cpc CHIEFTEK PRECISION CO., LTD.

# Will1/Will1-B Servo Drive

# Fieldbus

# Programming Manual

Revision 1.0

# Table of Contents

# 1. About this manual

This manual describes cpc's interpretation and implementation of the DS402 standard. It should not be used as the foundation to design generic DS402 master controllers, with the assumption that servo drives from other manufactures will have identical behavior.

This manual describes the objects and operation modes used in cpc drivers and is based on the CiA® 402 Draft Standard Proposal (DSP).

## 1.1 Revision History

| Revision | Date | Description | Remarks |
|---|---|---|---|
| 1.0 | December 2017 | Initial release | -- |

## 1.2 Copyright Notice

## 1.3 Trademarks

CANopen and CiA are registered trademarks of the CAN in Automation User's Group. Windows is a registered trademark of Microsoft Corporation.

# 1.4 Disclaimer

1. Information furnished by cpc is believed to be accurate and reliable. However, no responsibility is assumed by cpc for its use, nor for any infringements of patents or other rights of third parties which may result from its use. cpc doesn't grant any license under its patent rights, nor the rights of others.
2. In addition, cpc assumes no responsibility for any errors that may appear in this document and for any claims or damages arising from information contained in this document.
3. The product specified in this document has been developed, produced, tested and documented in accordance with the relevant standards. cpc is not responsible for damages, accidents, or injuries caused by any deviation from the configuration and installation described in this guide;
4. Furthermore, cpc is not responsible for the performance of new measurements or ensuring that regulatory requirements are met.
5. The product specified in this document is not assumed to be used in critical application including, but not limited to, medical equipment, transportation, aerospace and nuclear instruments, undersea equipment, power plant equipment, as well as disaster prevention and crime prevention equipment.
6. We reserve the right to modify our products, including its hardware and software design, in order to improve its design and/or performance. The information in this document is subject to change without notice and does not represent a commitment by cpc.
7. Specifications are subject to change without notice.
8. Performance specification beyond those specified by safety regulations are guaranteed by design and not subject to production test.
9. Customers should obtain the latest relevant information before placing orders and should verify that such information is current and complete.
10. cpc assumes no liability for applications assistance or customer product design. Customers are responsible for their products and applications using cpc products.

# 1.5 Contact Us

## Headquarters

## Chieftek Precision Co., Ltd.

NO.3, Dali 1st Rd., Xinshi Dist., Southern Taiwan Science Park,

Tainan City. 741-45, Taiwan (R.O.C.) 3

TEL: +886-6-505-5858

FAX: +886-6-505-5959

Email : service@mail.chieftek.com

## China

## Chieftek Machinery Kunshan Co., Ltd.

No.1188, Hongqiao Rd, Kunshan, Jiangsu, P.R. China

Tel : +86-512-55252831

Fax : +86-512-55252851

Email : cn.service@mail.chieftek.com

## Europe

cpc Europa GmbH

Industriepark 314, D-78244 Gottmadingen, Germany

Tel : +49-7731-59130-38

Fax : +49-7731-59130-28

Email : info@cpc-europa.de

## USA

Chieftek Precision USA Co., Ltd.

2280 E. Locust Court. Ontario, CA 91761, USA

TEL: +1-909-773-1200

FAX: +1-909-773-1202

Email : info@usa.chieftek.com

# 2. Introduction

## 2.1 Abbreviations and terms

| Term / Abbrev. | Stands for: |
|---|---|
| AC | Alternating current |
| C | Constant |
| COB | Communication object |
| csp | Cyclic synchronous position mode |
| cst | Cyclic synchronous torque mode |
| csv | Cyclic synchronous velocity mode |
| DC | Direct current |
| FE | Functional element |
| hm | Homing mode |
| I/O | Input/output |
| ms | Manufacturer-specific |
| NMT | Network management |
| PDO | Process data object |
| PDS | Power drive system |
| pp | Profile position mode |
| pv | Profile velocity mode |
| r | Reserved |
| r.m.s. | Root mean square |
| RO | Read only |
| RW | Read-write |
| tq | Torque mode (= profile torque) |

## 2.2 Operation modes

The cpc device profile specifies the modes of operation, including:

- Profile position mode
- Profile velocity mode
- Profile torque mode
- Homing mode
- Cyclic synchronous position mode
- Cyclic synchronous velocity mode
- Cyclic synchronous torque mode

# 2.3 Standard Servo Drive Objects (0x60nn)

(The “ * ” sign refers to the numeric range of the data type.)

| Index | Sub-index | Name | Type | Access | Default | Max | Min | Unit | PDO mapping |
|---|---|---|---|---|---|---|---|---|---|
| 0x6007 | 0x00 | Abort connection option code | INT16 | RW | 0 | 3 | 0 | – | X |
| 0x603F | 0x00 | Error Code | UINT16 | RO | 0 | * | * | – | O |
| 0x6040 | 0x00 | Controlword | UINT16 | RW | 0 | * | * | – | O |
| 0x6041 | 0x00 | Statusword | UINT16 | RO | – | * | * | – | O |
| 0x605A | 0x00 | Quick stop option code | INT16 | RW | 2 | 4 | -1 | – | X |
| 0x605B | 0x00 | Shutdown option code | INT16 | RW | 0 | 1 | -1 | – | X |
| 0x605C | 0x00 | Disable operation option code | INT16 | RW | 0 | 1 | -1 | – | X |
| 0x605D | 0x00 | Halt option code | INT16 | RW | 1 | 4 | -1 | – | X |
| 0x605E | 0x00 | Fault reaction option code | INT16 | RW | 0 | 4 | -1 | – | X |
| 0x6060 | 0x00 | Modes of operation | INT8 | RW | 0 | 10 | -6 | – | O |
| 0x6061 | 0x00 | Modes of operation display | INT8 | RO | 0 | * | * | – | O |

| Index | Sub-index | Name | Type | Access | Default | Max | Min | Unit | PDO mapping |
|---|---|---|---|---|---|---|---|---|---|
| 0x6062 | 0x00 | Position demand value | INT32 | RO | – | * | * | count | O |
| 0x6063 | 0x00 | Position actual internal value | INT32 | RO | 0 | * | * | count | O |
| 0x6064 | 0x00 | Position actual value | INT32 | RO | 0 | * | * | count | O |
| 0x6065 | 0x00 | Following error window | UINT32 | RW | 0 | * | * | count | O |
| 0x6066 | 0x00 | Following error time out | UINT16 | RW | 0 | * | * | ms | O |
| 0x6067 | 0x00 | Position window | UINT32 | RW | 0 | * | * | count | O |
| 0x6068 | 0x00 | Position window time | UINT16 | RW | 0 | * | * | ms | O |
| 0x606B | 0x00 | Velocity demand value | INT32 | RO | - | * | * | count/s | O |
| 0x606C | 0x00 | Velocity actual value | INT32 | RO | 0 | * | * | count/s | O |
| 0x606D | 0x00 | Velocity window | UINT16 | RW | 0 | * | * | count/s | O |
| 0x606E | 0x00 | Velocity window time | UINT16 | RW | 0 | * | * | ms | O |
| 0x606F | 0x00 | Velocity threshold | UINT16 | RW | 0 | * | * | count/s | O |

| Index | Sub-index | Name | Type | Access | Default | Max | Min | Unit | PDO mapping |
|---|---|---|---|---|---|---|---|---|---|
| 0x6070 | 0x00 | Velocity threshold time | UINT16 | RW | 0 | * | * | ms | O |
| 0x6071 | 0x00 | Target torque | UINT16 | RW | 0 | * | * | 0.10% | O |
| 0x6073 | 0x00 | Max current | UINT16 | RW | 0 | * | * | 0.1% | O |
| 0x6074 | 0x00 | Torque demand | INT16 | RO | – | | | 0.1% | O |
| 0x6075 | 0x00 | Motor rated current | UINT32 | RW | 0 | * | * | mA | X |
| 0x6076 | 0x00 | Motor rated torque | UINT32 | RW | 0 | * | * | mNm (milli Newton metre) | X |
| 0x6077 | 0x00 | Torque actual value | INT16 | RO | 0 | * | * | 0.1% | O |
| 0x6078 | 0x00 | Current actual value | INT16 | RO | 0 | * | * | 0.1% | O |
| 0x6079 | 0x00 | DC link circuit voltage | UINT32 | RO | 0 | * | * | mV. | X |
| 0x6080 | 0x00 | Max motor speed | UINT32 | RW | 1,500,000 | * | * | count/s | O |
| 0x607A | 0x00 | Target position | INT32 | RW | 0 | * | * | count | O |
| 0x607B | 0x00 | Highest sub-index supported | 2 | C | 2 | 2 | | – | X |
| | 0x01 | Min position range limit | INT32 | RW | $-2^{31}$ | * | * | count | X |

