deceleration is Acc_ParamDown, the increase in deceleration is calculated at DEC_M <> 0.

2: Sin^2 deceleration
With DEC_M = 0, the period for deceleration is Acc_ParamDown, the increase in deceleration is calculated at DEC_M <> 0.

3: Reserved

DEC_M 0: no modification

1: Acc_ParamDown interpreted as the decceleration period

2: Acc_ParamDown interpreted as the decceleration path

3: Reserved

Response								
Byte	**2^7**	**2^6**	**2^5**	**2^4**	**2^3**	**2^2**	**2^1**	**2^0**
MB0	**0x40**							
MB1	T	Return Code						
MB2	**0x21**							
MB3	Reserved							
MB4	Reserved							
MB5	Reserved							

4.3.4.1.15 SET_ACC (0x22)

Sets the type and rate of acceleration; valid as of the next positioning process (see also positioning commands MOVE…, 0x02, 0x03, 0x04, 0x05, 0x06, 0x08)

Acceleration range: 1 … 32767.

Request								
Byte	**2^7**	**2^6**	**2^5**	**2^4**	**2^3**	**2^2**	**2^1**	**2^0**
MB0	**0x40**							
MB1	T	-						
MB2	**0x22**							
MB3	Acceleration (LSB)							
MB4	Acceleration (MSB)							
MB5	Reserved						SEL	

SEL	0:	Sets the value for acceleration and brake phase.
	1:	Sets the value for acceleration phase only.
	2:	Sets the value for brake phase only.
	3:	Sets the value for acceleration and brake phase.

Response								
Byte	**2^7**	**2^6**	**2^5**	**2^4**	**2^3**	**2^2**	**2^1**	**2^0**
MB0	**0x40**							
MB1	T	Return Code						
MB2	**0x22**							
MB3	Reserved							
MB4	Reserved							
MB5	Reserved							

4.3.4.1.16 SET_ACC_PARAM_UP (0x23)

Set the Acc_ParamUp parameter for acceleration; valid as of the next positioning process.

The function for the acceleration parameter as a function of the set acceleration modification is shown in the following table.

Acceleration modification (SET_ACC_MODE → ACC_M)	**Acceleration parameter Acc_ParamUp**
none	Time constant for acceleration increase with linear or $\sin^2$*t acceleration
constant acceleration period	Acceleration time
constant acceleration path	Acceleration path

Acceleration parameter range: 1–16777215.

Request								
Byte	**2^7**	**2^6**	**2^5**	**2^4**	**2^3**	**2^2**	**2^1**	**2^0**
MB0	**0x40**							
MB1	T	-						
MB2	**0x23**							
MB3	Acceleration parameter (LSB)							
MB4	Acceleration parameter							
MB5	Acceleration parameter (MSB)							

Response								
Byte	**2^7**	**2^6**	**2^5**	**2^4**	**2^3**	**2^2**	**2^1**	**2^0**
MB0	**0x40**							
MB1	T	Return Code						
MB2	**0x23**							
MB3	Reserved							
MB4	Reserved							
MB5	Reserved							

4.3.4.1.17 SET_ACC_PARAM_DOWN (0x24)

Set the Acc_ParamDown parameter for delay (deceleration); valid as of the next positioning process.

The function for the delay parameter as a function of the set delay modification is shown in the following table.

Deceleration modification (SET_ACC_MODE → DEC_M)	**Delay parameter Acc_ParamDown**
none	Time constant for decceleration increase with linear or sin^2*t deceleration
constant delay period	Delay time
constant delay path	Delay path

Request								
Byte	**2^7**	**2^6**	**2^5**	**2^4**	**2^3**	**2^2**	**2^1**	**2^0**
MB0	**0x40**							
MB1	T	-						
MB2	**0x24**							
MB3	Decceleration parameter (LSB)							
MB4	Decceleration parameter							
MB5	Decceleration parameter (MSB)							

Response								
Byte	**2^7**	**2^6**	**2^5**	**2^4**	**2^3**	**2^2**	**2^1**	**2^0**
MB0	**0x40**							
MB1	T	Return Code						
MB2	**0x24**							
MB3	Reserved							
MB4	Reserved							
MB5	Reserved							

4.3.4.1.18 SET_VELOCITY (0x25)

Sets the positioning velocity; valid as of the next positioning command (see also positioning commands MOVE…, 0x02, 0x03, 0x04, 0x05, 0x06, 0x08)

Velocity range: 1 .. 25000.

Request								
Byte	**2^7**	**2^6**	**2^5**	**2^4**	**2^3**	**2^2**	**2^1**	**2^0**
MB0	**0x40**							
MB1	T	-						
MB2	**0x25**							
MB3	Velocity (LSB)							
MB4	Velocity (MSB)							
MB5	Reserved							

Response								
Byte	**2^7**	**2^6**	**2^5**	**2^4**	**2^3**	**2^2**	**2^1**	**2^0**
MB0	**0x40**							
MB1	T	Return Code						
MB2	**0x25**							
MB3	Reserved							
MB4	Reserved							
MB5	Reserved							

4.3.4.1.19 SET_VELOCITY_TARGET (0x2B)

Sets the target velocity for the next positioning process. The target velocity is automatically reset to zero after the next positioning process.

Velocity range: 1 .. 25000.

Request								
Byte	**2^7**	**2^6**	**2^5**	**2^4**	**2^3**	**2^2**	**2^1**	**2^0**
MB0	**0x40**							
MB1	T	-						
MB2	**0x2B**							
MB3	Velocity (LSB)							
MB4	Velocity (MSB)							
MB5	Reserved							

Response								
Byte	**2^7**	**2^6**	**2^5**	**2^4**	**2^3**	**2^2**	**2^1**	**2^0**
MB0	**0x40**							
MB1	T	Return Code						
MB2	**0x2B**							
MB3	Reserved							
MB4	Reserved							
MB5	Reserved							

4.3.4.1.20 SET_ACTUALPOSITON (0x2E)

The current position is set to the transferred value. For this, the logical point Zero is modified accordingly.

The position is given as a 24-bit value, including sign.

Request								
Byte	**2^7**	**2^6**	**2^5**	**2^4**	**2^3**	**2^2**	**2^1**	**2^0**
MB0	**0x40**							
MB1	T	-						
MB2	**0x2E**							
MB3	Position (LSB)							
MB4	Position							
MB5	Position (MSB)							

Response								
Byte	**2^7**	**2^6**	**2^5**	**2^4**	**2^3**	**2^2**	**2^1**	**2^0**
MB0	**0x40**							
MB1	T	Return Code						
MB2	**0x2E**							
MB3	Reserved							
MB4	Reserved							
MB5	Reserved							

4.3.4.1.21 SET_ACTUALPOSITION_ZERO (0x2F)

Sets the position of the logical zero point to the current position.

Request								
Byte	**2^7**	**2^6**	**2^5**	**2^4**	**2^3**	**2^2**	**2^1**	**2^0**
MB0	**0x40**							
MB1	T	-						
MB2	**0x2F**							
MB3	Reserved							
MB4	Reserved							
MB5	Reserved							

Response								
Byte	**2^7**	**2^6**	**2^5**	**2^4**	**2^3**	**2^2**	**2^1**	**2^0**
MB0	**0x40**							
MB1	T	Return Code						
MB2	**0x2F**							
MB3	Reserved							
MB4	Reserved							
MB5	Reserved							

4.3.4.1.22 SET_CURRENT (0x39)

Sets the motor current for drive movement.

The corresponding bit in the valid range must be set to 1 for the working range for which the motor current is to be set. Several bits may be set simultaneously. If the corresponding bit is set to 0, the value for the motor current valid up to then is retained for this range.

Motor current range: 0 … 150 % module rated current

Request								
Byte	**2^7**	**2^6**	**2^5**	**2^4**	**2^3**	**2^2**	**2^1**	**2^0**
MB0	**0x40**							
MB1	T	-						
MB2	**0x39**							
MB3	Motor current							
MB4	Reserved				Scope of Validity			
MB5	Reserved							

Valid range	Bit 0:	Set motor current for standstill
	Bit 1:	Set motor current for acceleration
	Bit 2:	Set motor current for drive movement
	Bit 3:	Set motor current for deceleration

Response								
Byte	**2^7**	**2^6**	**2^5**	**2^4**	**2^3**	**2^2**	**2^1**	**2^0**
MB0	**0x40**							
MB1	T	Return Code						
MB2	**0x39**							
MB3	Reserved							
MB4	Reserved							
MB5	Reserved							

4.3.4.2 Mathmatic Commands

4.3.4.2.1 VAR_SET (0x50)

Sets a variable to the defined value.

Request								
Byte	**2^7**	**2^6**	**2^5**	**2^4**	**2^3**	**2^2**	**2^1**	**2^0**
MB0	**0x40**							
MB1	T	-						
MB2	**0x50**							
MB3	1 … 8 (corresponds to FILT1 … FILT8)							
MB4	16 bit value (LSB)							
MB5	16 bit value (MSB)							

Response								
Byte	**2^7**	**2^6**	**2^5**	**2^4**	**2^3**	**2^2**	**2^1**	**2^0**
MB0	**0x40**							
MB1	T	Return Code						
MB2	**0x50**							
MB3	Reserved							
MB4	Reserved							
MB5	Reserved							

4.3.4.2.2 VAR_INC (0x51)

Adds the given value to a variable.

Request								
Byte	**2^7**	**2^6**	**2^5**	**2^4**	**2^3**	**2^2**	**2^1**	**2^0**
MB0	**0x40**							
MB1	T	-						
MB2	**0x51**							
MB3	1 … 8 (corresponds to FILT1 … FILT8)							
MB4	16 bit value (LSB)							
MB5	16 bit value (MSB)							

Response								
Byte	**2^7**	**2^6**	**2^5**	**2^4**	**2^3**	**2^2**	**2^1**	**2^0**
MB0	**0x40**							
MB1	T	Return Code						
MB2	**0x51**							
MB3	Reserved							
MB4	Reserved							
MB5	Reserved							

4.3.4.2.3 VAR_DEC (0x52)

Subtracts the given value from a variable.

Request								
Byte	**2^7**	**2^6**	**2^5**	**2^4**	**2^3**	**2^2**	**2^1**	**2^0**
MB0	**0x40**							
MB1	T	-						
MB2	**0x52**							
MB3	1 … 8 (corresponds to FILT1 … FILT8)							
MB4	16 bit value (LSB)							
MB5	16 bit value (MSB)							

Response								
Byte	**2^7**	**2^6**	**2^5**	**2^4**	**2^3**	**2^2**	**2^1**	**2^0**
MB0	**0x40**							
MB1	T	Return Code						
MB2	**0x52**							
MB3	Reserved							
MB4	Reserved							
MB5	Reserved							

4.3.4.2.4 VAR_ADD (0x53)

Adds two variables and writes the results to a third variable.

Request								
Byte	**2^7**	**2^6**	**2^5**	**2^4**	**2^3**	**2^2**	**2^1**	**2^0**
MB0	**0x40**							
MB1	T	-						
MB2	**0x53**							
MB3	Result (1 … 8 corresponds to FILT1 … FILT8)							
MB4	Summand 2 (1 … 8 corresponds to FILT1 … FILT8)							
MB5	Summand 1 (1 … 8 corresponds to FILT1 … FILT8)							

Response								
Byte	**2^7**	**2^6**	**2^5**	**2^4**	**2^3**	**2^2**	**2^1**	**2^0**
MB0	**0x40**							
MB1	T	Return Code						
MB2	**0x53**							
MB3	Reserved							
MB4	Reserved							
MB5	Reserved							

4.3.4.2.5 VAR_SUB (0x54)

Subtracts one variable from another one and writes the results to a third variable.

Request								
Byte	**2^7**	**2^6**	**2^5**	**2^4**	**2^3**	**2^2**	**2^1**	**2^0**
MB0	**0x40**							
MB1	T	-						
MB2	**0x54**							
MB3	Difference (1 … 8 corresponds to FILT1 … FILT8)							
MB4	Minuend (1 … 8 corresponds to FILT1 … FILT8)							
MB5	Subtrahend (1 … 8 corresponds to FILT1 … FILT8)							

Response								
Byte	**2^7**	**2^6**	**2^5**	**2^4**	**2^3**	**2^2**	**2^1**	**2^0**
MB0	**0x40**							
MB1	T	Return Code						
MB2	**0x54**							
MB3	Reserved							
MB4	Reserved							
MB5	Reserved							

4.3.4.2.6 VAR_MUL (0x55)

Multiplies one variable by another one and writes the results to a third variable.

Request								
Byte	**2^7**	**2^6**	**2^5**	**2^4**	**2^3**	**2^2**	**2^1**	**2^0**
MB0	**0x40**							
MB1	T	-						
MB2	**0x55**							
MB3	Product (1 … 8 corresponds to FILT1 … FILT8)							
MB4	Multiplicand 2 (1 … 8 corresponds to FILT1 … FILT8)							
MB5	Multiplicand 1 (1 … 8 corresponds to FILT1 … FILT8)							

Response								
Byte	**2^7**	**2^6**	**2^5**	**2^4**	**2^3**	**2^2**	**2^1**	**2^0**
MB0	**0x40**							
MB1	T	Return Code						
MB2	**0x55**							
MB3	Reserved							
MB4	Reserved							
MB5	Reserved							

4.3.4.2.7 VAR_COPY (0x56)

Copes one variable to another variable.

Request								
Byte	**2^7**	**2^6**	**2^5**	**2^4**	**2^3**	**2^2**	**2^1**	**2^0**
MB0	**0x40**							
MB1	T	-						
MB2	**0x56**							
MB3	Target (1 … 8 corresponds to FILT1 … FILT8)							
MB4	Source (1 … 8 corresponds to FILT1 … FILT8)							
MB5	Reserved							

Response								
Byte	**2^7**	**2^6**	**2^5**	**2^4**	**2^3**	**2^2**	**2^1**	**2^0**
MB0	**0x40**							
MB1	T	Return Code						
MB2	**0x56**							
MB3	Reserved							
MB4	Reserved							
MB5	Reserved							

4.3.4.2.8 VAR_DIV (0x57)

Divides one variable by another one and writes the results to a third variable.

Request								
Byte	**2^7**	**2^6**	**2^5**	**2^4**	**2^3**	**2^2**	**2^1**	**2^0**
MB0	**0x40**							
MB1	T	-						
MB2	**0x57**							
MB3	Quotient (1 … 8 corresponds to FILT1 … FILT8)							
MB4	Dividend (1 … 8 corresponds to FILT1 … FILT8)							
MB5	Divisor (1 … 8 corresponds to FILT1 … FILT8)							

Response								
Byte	**2^7**	**2^6**	**2^5**	**2^4**	**2^3**	**2^2**	**2^1**	**2^0**
MB0	**0x40**							
MB1	T	Return Code						
MB2	**0x57**							
MB3	Reserved							
MB4	Reserved							
MB5	Reserved							

4.3.4.3 Wait Commands

4.3.4.3.1 WAIT_TIME (0x70)

Waits for some time before processing the next command

Waiting time range: 0 … 16777215 ms.

Request								
Byte	**2^7**	**2^6**	**2^5**	**2^4**	**2^3**	**2^2**	**2^1**	**2^0**
MB0	**0x40**							
MB1	T	-						
MB2	**0x70**							
MB3	Waiting time (LSB)							
MB4	Waiting time							
MB5	Waiting time (MSB)							

Response								
Byte	**2^7**	**2^6**	**2^5**	**2^4**	**2^3**	**2^2**	**2^1**	**2^0**
MB0	**0x40**							
MB1	T	Return Code						
MB2	**0x70**							
MB3	Reserved							
MB4	Reserved							
MB5	Reserved							

4.3.4.3.2 WAIT_TEST_BIT (0x71)

Before processing the next command waits until the specified bit has the specified status 0 or 1.
Refer to chapter 4.5, "Bit Field for I/O Driver" for the bit number.

Request								
Byte	**2^7**	**2^6**	**2^5**	**2^4**	**2^3**	**2^2**	**2^1**	**2^0**
MB0	**0x40**							
MB1	T	-						
MB2	**0x71**							
MB3	Bit No.							
MB4	Specified status of bit (0 or 1)							
MB5	Reserved							

Response								
Byte	**2^7**	**2^6**	**2^5**	**2^4**	**2^3**	**2^2**	**2^1**	**2^0**
MB0	**0x40**							
MB1	T	Return Code						
MB2	**0x71**							
MB3	Reserved							
MB4	Reserved							
MB5	Reserved							

4.3.4.4 Auxiliary commands

4.3.4.4.1 WR_BIT (0x78)

Sets bit to 0 or 1.
Refer to chapter 4.5, „Bit Field for I/O Driver“ for the bit number.

Request								
Byte	**2^7**	**2^6**	**2^5**	**2^4**	**2^3**	**2^2**	**2^1**	**2^0**
MB0	**0x40**							
MB1	T	-						
MB2	**0x78**							
MB3	Bit No.							
MB4	Specified status of bit (0 or 1)							
MB5	Reserved							

Response								
Byte	**2^7**	**2^6**	**2^5**	**2^4**	**2^3**	**2^2**	**2^1**	**2^0**
MB0	**0x40**							
MB1	T	Return Code						
MB2	**0x78**							
MB3	Reserved							
MB4	Reserved							
MB5	Reserved							

4.3.4.4.2 NOP (0xF0)

Function not defined.

Request								
Byte	**2^7**	**2^6**	**2^5**	**2^4**	**2^3**	**2^2**	**2^1**	**2^0**
MB0	**0x40**							
MB1	T	-						
MB2	**0xF0**							
MB3	Reserved							
MB4	Reserved							
MB5	Reserved							

Response								
Byte	**2^7**	**2^6**	**2^5**	**2^4**	**2^3**	**2^2**	**2^1**	**2^0**
MB0	**0x40**							
MB1	T	Return Code						
MB2	**0xF0**							
MB3	Reserved							
MB4	Reserved							
MB5	Reserved							

4.3.4.4.3 PROG_STOP (0xF1)

Ends the table processing. Sets speed to Zero, deactivates final step, ends table processing.

Request								
Byte	**2^7**	**2^6**	**2^5**	**2^4**	**2^3**	**2^2**	**2^1**	**2^0**
MB0	**0x40**							
MB1	T	-						
MB2	**0xF1**							
MB3	Error message							
MB4	Reserved							
MB5	Reserved							

Error message	0:	No error message
	1 ... 8:	Error message ERROR_TBL_PROGRAM_STOP1 ... 8
	9 ... 255	Reserved

Response								
Byte	**2^7**	**2^6**	**2^5**	**2^4**	**2^3**	**2^2**	**2^1**	**2^0**
MB0	**0x40**							
MB1	T	Return Code						
MB2	**0xF1**							
MB3	Reserved							
MB4	Reserved							
MB5	Reserved							

4.3.4.4.4 PROG_END (0x00 or 0xFF)

End of table (default command for a blank / deleted table). Sets speed to Zero, deactivates the final step, ends table processing and indicates an error ERR_PROG_END.

Request								
Byte	**2^7**	**2^6**	**2^5**	**2^4**	**2^3**	**2^2**	**2^1**	**2^0**
MB0	**0x40**							
MB1	T	-						
MB2	**0x00 or 0xFF**							
MB3	Reserved							
MB4	Reserved							
MB5	Reserved							

Response								
Byte	**2^7**	**2^6**	**2^5**	**2^4**	**2^3**	**2^2**	**2^1**	**2^0**
MB0	**0x40**							
MB1	T	Return Code						
MB2	**0x00 or 0xFF**							
MB3	Reserved							
MB4	Reserved							
MB5	Reserved							

4.3.4.4.5 GOTO (0xF5)

Continues table processing at the addressed entry.

Command number range: 1 … 500

Request								
Byte	**2^7**	**2^6**	**2^5**	**2^4**	**2^3**	**2^2**	**2^1**	**2^0**
MB0	**0x40**							
MB1	T	-						
MB2	**0xF5**							
MB3	Number of next command (LSB)							
MB4	Number of next command (MSB)							
MB5	Reserved							

Response								
Byte	**2^7**	**2^6**	**2^5**	**2^4**	**2^3**	**2^2**	**2^1**	**2^0**
MB0	**0x40**							
MB1	T	Return Code						
MB2	**0xF5**							
MB3	Reserved							
MB4	Reserved							
MB5	Reserved							

4.3.4.4.6 GOTO_IF (0xF6)

If a bit has been set, table processing is continued at the addressed entry; otherwise the next table entry is used.

Command number range: 1 … 500

Request								
Byte	**2^7**	**2^6**	**2^5**	**2^4**	**2^3**	**2^2**	**2^1**	**2^0**
MB0	**0x40**							
MB1	T	-						
MB2	**0xF6**							
MB3	Number of next command (LSB)							
MB4	Number of next command (MSB)							
MB5	Number of bit to be checked							

Response								
Byte	**2^7**	**2^6**	**2^5**	**2^4**	**2^3**	**2^2**	**2^1**	**2^0**
MB0	**0x40**							
MB1	T	Return Code						
MB2	**0xF6**							
MB3	Reserved							
MB4	Reserved							
MB5	Reserved							

4.3.4.4.7 GOTO_IF_NOT (0xF7)

If a bit has not been set, table processing is continued at the addressed entry; otherwise the next table entry is used.

Command number range: 1 … 500

Request								
Byte	**2^7**	**2^6**	**2^5**	**2^4**	**2^3**	**2^2**	**2^1**	**2^0**
MB0	**0x40**							
MB1	T	-						
MB2	**0xF7**							
MB3	Number of next command (LSB)							
MB4	Number of next command (MSB)							
MB5	Number of bit to be checked							

Response								
Byte	**2^7**	**2^6**	**2^5**	**2^4**	**2^3**	**2^2**	**2^1**	**2^0**
MB0	**0x40**							
MB1	T	Return Code						
MB2	**0xF7**							
MB3	Reserved							
MB4	Reserved							
MB5	Reserved							

4.3.4.4.8 GOTO_LABEL (0xF8)

Continues table processing at the addressed entry.

Label number range: 1 … 65536

Request								
Byte	**2^7**	**2^6**	**2^5**	**2^4**	**2^3**	**2^2**	**2^1**	**2^0**
MB0	**0x40**							
MB1	T	-						
MB2	**0xF8**							
MB3	Label number (LSB)							
MB4	Label number (MSB)							
MB5	Reserved							

Response								
Byte	**2^7**	**2^6**	**2^5**	**2^4**	**2^3**	**2^2**	**2^1**	**2^0**
MB0	**0x40**							
MB1	T	Return Code						
MB2	**0xF8**							
MB3	Reserved							
MB4	Reserved							
MB5	Reserved							

4.3.4.4.9 GOTO_LABEL_IF (0xF9)

If a bit has been set, table processing is continued at the addressed entry; otherwise the next table entry is used.

Label number range: 1 … 65536

Request								
Byte	**2^7**	**2^6**	**2^5**	**2^4**	**2^3**	**2^2**	**2^1**	**2^0**
MB0	**0x40**							
MB1	T	-						
MB2	**0xF9**							
MB3	Label number (LSB)							
MB4	Label number (MSB)							
MB5	Number of bit to be checked							

Response								
Byte	**2^7**	**2^6**	**2^5**	**2^4**	**2^3**	**2^2**	**2^1**	**2^0**
MB0	**0x40**							
MB1	T	Return Code						
MB2	**0xF9**							
MB3	Reserved							
MB4	Reserved							
MB5	Reserved							

4.3.4.4.10 GOTO_LABEL_IF_NOT (0xFA)

If a bit has not been set, table processing is continued at the addressed entry; otherwise the next table entry is used.

Label number range: 1 … 65536

Request								
Byte	**2^7**	**2^6**	**2^5**	**2^4**	**2^3**	**2^2**	**2^1**	**2^0**
MB0	**0x40**							
MB1	T	-						
MB2	**0xFA**							
MB3	Label number (LSB)							
MB4	Label number (MSB)							
MB5	Number of bit to be checked							

Response								
Byte	**2^7**	**2^6**	**2^5**	**2^4**	**2^3**	**2^2**	**2^1**	**2^0**
MB0	**0x40**							
MB1	T	Return Code						
MB2	**0xFA**							
MB3	Reserved							
MB4	Reserved							
MB5	Reserved							

4.3.4.4.11 LABEL (0xFB)

Defines a label as a step target for a GOTO command; no further function. If more than one identical label numbers are defined, the one at the lowest address in the table shall be valid.

Label number range: 1 … 65536

Request								
Byte	**2^7**	**2^6**	**2^5**	**2^4**	**2^3**	**2^2**	**2^1**	**2^0**
MB0	**0x40**							
MB1	T	-						
MB2	**0xFB**							
MB3	Label number (LSB)							
MB4	Label number (MSB)							
MB5	Reserved							

Response								
Byte	**2^7**	**2^6**	**2^5**	**2^4**	**2^3**	**2^2**	**2^1**	**2^0**
MB0	**0x40**							
MB1	T	Return Code						
MB2	**0xFB**							
MB3	Reserved							
MB4	Reserved							
MB5	Reserved							

4.4 Error Blink Codes

The errors and warnings of all subsystems are analyzed at a central location. The occurrence of errors and warnings is recorded and leads the device into an error condition.

In the case of error messages, the output stage is terminated immediately. The motor has no torque in this state and is at risk of uncontrolled movements, which can be prevented with an external brake.

The error or warning condition continues until the cause is remedied.

Attention
The bit error (S2.7) follows statically the bit Error_Quit (C2.7). If the error cause is not eliminated, the bit Error_Quit is set again!

If no more errors or warnings are reported by the hardware, the error or warning condition can be acknowledged with the positive edge by the control bit Error_Quit or with a mailbox command DIAG_QUIT_ERROR.

The error message consists of a 4-digit number.

4.4.1 Error Codes 1111 through 1999

Error codes 1111 through 1999 are errors which can be confirmed by a simple acknowledgement.

4.4.2 Error Codes 2111 through 2999

Error codes 2111 through 2999 are errors which require a warm start of the module. This causes unsaved settings to be lost, requiring them to be reconfigured by the controller. This is indicated by the Reset status bit.

4.4.3 Error Codes 3111 through 3999

Error codes 3111 through 3999 are serious system errors which, for safety reasons, do not allow further operation of the module.
When such an error is reported, all further program processing is blocked and is branched to a routine, in which the error code is reported exclusively with LED A.
These errors require a reset of the module by switching it off and then switching it back on.
An error is indicated by the Error status bit.

4.4.4 Warning Codes 5111 through 5999

Warning codes 5111 through 5999 do not lead to an error condition--they are merely signaled.
A warning is indicated by the Warning status bit.

4.4.5 Signaling

The error display starts with the first blinking sequence (approx. 10 Hz).
Shortly afterward, the second blinking sequence starts (approx. 2 Hz). This sequence represents the highest digit for the 4-place error code.
After a 1 second pause each, all other numbers appear down to the number with the lowest value.
Then, the flash sequence repeats.

4.4.6 Internal Data Bus Parameterization: Accumulative Diagnostics

Errors can also be reported in status byte S0, bit 6 (ERR) and thus initiate an acyclic diagnostics message.
This bit is addressed continuously as long as the error is present and Error_Quit is not set permanently.
Enabling for the indication is set using the configuration parameter ErrorNotificationMode.SystemFlagEnable (parameter 113, bit 0).
This bit has the following meaning:

0: Errors are not reported via status byte S0, bit 6 (ERR).

1: Errors are reported via status byte S0, bit 6 (ERR).

Warning displays are enabled using the configuration parameter WarningNotificationMode.SystemFlagEnable (parameter 116, bit 0).
This bit has the following meaning:

0: Warnings are not reported via status byte S0, bit 6 (ERR).
1: Warnings are reported via status byte S0, bit 6 (ERR).

An explanation of the individual error numbers is given in the following table.

4.4.7 Overview of Error Blink Codes

Error No.:	Designation	Description	Possible cause/Solution
1111	CI_UNKNOWN COMMAND	internal	Internal
1112	CI_UNKNOWN COMMAND2	Unknown command	A drive command was received with an unknown command. Check command (and possibly the run program).
1113	CI_ERR1	Unknown command, like Error 1112	Same as for Error 1112.
1114	CI_NOT IMPLEMENTED	Command not implemented	Command has been prepared but has not (yet) been implemented, otherwise same as Error 1112.
1115	CI_COMMAND_DE NIED	Command could not be executed	It was not possible to execute a command, as the marginal conditions do not permit it. (incorrect mode, incorrect parameter, ... for this command)
1116	CI_SPEED1	Configuration: maximum frequency invalid	Unacceptable value specified for maximum frequency.
1117	CI_SPEED2	Frequency setting invalid	Unacceptable value specified for set frequency.
1118	CI_SPEED3	internal	Internal
1121	CI_ACC1	Acceleration value invalid	Check the specified acceleration value (configuration, Move program, process data ….).
1122	CI_ACC2	Acceleration value invalid	Same as for Error 1121.
1123	CI_ ROTARY_AXIS_ PERIOD_SETPOINT	Operation with round axis: absolute position set point value exceeds the round axis range	During operation with the rotary shaft, the absolute position must lie within the range of 0 … RotaryShaftRange. Check the position data and parameters for the rotary shaft.
1124	CI_POS_TABLE	Invalid table selected for positioning on positioning table	Check Parameter 2 for a command "MOVE_L".

Error No.:	Designation	Description	Possible cause/Solution
1125	CI_SET_POS	Actual value can not be set (e.g. while positioning task in progress)	The drive must be motionless for one SET_POS command.
1141	CM_UNKNOWN_CMDSOURCE	internal	internal
1142	CM_AUTOSTART_NOT_POSSIBLE	Autostart of a Move program not possible (not available)	Load the Move program to the module, or de-activate the autostart.
1161	CONFIG_WRITE_SIZE	Illegal variable when writing the configuration	An attempt was made to enter an element into the configuration with a length less than 1 byte or greater than 4 bytes.
1162	CONFIG_WRONGPASSWORD	Password not accepted	Specify a correct password
1211	CTRLOUT_FREF	Configuration: maximum frequency invalid	The maximum frequency must be between 1…..25000.
1212	CTRLOUT_F_FREF	Maximum frequency reached	An attempt was made to specify a frequency that is greater than the maximum frequency. Check the specified target frequency and the maximum frequency values.
1213	CTRLOUT_CURRENT_SCALE	Configuration: nominal current invalid	Incorrect rated motor current specified. Check the configuration.
1214	CTRLOUT_OPERATION_MODE	Configuration: The selected application is not present in this module	Check the configuration value for Mode 1.
1215	CTRLOUTP_CURR_PARAM	Configuration: current controller password does not agree with current controller parameters	Configuration: Check the parameters for current regulation with the password for current regulation.
1216	CTRLOUTP_F_DIV	Configuration: frequency prescaler faulty	Configuration: Check parameters for frequency prescaler.
1217	CTRLOUT_CURRENT_FACTOR	Current setting incorrect (greater than 150%)	Invalid parameterization for current setting, check configuration, check Move program of Mailbox commands where applicable

Error No.:	Designation	Description	Possible cause/Solution
1218	CTRLOUTP_CURR_ PARAM2	Configuration: current controller password does not agree with current controller parameters (750-673).	Configuration: Check the parameters for current regulation (750-673) with Password2 for current regulation.
1241	IO_BITINDEX TOLARGE	Access to non-existing bit (internal)	Configuration, check Move program and Mailbox commands for incorrect bit addresses.
1242	IO_ILLEGAL_USER BITNR	Bit cannot be modified by user	An attempt was made to change a bit not linked to MZERO or MONE using the Move program of Mailbox command
1243	IO_USERBIT_READ ONLY	Bit cannot be written	Bit not authorized for external write access.
1245	IO_RECURSIVE_LIN K	Nesting of linked bits exceeds parameters	A linkable bit was linked to a linkable bit that was linked to a linkable bit, etc.; A linkable bit may be linked to itself.
1246	IO_TIMER_1	Configuration: filter function not defined	Check configuration for filter functions.
1247	IO_UNKNOWN_TIM ER	A filter with this number does not exist	Check configuration for filter functions.
1248	IO_TIMERMODE	Filter function does not permit write access	The current filter function configuration does not permit writing of the filter.
1249	IO_BITNOT IMPLEMENTED	Requested bit not implemented (internal)	A bit being used can not be queried (internal error).
1251	IO_DIVISION_ZERO	Filter function division: division by zero	Analyze run program and ensure that the denominator can never be 0.
1261	MEAS_ SETPOSITION	For operation with the rotary axis, the command SET_POSITION was used with a position outside the rotary axis.	For this parameter, the precondition Drive_Range_Neg ≤ setpoint position ≤ Drive_Range_Pos must be fulfilled.
1311	TBL_PROGRAM_ST OP1	Motion program terminated with error message 1	Check the termination condition of the Move program.
1312	TBL_PROGRAM_ST OP2	Motion program terminated with error message 2	Check the termination condition of the Move program.