| Index | Sub-index | Name | Type | Access | Default | Max | Min | Unit | PDO mapping |
|---|---|---|---|---|---|---|---|---|---|
|  | 0x02 | Max position range limit | INT32 | RW | $2^{31}$-1 | * | * | count | X |
| 0x607C | 0x00 | Home offset | INT32 | RW | 0 | * | * | count | O |
| 0x607D | 0x00 | Highest sub-index supported | INT32 | C | 2 | 2 |  | – | X |
|  | 0x01 | Min position limit | INT32 | RW | -$2^{31}$ | * | * | count | X |
|  | 0x02 | Max position limit | INT32 | RW | $2^{31}$-1 | * | * | count | X |
| 0x607F | 0x00 | Max profile velocity | UINT32 | RW | 1,500,000 | * | * | count/s | O |
| 0x6081 | 0x00 | Profile velocity | UINT32 | RW | 2,000,000 | 0x7FFFFFFF | 1 | count/s | O |
| 0x6083 | 0x00 | Profile acceleration | UINT32 | RW | 1,000,000 | 0x7FFFFFFF | 1 | count/$s^2$ | X |
| 0x6084 | 0x00 | Profile deceleration | UINT32 | RW | 1,000,000 | 0x7FFFFFFF | 1 | count/$s^2$ | O |
| 0x6085 | 0x00 | Quick stop deceleration | UINT32 | RW | 100,000,000 | 0x7FFFFFFF | 1 | count/$s^2$ | O |
| 0x6087 | 0x00 | Torque slope | UINT32 | RW | 100,000 | * | * | 0.1%/s | O |
| 0x6098 | 0x00 | Homing method | INT8 | RW | 35 | 37 | -12 | – | O |
| 0x6099 | 0x00 | Highest sub-index supported | UINT32 | C | 2 | * | * | – | X |
|  | 0x01 | Speed during search for switch | UINT32 | RW | 20,000 | * | * | count/s | X |

| Index | Sub-index | Name | Type | Access | Default | Max | Min | Unit | PDO mapping |
|---|---|---|---|---|---|---|---|---|---|
| | 0X02 | Speed during search for zero | UINT32 | RW | 20,000 | * | * | count/s | X |
| 0x609A | 0x00 | Homing acceleration | UINT32 | RW | 20,000 | 0x7FFFFFFF | 1 | count/s$^2$ | O |
| 0x60B0 | 0x00 | Position offset | INT32 | RW | 0 | * | * | count | O |
| 0x60B1 | 0x00 | Velocity offset | INT32 | RW | 0 | * | * | count/s | O |
| 0x60B2 | 0x00 | Torque offset | INT16 | RW | 0 | * | * | 0.1% | O |
| 0x60B8 | 0x00 | Touch probe function | UINT16 | RW | 1 | * | * | – | O |
| 0x60B9 | 0x00 | Touch probe status | UINT16 | RO | – | – | | – | O |
| 0x60BA | 0x00 | Touch probe 1 positive edge | INT32 | RO | – | * | * | count | O |
| 0x60BB | 0x00 | Touch probe 1 negative edge | INT32 | RO | – | * | * | count | O |
| 0x60BC | 0x00 | Touch probe 2 positive edge | INT32 | RO | – | * | * | count | O |
| 0x60BD | 0x00 | Touch probe 2 negative edge | INT32 | RO | – | * | * | count | O |

| Index | Sub-index | Name | Type | Access | Default | Max | Min | Unit | PDO mapping |
|---|---|---|---|---|---|---|---|---|---|
| 0x60C2 | 0x00 | Highest sub-index supported | UINT8 | C | 2 | 2 | | – | X |
| | 0x01 | Interpolation time period value | UINT8 | RW | 1 | 255 | 1 | – | X |
| | 0x02 | Interpolation time index | INT8 | RW | -3 | 1 | -3 | – | X |
| 0x60D5 | 0x00 | Touch probe 1 positive edge counter | UINT16 | RO | – | * | * | – | O |
| 0x60E3 | 0x00 | Highest sub-index supported | INT8 | C | 44 | – | | – | X |
| | 0x01 | 1st supported homing method | INT8 | C | (-12 ~ -1) and all CiA402 standard mode | – | | – | X |
| | 0x02 | 2nd supported homing method | INT8 | C | (-12 ~ -1) and all CiA402 standard mode | – | | – | X |
| | 0xFE | 254 supported homing method | INT8 | C | (-12 ~ -1) and all CiA402 standard mode | – | | – | X |

| Index | Sub-index | Name | Type | Access | Default | Max | Min | Unit | PDO mapping |
|---|---|---|---|---|---|---|---|---|---|
| 0x60D6 | 0x00 | Touch probe 1 negative edge counter | UINT16 | RO | – | * | * | – | O |
| 0x60F4 | 0x00 | Following error actual value | INT32 | RO | – | * | * | count | O |
| 0x60FC | 0x00 | Position demand internal value | INT32 | RO | – | * | * | in increments of the position encoder | O |
| 0x60FD | 0x00 | Digital inputs | UINT32 | RO | – | – |  | – | O |
| 0x60FE | 0x00 | Highest sub-index supported | UINT32 | C | 1 | 0x01 |  | – | X |
|  | 0x01 | Physical outputs | UINT32 | RW | 0000 0000h | * | * | – | X |
| 0x60FF | 0x00 | Target Velocity | INT32 | RW | 0 | * | * | count/s$^2$ | O |

| Index | Sub-index | Name | Type | Access | Default | Max | Min | Unit | PDO mapping |
|---|---|---|---|---|---|---|---|---|---|
| 0x6502 | 0x00 | Supported drive modes | UINT32 | RO | 0x3AD | cst, csv, csp, hm, tq, pv, and pp bits:<br>1 = mode is supported.<br>0 = mode is not supported<br><br>manufacturer-specific bits:<br>No.<br><br>r(eserved) bits: 0 | | – | O |
| 0x67FE | 0x00 | Version number | UINT32 | C | 3 | – | | – | X |

# 2.4 Format of Object Dictionary

The format of the object description and the entry description in this manual is as follows:

- Object description

| | |
|---|---|
| Index | nnnn |
| Name | Name of the object |
| Object code | Variable / Array / Record |
| Data type | Integer8 / Integer16 / Integer32<br>Unsigned8 / Unsigned16 / Unsigned32 |

- Entry description

| | |
|---|---|
| Sub-index | 0xnn |
| Description | Description of the sub-index |
| Access | RW / RO / C<br>**(Read/Write / Read Only / Constant)** |
| PDO mapping | Yes / No |
| Value range | Number or INT or UINT |
| Default value | The object’s default value |
| Units | When the object involves measurement, unit is applied. |

- The “value range” in the entry description:

| Description | Numeric range |
|---|---|
| INT8 | $-2^{7}$ ~ $2^{7}-1$ |
| UINT8 | 0 ~ $2^{7}-1$ |
| INT16 | $-2^{15}$ ~ $2^{15}-1$ |
| UINT16 | 0 ~ $2^{15}-1$ |
| INT32 | $-2^{31}$ ~ $2^{31}-1$ |
| UINT32 | 0 ~ $2^{31}-1$ |

# 3. General Entries

## 3.1 Drive Error

### Object 0x6007: Abort connection option code

This object indicates what the reaction will be when one of the following events occurs:
CAN bus-off, heartbeat, fieldbus stopped state entered, and reset communication.

- Object description

| | |
|---|---|
| Index | 6007 |
| Name | Abort connection option code |
| Object code | Variable |
| Data type | Interger16 |

- Entry description

| | |
|---|---|
| Sub-index | 0x00 |
| Access | RW |
| PDO mapping | No |
| Value range | 0 ~ 3 |
| Default value | 0 |
| Units | No |

- Value definition

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

# Object 0x603F: Error code

This object indicates the last error that appears in the drive device.

- Object description

| | |
|---|---|
| Index | 603F |
| Name | Error code |
| Object code | Variable |
| Data type | Unsigned16 |

- Entry description

| | |
|---|---|
| Sub-index | 0x00 |
| Access | RO |
| PDO mapping | Yes |
| Value range | Unsigned16 |
| Default value | 0 |
| Units | No |

# 3.2 Drive Data

## Object 0x6502: Supported drive modes

This object indicates what modes are supported. See bit definitions below.

- Object description

| | |
|---|---|
| Index | 6502 |
| Name | Supported drive modes |
| Object code | Variable |
| Data type | Unsigned32 |

- Entry description

| | |
|---|---|
| Sub-index | 0x00 |
| Access | RO |
| PDO mapping | Yes |
| Value range | cst, csv, csp, hm, tq, pv, and pp bits:<br>1 = mode is supported.<br>0 = mode is not supported<br><br>manufacturer-specific bits:<br>No.<br><br>r(eserved) bits: 0 |
| Default value | 0x3AD |
| Units | No |

- Bit definition

| Bit | Function |
|---|---|
| 0 | pp |
| 1 | Reserved |
| 2 | pv |
| 3 | tq |
| 4 | Reserved |
| 5 | hm |
| 6 | Reserved |
| 7 | csp |
| 8 | csv |
| 9 | cst |
| 10 | Reserved |
| 11-15 | Reserved |
| 16-31 | Reserved |

# 4. Device Control

## 4.1 General

The PDS finite state machine is an abstract concept to define the behavior of a black box when a control device interacts with the PDS. It defines the application behavior of the PDS.

The PDS finite state machine is operated by these means:

- Controlword from control device sent via network;
- Local signals, such as script, faults, or signals sent via RS232.
- The state of the PDS reported by the statusword produced by the drive device.
- Error detection signals.