Error No.:	Designation	Description	Possible cause/Solution
1313	TBL_PROGRAM_STOP3	Motion program terminated with error message 3	Check the termination condition of the Move program.
1314	TBL_PROGRAM_STOP4	Motion program terminated with error message 4	Check the termination condition of the Move program.
1315	TBL_PROGRAM_STOP5	Motion program terminated with error message 5	Check the termination condition of the Move program.
1316	TBL_PROGRAM_STOP6	Motion program terminated with error message 6	Check the termination condition of the Move program.
1317	TBL_PROGRAM_STOP7	Motion program terminated with error message 7	Check the termination condition of the Move program.
1319	TBL_PROGRAM_END	Move program not ended properly	Move program ended without the regular Stop command.
1321	TBL_UNKNOWN_CMD	Unknown command for table processing (internal)	(internal)
1322	TBL_LABELNOTFOUND	Label as jump target not available in motion program	Check the definition of the label in the Move program.
1323	TBL_ENDOFTABLE	Jump target exceeds motion program	Check the step targets in the Move program.
1331	TBL_CAM9PARMS	Configuration: faulty parameter for camshaft channel 9	Check configuration for camshaft channel 9, the cycle may not be 0.
1332	TBL_INDEX_OUT_OF_RANGE	Table access out of table range	Check the tables and table access.
1333	TBL_INVALID	Access to invalid table	Check access to tables.
1334	TBL_COPY_FAILED	Version can not be written to EEPROM.	Component error
1351	OPC_START	START command not accepted	Start may only be set when a mode is active.
1352	OPC_TBL_START	Motion program cannot be started (not available).	Motion program cannot be started (not available).
1353	INV_CONTR_IN_PULSE_MODE	Mode not available in selected application	Check the activation of operating modes and configuration of mode 1.
1354	OPC_MULTIMODE_1	Multiple modes selected	Selection of mode is ambiguous.

Error No.:	Designation	Description	Possible cause/Solution
1355	OPC_MULTIMODE_2	Multiple modes selected	Selection of mode is ambiguous.
1356	OPC_WHOOPS1	Unknown mode selected (internal).	Unknown mode selected (internal).
1358	OPC_EXT_ERROR	User bit Error_Set is set	Reset user bit Error_Set.
1359	OPC_MODE	The selected operating mode is not available in the current application	Ensure that a valid operating mode is selected.
1361	OPC_QUIT_START	Start not accepted as long as Quit is active	Set the control bit Error_Quit to 0.
1362	OPC_TMS_RDY	Start/selection of an operating mode is not accepted when TMS is not ready	Monitor supply voltages.
1363	OPC_TMS_SPI	Start/selection of an operating mode is not accepted when SPI contact is not present	24 V control voltage dropped
1411	PARTMODL_CURRENT	internal	internal
1412	PARTMODL_CURRENT_SET	Current setting could not be executed.	Check parameter for Move command SET_CURRENT
1413	PARTMODL_CURRENT_TIME	Time limit for overcurrent reached	The drive has been operated too long at a current >150%. Check move profile and current setting.
1414	PARTMODL_FIFONOTREADY	internal	internal
1415	PARTMODL_POSITION_RANGE	The movement calculator has determined partial movement that exceeds the internal 32-bit position range.	Check movement parameters. This error occurs on unrealistic settings for velocity, acceleration or positions. With extreme parameters, braking from a high speed, for example, at the lowest deceleration yields a brake path that far exceeds the internal value range.
1416	PARTMODL_SPEED_RANGE	The movement calculator has determined partial movement that exceeds the permissible velocity range.	Check the specified velocities

Error No.:	Designation	Description	Possible cause/Solution
1417	PARTMODL_INTERN1	Unknown status of internal FIFO: internal	internal
1418	PARTMODL_ ALIGNTIMOUT	Timeout in the motor alignment (750-673)	Check drive parameters.
1431	PROT_REF_DIR	Reference motion without direction setting	The reference run via the process image must be informed of the starting direction through Direction_Neg or Direction_Pos.
1432	PULSE_TRAIN_NO_ CAM_ACTIVE	Pulse train generator cannot be started when camshaft is not defined.	Download the definition for the pulse chain from the camshaft table.
1433	PROT_UNKNOW_MODE	No application selected	Check configuration of mode 1.
1434	PROT_TEST_ MODE	Special function Integration Test active	The module is switched to the test mode via register 32.
1435	PROT_CURR	Current setpoint outside acceptable range	Run commands TORQ or TORQ_IMM were called up with unacceptable current setpoint.
1451	REF_SWITCH_NOT_ FOUND	Reference switch not found	Check reference switch.
1452	REF_LIM_SWITCH	Reference switch not unique at start of reference motion.	Both limit switches active simultaneously during search for reference switch.
1453	REF_SPEED	Speed setting missing for reference run	Reference speed of 0 is not accepted.
1454	REF_START_DIR_LIMIT	Reference motion at Limit switch. Limit switch already actuated.	Reference run at Limit switch: limit switch already actuated.
1455	ERROR_REF_LIM_ SWITCH_NOT_ EXPECTED	Reference motion: limit switch unexpected	Check limit switch wiring; was the reference run started beyond the limit switch?
1456	REF_ALIGNMENT_ NEEDS_ENCODER	Command for motor alignment was started, but incremental encoder is not active.	Motor alignment requires an incremental encoder to be activated using parameter 413, bit 1.
1511	UNITS_POS_INT_RESULT	Conversion of position in user-specific unit to internal unit: range exceeded	Configuration: Check units conversion..

Error No.:	Designation	Description	Possible cause/Solution
1512	UNITS_POS_USER_RESULT	Conversion of position in internal unit to user-specific unit: range exceeded	Configuration: Check units conversion.
1513	UNITS_SPEED_INT_RESULT	Conversion of speed in user-specific unit to internal unit: range exceeded	Configuration: Check units conversion.
1514	UNITS_SPEED_USER_RESULT	Conversion of speed in internal unit to user-specific unit: range exceeded	Configuration: Check units conversion.
1515	UNITS_ACC_INT_RESULT	Conversion of acceleration in user-specific unit to internal unit: range exceeded	Configuration: Check units conversion.
1516	UNITS_ACC_USER_RESULT	Conversion of acceleration in internal unit to user-specific unit: range exceeded	Configuration: Check units conversion.
1517	UNITS_PARAM_ZERO	Parameter for conversion is zero	Configuration: Units conversion: Divisor is zero
1521	SYS_MODE	Configuration: application cannot be executed on this terminal. Acceptable value range: 1, 5	Check value ranges 1, 5
1551	MCALC_SPEED1	internal	internal
1552	MCALC_SPEED2	internal	internal
1553	MCALC_SPEED3	internal	internal
1554	MCALC_ACC1	During the ramp run at a defined ramp time the movement calculator has determined partial movement that exceeds the internal value range for acceleration	Adapt ramp time setting.
1555	MCALC_ACC2	Same as MCALC_ACC1	Same as MCALC_ACC1
1556	MCALC_ACC3	During the ramp run at a defined ramp path the movement calculator has determined partial movement that exceeds the internal value range for acceleration	Adapt ramp path setting.

Error No.:	Designation	Description	Possible cause/Solution
1557	MCALC_PARA	The movement calculator has received invalid parameters	Check all parameters: acceleration, moving velocity, ramp time, ramp path, ramp type, starting position, target position
1561	MCALC_TIME1	Internal motion time range exceeded	Check parameterization. *
1562	MCALC_TIME2	The movement calculator has determined partial movement that exceeds the internal time range (>500 h)	Check parameterization. *
1563	MCALC_TIME3	Internal motion time range exceeded	Check parameterization. *
1564	MCALC_TIME4	The movement calculator has determined a movement sequence that exceeds the internal time range (>500 h).	Check parameterization. *
1565	MCALC_DIST1	Internal motion travel range exceeded	Check parameterization. *
1566	MCALC_DIST2	Internal motion travel range exceeded	Check parameterization. *
1567	MCALC_DIST3	Same as PARTMODL_ POSITION_RANGE	Check parameterization. *
1568	MCALC_DIST4	Internal motion travel range exceeded	Check parameterization. *
1569	MCALC_DIST5	Same as PARTMODL_ POSITION_RANGE	Check parameterization. *
1571	MCALC_MOVE1	Internal: Movement calculator can find no solution	Check parameterization. *
1572	MCALC_MOVE2	Internal: Movement calculator can find no solution	Check parameterization. *
1573	MCALC_MOVE3	Internal: Movement calculator can find no solution	Check parameterization. *
1574	MCALC_CURR	Current setpoint outside acceptable range	Reduce default value
1611	ERR_ILLEGAL_ERR ORCODE	Illegal error code to be signaled	internal
1715	SPI_TELEGRAM3_ FAIL	Timeout for data transfer (750-673)	internal

Error No.:	Designation	Description	Possible cause/Solution
1716	SPI_VAR_READ_ FAIL	Internal buffer of SPI data transfer is too small	internal
1717	SPI_VAR_READ_ FAIL2	Internal error of SPI data transfer	internal
1718	SPI_VAR_LAST_ READ	internal error	internal
1811	TMS	internal TMS error	internal
1812	TMS_ OVERCURRENT_ HW	Overcurrent output stage (750-673)	Overcurrent output stage (750-673): Short circuit or ground leak on the motor circuit
1813	TMS_ OVERVOLTAGE_ 24V	Overvoltage in the 24 V control voltage	24 V control voltage shall not exceed 36 V.
1814	TMS_ UNDERVOLTAGE_ 24V	Undervoltage in the 24 V control voltage	24 V control voltage shall not exceed 16 V.
1815	TMS_ OVERVOLTAGE_V_ DC	Overvoltage in the output voltage supply	Output voltage supply shall not be larger than that configured in parameter 410.
1816	TMS_ UNDERVOLTAGE_ V_DC	Undervoltage in the output voltage supply	Output voltage supply shall not be smaller than that configured in parameter 408.
1817	TMS_V_DC	Error in parameterization of output voltage supply monitoring	Parameter 408 or 410 is configured incorrectly.
1818	TMS_ENC_ RESOLUTION	Unacceptable resolution of incremental encoder configured	Resolution of 0 is not allowed.
1819	TMS_MOTOR_ FULL_STEPS	Unacceptable number of motor full steps configured	Number of motor full steps shall not be 0 and must be divisible by four.
1823	TMS_ SPEEDMEASURETI MEA	Unacceptable gate time for absolute rotational speed measurement	Parameter 428 is configured incorrectly.
1824	TMS_ SPEEDMEASURETI MEOUTA	Unacceptable timeout period for absolute rotational speed measurement	Parameter 432 is configured incorrectly.
1825	TMS_ SPEEDMEASURETI MEV	Unacceptable gate time for internal rotational speed measurement	Parameter 436 is configured incorrectly.

Error No.:	Designation	Description	Possible cause/Solution
1826	TMS_ SPEEDMEASURETIMEOUTV	Unacceptable timeout period for internal rotational speed measurement	Parameter 440 is configured incorrectly.
1827	TMS_SENSOR_ SUPPLY_SHORT	Short circuit/overload of incremental encoder supply	Check incremental encoder connection.
1831	TMS_ SPEEDMEASUREDIVZERO_1	internal	internal
1832	TMS_ SPEEDMEASUREDIVZERO_2	internal	internal
1833	TMS_ENCODER_ RESOLUTION	Unacceptable incremental encoder resolution	Incremental encoder resolution shall not be 0.
1854	TMS_MODE_ERROR	Unacceptable TMS mode	internal
1859	TMS_CURRENT_ CONTR_TYPE	Unacceptable current controller type	Correct parameter 396.
1881	TMS_PARAM_ CHANGED	Parameter shall not be changed while motor is running	Change parameter only when motor is stopped.
1882	TMS_PARAM_NOT_ OK	internal	internal
1884	TMS_NOT_ ALIGNED	Operation with field orientation without previous motor alignment	Perform motor alignment.
1886	TMS_SWITCHING_ FREQUENCY	Maximum acceptable switching frequency of output stage exceeded	Configure current controller.
1887	TMS_BRIDGE_A_ LOW	No load detected on motor circuit A	Check motor connection.
1889	TMS_BRIDGE_B_ LOW	No load detected on motor circuit B	Check motor connection.
1891	TMS_TRACKING_V	Speed contouring error	Check speed contouring error.
1892	TMS_TRACKING_S	Position contouring error	Check position contouring error.
1893	TMS_TEMP_HIGH	Output stage temperature higher than max. acceptable temp.	Decrease load cycle.