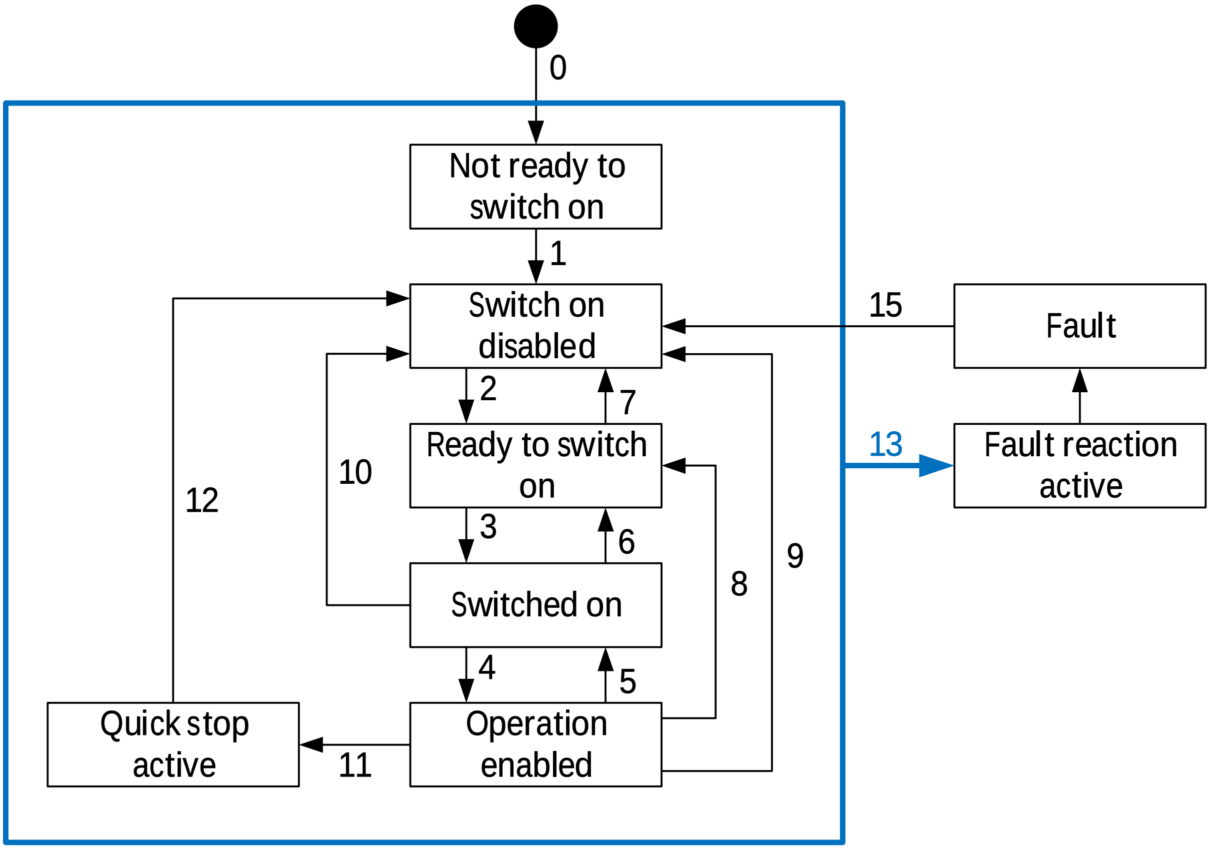
## 4.2 Finite State Machine and States

- The state machine describes the device status and the possible control sequence of the drive.
- A single state represents a specific internal or external behavior.
- The state of the drive also determines which commands are accepted; for example, a point-to-point motion can be started only when the drive is in OPERATION ENABLED state.

The device states and possible control sequence of the drive are described by the state machine, as depicted in the following figure:

### 4.2.1 Diagram of Power Drive System Finite State Machine

< Diagram of Power Drive System Finite State Machine >



## 4.2.2 Description of the states of finite state machine

| State | Description |
|---|---|
| Not Ready to Switch On | l Low-level power (e.g. 24V, 5V) has been applied to the drive.<br>l Drive is being initialized or is running self-test;<br>The communication channel is opened after this state hence users won't be able to encounter this state in practice.<br>l A brake, if present, is applied in this state.<br>l Drive function is disabled. |
| Switch On Disabled | l Drive initialization is complete.<br>l Drive parameters have been set up.<br>l Drive parameter may be changed.<br>l Drive function is disabled. |
| Ready to Switch On | l Drive parameters may be changed.<br>l Drive function is disabled. |
| Switched On | l High voltage has been applied to the drive.<br>l Drive parameter may be changed.<br>l Drive function is disabled. |
| Operation Enabled | l No faults have been detected.<br>l Drive function is enabled and power is applied to the motor.<br>l Motor operation related parameters cannot be changed. |
| Quick Stop Active | l Drive parameters may be changed.<br>l Quick stop function is being executed.<br>l Drive function is enabled and power is applied to the motor. |
| Fault Reaction Active | l Drive parameters may be changed.<br>l A non-fatal fault has occurred in the drive.<br>l Quick stop function is being executed if fault reaction option is set to quick stop.<br>l Drive function is enabled and power is applied to the motor. |
| Fault | l Drive parameters may be changed.<br>l A fault has occurred in the drive.<br>l Drive function is disabled. |

## 4.2.3 Descriptions of the transitions of finite state machine

|  | From state | To state | Event/Action |  |
|---|---|---|---|---|
| 0 | Start | Not Ready to Switch On | Event:<br>Action: | Power-on reset.<br>The drive self-tests and/or self-initializes. |
| 1 | Not Ready to Switch On | Switch On Disabled | Event:<br>Action: | The drive has self-tested and/or initialized successfully.<br>Activate communication and process data monitoring |
| 2 | Switch On Disabled | Ready to Switch On | Event:<br>Action: | "Shutdown" command received from controlword.<br>None |
| 3 | Ready to Switch On | Switched On | Event:<br>Action: | "Switch On" command received from controlword.<br>None. |
| 4 | Switched On | Operation Enabled | Event:<br>Action: | "Enable Operation" command received from controlword.<br>The drive function is enabled. |
| 5 | Operation Enabled | Switched On | Event:<br>Action: | "Disable operation" command received from controlword.<br>The drive operation is disabled. |
| 6 | Switched On | Ready to Switch On | Event:<br>Action: | "Shutdown" command received from controlword.<br>None. |
| 7 | Ready to Switch On | Switch On Disabled | Event:<br>Action: | "Quick Stop" command received from controlword.<br>None. |
| 8 | Operation Enabled | Ready to Switch On | Event:<br>Action: | "Shutdown" command received from controlword.<br>Drive function is disabled and the motor is free to rotate if unbraked. |
| 9 | Operation Enable | Switch On Disabled | Event:<br>Action: | "Disable Voltage" command received from controlword.<br>Drive function is disabled and the motor is free to rotate if unbraked. |
| 10 | Switched On | Switch On Disabled | Event:<br>Action: | "Disable Voltage" or "Quick Stop" command received from controlword.<br>Drive function is disabled and the motor is free to rotate if unbraked. |

|  | From state | To state | Event/Action |  |
|---|---|---|---|---|
| 11 | Operation Enable | Quick Stop Active | Event:<br>Action: | “Quick Stop” command received from controlword.<br>The Quick Stop function is executed. |
| 12 | Quick Stop Active | Switch On Disabled | Event:<br>Action: | “Quick Stop” function is completed or “Disable Voltage” command received from controlword.<br>Drive function is disabled |
| 13 | Any state | Fault Reaction Active | Event:<br>Action: | A fatal fault has occurred in the drive.<br>Execute appropriate fault reaction. |
| 14 | Fault Reaction Active | Fault | Event:<br>Action: | The fault reaction is completed.<br>The drive function is disabled. |
| 15 | Fault | Switch On Disabled | Event:<br>Action: | “Fault reset” command received from controlword.<br>l If no fault exists currently on the drive, a reset of the fault condition will be carried out.<br>l After leaving the “Fault” state, the “Fault Reset” bit in controlword should be cleared to 0 for future fault reset command. |

## 4.3 Detailed Objects specifications

The following chapter describes

- Controlword and Statusword – Ch.4.3.1 and 4.3.2

  The state of the device is controlled by the controlword, while the status of the device is indicated by the statusword.

  The following content includes:

  - Structure of controlword and statusword
  - Command coding and state coding
  - Statusword bit interpretations

- Objects of stop, halt, and fault – Ch. 4.3.3 to 4.3.7

### 4.3.1 Object 0x6040: Controlword

This object indicates the received command that controls the PDS FSA.

Please refer to the chart of <Controlword Structure>, its bits are structured as described in the chart below.

- Bits 0, 1, 2, 3, and 7 are supported.
- Bits 4, 5, 6, and 9 are operation mode specific.
- Bit 8 – the action of bit 8 is mode-specific.

  If bit 8 = 1, the commanded motion will be interrupted and the PDS will act according to the halt option code.
- Bit 10 is reserved.
- Bits 11 to 15 are manufacturer-specific according to 402 DSP; these bits are reserved by cpc.