Error No.:	Designation	Description	Possible cause/Solution
1894	TMS_TEMP_HIGH_ WARN	Output stage temperature higher than parameterized warning threshold	Check parameters, decrease load cycle.
1911	COMMAND_IS_ RUNNING	Command can not be executed, as another command is currently being processed	internal
1912	HIGH_PRIO_ COMMAND_IS_ RUNNING	Command can not be executed, as another command of higher priority is currently being processed	internal
1931	PARTMODL_ LIMITSWITCH	Command can not be executed, as a limit switch is active	Move drive away from limit switch.
2811	KBUS	internal	Internal
2821	CFG_FACTORY_ LOAD	A Factory_Default data set has been copied for upload to RAM	A CONFIG_RESTORE command has been executed without a warm start. The module is not operational is this state. → Conduct a warm start or Power-on reset.
2831	MEASURE_ERR1	internal: hardware unknown	internal
2832	MEASURE_ERR2	same as MEASURE_ERR1	internal
2833	MCALC_INTERN1	Internal: error in travel calculation	internal
2834	MCALC_INTERN2	Internal: error in travel calculation	internal
2835	MCALC_INTERN3	Internal: unknown acceleration profile	internal
2836	MCALC_INTERN4	MCALC_INTERN3	internal
2837	MCALC_INTERN5	MCALC_INTERN3	internal
2838	MCALC_INTERN6	MCALC_INTERN3	internal
2839	MCALC_INTERN7	MCALC_INTERN3	internal
2841	MCALC_INTERN8	MCALC_INTERN3	internal
2842	MCALC_INTERN9	MCALC_INTERN3	internal
2843	MCALC_INTERN10	MCALC_INTERN3	internal

Error No.:	Designation	Description	Possible cause/Solution
2844	MCALC_INTERN11	MCALC_INTERN3	internal
2845	MCALC_BUFFER_FULL	internal: buffer overflow	internal
2846	MOVECALC_ACC2	internal	internal
2863	TEST_EERPOM_FAILURE	Auto test: EEPROM failure	Hardware defective
2864	TEST_CPLD_FAILURE	Auto test: CPLD failure	Hardware defective
2865	TEST_INVALID_MODULE	Auto test: unknown hardware	Hardware defective
2866	GENERIC_TEST	Auto test: invalid hardware	Hardware defective
2871	RS232_TX_TIMEOUT	internal:time limit exceeded in debug interface	internal
2881	SYS_IDLE_RECURSIVE	internal	internal
2882	SYS_SPI_TIMEOUT	internal	internal
2883	SPI_CONTACTLOST	No more contact to TMS	24 V control voltage possibly dropped
2891	VERSION_UNKNOWN_IDENT	Unknown hardware	Hardware defective
2892	VERSION_NOT_COMPATIBLE_HW	Hardware not compatible with software	Hardware defective
2893	VERSION_WRONG_TMS_VERSION	Wrong version of TMS firmware	Update TMS to correct version.
2911	TMS_ NOTSPECIFIED	unspecified TMS error	internal
2912	TMS_RESET	TMS restart detected	Check 24 V control voltage.
2913	SPI_TMS_ PARAMINDEX	TMS: wrong parameter index in SPI transfer	internal
2914	SPI_TMS_ TOGGLETIMEOUT	TMS: SPI transfer timeout	internal
2915	SPI_TMS_ SENSORDIRECTION	TMS: wrong rotational direction of motor or encoder	Reconnect motor circuits or incremental encoder connector.

Error No.:	Designation	Description	Possible cause/Solution
2916	SPI_TMS_NO_ SENSOR	TMS: no incremental encoder during alignment	Check incremental encoder connector
2917	SPI_TMS_Z_JITTER	TMS: Z pulse outside the acceptable jitter range	The Z pulse was recorded at various positions. -> Check incremental encoder connector -> Check parameter 426
2918	ALIGN_WITHOUT_ Z	TMS: Motor alignment with stored measurement values requires Z pulse configuration	Configure Z pulse -> Check parameter 413
3111	INT_KBUS	internal	internal
3112	INT_WATCHDOG	internal: watchdog	internal
3113	INT_SPURIOUS	internal	internal
3114	INT_UNUSED	internal	internal
3115	INT_FIQ	internal	internal
3116	INT_SWI	internal	internal
3117	INT_UNDEF_INST	internal	internal
3118	INT_FETCH	internal	internal
3119	INT_DATA_ACESS	internal	internal
3121	INT_ROM_ISR	internal	internal
3122	INT_STACK_ OVERFLOW	internal	internal
3142	SYS_PLL_NOT_LOCKED	internal	internal
3143	SYS_ADC_TIMEOUT	internal	internal
3144	SYSTEMEXIT	internal	internal
3155	ERR_ILLEGAL_ERRORCODE	Illegal error code to be signaled	internal
3166	OPC_MULTIMODE_ 1	internal	internal
3167	OPC_MULTIMODE_ 2	internal	internal
3168	OPC_WHOOPS1	internal	internal
3179	TBL_COPY_FAILED	Error when writing to EEPROM	internal

Error No.:	Designation	Description	Possible cause/Solution
3211	PARTMODL_FIFONOTREADY	internal	internal
3212	PARTMODL_POSITION_RANGE	internal	internal
3213	PARTMODL_SPEED_RANGE	internal	internal
3214	PARTMODL_INTERN1	internal	internal
3215	PARTMODL_ERR4	internal	internal
3216	PARTMODL_ERR5	internal	internal
3231	TBL_INVALID	internal	internal
3232	TBL_CP2EEPROM_FAIL	internal	internal
3233	TBL_COPY_INVALID	internal	internal
3234	TBL_UNKNOWN_TTYPE	internal	internal
3271	TEST_FLASH	Error in checksum of program memory	internal
3272	TEST_FLASH_CRCGEN	Checksum for program memory not available	internal
3273	TEST_EERPOM_FAILURE	Auto test: EEPROM failure	internal
3274	TEST_CPLD_FAILURE	Auto test: CPLD failure	internal
3275	TEST_INVALID_MODULE	Auto test: unknown hardware	internal
3276	GENERIC_TEST	Auto test: invalid hardware	internal
3277	TEST_ENDOF_ FUNCTION_TEST	Self test completed	internal A reset is required after a self test.
3811	TMS_SWITCH_ ALIGN	internal	internal
5887	ERROR_WARN_ BRIDGE_A_LOW	No motor detected at output stage A	Check motor connection.
5889	ERROR_WARN_ BRIDGE_B_LOW	No motor detected at output stage B	Check motor connection

Error No.:	Designation	Description	Possible cause/Solution
5891	ERROR_WARN_ TMS_TRACKING_V	Contouring error speed	Increase parameter 482 (TrackingError_Range_Speed).
5892	ERROR_WARN_ TMS_TRACKING_S	Position contouring error	Increase parameter 484 (TrackingError_Range_Position).
5893	ERROR_WARN_ TMS_TEMP_HIGH_ WARN	Output stage temperature higher than parameterized warning threshold	Either decrease current or increase parameter 406 (Warn_Threshold_Temperature)
5917	ERROR_WARN_SPI _TMS_Z_JITTER	TMS: Z pulse outside the acceptable jitter range	Increase parameter 426 (Encoder_IndexMeasureJitterRange).
* Check parameterization of movement. This error occurs if the specifications for speed, acceleration or position are unrealistic. No numerical solution exists for the desired move profile. As long as the parameter values are iterative, they will decrease until no error message is issued.			

4.5 Bit Field for I/O Driver

The bit functions described in this table relate to the standard application stepper positioning control

In the description of respective application it is shown when bits have significance in other applications.

The following conventions apply:

- Source bits are assigned numbers 0 to 127 and may not be used as target bits. A source bit may reference several target bits.
- Target bits are assigned numbers 128 to 255 and may also be used as source bits. Target bits have exactly one source.
- References are stored in the configuration table. The names of the table entries correspond to those in the bit table. The prefix Ptr is placed in front of the identifier.
- The standard link between the source and target is entered in the column "Target/Source". This corresponds to the WAGO default settings (FACTORY_DEFAULT_1).

Designation	Bit number		Type	Default		Description
	Dec.	Hex.		Target/Source	Bit no.	
ZERO	0	0x00	SRC	0	-	Bit is always cleared
ONE	1	0x01	SRC	1	-	Bit is always set
MZERO	2	0x02	SRC	0	-	A bit linked to MZERO is originally cleared after reset, but may be manipulated by mailbox commands or motion programs.
MONE	3	0x03	SRC	1	-	A bit linked to MZERO is originally set after reset, but may be manipulated by mailbox commands or motion programs.
Reset	4	0x04	SRC	KBUS_ST3_7	0x97	With this bit, the control can detect the reset of the terminal. The bit is true after a reset and is confirmed and false by Reset_Quit.
						0: No reset since last confirmation.
						1: A reset has been carried out but not yet confirmed with Reset_Quit. Parameters, data or tables not stored in the EEPROM are no longer valid.

Designation	Bit number		Type	Default		Description	
	Dec.	Hex.		Target/Source	Bit no.		
Warning	5	0x05	SRC	KBUS_ST3_6	0x96	Warning status of driver A warning can be acknowledged using Error_Quit.	
						0	The drive has no warning.
						1	The drive has a warning.
KBUS_ Active	6	0x06	SRC	-	-	Internal data bus communication active	
						0:	Internal data bus communication for more than 100 ms
						1:	Internal data bus communication present
	7	0x07					
On_Target	8	0x08	SRC	KBUS_ST2_0 OUT1	0x88 0xA0	Target reached The significance of this bit depends on the selected operating mode.	
						Step positioning:	
						0:	The specified target position has not been reached.
						1:	The specified setpoint within the target window.
						Run program:	
						0:	TargetWindowPosition has not been reached.
						1:	TargetWindowPosition has been reached.
						Referencing	
						0:	The reference point has not been approached and set.
						1:	The reference point has been moved to and set successfully.
						Jog Mode:	
						0:	The bit is not used in this mode and remains at 0.
						1:	The bit is not used in this mode and remains at 0.
						Mailbox mode:	
						0:	Function of mailbox command.
						1:	Function of mailbox command.

Designation	Bit number		Type	Default		Description	
	Dec.	Hex.		Target/Source	Bit no.		
Busy	9	0x09	SRC	KBUS_ST2_1 LED E	0x89 0xD0	Busy: the selected operating mode is active and not yet finished. It is possible that the operating mode was cancelled.	
						Step positioning:	
						0:	Step positioning not running.
						1:	Step positioning running.
						Run program:	
						0:	The Run program is not running.
						1:	The Run program is running.
						Reference run:	
						0:	Reference run not in operation.
						1:	Reference run in operation.
						Jog Mode:	
						0:	Motor at standstill.
						1:	The Jog mode is running, i.e. the motor has been started using Jog_Pos or Jog_Neg.
						Mailbox mode:	
						0:	No command is active.
						1:	A command is active.
StandStill	10	0x0A	SRC	KBUS_ST2_2	0x8A	Drive standstill, frequency output at 0.	
						0:	Motor is turning.
						1:	Motor at standstill.
On_Speed	11	0x0B	SRC	KBUS_ST2_3	0x8B	Drive speed reached	
						0:	The drive has not reached its setpoint speed.
						1:	The drive has reached its setpoint speed.
Direction	12	0x0C	SRC	KBUS_ST2_4	0x8C	Direction of rotation is valid only when StandStill is not set to 1.	
						0:	Drive moving in the negative direction.
						1:	Drive moving in the positive direction.

Designation	Bit number		Type	Default		Description	
	Dec.	Hex.		Target/Source	Bit no.		
Reference_OK	13	0x0D	SRC	KBUS_ST2_5	0x8D	Set when reference run has been successfully concluded.	
						0:	Upon switching on of the module, the bit is set to 0. It is also set to 0 when the reference motion is initiated.
						1:	The reference point has been successfully located in the reference run mode.
PreCalc_Ack	14	0x0E	SRC	KBUS_ST2_6	0x8E	Setpoints from Mode 2.2 saved. This bit is set when the setpoint save mode has been requested with PreCalc and precalculation of a movement has been successfully completed.	
						0:	Precalculation not yet performed.
						1:	Precalculation performed.
Error	15	0x0F	SRC	KBUS_ST2_7 OUT2	0x8F 0xA1	Drive error status. An error can be acknowledged using Error_Quit.	
						0:	No error present for the drive.
						1:	Error present for the drive.
Ready	16	0x10	SRC	KBUS_ST1_0	0x80	Ready	
						0:	The module is not ready for operation. Either a corresponding request is present via Enable, or an error has resulted in cancellation of Ready. When the bit switches from 1 to 0 the output stage is deactivated, or the output frequency is set to 0.
						1:	Readiness for operation has been requested via Enable and no error is present.
Stop_N_ACK	17	0x11	SRC	KBUS_ST1_1 LED G	0x81 0xD2	Drive stop inverted	
						0:	The bit Stop1_N or Stop2_N is set to 0. The motor is set to 0 (StandStill set to 1). Start can not be used to start up the unit.
						1:	The bits Stop1_N and Stop2_N are both set to 1, or the drive is braking the unit.

Designation	Bit number		Type	Default		Description	
	Dec.	Hex.		Target/Source	Bit no.		
Start_ACK	18	0x12	SRC	KBUS_ST1_2	0x82	Start process in operating mode	
						0:	This bit is also set to 0 when the Start request is canceled.
						1:	The rising edge function is a function of the selected operating mode.
						Single positioning Speed control The specified setpoint have been accepted from the process image.	
						Run program The run program has been started.	
						Referencing The reference run has been started.	
						JogMode No effect. Handshake not performed.	
						Mailbox mode Handshake not performed. As soon as the mailbox mode has been enabled, the corresponding commands can be transmitted via the mailbox.	
Command_ACK[1]	19	0x13	SRC	KBUS_ST1_3	0x83	Selecting the operating mode 0: Idle mode 1: Single positioning For this operating mode, the mailbox must be disabled. 2: Run program 3: Speed Control 4: Reference motion 8: JogMode The drive can be operated manually at the setup speed. Control is performed via Input3 (Jog_Pos) and Input4 (Jog_Neg) 16: Run commands via mailbox In this mode, all movement commands are issued directly via mailbox.	
Command_ACK[2]	20	0x14	SRC	KBUS_ST1_4	0x84		
Command_ACK[3]	21	0x15	SRC	KBUS_ST1_5	0x85		
Command_ACK[4]	22	0x16	SRC	KBUS_ST1_6	0x86		
Command_ACK[5]	23	0x17	SRC	KBUS_ST1_7	0x87		
Break	24	0x18	SRC	-	-	Brake	
						0:	The drive is being braked.
						1:	The brake is vented.
ERR_Code	25	0x19	SRC	LED H	0xD3	This bit is usually connected to a LED. IF an error occurs, the errors is output as flash code.	

Designation	Bit number		Type	Default		Description
	Dec.	Hex.		Target/Source	Bit no.	
SetupSpeed_Active_ACK	26	0x1A	SRC	-	-	Setting-up operation is active. When this bit is set, the drive speed is limited to the defined setup speed. Acceleration is not limited. The currently valid acceleration value is applied. 0: Setup mode is not active. 1: Setting-up operation is active.
Program Running	27	0x1B	SRC	LED C	0xD1	A motion program is in process.
Ramp_Up	28	0x1C	SRC	-	-	Set during acceleration phase
Ramp_Down	29	0x1D	SRC	-	-	Set during delay phase
	30	0x1E				
	31	0x1F				
Trace_Stored	32	0x20	SRC	-	-	This bit is set when all data sets TRACE_VAR1/2 have been saved in the Trace table. Trace_Stored is cleared each time trace recording is started (0→1 to Trace_Trigger when Trace_Armed is also set). A trace can be read out via the table commands DLD_START, DLD_CONT and DLD_END.
	33	0x21				
	34	0x22				
	35	0x23				
	36	0x24				
	37	0x25				
Encoder_A	38	0x26		LED_E	0xD4	Internal signal: status of incremental encoder, track A.
Encoder_B	39	0x27		LED_F	0xD5	Internal signal: status of incremental encoder, track B.
Err_Range_Neg	40	0x28	SRC	-	-	Moving range exceeded when moving in negative direction. 0: The bottom limit for the movement range has not been violated. 1: The bottom limit for the movement range has been violated.

Designation	Bit number		Type	Default		Description
	Dec.	Hex.		Target/Source	Bit no.	
Err_Range_Pos	41	0x29	SRC	-	-	Moving range exceeded when moving in positive direction. 0: The top limit for the movement range has not been violated. 1: The top limit for the movement range has been violated.
Err_Range	42	0x2A	SRC	-	-	This parameter is set when it has been detected by Err_Range_Neg, Err_Range_Pos, LimitSwitch_Pos or LimitSwitch_Neg that the permissible movement range has been violated.
	43	0x2B				
	44	0x2C				
	45	0x2D				
Encoder_Z_Toggle	46	0x2E	SRC	-	-	This bit toggles the Z signal from the incremental encoder at the rising edge.
CAM9	47	0x2F	SRC	-	-	Camshaft 9
Input1	48	0x30	SRC	KBUS_ST3_0 Stop1_N	0x90 0xC2	Input 1
Input2	49	0x31	SRC	KBUS_ST3_1 Set_Reference	0x91 0xBC	Input 2
Input3	50	0x32	SRC	KBUS_ST3_2 Set_Reference	0x92 0xCA	Input 3
Input4	51	0x33	SRC	KBUS_ST3_3 Jog_Neg	0x93 0xCB	Input 4
Input5	52	0x34	SRC	KBUS_ST3_4 LimitSwitch_Pos	0x94 0xC0	Input 5
Input6	53	0x35	SRC	KBUS_ST3_5 LimitSwitch_Neg	0x95 0xC1	Input 6
Input7	54	0x36	SRC	-	-	Input 7
Input8	55	0x37	SRC	-	-	Input 8
CAM1	56	0x38	SRC	-	-	Camshaft 1
CAM2	57	0x39	SRC	-	-	Camshaft 2
CAM3	58	0x3A	SRC	-	-	Camshaft 3
CAM4	59	0x3B	SRC	-	-	Camshaft 4

Designation	Bit number		Type	Default		Description
	Dec.	Hex.		Target/Source	Bit no.	
CAM5	60	0x3C	SRC	-	-	Camshaft 5
CAM6	61	0x3D	SRC	-	-	Camshaft 6
CAM7	62	0x3E	SRC	-	-	Camshaft 7
CAM8	63	0x3F	SRC	-	-	Camshaft 8
KBUS_CTRL1_0	64	0x40	SRC	Enable	0xB0	Internal bus control byte 1 bit 0
KBUS_CTRL1_1	65	0x41	SRC	Stop2_N	0xB1	Internal bus control byte 1 bit 1
KBUS_CTRL1_2	66	0x42	SRC	Start	0xB2	Internal bus control byte 1 bit 2
KBUS_CTRL1_3	67	0x43	SRC	M_Positioning	0xB3	Internal bus control byte 1 bit 3
KBUS_CTRL1_4	68	0x44	SRC	M_Program	0xB4	Internal bus control byte 1 bit 4
KBUS_CTRL1_5	69	0x45	SRC	M_Reference	0xB5	Internal bus control byte 1 bit 5
KBUS_CTRL1_6	70	0x46	SRC	M_Jog	0xB6	Internal bus control byte 1 bit 6
KBUS_CTRL1_7	71	0x47	SRC	M_DriveBxMbx	0xB7	Internal bus control byte 1 bit 7
KBUS_CTRL2_0	72	0x48	SRC	Freq_ Range_Sel_0	0xC4	Internal bus control byte 2 bit 0
KBUS_CTRL2_1	73	0x49	SRC	Freq_ Range_Sel_1	0xC5	Internal bus control byte 2 bit 1
KBUS_CTRL2_2	74	0x4A	SRC	Acc_ Range_Sel_0	0xC6	Internal bus control byte 2 bit 2
KBUS_CTRL2_3	75	0x4B	SRC	Acc_ Range_Sel_1	0xC7	Internal bus control byte 2 bit 3
KBUS_CTRL2_4	76	0x4C	SRC	-	-	Internal bus control byte 2 bit 4
KBUS_CTRL2_5	77	0x4D	SRC	-	-	Internal bus control byte 2 bit 5
KBUS_CTRL2_6	78	0x4E	SRC	PreCalc	0xBD	Internal bus control byte 2 bit 6
KBUS_CTRL2_7	79	0x4F	SRC	Error_Quit	0xBF	Internal bus control byte 2 bit 7
KBUS_CTRL3_0	80	0x50	SRC	Set_Actual_Pos	0xC8	Internal bus control byte 3 bit 0
KBUS_CTRL3_1	81	0x51	SRC	-	-	Internal bus control byte 3 bit 1
KBUS_CTRL3_2	82	0x52	SRC	Direction_Pos	0xBA	Internal bus control byte 3 bit 2

Designation	Bit number		Type	Default		Description
	Dec.	Hex.		Target/Source	Bit no.	
KBUS_CTRL3_3	83	0x53	SRC	Direction_Neg	0xBB	Internal bus control byte 3 bit 3
KBUS_CTRL3_4	84	0x54	SRC	-	-	Internal bus control byte 3 bit 4
KBUS_CTRL3_5	85	0x55	SRC	-	-	Internal bus control byte 3 bit 5
KBUS_CTRL3_6	86	0x56	SRC	-	-	Internal bus control byte 3 bit 6
KBUS_CTRL3_7	87	0x57	SRC	Reset_Quit	0xB9	Internal bus control byte 3 bit 7
	88	0x58				
	89	0x59				
	90	0x5A				
	91	0x5B				
	92	0x5C				
	93	0x5D				
	94	0x5E				
	95	0x5F				
	96	0x60				
	97	0x61				
	98	0x62				
	99	0x63				
	100	0x64				
	101	0x65				
	102	0x66				
	103	0x67				
	104	0x68				
	105	0x69				
	106	0x6A				
	107	0x6B				
	108	0x6C				
	109	0x6D				
	110	0x6E				
	111	0x6F				
	112	0x70				
	113	0x71				
	114	0x72				

Designation	Bit number		Type	Default		Description
	Dec.	Hex.		Target/Source	Bit no.	
	115	0x73				
	116	0x74				
	117	0x75				
	118	0x76				
	119	0x77				
	120	0x78				
	121	0x79				
	122	0x7A				
	123	0x7B				
	124	0x7C				
	125	0x7D				
	126	0x7E				
	127	0x7F				
KBUS_ ST1_0	128	0x80	DST/ SRC	Ready	0x10	Internal data bus status byte 1 bit 0 Ready
KBUS_ ST1_1	129	0x81	DST/ SRC	Stop_N_ACK	0x11	Internal data bus status byte 1 bit 1 Stop_N_ACK
KBUS_ ST1_2	130	0x82	DST/ SRC	Start_ACK	0x12	Internal data bus status byte 1 bit 2 Start_ACK
KBUS_ ST1_3	131	0x83	DST/ SRC	Command_Ack [1]	0x13	Internal data bus status byte 1 bit 3 Command_ACK[1]
KBUS_ ST1_4	132	0x84	DST/ SRC	Command_Ack [2]	0x14	Internal data bus status byte 1 bit 4 Command_ACK[2]
KBUS_ ST1_5	133	0x85	DST/ SRC	Command_Ack [3]	0x15	Internal data bus status byte 1 bit 5 Command_ACK[3]
KBUS_ ST1_6	134	0x86	DST/ SRC	Command_Ack [4]	0x16	Internal data bus status byte 1 bit 6 Command_ACK[4]
KBUS_ ST1_7	135	0x87	DST/ SRC	Command_Ack [5]	0x17	Internal data bus status byte 1 bit 7 Command_ACK[5]
KBUS_ ST2_0	136	0x88	DST/ SRC	On_Target	0x08	Internal data bus status byte 2 bit 0 On_Target
KBUS_ ST2_1	137	0x89	DST/ SRC	Busy	0x09	Internal data bus status byte 2 bit 1 Busy
KBUS_ ST2_2	138	0x8A	DST/ SRC	StandStill	0x0A	Internal data bus status byte 2 bit 2 StandStill
KBUS_ ST2_3	139	0x8B	DST/ SRC	On_Speed	0x0B	Internal data bus status byte 2 bit 3 On_Speed
KBUS_ ST2_4	140	0x8C	DST/ SRC	Direction	0x0C	Internal data bus status byte 2 bit 4 Direction

Designation	Bit number		Type	Default		Description
	Dec.	Hex.		Target/Source	Bit no.	
KBUS_ ST2_5	141	0x8D	DST/ SRC	Reference_OK	0x0D	Internal data bus status byte 2 bit 5 Reference_OK
KBUS_ ST2_6	142	0x8E	DST/ SRC	PreCalc_ACK	0x0E	Internal data bus status byte 2 bit 6 Precalc_ACK
KBUS_ ST2_7	143	0x8F	DST/ SRC	Error	0x0F	Internal data bus status byte 2 bit 7 Error
KBUS_ ST3_0	144	0x90	DST/ SRC	Input1	0x30	Internal data bus status byte 3 bit 0 INP1
KBUS_ ST3_1	145	0x91	DST/ SRC	Input2	0x31	Internal data bus status byte 3 bit 1 INP2
KBUS_ ST3_2	146	0x92	DST/ SRC	Input3	0x32	Internal data bus status byte 3 bit 2 INP3
KBUS_ ST3_3	147	0x93	DST/ SRC	Input4	0x33	Internal data bus status byte 3 bit 3 INP4
KBUS_ ST3_4	148	0x94	DST/ SRC	Input5	0x34	Internal data bus status byte 3 bit 4 INP5
KBUS_ ST3_5	149	0x95	DST/ SRC	Input6	0x35	Internal data bus status byte 3 bit 5 INP6
KBUS_ ST3_6	150	0x96	DST/ SRC	Warning	0x05	Internal bus status byte 3 bit 6
KBUS_ ST3_7	151	0x97	DST/ SRC	Reset	0x04	Internal bus status byte 3 bit 7
UserBit_0	152	0x98	SRC	MZERO	0x02	User bit; e.g., for run program
UserBit_1	153	0x99	SRC	MZERO	0x02	User bit; e.g., for run program
UserBit_2	154	0x9A	SRC	MZERO	0x02	User bit; e.g., for run program
UserBit_3	155	0x9B	SRC	MZERO	0x02	User bit; e.g., for run program
UserBit_4	156	0x9C	SRC	MZERO	0x02	User bit; e.g., for run program
UserBit_5	157	0x9D	SRC	MZERO	0x02	User bit; e.g., for run program
UserBit_6	158	0x9E	SRC	MZERO	0x02	User bit; e.g., for run program
UserBit_7	159	0x9F	SRC	MZERO	0x02	User bit; e.g., for run program
OUT1	160	0xA0	DST/ SRC	On_Target	0x08	Output 1
OUT2	161	0xA1	DST/ SRC	Error	0x0F	Output 2
	162	0xA2				
	163	0xA3				
	164	0xA4				
	165	0xA5				
	166	0xA6				
	167	0xA7				

Designation	Bit number		Type	Default		Description
	Dec.	Hex.		Target/Source	Bit no.	
FILT1	168	0xA8	FILT	ZERO	0x00	Timer / Filter 1
FILT2	169	0xA9	FILT	ZERO	0x00	Timer / Filter 2
FILT3	170	0xAA	FILT	ZERO	0x00	Timer / Filter 3
FILT4	171	0xAB	FILT	ZERO	0x00	Timer / Filter 4
FILT5	172	0xAC	FILT	ZERO	0x00	Timer / Filter 5
FILT6	173	0xAD	FILT	ZERO	0x00	Timer / Filter 6
FILT7	174	0xAE	FILT	ZERO	0x00	Timer / Filter 7
FILT8	175	0xAF	FILT	ZERO	0x00	Timer / Filter 8
Enable	176	0xB0	DST/ SRC	KBUS_CTRL1 _0	0x40	Module enable. Contrary to Enable_Drive, this bit must be set to activate an operating mode.
						0: The module is blocked. When this bit is reset during ongoing operation, the power output stage is switched off. This bit terminates the current operating mode.
						1: The module is enabled and can be started when the corresponding return message is also available in the status.
Stop2_N	177	0xB1	DST/ SRC	KBUS_CTRL1 _1	0x41	Drive Stop 2 inverted. With this bit, the drive can be turned off from the control. This bit must be set to activate an operating mode. The return message is transmitted via the Stop_N_ACK bit. Stop1_N and Stop2_N are always taken into account at that bit.
						0: Current is being supplied to the motor, but it is at standstill. If the motor still rotates, it will be driven to stand still by the STOP acceleration. The motor can not be started up.
						1: The drive may be started.

Designation	Bit number		Type	Default		Description
	Dec.	Hex.		Target/Source	Bit no.	
Start	178	0xB2	DST/ SRC	KBUS_CTRL1 _2	0x42	Start-up of drive The drives, or frequency output, are started in the selected mode on the positive edge. If the edge is not accepted (in the Jog or Mailbox mode), an error message is generated.
						0→1: The drive is started accordingly on the rising edge.
						Single positioning Speed control The specified setpoints are accepted from the process image. Movement is made directly to the new target, even if the drive is already turning. A previously calculated movement sequence is started immediately when the PreCalc_ACK bit is set (instant setpoint switch).
						Run Program The run program is started at the Start_Instruction_Counter address in the process image. A run program that is currently running is interrupted, and the program flow is started at the new address. This allows various program parts to be transferred via the process image.
						Referencing The reference run is initiated. If the reference run is still in operation, the (new) setpoints are again accepted and calculated (same procedure as for positioning). The reference run is newly initiated.
						JogMode No effect. The drive is started in JogMode by default via Input3 (Jog_Pos) or Input4 (Jog_Neg). The inputs and outputs can be freely reconfigured. An error message is generated.
						Mailbox mode No effect in this operating mode. The command is carried out via the mailbox. If the bit is set nonetheless, neither an acknowledgment nor an error message is generated.

Designation	Bit number		Type	Default		Description	
	Dec.	Hex.		Target/Source	Bit no.		
Command[1]	179	0xB3	DST/ SRC	KBUS_CTRL1 _3	0x43	Operating mode Single positioning For this operating mode, the mailbox must be disabled.	
						0:	The Step positioning mode is not active (selected).
						1:	Operating mode Single positioning is selected.
Command[2]	180	0xB4	DST/ SRC	KBUS_CTRL1 _4	0x44	Operating mode Run program	
						0:	The Run program mode is not active (selected).
						1:	The Run program mode has been selected.
Command[3]	181	0xB5	DST/ SRC	KBUS_CTRL1 _5	0x45	Reference run operating mode	
						0:	Reference run operating mode is not selected.
						1:	Reference run operating mode is selected.
Command[4]	182	0xB6	DST/ SRC	KBUS_CTRL1 _6	0x46	Jog mode. The drive can be run manually at the setup speed when the Jog mode is active. Control is implemented using Direction_Pos and Direction_Neg.	
						0:	The Jog mode is not active (selected).
						1:	The Jog mode has been selected.
Command[5]	183	0xB7	DST/ SRC	KBUS_CTRL1 _7	0x47	Mailbox mode. For this operating mode, all motion commands are given directly via the mailbox.	
						0:	The Mailbox mode is not active (selected).
						1:	The Mailbox mode has been selected.
Enable_Drive	184	0xB8	DST/ SRC	ONE	0x01	Enabling output stage. The output stage can be inhibited directly by deleting this bit. Contrary to Enable, this bit does not have any further effects on internal processing. The output stage is only enabled when Enable has been set and all other enable conditions are fulfilled. This bit is linked to ONE by default.	
						0:	Output stage inhibited.
						1:	Output stage can be enabled.

Designation	Bit number		Type	Default		Description	
	Dec.	Hex.		Target/Source	Bit no.		
Reset_Quit	185	0xB9	DST/ SRC	KBUS_CTRL3 _7	0x57	Reset acknowledgement	
						0:	Function not defined.
						1:	The Reset signal is reset.
Direction_ Pos	186	0xBA	DST/ SRC	KBUS_CTRL3 _2	0x52	In reference run operating mode, this bit establishes that the reference switch is searched for in positive direction.	
						0:	Drive not to move in a positive direction.
						1:	Drive should move in a positive direction. The drive is deactivated when the bit Direction_Neg is set at the same time.
Direction_ Neg	187	0xBB	DST/ SRC	KBUS_CTRL3 _3	0x53	In reference run operating mode, this bit establishes that the reference switch is searched for in negative direction.	
						0:	Drive not to move in a negative direction.
						1:	Drive should move in a negative direction. The drive is deactivated when the bit Direction_Pos is set at the same time.
Set_ Reference	188	0xBC	DST/ SRC	Input2	0x31	Reference input Input 2 is set to this bit in the standard configuration.	
						0:	The reference switch is not actuated.
						1:	The reference switch is actuated.
PreCalc	189	0xBD	DST/ SRC	KBUS_CTRL2 _6	0x4E	Save setpoints from Module 2.2 and, where applicable, calculate a movement sequence in advance.	
						0:	Each setpoint that is transmitted via cyclic telegram traffic must be accepted and processed. A possibly pre-calculated movement process will be discarded. A movement sequence can be calculated and started using Start.
						1:	The setpoints from the cyclic telegram traffic are ignored and the setpoint saved for the 0→1 edge used instead. If the starting speed is zero, a movement sequence will be calculated in advance using this setpoint; this sequence can then be started with the normal delay using Start.