- Object description

| | |
|---|---|
| Index | 6040 |
| Name | Controlword |
| Object code | Variable |
| Data type | Unsigned16 |

- Entry description

| | |
|---|---|
| Sub-index | 0x00 |
| Access | RW |
| PDO mapping | Yes |
| Value range | Unsigned16 |
| Default value | 0 |
| Units | No |

## 4.3.1.1 Controlword structure

< Controlword Structure >

| Bit | Keys | Name |
|---|---|---|
| 0 | so | Switch on |
| 1 | ev | Enable voltage |
| 2 | qs | Quick stop |
| 3 | eo | Enable operation |
| 4~6 | oms | Operation mode specific |
| 7 | fr | Fault reset |
| 8 | h | Halt |
| 9 | oms | Operation mode specific |
| 10 | r | Reserved |
| 11~15 | (ms) | Manufacturer-specific; these are reserved by cpc. |

## 4.3.1.2 Command coding

The controlword contains the bits controlling the states of PDS, these commands are coded in the way as described in the chart here:

< Command Coding >

| Command | Bits of the controlword | | | | | Transitions |
|---|---|---|---|---|---|---|
| | 7 | 3 | 2 | 1 | 0 | |
| | fault reset | enable operation | quick stop | enable voltage | switch on | |
| Shut down | 0 | x | 1 | 1 | 0 | 2, 6, 8 |
| Switch on | 0 | 0 | 1 | 1 | 1 | 3 |
| Switch on + enable operation | 0 | 1 | 1 | 1 | 1 | 3 + 4<br>(*note) |
| Disable voltage | 0 | x | x | 0 | x | 7, 9, 10, 12 |
| Quick stop | 0 | x | 0 | 1 | x | 7, 10, 11 |
| Disable operation | 0 | 0 | 1 | 1 | 1 | 5 |
| Enable operation | 0 | 1 | 1 | 1 | 1 | 4, 16 |
| Fault reset | ↑ | x | x | x | x | 15 |

*note:

Automatically transit to the “operation enabled” state after performing the “switched on” state functionality.

## 4.3.2 Object 0x6041: Statusword

This object indicates the status of PDS FSA.

It is structured as defined in the chart below:

- Bits 0 to 10 are supported
- Bit 7, originally defined as warning by 402 DSP, is hereby reserved by cpc.
- Bit 8 and 15, originally defined as manufacturer-specific, is hereby reserved by cpc.
- Bits 12 and 13:

  If the related functionality of the oms bits is not available, the bit will set to 0.

- Object description

| | |
|---|---|
| Index | 6041 |
| Name | Statusword |
| Object code | Variable |
| Data type | Unsigned16 |

- Entry description

| | |
|---|---|
| Sub-index | 0x00 |
| Access | RO |
| PDO mapping | Yes |
| Value range | Unsigned16 |
| Default value | No |
| Units | No |

## 4.3.2.1 Statusword structure

< Statusword Structure >

| Bit | Keys | Name |
|---|---|---|
| 0 | rtso | Ready to switch on |
| 1 | so | Switched on |
| 2 | oe | Operation enabled |
| 3 | F | Fault |
| 4 | ve | Voltage enabled |
| 5 | qs | Quick stop |
| 6 | sod | Switch on disabled |
| 7 ~ 8 | (r) | Reserved |
| 9 | rm | Remote |
| 10 | tr | Target reached |
| 11 | ila | Internal limit active |
| 12-13 | oms | Operation mode specific |
| 14 | (ms) | When bit 14 is set to 1, it means<br>(1) The transition from power-off to operation enabled state; or<br>(2) Quick-stop is in process, or, waiting for external enable. |
| 15 | (r) | Reserved |

## 4.3.2.2 State coding

The combinations of bits 0 to 7 will code the states of PDS FSA. See the chart below:

< State Coding >

| Statusword | | | | | | | | | PDS FSA state |
|---|---|---|---|---|---|---|---|---|---|
| | w | sod | qs | ve | f | oe | so | rtso | |
| | Warning | Switch On Disabled | Quick Stop | Voltage Enabled | Fault | Operation Enabled | Switched On | Ready To Switch On | |
| xxxx xxxx | x | 0 | x | x | 0 | 0 | 0 | 0 | Not ready to switch on |
| xxxx xxxx | x | 1 | x | x | 0 | 0 | 0 | 0 | Switch on disabled |
| xxxx xxxx | x | 0 | 1 | x | 0 | 0 | 0 | 1 | Ready to switch on |
| xxxx xxxx | x | 0 | 1 | x | 0 | 0 | 1 | 1 | Switched on |
| xxxx xxxx | x | 0 | 1 | x | 0 | 1 | 1 | 1 | Operation enabled |
| xxxx xxxx | x | 0 | 0 | x | 0 | 1 | 1 | 1 | Quick stop active |
| xxxx xxxx | x | 0 | x | x | 1 | 1 | 1 | 1 | Fault reaction active |
| xxxx xxxx | x | 0 | x | x | 1 | 0 | 0 | 0 | Fault |

## 4.3.2.3 Statusword bit interpretations

| Bit | Name | Value | Explanations |
|---|---|---|---|
| 0 | Ready to switch on | -- | -- |
| 1 | Switched on | -- | -- |
| 2 | Operation enabled | -- | -- |
| 3 | Fault | -- | -- |
| 4 | Voltage enabled | 1 | High voltage is applied to the PDS. |
| 5 | Quick stop | 0 | The PDS is reacting on a quick stop request. |
| 6 | Switch on disabled | -- | -- |
| 7~8 | Reserved | -- | -- |
| 9 | Remote | 0 (local) | Controlword is not processed. |
| | | 1 (remote) | Controlword is processed. |
| 10 | Target reached | 1 | ① Indicates that the PDS has reached the set-point. Bit 10 is operation mode specific, please see related chapters.<br>② The operation mode is changed.<br><br>*Note:<br>Changing the target value via software will alter this bit (0 à 1 or 1 à 0). |
| 11 | Internal limit active | 1 | An internal limit is active.<br>(example: position range limit). |
| 12~13 | Operation mode specific | -- | Operation mode specific |
| 14 | Manufacturer-specific | -- | See <Statusword structure> above. |
| 15 | Reserved | -- | -- |

### 4.3.3 Object 0x605A: Quick stop option code

This object indicates what action should be taken when the quick stop function is activated.

The slow down ramp is the deceleration value of the mode being used in operation.

- Object description

| | |
|---|---|
| Index | 605A |
| Name | Quick stop option code |
| Object code | Variable |
| Data type | Unsigned16 |

- Entry description

| | |
|---|---|
| Sub-index | 0x00 |
| Access | RW |
| PDO mapping | No |
| Value range | -1 ~ 4 |
| Default value | 2 |
| Units | No |

- Value definition

| Value | Definition |
|---|---|
| -1 | Dynamic brake. (Brake motor by means of a controlled motor short circuit) |
| 0 | Disable drive function |
| 1 | Slow down on slow down ramp and transit into switch on disabled |
| 2 | Slow down on quick stop ramp and transit into switch on disabled |
| 3 | Slow down on current limit and transit into switch on disabled |
| 4 | Slow down on voltage limit and transit into switch on disabled |

## 4.3.4 Object 0x605B: Shutdown option code

This object indicates what action should be taken when there is a state transition of: Operation enabled è Ready to switch on.

- Object description

| | |
|---|---|
| Index | 605B |
| Name | Shutdown option code |
| Object code | Variable |
| Data type | INT16 |

- Entry description

| | |
|---|---|
| Sub-index | 0x00 |
| Access | RW |
| PDO mapping | No |
| Value range | -1 ~ 1 |
| Default value | 0 |
| Units | No |

- Value definition

| Value | Definition |
|---|---|
| -1 | Dynamic brake. (Brake motor by means of a controlled motor short circuit) |
| 0 | Disable drive function (switch-off the drive power stage) |
| 1 | Slow down on slow down ramp; disable of the drive function |

### 4.3.5 Object 0x605C: Disable operation option code

This object indicates what action should be taken when there is a state transition of: Operation enabled è Switched on.

- Object description

| | |
|---|---|
| Index | 605C |
| Name | Disable operation option code |
| Object code | Variable |
| Data type | INT16 |

- Entry description

| | |
|---|---|
| Sub-index | 0x00 |
| Access | RW |
| PDO mapping | No |
| Value range | -1 ~ 1 |
| Default value | 0 |
| Units | No |

- Value definition

| Value | Definition |
|---|---|
| -1 | Dynamic brake. (Brake motor by means of a controlled motor short circuit) |
| 0 | Disable drive function (switch-off the drive power stage) |
| 1 | Slow down on slow down ramp; disable of the drive function |

## 4.3.6 Object 0x605D: Halt option code

This object indicates what action should be taken when halt is executed
The slow down ramp is the deceleration value of the mode being used in operation.

- Object description

| | |
|---|---|
| Index | 605D |
| Name | Halt option code |
| Object code | Variable |
| Data type | INT16 |

- Entry description

| | |
|---|---|
| Sub-index | 0x00 |
| Access | RW |
| PDO mapping | No |
| Value range | -1 ~ 4 |
| Default value | 1 |
| Units | No |

- Value definition

| Value | Definition |
|---|---|
| -1 | Dynamic brake. (Brake motor by means of a controlled motor short circuit) |
| 0 | Disable drive function (Immediately power off) |
| 1 | Slow down on slow down ramp and stay in operation enabled |
| 2 | Slow down on quick stop ramp and stay in operation enabled |
| 3 | Slow down on current limit and stay in operation enabled |
| 4 | Slow down on voltage limit and stay in operation enabled |

### 4.3.7 Object 0x605E: Fault reaction option code

This object indicates what action should be taken when fault is detected in the PDS. The slow down ramp is the deceleration value of the mode being used in operation.