Designation	Bit number		Type	Default		Description
	Dec.	Hex.		Target/Source	Bit no.	
SetupSpeed_ Active	190	0xBE	DST/ SRC	MZERO	0x02	Setup mode selected. When the bit SetupSpeed_Active_ACK is set the drive speed is limited to the defined setup speed. 0: Setup mode not selected. 1: Setup mode selected.
Error_Quit	191	0xBF	DST/ SRC	KBUS_CTRL2 _7	0x4F	Acknowledge error. All errors that are present are acknowledged at the rising edge from 0 to 1. After acknowledgement, the error switches to 0, or a new error is present:
LimitSwitch_ Pos	192	0xC0	SRC	Input5	0x34	Limit switch input on movement in positive direction. A direct input exists for this function. 0: The positive direction limit switch is not actuated. 1: The positive direction limit switch is actuated. The drive is ramped down.
LimitSwitch_ Neg	193	0xC1	SRC	Input6	0x35	Limit switch input on movement in negative direction. A direct input exists for this function. 0: The positive direction limit switch is not actuated. 1: The positive direction limit switch is actuated. The drive is ramped down.
Stop1_N	194	0xC2	DST/ SRC	Input1	0x30	Drive Stop 1 inverted. This bit is linked to input 1 of the terminal. The return message is transmitted via Stop_N bit. Stop1_N and Stop2_N are always taken into account at that bit. 0: Current is being supplied to the motor, but it is at standstill. If the motor still rotates, it will be driven to stand still by the STOP acceleration. The motor can not be started up. 1: The drive may be started.
Brake_ Manual	195	0xC3	DST/ SRC	MZERO	0x02	Manual actuation of brake

<table>
<tr><th rowspan="2">Designation</th><th colspan="2">Bit number</th><th rowspan="2">Type</th><th colspan="2">Default</th><th rowspan="2" colspan="2">Description</th></tr>
<tr><th>Dec.</th><th>Hex.</th><th>Target/Source</th><th>Bit no.</th></tr>
<tr><td>Freq_Range_ Sel_0</td><td>196</td><td>0xC4</td><td>DST/ SRC</td><td>KBUS_CTRL2 _0</td><td>0x48</td><td rowspan="2"></td><td rowspan="2">Configuration of velocity prescaler.
The prescaler Frq_Prescaler is set for speed using these two bits when the module is to be operated without configuration via the mailbox. These values are accepted only when Enable is set to 0.
<table>
<tr><th>Freq_ Range_ Sel_0</th><th>Freq_ Range_ Sel_1</th><th></th></tr>
<tr><td>0</td><td>0</td><td>The Freq_Prescaler prescaler is loaded with the parameter Freq_Div from the current configuration data set (*).</td></tr>
<tr><td>0</td><td>1</td><td>Freq_Prescaler = 80
Fmax = 25 kHz</td></tr>
<tr><td>1</td><td>0</td><td>Freq_Prescaler = 20
Fmax = 100 kHz</td></tr>
<tr><td>1</td><td>1</td><td>Freq_Prescaler = 4
Fmax = 500 kHz</td></tr>
</table>
(*) If the parameter Freq_Div in the configuration data set has been assigne a value of zero, the prescaler will be set to Freq_Prescaler = 200 (Fmax = 10 kHz) for Freq_Range_Sel_0 = 0 and Freq_Range_Sel_1 = 0.</td></tr>
<tr><td>Freq_Range_ Sel_1</td><td>197</td><td>0xC5</td><td>DST/ SRC</td><td>KBUS_CTRL2 _1</td><td>0x49</td></tr>
</table>

<table>
<tr><th rowspan="2">Designation</th><th colspan="2">Bit number</th><th rowspan="2">Type</th><th colspan="2">Default</th><th rowspan="2">Description</th></tr>
<tr><th>Dec.</th><th>Hex.</th><th>Target/Source</th><th>Bit no.</th></tr>
<tr><td>Acc_Range_ Sel_0</td><td>198</td><td>0xC6</td><td>DST/ SRC</td><td>KBUS_CTRL2 _0</td><td>0x48</td><td rowspan="2"><table>
<tr><td colspan="3">Configuration factor Acceleration.
These two bits are used to set the ACC_Multiplier factor for acceleration.
These values are accepted only when Enable is set to 0.</td></tr>
<tr><td>Acc_ Range_ Sel_0</td><td>Acc_ Range_ Sel_1</td><td></td></tr>
<tr><td>0</td><td>0</td><td>The factor Acc_Multiplier is loaded with the parameter Acc_Fact from the current configuration data set (*).</td></tr>
<tr><td>0</td><td>1</td><td>Acc_Multiplier = 80
T = 760 ms</td></tr>
<tr><td>1</td><td>0</td><td>Acc_Multiplier = 800
T = 76 ms</td></tr>
<tr><td>1</td><td>1</td><td>Acc_Multiplier = 8000
T = 7.6 ms</td></tr>
<tr><td colspan="3">(*) If the parameter Acc_Fact in the configuration data set has been assigned a value of zero, the factor is set to Acc_Multiplier = 8 (T = 7.6s) for Acc_Range_Sel_0 = 0 and Acc_Range_Sel_1 = 0.</td></tr>
</table></td></tr>
<tr><td>Acc_Range_ Sel_1</td><td>199</td><td>0xC7</td><td>DST/ SRC</td><td>KBUS_CTRL2 _1</td><td>0x49</td></tr>
<tr><td>Set_Actual_ POS</td><td>200</td><td>0xC8</td><td>DST/ SRC</td><td>KBUS_CTRL3 _0</td><td>0x50</td><td>The actual value is set to the reference position (configuration parameter Reference_Offset) on a rising edge from the bit Set_Actual_POS. This function can not be performed while a positioning run is ongoing.</td></tr>
<tr><td></td><td>201</td><td>0xC9</td><td></td><td></td><td></td><td></td></tr>
<tr><td>Jog_Pos</td><td>202</td><td>0xCA</td><td>DST/S RC</td><td>Input3</td><td>0x32</td><td><table>
<tr><td colspan="2">Move in positive direction.
This bit is required for the JogMode. The drive is controlled via these bits, after the selection of the corresponding operating mode.</td></tr>
<tr><td>0:</td><td>Drive not to move in a positive direction.</td></tr>
<tr><td>1:</td><td>Drive should move in a positive direction. The drive is deactivated when the bit Jog_Neg is set at the same time.</td></tr>
</table></td></tr>
</table>

Designation	Bit number		Type	Default		Description
	Dec.	Hex.		Target/Source	Bit no.	
Jog_Neg	203	0xCB	DST/SRC	Input4	0x33	Move in negative direction. This bit is required for the JogMode. The drive is controlled via these bits, after the selection of the corresponding operating mode. In reference run operating mode, this bit establishes that the reference switch is searched for in negative direction. 0: Drive not to move in a negative direction. 1: Drive should move in a negative direction. The drive is deactivated when the bit Jog_Neg is set at the same time.
	204	0xCC				
	205	0xCD				
	206	0xCE				
	207	0xCF				
LED_B	208	0xD0	DST/ SRC	Busy	0x09	Light-emitting diode B
LED_C	209	0xD1	DST/ SRC	ProgramRunning	0x1B	Light-emitting diode C
LED_D	210	0xD2	DST/ SRC	Stop_N_ACK	0x11	Light-emitting diode D
LED_A	211	0xD3	DST/ SRC	Error	0x19	Light-emitting diode A
LED_E	212	0xD4	DST/ SRC	Encoder_A	0x26	Light-emitting diode E
LED_F	213	0xD5	DST/ SRC	Encoder_B	0x27	Light-emitting diode F
	214	0xD6				
	215	0xD7				
	216	0xD8				
	217	0xD9				
	218	0xDA				
	219	0xDB				
	220	0xDC				
Error_Set	221	0xDD	DST/ SRC	ZERO	0x00	Setting the bit generates error message 1358. This bit can be used to trigger an error message with another arbitrary bit via appropriate linking. This bit can also be set by a run program.

Designation	Bit number		Type	Default		Description
	Dec.	Hex.		Target/Source	Bit no.	
TRACE_TRIGGER	222	0xDE	DST/ SRC	Busy	0x09	A trace is started with a positive edge when Trace_Armed is set. The Trace_Var1/2 variables given in the configuration are recorded using the time frame specified in Trace_MsecCycleTime.
Trace_Armed	223	0xDF	DST/ SRC	MONE	0x03	Trace activation
	224	0xE0				
	225	0xE1				
	226	0xE2				
	227	0xE3				
	228	0xE4				
	229	0xE5				
	230	0xE6				
	231	0xE7				
	232	0xE8				
	233	0xE9				
	234	0xEA				
	235	0xEB				
	236	0xEC				
	237	0xED				
	238	0xEE				
	239	0xEF				
	240	0xF0	INT			
	241	0xF1	INT			
	242	0xF2	INT			
	243	0xF3	INT			
	244	0xF4	INT			
	245	0xF5	INT			
	246	0xF6	INT			
	247	0xF7	INT			
	248	0xF8	INT			
	249	0xF9	INT			
	250	0xFA	INT			
	251	0xFB	INT			
	252	0xFC	INT			
	253	0xFD	INT			

Designation	Bit number		Type	Default		Description
	Dec.	Hex.		Target/Source	Bit no.	
	254	0xFE	INT			
	255	0xFF	INT			

4.6 Configuration Variables

Configuration variable	Address		Data type:	Default:	Range:	Description
	Dec.	Hex.				
User_Conf_Id	0	0x00	UINT16	0	0 ... 50000	The dataset number can be freely assigned by the user. Numbers above 50000 are reserved.
ConfVersion	2	0x02	UINT8	4	0 ...254	Configuration version number. This number identifies the structure of the dataset. (Order and length of the data.) If it becomes necessary to expand the dataset within the framework of future expansions, the number will be adjusted accordingly. The software can only read a dataset that is known. Other dataset numbers are rejected.
Application_ Selector	3	0x03	UINT8	1	1	Switching of the application The appropriate process image is activated when a new application is selected. 0: Reserved 1: Positioning controller
Freq_Div	4	0x04	UINT16	40	4 ...32767	Sets the prescaler for maximum speed, only valid when the bits Freq_Range_Sel_0 and Freq_Range_Sel_1 are zero.
Acc_Fact	6	0x06	UINT16	1000	1 ... 32767	Sets the factor for maximum acceleration, only valid if bits Acc_Range_Sel_0 and Acc_Range_Sel_1 are zero.
Current	14	0x0E	UINT8	50	0 ...50	Motor rated current in [0.1 A]
Current_Ratio_ StandStill	15	0x0F	UINT8	33	0 ...150	Current factor at standstill in [%], based on motor rated current "Current". A value of "0" blocks the modulation.
Current_Ratio_ RampUp	16	0x10	UINT8	100	0 ...150	Current factor for ramp-down in [%], based on motor rated current "Current". A value of "0" blocks the modulation.
Current_Ratio_ Drive	17	0x11	UINT8	50	0 ...150	Current factor for drive in [%], based on motor rated current "Current". A value of "0" blocks the modulation.
Current_Ratio_ RampDown	18	0x12	UINT8	100	0 ...150	Current factor for ramp-down in [%], based on motor rated current "Current". A value of "0" blocks the modulation.

Configuration variable	Address		Data type:	Default:	Range:	Description
	Dec.	Hex.				
HwSwConfig	19	0x13	UINT8	0		Bit 0 ... 1: reserved
						Bit 2: Drive_Direction (Direction of rotation inversion)
						0: Output signal is processed directly
						1: Output signal: rotary direction is inverted
						Bits 3 ... 6: reserved
						Bit 7: Program_Autostart (Move program Autostart – Normal mode)
						0: Move program activated only via Move program or Mailbox mode.
						1: Move program activated immediately after startup, see description.
Pos_Mult	20	0x14	UINT16	1	1 ...65535	Scaling factors for positions
Pos_Div	22	0x16	UINT16	1	1 ...65535	
Enc_Mult	24	0x18	UINT16	1	1 ...65535	Reserved
Enc_Div	26	0x1A	UINT16	1	1 ...65535	
Speed_Mult	28	0x1C	UINT16	1	1 ...65535	Scaling factors for speeds
Speed_Div	30	0x1E	UINT16	1	1 ...65535	Scaling factors for speeds
Acc_Mult	32	0x20	UINT16	1	1 ...65535	Scaling factors for acceleration
Acc_Div	34	0x22	UINT16	1	1 ...65535	Scaling factors for acceleration
Reserved_36	36	0x24	UINT16	0		Reserved
Reserved_38	38	0x26	UINT16	0		Reserved
Speed	40	0x28	INT16	3000	1 ...25000	Default speed
Speed_Limit	42	0x2A	INT16	25000	1 ...25000	Default maximum speed. When exceeding this speed, the drive is turned off.
SetupSpeed	44	0x2C	INT16	3000	1 ...25000	Default setup speed. When the value 0 is specified, a 1 value is automatically accepted.
Acceleration_ Stop_Fast	46	0x2E	INT16	30000	0 ...32767	Default acceleration for STOP operation. If this parameter is set to 0, the current acceleration speed is used.
Acceleration_ RampUp	48	0x30	INT16	1000	0 ...32767	Default acceleration in acceleration phase
Acceleration_ RampDown	50	0x32	INT16	1000	0 ...32767	Default acceleration in delay phase

Configuration variable	Address		Data type:	Default:	Range:	Description
	Dec.	Hex.				
Acceleration_ RampUp_Param	52	0x34	INT32	300	0 ...16777216	Default acceleration time or acceleration path Default value for Acc_ParamUp for acceleration No acceleration modification: time constant for acceleration increase with linear or sin^2. Constant acceleration modification: acceleration time Constant acceleration path modification: acceleration path Parameter 1-3: value 1..16777215
Acceleration_ RampDown_ Param	56	0x38	INT32	300	0 ...16777216	Default deceleration time or deceleration path Default value for Acc_ParamDown for deceleration No deceleration modification: time constant for deceleration increase with linear or sin^2. Constant deceleration time modification: deceleration time Constant deceleration path modification: deceleration path Parameter 1-3: value 1..16777215

Configuration variable	Address		Data type:	Default:	Range:	Description	
	Dec.	Hex.					
Acceleration_ Modes	60	0x3C	UINT8	0		Bit 0 ... 1: AccType (acceleration type)	
						0:	constant acceleration
						1:	linear rise in acceleration; the period for acceleration increase is Acceleration_RampUp_Param
						2:	Sin^2 rise in acceleration; the period for acceleration increase is Acceleration_RampUp_Param
						3:	Reserved
						Bit 2 ... 3: AccParam (acceleration parameter)	
						0:	no modification
						1:	Acceleration_RampUp_Param interpreted as the acceleration period
						2:	Acceleration_RampUp_Param interpreted as the acceleration path
						3:	Reserved
						Bit 4 ... 5: DecType (deceleration type)	
						0:	constant acceleration
						1:	linear rise in acceleration; the period for acceleration increase is Acceleration_RampUp_Param
						2:	Sin^2 rise in acceleration; the period for acceleration increase is Acceleration_RampUp_Param
						3:	Reserved
						Bit 6 .. 7: DecParam (decelration parameter)	
						0:	no modification
						1:	Acceleration_RampUp_Param interpreted as the acceleration period
						2:	Acceleration_RampUp_Param interpreted as the acceleration path
						3:	Reserved