- Object description

| | |
|---|---|
| Index | 605E |
| Name | Fault reaction option code |
| Object code | Variable |
| Data type | INT16 |

- Entry description

| | |
|---|---|
| Sub-index | 0x00 |
| Access | RW |
| PDO mapping | No |
| Value range | -1 ~ 4 |
| Default value | 0 |
| Units | No |

- Value definition

| Value | Definition |
|---|---|
| -1 | Dynamic brake. (Brake motor by means of a controlled motor short circuit) |
| 0 | Disable drive function (Immediately power off) |
| 1 | Slow down on slow down ramp and stay in operation enabled |
| 2 | Slow down on quick stop ramp and stay in operation enabled |
| 3 | Slow down on current limit and stay in operation enabled |
| 4 | Slow down on voltage limit and stay in operation enabled |

# 5. Modes of Operation

## 5.1 Functional Description

The behavior of the PDS depends on the mode chosen for operation. Though the cpc PDS implements several modes of operation, the modes cannot be changed during Operation Enable state.

- On one hand, the control device writes to the object 0x6060 (modes of operation);

  on the other hand, the drive device provides object 0x6061 (modes of operation display) to indicate what operation mode is actually activated.
- Controlword, statusword, and set-point are mode-specific.
- The switching between modes of operation requires that no automatic reconfiguration of COBs (communication objects) for real-time data transmission is necessary.

  Namely, all necessary data objects (for the mode) that may be used during Operation Enabled state is configured before cyclic communication function is enabled.
- For cpc drives, it is possible to switch modes in any FSA state except for the Operation Enabled state.

The following modes of operation are implemented in cpc's servo drive:

| | |
|---|---|
| Profile position mode | (pp) |
| Homing mode | (hm) |
| Profile velocity mode | (pv) |
| Profile torque mode | (tq) |
| Cyclic synchronous position mode | (csp) |
| Cyclic synchronous velocity mode | (csv) |
| Cyclic synchronous torque mode | (cst) |

# 5.2 Objects

## Object 0x6060: Modes of operation

This object indicates the required operation mode.

- Object description

| | |
|---|---|
| Index | 6060 |
| Name | Modes of operation |
| Object code | Variable |
| Data type | INT8 |

- Entry description

| | |
|---|---|
| Sub-index | 0x00 |
| Access | RW |
| PDO mapping | Yes |
| Value range | -6 ~ 10 |
| Default value | 0 |
| Units | No |

- Value definition

| Value | Definition |
|---|---|
| -6 | for cpc internal use only |
| -5 | for cpc internal use only |
| -4 | Direct position |
| -3 | Direct velocity |
| -2 | Direct torque |
| -1 | Direct voltage |
| 0 | Drive is disabled |
| 1 | Profile position mode |
| 2 | Reserved |
| 3 | Profile velocity mode |
| 4 | Profile torque mode |
| 5 | Reserved |
| 6 | Homing mode |
| 7 | Reserved |
| 8 | Cyclic synchronous position mode |
| 9 | Cyclic synchronous velocity mode |
| 10 | Cyclic synchronous torque mode |

# Object 0x6061: Modes of operation display

This object provides the actual operation mode.

- Object description

| | |
|---|---|
| Index | 6061 |
| Name | Modes of operation display |
| Object code | Variable |
| Data type | INT8 |

- Entry description

| | |
|---|---|
| Sub-index | 0x00 |
| Access | RO |
| PDO mapping | Yes |
| Value range | INT8 |
| Default value | 0 |
| Units | No |

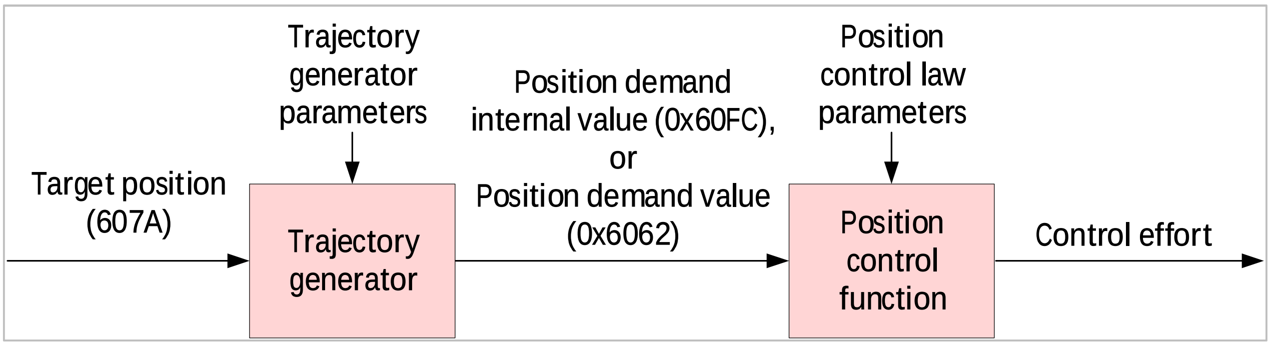
# 6. Profile Position Mode

## 6.1 General information

This chapter describes about how to configure a point-to-point move in a profiled motion.

The driver receives a target position value, then the trajectory generator converts this target position value into a position demand value to the position control loop.

The overall structure of this mode is shown as the diagram below.


Trajectory generator parameters
Position demand internal value (0x60FC), or Position demand value (0x6062)
Position control law parameters
Target position (607A)
Trajectory generator
Position control function
Control effort

At the input to the trajectory generator, parameters are limited before being normalized to the internal units.

The trajectory input includes:

- Position range limit
- Software position limit
- Profile velocity
- End velocity
- Max profile velocity
- Max motor speed
- Profile acceleration
- Profile deceleration
- Quick-stop deceleration
- Quick-stop option code

# 6.2 Structure of Controlword and Statusword

## 6.2.1 Controlword of the Profile Position Mode.

| Bit | Function |
|---|---|
| 0~3 | *See chapter 4.3.1.1 - Device Control/Controlword structure |
| 4 | New set-point |
| 5 | Change set immediately |
| 6 | Abs/rel |
| 7 | *See chapter 4.3.1.1 - Device Control/Controlword structure |
| 8 | Halt |
| 9 | Reserved |
| 10~15 | *See chapter 4.3.1.1 - Device Control/Controlword structure |

- Definition of bit 4, 5, 6, and 8

| Bit | Name | Value | Description |
|---|---|---|---|
| 4 | New set-point | 0 | There is No target position given |
| | | 1 | There is target point given. |
| 5 | Change set immediately | 0 | Fully complete the present positioning (target reached) before the next set-point gets started.<br>See ch. 6.3.2 – set of set-point. |
| | | 1 | Interrupt the present positioning and start the next set-point immediately.<br>See ch. 6.3.2 – single set-point. |
| 6 | Absolute / Relative | 0 | The target position is an absolute value. |
| | | 1 | The target position is a relative value. |
| 8 | Halt | 0 | Perform or continue positioning. |
| | | 1 | Motor is stopped according to halt option code (0x605D). |

## 6.2.2 Statusword of the Profile Position Mode

| Bit | Function |
|---|---|
| 0~9 | *See chapter 4.3.2.1 - Device Control/Statusword structure |
| 10 | Target reached |
| 11 | *See chapter 4.3.2.1 - Device Control/Statusword structure |
| 12 | Set-point acknowledge |
| 13 | Following error |
| 14~15 | *See chapter 4.3.2.1 - Device Control/Statusword structure |

- Definition of bit 10, 12 and 13

| Bit | Name | Value | Description |
|---|---|---|---|
| 10 | Target reached | 0 | Halt = 0: Target position not reached.<br>Halt = 1: motor decelerates. |
| | | 1 | Halt = 0: Target position reached<br>Halt = 1: motor decelerates to 0 velocity. |
| 12 | Set-point acknowledge | 0 | Previous set-point has been processed, ready to accept new set-point. |
| | | 1 | Detects that “new set-point bit” in controlword is 1; or,<br>set-point buffer is busy. |
| 13 | Following error | 0 | No following error. |
| | | 1 | Following error occurs. |

# 6.3 Functional Descriptions

## 6.3.1 General

The setting of set-point is determined by these three together:

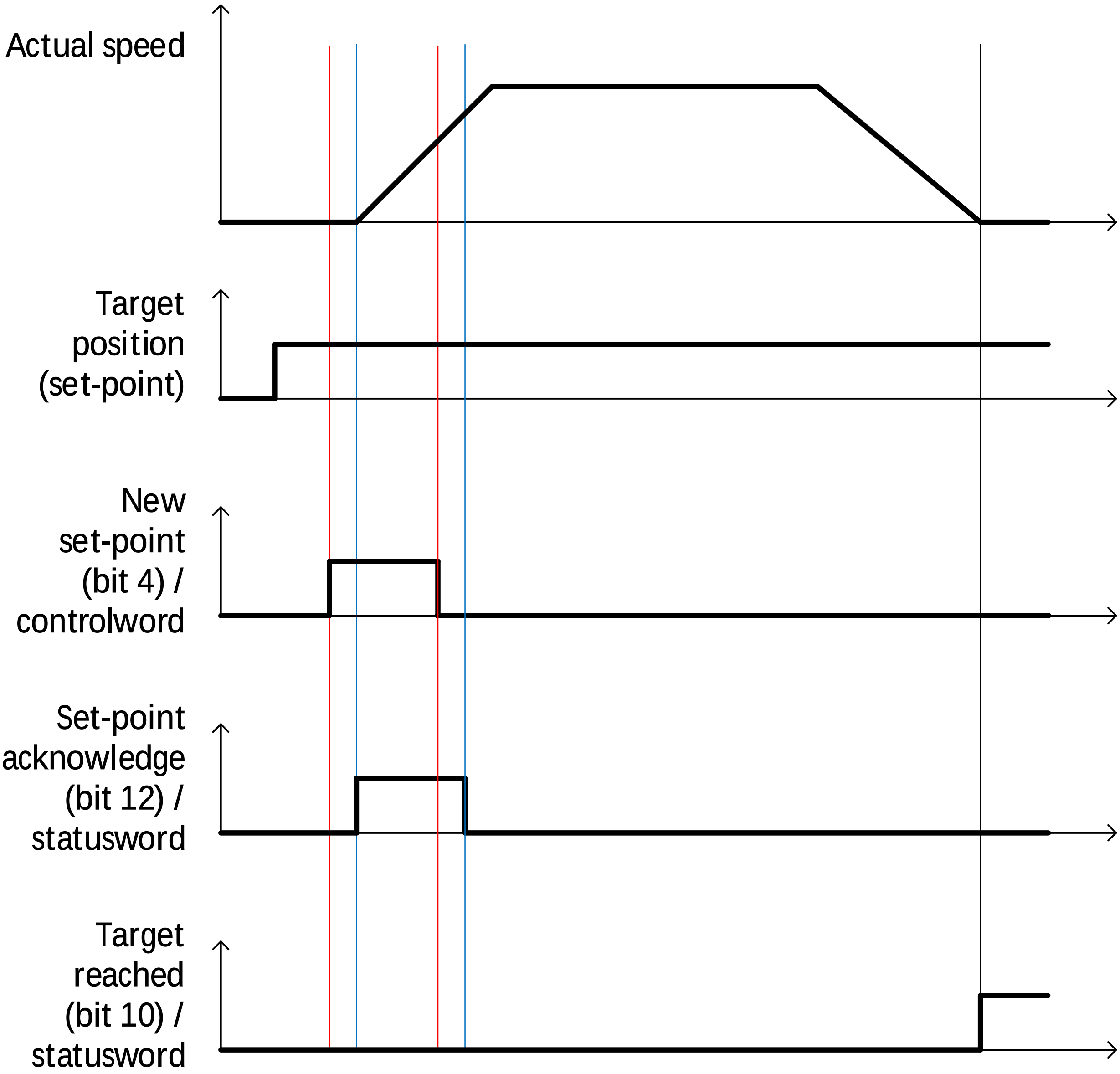
- The timing of the new set-point bit (bit 4 in controlword),
- The change set immediately bit (bit 5 in controlword), and
- The set-point acknowledgement bit (bit 12 in statusword).

The setting procedure is as follows:

1. A set-point is applied to the driver.
2. The control device, such as a computer, signals with a rising edge of the new set-point bit (bit 4 in controlword) that the set-point data is completely transmitted.
3. The driver sets the set-point acknowledgement bit (bit 12 in statusword) to 1.
4. The control device makes the new set-point bit to 0.
5. The driver signals that the set-point acknowledgement bit is set to 0, indicating its ability to accept new set-points.

Please refer to the diagram on the next page.

# Procedure of setting Set-point


Actual speed
Target position (set-point)
New set-point (bit 4) / controlword
Set-point acknowledge (bit 12) / statusword
Target reached (bit 10) / statusword

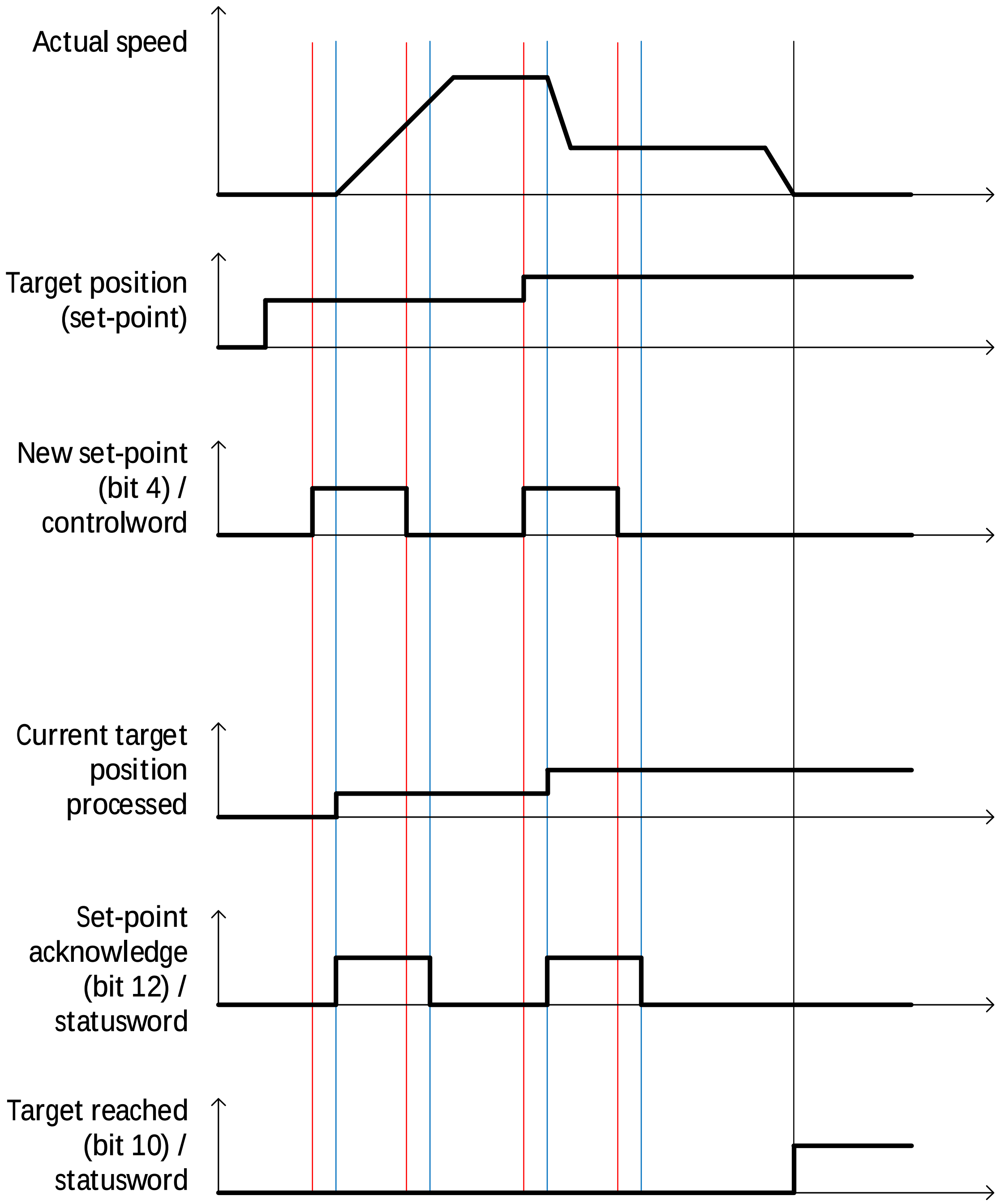
## 6.3.2 Single Set-point & Set of Set-point

There are 2 ways of applying target positions to a driver:

- Single set-point (the “change set immediately” bit of controlword = 1)

  When a set-point is in progress and a new set-point is validated via the new set-point (bit 4) in the controlword, the new set-point will be processed immediately.