Configuration variable	Address		Data type:	Default:	Range:	Description
	Dec.	Hex.				
Current_Ratio_ Stop	61	0x3D	UINT8	150	0 ...150	Current factor for STOP mode in [%], based on motor rated current "Current"
Setup Acceleration	62	0x3E	UINT16	1000	0 ...32767	Acceleration for JOG operation
Rotary_Axis_ Period	64	0x40	INT32	0	0 ...16777216	Sets period P for a round axis, Zero is entered for a linear axis.
Drive_Range_ Neg	68	0x44	INT32	- 8388607	-8388607 ... 8388607	Permissible travel range in negative direction
Drive_Range_ Pos	72	0x48	INT32	+ 8388607	+0x7fffffff ... 0x7fffffff	Permissible travel range in positive direction
Reserved_76	76	0x4C	UINT32	0		Reserved
Reserved_80	80	0x50	UINT32	0		Reserved
PWM_Period	84	0x54	UINT32	0	0 ...4294967295	Parameter has no meaning
Camshaft_Ch9_ Start	88	0x58	INT32	0	±8388607	Camshaft channel 9 position, starting edge in path increments
Camshaft_Ch9_ Period	92	0x5C	INT32	100	1 ...8388607	Camshaft channel 9 period in path increments
Camshaft_Ch9_ Pulsewidth	96	0x60	INT32	50	0 ...8388607	Camshaft channel 9 pulse width in path increments
Braketime_Turn_ On	100	0x64	UINT32	0	0 ...8388607	Brake make time in [ms]
Braketime_Turn_ Off	104	0x68	UINT32	0	0 ...8388607	Brake breaktime in [ms]
Reference_Offset	108	0x6C	UINT32	0	±8388607	Reference switch position

Configuration variable	Address		Data type:	Default:	Range:	Description	
	Dec.	Hex.					
Reference_Mode	112	0x70	UINT8	0		Mode for reference run at the start of a reference run via the cyclic process image (for starting a reference run using the move command START_REFERENCING, call parameters are used, but NOT the following configuration bits)	
						Bit 0:	
						0:	Reference run to reference switch
						1:	Reference run to limit switch
						Bit 1:	
						0:	Reference run to negative end of a reference switch
						1:	Reference run to positive end of a reference switch
						Bit 2 .. 7: reserved	
ErrorNotification Mode	113	0x71	UINT8	0x06		Bit 0: SystemFlagEnable: This parameter bit affects all error messages.	
						0:	Errors are not reported via internal bus status bit S0.6
						1:	Errors are reported via internal bus status bit S0.6
						Bit 1: OverTemperature: Overtemperature warning threshold (parameter 406) is analyzed as an error message	
						0:	Overtemperature is not reported as an error
						1:	Overtemperature is reported as an error
						Bit 2: MotorOff: Motor connection error message	
						0:	Faulty motor connection is not reported as an error
						1:	Faulty motor connection is reported as an error
						Bits 3 ... 7: reserved	
Reserved_114	114	0x72	INT16	0		Reserved	

Configuration variable	Address		Data type:	Default:	Range:	Description	
	Dec.	Hex.					
WarningNotification_Mode	116	0x74	INT 8	0x18		Bit 0: SystemFlagEnable: This parameter bit affects all warning messages	
						0:	Warning messages are not reported via internal bus status bit S0.6
						1:	Warning messages are reported via internal bus status bit S0.6
						Bit 1: OverTemperature: Overtemperature warning threshold (parameter 406) is analyzed.	
						0:	Overtemperature is not reported as a warning.
						1:	Overtemperature is reported as a warning.
						Bit 2 .. 7: reserved	
Reserved_117	117	0x75	INT8	0		Reserved	
Reserved_118	118	0x76	INT16	0	0 ...32767	Reserved	
Reserved_120	120	0x78	INT16	0	0 ...32767	Reserved	
Reserved_122	122	0x7A	INT16	0	0 ...32767	Reserved	
Reserved_124	124	0x7C	INT32	0		Reserved	
Ptr_KBUS_ ST1_0	128	0x80	UINT8	0x10	0 ...255	Source for linkable bit 0x80	
Ptr_KBUS_ ST1_1	129	0x81	UINT8	0x11	0 ...255	Source for linkable bit 0x81	
Ptr_KBUS_ ST1_2	130	0x82	UINT8	0x12	0 ...255	Source for linkable bit 0x82	
Ptr_KBUS_ ST1_3	131	0x83	UINT8	0x13	0 ...255	Source for linkable bit 0x83	
Ptr_KBUS_ ST1_4	132	0x84	UINT8	0x14	0 ...255	Source for linkable bit 0x84	
Ptr_KBUS_ ST1_5	133	0x85	UINT8	0x15	0 ...255	Source for linkable bit 0x85	
Ptr_KBUS_ ST1_6	134	0x86	UINT8	0x16	0 ...255	Source for linkable bit 0x86	
Ptr_KBUS_ ST1_7	135	0x87	UINT8	0x17	0 ...255	Source for linkable bit 0x87	
Ptr_KBUS_ ST2_0	136	0x88	UINT8	0x08	0 ...255	Source for linkable bit 0x88	
Ptr_KBUS_ ST2_1	137	0x89	UINT8	0x09	0 ...255	Source for linkable bit 0x89	

Configuration variable	Address		Data type:	Default:	Range:	Description
	Dec.	Hex.				
Ptr_KBUS_ ST2_2	138	0x8A	UINT8	0x0A	0 ...255	Source of linkable bit 0x8A
Ptr_KBUS_ ST2_3	139	0x8B	UINT8	0x0B	0 ...255	Source of linkable bit 0x8B
Ptr_KBUS_ ST2_4	140	0x8C	UINT8	0x0C	0 ...255	Source of linkable bit 0x8C
Ptr_KBUS_ ST2_5	141	0x8D	UINT8	0x0D	0 ...255	Source of linkable bit 0x8D
Ptr_KBUS_ ST2_6	142	0x8E	UINT8	0x0E	0 ...255	Source of linkable bit 0x8E
Ptr_KBUS_ ST2_7	143	0x8F	UINT8	0x0F	0 ...255	Source of linkable bit 0x8F
Ptr_KBUS_ ST3_0	144	0x90	UINT8	0x30	0 ...255	Source for linkable bit 0x90
Ptr_KBUS_ ST3_1	145	0x91	UINT8	0x31	0 ...255	Source for linkable bit 0x91
Ptr_KBUS_ ST3_2	146	0x92	UINT8	0x32	0 ...255	Source for linkable bit 0x92
Ptr_KBUS_ ST3_3	147	0x93	UINT8	0x33	0 ...255	Source for linkable bit 0x93
Ptr_KBUS_ ST3_4	148	0x94	UINT8	0x34	0 ...255	Source for linkable bit 0x94
Ptr_KBUS_ ST3_5	149	0x95	UINT8	0x35	0 ...255	Source for linkable bit 0x95
Ptr_KBUS_ ST3_6	150	0x96	UINT8	0x05	0 ...255	Source for linkable bit 0x96
Ptr_KBUS_ ST3_7	151	0x97	UINT8	0x04	0 ...255	Source for linkable bit 0x97
Ptr_UserBit_0	152	0x98	MZERO	0x02	0 ...1	User bits; e.g., for run program
Ptr_UserBit_1	153	0x99	MZERO	0x02	0 ...1	User bits; e.g., for run program
Ptr_UserBit_2	154	0x9A	MZERO	0x02	0 ...1	User bits; e.g., for run program
Ptr_UserBit_3	155	0x9B	MZERO	0x02	0 ...1	User bits; e.g., for run program
Ptr_UserBit_4	156	0x9C	MZERO	0x02	0 ...1	User bits; e.g., for run program
Ptr_UserBit_5	157	0x9D	MZERO	0x02	0 ...1	User bits; e.g., for run program
Ptr_UserBit_6	158	0x9E	MZERO	0x02	0 ...1	User bits; e.g., for run program
Ptr_UserBit_7	159	0x9F	MZERO	0x02	0 ...1	User bits; e.g., for run program
Ptr_OUT1	160	0xA0	UINT8	0x08	0 ...255	Source for linkable bit 0xA0
Ptr_OUT2	161	0xA1	UINT8	0x0F	0 ...255	Source for linkable bit 0xA1
Reserved_162	162	0xA2	UINT8	0x00	0 ...255	Source for linkable bit 0xA2
Reserved_163	163	0xA3	UINT8	0x00	0 ...255	Source for linkable bit 0xA3

Configuration variable	Address		Data type:	Default:	Range:	Description
	Dec.	Hex.				
Reserved_164	164	0xA4	UINT8	0x00	0 ...255	Source for linkable bit 0xA4
Reserved_165	165	0xA5	UINT8	0x00	0 ...255	Source for linkable bit 0xA5
Reserved_166	166	0xA6	UINT8	0x00	0 ...255	Source for linkable bit 0xA6
Reserved_167	167	0xA7	UINT8	0x00	0 ...255	Source for linkable bit 0xA7
Ptr_FILT1	168	0xA8	UINT8	0x00	0 ...255	Source for linkable bit 0xA8
Ptr_FILT2	169	0xA9	UINT8	0x00	0 ...255	Source for linkable bit 0xA9
Ptr_FILT3	170	0xAA	UINT8	0x00	0 ...255	Source of linkable bit 0xAA
Ptr_FILT4	171	0xAB	UINT8	0x00	0 ...255	Source of linkable bit 0xAB
Ptr_FILT5	172	0xAC	UINT8	0x00	0 ...255	Source of linkable bit 0xAC
Ptr_FILT6	173	0xAD	UINT8	0x00	0 ...255	Source of linkable bit 0xAD
Ptr_FILT7	174	0xAE	UINT8	0x00	0 ...255	Source of linkable bit 0xAE
Ptr_FILT8	175	0xAF	UINT8	0x00	0 ...255	Source of linkable bit 0xAF
Ptr_Enable	176	0xB0	UINT8	0x40	0 ...255	Source for linkable bit 0xB0
Ptr_Stop2_N	177	0xB1	UINT8	0x41	0 ...255	Source for linkable bit 0xB1
Ptr_Start	178	0xB2	UINT8	0x42	0 ...255	Source for linkable bit 0xB2
Ptr_Command[1]	179	0xB3	UINT8	0x43	0 ...255	Source for linkable bit 0xB3
Ptr_Command[2]	180	0xB4	UINT8	0x44	0 ...255	Source for linkable bit 0xB4
Ptr_Command[3]	181	0xB5	UINT8	0x45	0 ...255	Source for linkable bit 0xB5
Ptr_Command[4]	182	0xB6	UINT8	0x46	0 ...255	Source for linkable bit 0xB6
Ptr_Command[5]	183	0xB7	UINT8	0x47	0 ...255	Source for linkable bit 0xB7
Ptr_Enable_Drive	184	0xB8	UINT8	0x01	0 ...255	Source for linkable bit 0xB8
Ptr_Reset_Quit	185	0xB9	UINT8	0x57	0 ...255	Source for linkable bit 0xB9
Ptr_Direction_Pos	186	0xBA	UINT8	0x52	0 ...255	Source of linkable bit 0xBA
Ptr_Direction_Neg	187	0xBB	UINT8	0x53	0 ...255	Source for linkable bit 0xBB
Ptr_Set_Reference	188	0xBC	UINT8	0x31	0 ...255	Source of linkable bit 0xBC
Ptr_PreCalc	189	0xBD	UINT8	0x4E	0 ...255	Source of linkable bit 0xBD
Ptr_SetupSpeed_Active	190	0xBE	UINT8	0x02	0 ...255	Source of linkable bit 0xBE
Ptr_Error_Quit	191	0xBF	UINT8	0x4F	0 ...255	Source of linkable bit 0xBF
Ptr_LimitSwitch_Pos	192	0xC0	UINT8	0x34	0 ...255	Source for linkable bit 0xC0

Configuration variable	Address		Data type:	Default:	Range:	Description
	Dec.	Hex.				
Ptr_ LimitSwitch_ Neg	193	0xC1	UINT8	0x35	0 ...255	Source for linkable bit 0xC1
Ptr_Stop1_N	194	0xC2	UINT8	0x30	0 ...255	Source for linkable bit 0xC2
Break_Manual	195	0xC3	UINT8	0x02	0 ...255	Source for linkable bit 0xC3
Ptr_Freq_Range_ Sel_0	196	0xC4	UINT8	0x48	0 ...255	Source for linkable bit 0xC4
Ptr_Freq_Range_ Sel_1	197	0xC5	UINT8	0x49	0 ...255	Source for linkable bit 0xC5
Ptr_Acc_Range_ Sel_0	198	0xC6	UINT8	0x4A	0 ...255	Source for linkable bit 0xC6
Ptr_Acc_Range_ Sel_1	199	0xC7	UINT8	0x4B	0 ...255	Source for linkable bit 0xC7
Ptr_Set_Actual_ POS	200	0xC8	UINT8	0x50	0 ...255	Source for linkable bit 0xC8
Reserved_201	201	0xC9	UINT8	0x00	0 ...255	Source for linkable bit 0xC9
Ptr_Jog_Pos	202	0xCA	UINT8	0x32	0 ...255	Source of linkable bit 0xCA
Ptr_Jog_Neg	203	0xCB	UINT8	0x33	0 ...255	Source of linkable bit 0xCB
Reserved_204	204	0xCC	UINT8	0x00	0 ...255	Source of linkable bit 0xCC
Reserved_205	205	0xCD	UINT8	0x00	0 ...255	Source of linkable bit 0xCD
Reserved_206	206	0xCE	UINT8	0x00	0 ...255	Source of linkable bit 0xCE
Reserved_207	207	0xCF	UINT8	0x00	0 ...255	Source of linkable bit 0xCF
Ptr_LED_B	208	0xD0	UINT8	0x09	0 ...255	Source of linkable bit 0xD0
Ptr_LED_C	209	0xD1	UINT8	0x1B	0 ...255	Source of linkable bit 0xD1
Ptr_LED_D	210	0xD2	UINT8	0x11	0 ...255	Source of linkable bit 0xD2
Ptr_LED_A	211	0xD3	UINT8	0x19	0 ...255	Source of linkable bit 0xD3
Ptr_LED_E	212	0xD4	UINT8	0x26	0 ...255	Source of linkable bit 0xD4
Ptr_LED_F	213	0xD5	UINT8	0x27	0 ...255	Source of linkable bit 0xD5
Reserved_214	214	0xD6	UINT8	0x00	0 ...255	Source of linkable bit 0xD6
Reserved_215	215	0xD7	UINT8	0x00	0 ...255	Source of linkable bit 0xD7
Reserved_216	216	0xD8	UINT8	0x00	0 ...255	Source of linkable bit 0xD8
Reserved_217	217	0xD9	UINT8	0x00	0 ...255	Source of linkable bit 0xD9
Reserved_218	218	0xDA	UINT8	0x00	0 ...255	Source of linkable bit 0xDA
Reserved_219	219	0xDB	UINT8	0x00	0 ...255	Source of linkable bit 0xDB
Reserved_220	220	0xDC	UINT8	0x00	0 ...255	Source of linkable bit 0xDC
Reserved_221	221	0xDD	UINT8	0x00	0 ...255	Source of linkable bit 0xDD
Ptr_Trace_ Trigger	222	0xDE	UINT8	0x09	0 ...255	Source of linkable bit 0xDE

Configuration variable	Address		Data type:	Default:	Range:	Description
	Dec.	Hex.				
Ptr_Trace_ Armed	223	0xDF	UINT8	0x03	0 ...255	Source of linkable bit 0xDF
Filter1_Function	224	0xE0	UINT8	0	0 ...11	Function of filter: 0: No filtering 1: Inversion 2: Start edge detection 3: Low pass 4: Pulse expansion 5: Monoflop 6: Delay 7: Arithmetic 8: Counter, incrementing 9: Counter, incrementing to Zero 10: Counter, decrementing 11: Counter, decrementing to Zero
Filter2_Function	225	0xE1	UINT8	0		
Filter3_Function	226	0xE2	UINT8	0		
Filter4_Function	227	0xE3	UINT8	0		
Filter5_Function	228	0xE4	UINT8	0		
Filter6_Function	229	0xE5	UINT8	0		
Filter7_Function	230	0xE6	UINT8	0		
Filter8_Function	231	0xE7	UINT8	0		
Filter1_Time	232	0xE8	UINT32	0	0 ...16777215	Filter time constant in [ms]
Filter2_Time	236	0xEC	UINT32	0		
Filter3_Time	240	0xF0	UINT32	0		
Filter4_Time	244	0xF4	UINT32	0		
Filter5_Time	248	0xF8	UINT32	0		
Filter6_Time	252	0xFC	UINT32	0		
Filter7_Time	256	0x100	UINT32	0		
Filter8_Time	260	0x104	UINT32	0		
TraceVar1	264	0x108	UINT32	1	0 ...16777215	Variable number 1 for trace memory
TraceVar2	268	0x10C	UINT32	2	0 ...16777215	Variable number 2 for trace memory
TraceMsecCycle Time	272	0x110	UINT32	1	0 ...16777215	Cycle time for recording the variables given in TraceVar1/2 in [ms]

Configuration variable	Address		Data type:	Default:	Range:	Description
	Dec.	Hex.				
IdentNumber	276	0x114	UINT32	750672	0 ..99999999	WAGO 8 digit ID number (numerical value)
Reserved_280	280	0x118	UINT32	0	0 ...16777215	Reserved
Reserved_284	284	0x11C	UINT32	0	0 ...16777215	Reserved
Reserved_288	288	0x120	UINT32	0	0 ...16777215	Reserved
Reserved_292	292	0x124	UINT32	0	0 ...16777215	Reserved
Reserved_296	296	0x128	UINT32	0	0 ...16777215	Reserved
Reserved_300	300	0x12C	UINT32	0	0 ...16777215	Reserved
Reserved_304	304	0x130	UINT32	0	0 ...16777215	Reserved
Reserved_308	308	0x134	UINT32	0	0 ...16777215	Reserved
Reserved_312	312	0x138	UINT32	0	0 ...16777215	Reserved
Reserved_316	316	0x13C	UINT32	0	0 ...16777215	Reserved
Reserved_320	320	0x140	UINT32	0	0 ...16777215	Reserved
Reserved_324	324	0x144	UINT32	0	0 ...16777215	Reserved
Reserved_328	328	0x148	UINT32	0	0 ...16777215	Reserved
Reserved_332	332	0x14C	UINT32	0	0 ...16777215	Reserved
Reserved_336	336	0x150	UINT32	0	0 ...16777215	Reserved
Reserved_340	340	0x154	UINT32	0	0 ...16777215	Reserved
Reserved_344	344	0x158	UINT32	0	0 ...16777215	Reserved
Reserved_348	348	0x15C	UINT32	0	0 ...16777215	Reserved
Reserved_352	352	0x160	UINT32	0	0 ...16777215	Reserved
Reserved_356	356	0x164	UINT32	0	0 ...16777215	Reserved

Configuration variable	Address		Data type:	Default:	Range:	Description
	Dec.	Hex.				
Reserved_360	360	0x168	UINT32	0	0 ...16777215	Reserved
Reserved_364	364	0x16C	UINT32	0	0 ...16777215	Reserved
Reserved_368	368	0x170	UINT32	0	0 ...16777215	Reserved
Reserved_372	372	0x174	UINT32	0	0 ...16777215	Reserved
Reserved_376	376	0x178	UINT32	0	0 ...16777215	Reserved
MotorAlign	380	0x17C	UINT8	0	0 ... 255	Parameter has no meaning
Fan_Config	381	0x17D	UINT8	0		Bit 0: Configuration fan
						0: Normal operation: temperature during output stage > 60 °C, overtravel time = 10 min.
						1: Fan disabled
						2: Fan always on
Reserved_382	382	0x17E	UINT16	0	1 ... 65535	Reserved
Currrent_Ctrl_Hysteresis	384	0x180	UINT16	50	0 ...5000	Current controller hysteresis The current controller hysteresis has the unit [mA]; the range is 0 ... 5000. The parameter defines the width of the current band.
Reserved_386	386	0x186	UINT16	0	1 ... 65535	Reserved
Currrent_Ctrl_TZMin	388	0x184	UINT16	2	1 ... 65535	Current three-point controller only: Waiting time at state zero before switching band in [us * 4]
Reserved_390	390	0x186	UINT16	0	1 ... 65535	Reserved
Currrent_Ctrl_TZMax	392	0x188	UINT16	30	1 ... 65535	Current three-point controller only: Maximum time from state zero in [us * 4], subsequent forced band switching
Current_Pass2	394	0x18A	UINT16	0xC0F0	1 ... 65535	Password for current controller parameters. Password = Currrent_Ctrl_Hysteresis XOR Currrent_Ctrl_TZMin XOR Currrent_Ctrl_TZMax XOR Current_Ctrl_Type XOR 0xC0DE. If the password is incorrect, the error message CONFIG_WRONGPASSWORD (1162) is generated.

Configuration variable	Address		Data type:	Default:	Range:	Description
	Dec.	Hex.				
Current_Ctrl_Type	396	0x18C	UINT16	0		Bit 0: Current controller type 0: Three-point controller 1: Two-point controller
Reserved_398	398	0x18E	UINT16	0	1 ... 65535	Reserved
Reserved_400	400	0x190	UINT32	0	1 ... 65535	Reserved
Reserved_404	404	0x194	UINT16	0	1 ... 65535	Reserved
Warn_Treshold_Temperature	406	0x196	UINT16	700	0 ...800	Warning temperature in [°C*0.1] If the temperature during output stage exceeds this value, a warning is triggered.
DC_Link_Umin	408	0x198	UINT16	150	150 ...800	Minimum intermediate circuit voltage in [V*0.1] If the intermediate circuit voltage becomes smaller than DC_Link_Umin, an error message is generated.
DC_Link_Umax	410	0x19A	UINT16	800	0 ...800	Maximum intermediate circuit voltage in [V*0.1] If the intermediate circuit voltage becomes larger than DC_Link_Umax, an error message is generated.
Trace_QuickVar_Index	412	0x19C	UINT8	0	0 ... 255	Index of fast DSP variables. The DSP variables are reported to the processor in a cyclic manner. This results in delays of up to 0.5 s. This parameter can be used to select a variable which is transmitted every 1 ms.
Encoder_Config	413	0x19D	UINT8	0	0 ... 255	Parameter has no meaning
Encoder_Config_QDECCTL	414	0x19E	UINT16	0	0 ... 0xFFFF	Parameter has no meaning
Encoder_Resolution	416	0x1A0	UINT32	2000	1 ...6777215	Parameter has no meaning
Motor_Steps_per_Rev	420	0x1A4	UINT16	200	4 ...10000	Number of full steps of the motor being used. This requires 360° / 1.8° =200 to be parameterized for a typical motor with a 1.8° step size. The value must be divisible by 4.
Reserved_422	422	0x1A6	UINT16	0	0 ...65535	Reserved
Reserved_424	424	0x1A8	UINT16	0	0 ...65535	Reserved
Encoder_IndexMeasureJitterRange	426	0x1AA	UINT16	20	0 ...65535	Parameter has no meaning
SpeedMeasureTimeA	428	0x1AC	UINT32	100	0 ...65535	Parameter has no meaning

Configuration variable	Address		Data type:	Default:	Range:	Description
	Dec.	Hex.				
SpeedMeasureTimeoutA	432	0x1B0	UINT32	100	0 ...65535	Parameter has no meaning
SpeedMeasureTimeV	436	0x1B4	UINT32	3	0 ...65535	Parameter has no meaning
SpeedMeasureTimeoutV	440	0x1B8	UINT32	0	0 ...65535	Parameter has no meaning
Controller_v_Kp	444	0x1BC	UINT32	32000	0 ... 0x7fffffff	Parameter has no meaning
Controller_v_ITi	448	0x1C0	UINT32	16000	0 ... 0x7fffffff	Parameter has no meaning
Controller_s_Kp	452	0x1C4	UINT32	2000	0 ... 0x7fffffff	Parameter has no meaning
Controller_s_ITi	456	0x1C8	UINT32	0	0 ... 0x7fffffff	Parameter has no meaning
Controller_s_Td	460	0x1CC	UINT32	0	0 ...16777215	Parameter has no meaning
Controller_s_Tddi	464	0x1D0	UINT32	0	0 ... 0xFFFF	Parameter has no meaning
Reserved_468	468	0x1D4	UINT16	0	0 ...65535	Reserved
TargetWindowSpeed	470	0x1D6	UINT16	2	0 ... 0x7fff	Target window for speed control. If the final speed after the termination of the acceleration ramp in speed control mode differs from the target speed by less than the parameterized target window, the bit On_Target is set.
TargetWindowPosition	472	0x1D8	UINT32	0	0 ... 0xffffffff	Target window for position control. If the position after the termination of movement control in position control mode differs from the target position by less than the parameterized target window, the bit On_Target and the bits linked to it are set.
Reserved_476	476	0x1DC	UINT32	0	0 ...16777215	Reserved
Speed_Filter	480	0x1E0	UINT16	16384	0 ... 0xFFFF	Parameter has no meaning
TrackingError_Range_Speed	482	0x1E2	UINT16	100	0 ... 0x7FFF	Parameter has no meaning
TrackingError_Range_Position	484	0x1E4	UINT32	100	0 ... 0xffffffff	Parameter has no meaning
TrackingError_Delay	488	0x1E8	UINT32	100	0 ... 0xffffffff	Parameter has no meaning
ControllerMinLimit $^{*)}$	492	0x1EC	UINT16	8	1 ... 0x7FF	System input limition for position regulator (internally multiplied by factor 16)

Configuration variable	Address		Data type:	Default:	Range:	Description
	Dec.	Hex.				
Reserved_496	496	0x1F0	UINT32	0	0 ...16777215	Reserved
Reserved_500	500	0x1F4	UINT32	0	0 ...16777215	Reserved
Reserved_504	504	0x1F8	UINT32	0	0 ...16777215	Reserved
Reserved_508	508	0x1FC	UINT32	0	0 ...16777215	Reserved

*) from version v3504

4.7 Internal State Variables

Index		Variable	Value
Dec.	Hex.		
0	0x00	Time since program start	[ms]
1	0x01	Actual position (measured value)	[user unit]
2	0x02	Actual output frequency	[user unit]
3	0x03	Number of state variables/ARM-CPU only, without TMS variables	
4	0x04	0	
5	0x05	0	
6	0x06	0	
7	0x07	0	
8	0x08	Software Version	ASCII
9	0x09	Hardware Version	
			8: Module 750-672
			16: Module 750-673
11	0x0B	CompilationMonth	mmmm ASCII
12	0x0C	CompilationDayYear	ddyy ASCII
13	0x0D	Compilationtime	hhmm ASCII
15	0x0F	Expecting configuration version	
16	0x10	Version of current configuration	
20	0x14	Current frequency prescaler	
21	0x15	Prior overcurrent load 0 ... 254; when 255 is reached, the error message PARTMODL_CURRENT_TIME is issued.	0 ...255
22	0x16	0	
24	0x18	Set point position of travel generator	
25	0x19	Set point speed of travel generator	
30	0x1E	Last target position/target speed	
31	0x1F	Current acceleration factor	
32	0x20	Position error	
37	0x25	0	
38	0x26	0	
39	0x27	0	
40	0x28	Current run command	internal
41	0x29	Run program command counter	

Index		Variable	Value
Dec.	Hex.		
43	0x2B	Setpoint value speed	
44	0x2C	Maximum speed	
45	0x2D	Final speed	
46	0x2E	0	
47	0x2F	Stop delay	
49	0x31	Actual acceleration	
50	0x32	Actual delay	
51	0x33	Acceleration modification (travel, time)	
52	0x34	Delay modification (travel, time)	
53	0x35	Ramp mode	
54	0x36	Maximum speed	internal
55	0x37	Current speed prescaler	
56	0x38	Actual current factor	
62	0x3E	0	
63	0x3F	0	
88	0x58	0	
89	0x59	0	
90	0x5A	0	
91	0x5B	State variable Filter1	
92	0x5C	State variable Filter2	
93	0x5D	State variable Filter3	
94	0x5E	State variable Filter4	
95	0x5F	State variable Filter5	
96	0x60	State variable Filter6	
97	0x61	State variable Filter7	
98	0x62	State variable Filter8	
99	0x63	0	
1024	0x400	Time since TMS program start	[ms]
1025	0x401	Actual position (measured value)	[user unit]
1026	0x402	Actual output frequency	[user unit]
1027	0x403	Content of rapid state variables	
1028	0x404	State of signal processing controller	
1029	0x405	0	

Index		Variable		Value
Dec.	Hex.			
1030	0x406	TMS enabling block		
		Bit 0:	ARM enabling	
		Bit 1:	Overcurrent	
		Bit 2:	Error acknowledgement still present	
		Bit 3:	Error	
		Bit 4:	Reset not yet completed	
		Bit 5:	Incomplete TMS parameters	
		Bit 6:	Faulty intermediate circuit voltage	
		Bit 7:	24 V faulty	
		Bits 8 - 15:	Reserved	
1031	0x407	0		
1032	0x408	Software Version		ASCII
1033	0x409	Position of last Z pulse		Incremental encoder increments
1034	0x40A	0		
1035	0x40B	CompilationMonth		mmmm ASCII
1036	0x40C	CompilationDayYear		ddyy ASCII
1037	0x40D	Compilationtime		hhmm ASCII
1038	0x40E	0		
1043	0x413	0		
1044	0x414	Switching frequency, maximum value since enabling		
1045	0x415	Switching frequency, current value		
1046	0x416	Output stage temperature		[°C*0.1]
1047	0x417	24 V voltage		[V*0.1]
1048	0x418	Intermediate circuit voltage		[V*0.1]
1049	0x419	Rotational speed measured value, scaled in microsteps, parameterized with SpeedMeasureTimeV and SpeedMeasureTimeOutV		
1050	0x41A	Rotational speed, absolute frequency, parameterized with SpeedMeasureTimeA and SpeedMeasureTimeOutA		[mHz]
1051	0x41B	Absolute position, incremental encoder increments		
1052	0x41C	Mechanical angle rotation of motor, incremental encoder increments		0 ... EncoderResolution
1053	0x41D	Electric angle of commutation		0 ... 0xFFFF = 0 ... 360°

Index		Variable	Value
Dec.	**Hex.**		
1057	0x421	Current setpoint	[%*0.1]
1058	0x422	0	
1052	0x426	0	
1059	0x42D	Rotational speed error	
1060	0x42E	Position error	

WAGO Kontakttechnik GmbH & Co. KG
Postfach 2880 • D-32385 Minden
Hansastraße 27 • D-32423 Minden
Telefon: 05 71/8 87 – 0
Telefax: 05 71/8 87 – 1 69
E-Mail: info@wago.com

Internet: http://www.wago.com