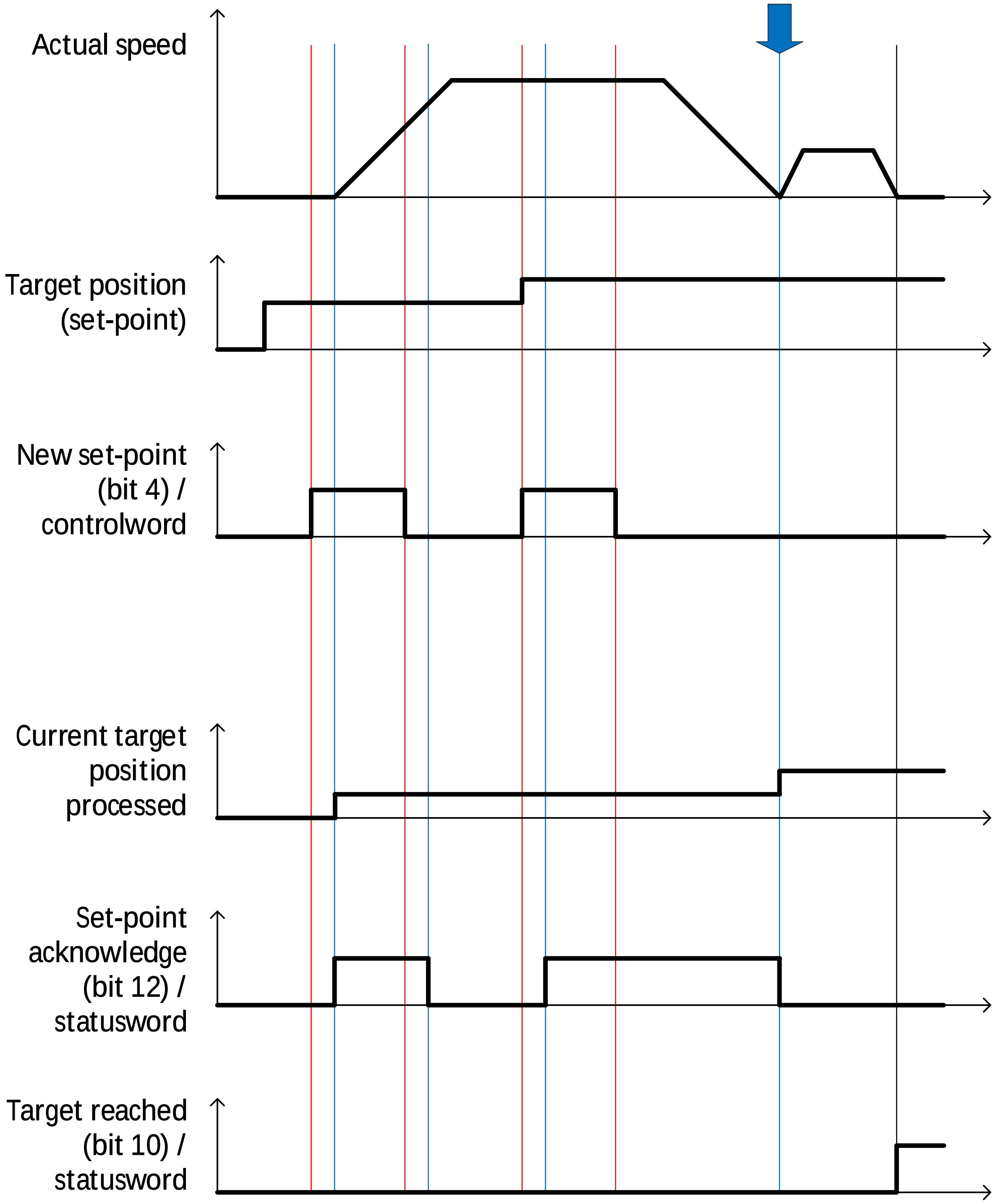
Single Set-point (change set immediately bit = 1)


Actual speed
Target position (set-point)
New set-point (bit 4) / controlword
Current target position processed
Set-point acknowledge (bit 12) / statusword
Target reached (bit 10) / statusword

- Set of set-points (the “change set immediately” bit of controlword = 0)

  When a set-point is in progress and a new set-point is validated via the new set-point (bit 4) in the controlword, the new set-point will be processed only after the previous one has been reached.

Set of Set-point (change set immediately bit = 0)


Actual speed
Target position (set-point)
New set-point (bit 4) / controlword
Current target position processed
Set-point acknowledge (bit 12) / statusword
Target reached (bit 10) / statusword

### 6.3.3 Buffered Set-point

When “set of set-point” is used (i.e., change set immediately bit = 0), the cpc driver supports two set-points:

- One is presently processed, and
- The other is buffered.

<u>Note</u>:

- The cpc driver can buffer one set-point.
- If there are two or more set-points to be buffered, the first will be stored while the second and the later ones will be ignored.

Please refer to the diagram below.

Buffered Set-point

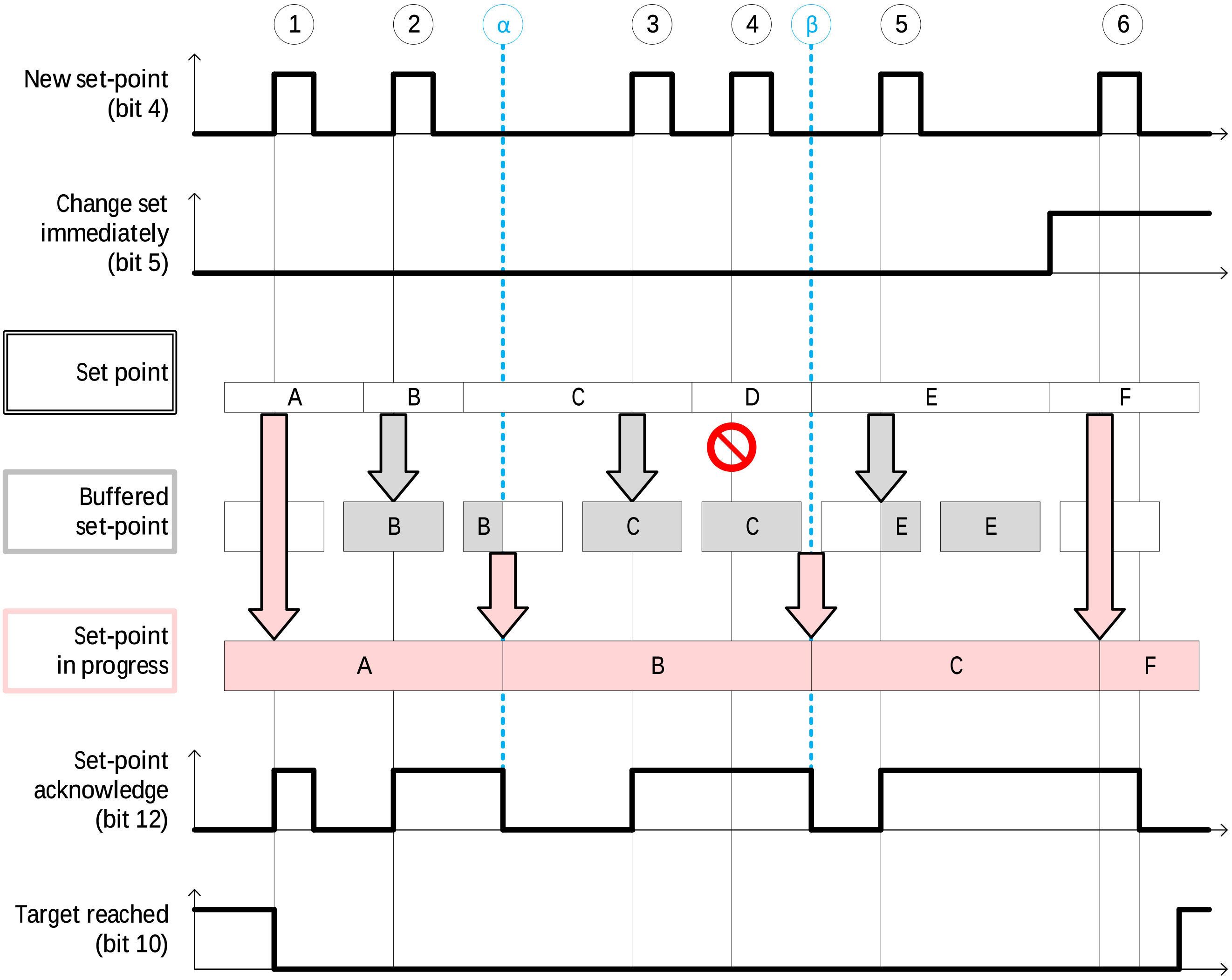


Diagram explanation:

| Item | Description |
|---|---|
| ① | If no set-point is in progress, the new set-point will become active immediately. |
| ② | If a set-point is in progress, the new set-point will be stored in the buffer that is free. |
| α | Set-point (A) has finished,<br>l The “set-point acknowledge (bit 12)” signals this status with falling edge which enables the buffered new set-point (B) to be active immediately.<br>l The buffer space is hence cleared. |
| ③ | Same as ②. |
| ④ | If the buffer is busy (set-point acknowledge bit = 1),<br>The new set-point (D) will be ignored and won’t be stored. |
| β | Same as α.<br>Set-point (B) has finished; the “set-point acknowledge bit” signals with falling edge, and the stored set-point (C) becomes active immediately.<br>The buffer hence becomes free. |
| ⑤ | Same as ②.<br>Note that the set-point which advances to be stored is the new set-point (E) instead of the previously ignored set-point (D). |
| ⑥ | The “change set immediately bit” is set to 1,<br>l The new set-point following after this event will be processed immediately as single set-point.<br>l All previously loaded set-points (the buffered and the one in progress) will be abandoned. |

# 6.4 Objects

## Object 0x607A: Target position

The target position is the position to which the drive should move in position profile mode, using the present settings of motion control parameters such as velocity, acceleration, deceleration and motion profile type. The target position is given in counts. The target position can be absolute or relative, depending on the Abs/Rel flag—the bit 6—in the controlword.

- Object description

| | |
|---|---|
| Index | 607A |
| Name | Target position |
| Object code | Variable |
| Data type | Integer32 |

- Entry description

| | |
|---|---|
| Sub-index | 0x00 |
| Access | RW |
| PDO mapping | Yes |
| Value range | Integer32 |
| Default value | 0 |
| Units | Count |

## Object 0x607B: Position range limit

This object contains two sub-parameters—namely, the minimum position range limit and the maximum position range limit—that limit the numerical range of the input value.

Upon reaching or exceeding these limits, the input value automatically wrap to the other end of the range.

To disable the position range limits, please set the two sub-parameters to 0.

- Object description

| | |
|---|---|
| Index | 607B |
| Name | Position range limit |
| Object code | Array |
| Data type | Integer32 |

See entry description on next page.

- Entry description

| | |
|---|---|
| Sub-index | 0x00 |
| Description | Highest sub-index supported |
| Access | C |
| PDO mapping | No |
| Value range | 2 |
| Default value | 2 |
| Units | No |

| | |
|---|---|
| Sub-index | 0x01 |
| Description | Min position range limit |
| Access | RW |
| PDO mapping | No |
| Value range | Integer32 |
| Default value | $-2^{31}$ |
| Units | Count |

| | |
|---|---|
| Sub-index | 0x02 |
| Description | Max position range limit |
| Access | RW |
| PDO mapping | No |
| Value range | Integer32 |
| Default value | $2^{31}-1$ |
| Units | Count |

# Object 0x607D: Software position limit

This object contains the 2 sub-indexes (min position limit and max position limit) which specify the actual position limits for both the position demand value and the position actual value.

Note:

Homing is required to validate these software position limits.

- Object description

| | |
|---|---|
| Index | 607D |
| Name | Software position limit |
| Object code | Array |
| Data type | Integer32 |

- Entry description

| | |
|---|---|
| Sub-index | 0x00 |
| Description | Highest sub-index supported |
| Access | C |
| PDO mapping | No |
| Value range | 2 |
| Default value | 2 |
| Units | No |

| | |
|---|---|
| Sub-index | 0x01 |
| Description | Min position limit |
| Access | RW |
| PDO mapping | No |
| Value range | Integer32 |
| Default value | $-2^{31}$ |
| Units | Count |

| | |
|---|---|
| Sub-index | 0x02 |
| Description | Max position limit |
| Access | RW |
| PDO mapping | No |
| Value range | Integer32 |
| Default value | $2^{31}$ -1 |
| Units | Count |

## Object 0x607F: Max profile velocity

The object is the configured maximum allowed velocity in either direction during a profiled move.

Note:

- For compatibility reason, the value of 0x607F (Max profile velocity) is internally equal to that of object 0x6080 (Max motor speed).
- Any change on object 0x607F will be applied to object 0x6080.

- Object description

| | |
|---|---|
| Index | 607F |
| Name | Max profile velocity |
| Object code | Variable |
| Data type | UINT32 |

- Entry description

| | |
|---|---|
| Sub-index | 0x00 |
| Access | RW |
| PDO mapping | Yes |
| Value range | UINT32 |
| Default value | 1,500,000 |
| Units | count/s |

## Object 0x6080: Max motor speed

This is the configured maximum allowed speed for the motor in either direction, it is for protection reason and is taken from the motor name-plate.

Note:

- For compatibility reason, the value of 0x607F (Max profile velocity) is internally equal to that of object 0x6080 (Max motor speed).
- Any change on object 0x607F will be applied to object 0x6080.

- Object description

| | |
|---|---|
| Index | 6080 |
| Name | Max motor speed |
| Object code | Variable |
| Data type | UINT32 |

- Entry description

| | |
|---|---|
| Sub-index | 0x00 |
| Access | RW |
| PDO mapping | Yes |
| Value range | 1 ~ 7FFFFFFF |
| Default value | 1,500,000 |
| Units | count/s |

# Object 0x6081: Profile velocity

The “Profile velocity” is the velocity normally attained at the end of the acceleration ramp during a profiled move and is valid for both directions of motion. The value is given in counts.

- Object description

| | |
|---|---|
| Index | 6081 |
| Name | Profile velocity |
| Object code | Variable |
| Data type | UINT32 |

- Entry description

| | |
|---|---|
| Sub-index | 0x00 |
| Access | RW |
| PDO mapping | Yes |
| Value range | 1 ~ 7FFFFFFF |
| Default value | 2,000,000 |
| Units | count/s |

## Object 0x6083: Profile acceleration

This object indicates the configured acceleration. The value is given in count/$s^2$.

- Object description

| | |
|---|---|
| Index | 6083 |
| Name | Profile acceleration |
| Object code | Variable |
| Data type | UINT32 |

- Entry description

| | |
|---|---|
| Sub-index | 0x00 |
| Access | RW |
| PDO mapping | No |
| Value range | 1 ~ 7FFFFFFF |
| Default value | 1,000,000 |
| Units | count/ $s^2$ |

## Object 0x6084: Profile deceleration

This object indicates the configured deceleration. The value is given in the same units as the profile acceleration object (0x6083).

- Object description

| | |
|---|---|
| Index | 6084 |
| Name | Profile deceleration |
| Object code | Variable |
| Data type | UINT32 |

- Entry description

| | |
|---|---|
| Sub-index | 0x00 |
| Access | RW |
| PDO mapping | No |
| Value range | 1 ~ 7FFFFFFF |
| Default value | 1,000,000 |
| Units | count/ $s^2$ |

# Object 0x6085: Quick stop deceleration

This object indicates the configured deceleration used to stop the motor when the quick stop function is triggered and the quick stop code object (605A) is set to 2. The quick stop deceleration is also used if the fault reaction code object (605E) is 2 and the halt option code object (605D) is 2.
The value is given in the same units as profile acceleration (6083).

- Object description

| | |
|---|---|
| Index | 6085 |
| Name | Quick stop deceleration |
| Object code | Variable |
| Data type | UINT32 |

- Entry description

| | |
|---|---|
| Sub-index | 0x00 |
| Access | RW |
| PDO mapping | Yes |
| Value range | 1 ~ 7FFFFFFF |
| Default value | 1,000,000 |
| Units | count/ $s^2$ |

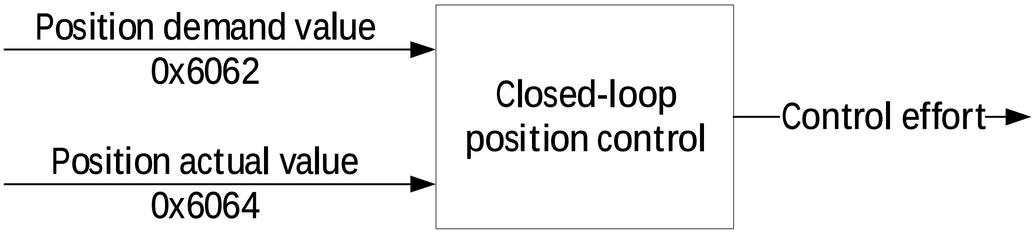
# 7. Position Control Function

## 7.1 General information

This chapter describes all the parameters required for closed-loop position control.

The control mainly relies on these 2 inputs:

- The position demand value (0x6062), and
- The position actual value (0x6064), e.g., encoder.



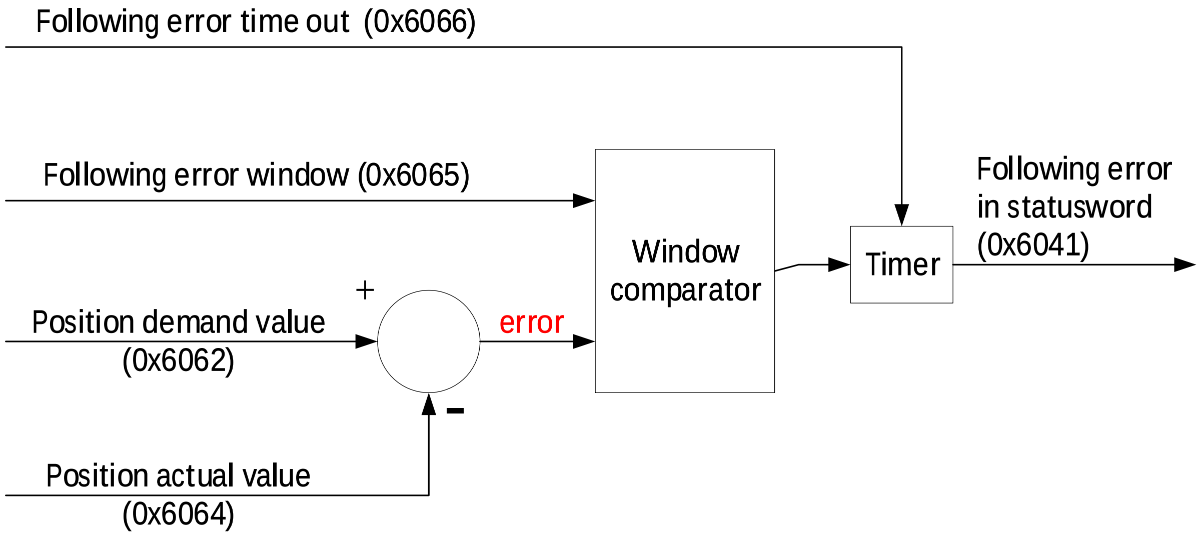
To make sure that the physical limits of a driver is not exceeded, an absolute function is implemented – the current limit function and the velocity limit function – for the position control effort.

The following terms are used in this chapter

- Following error

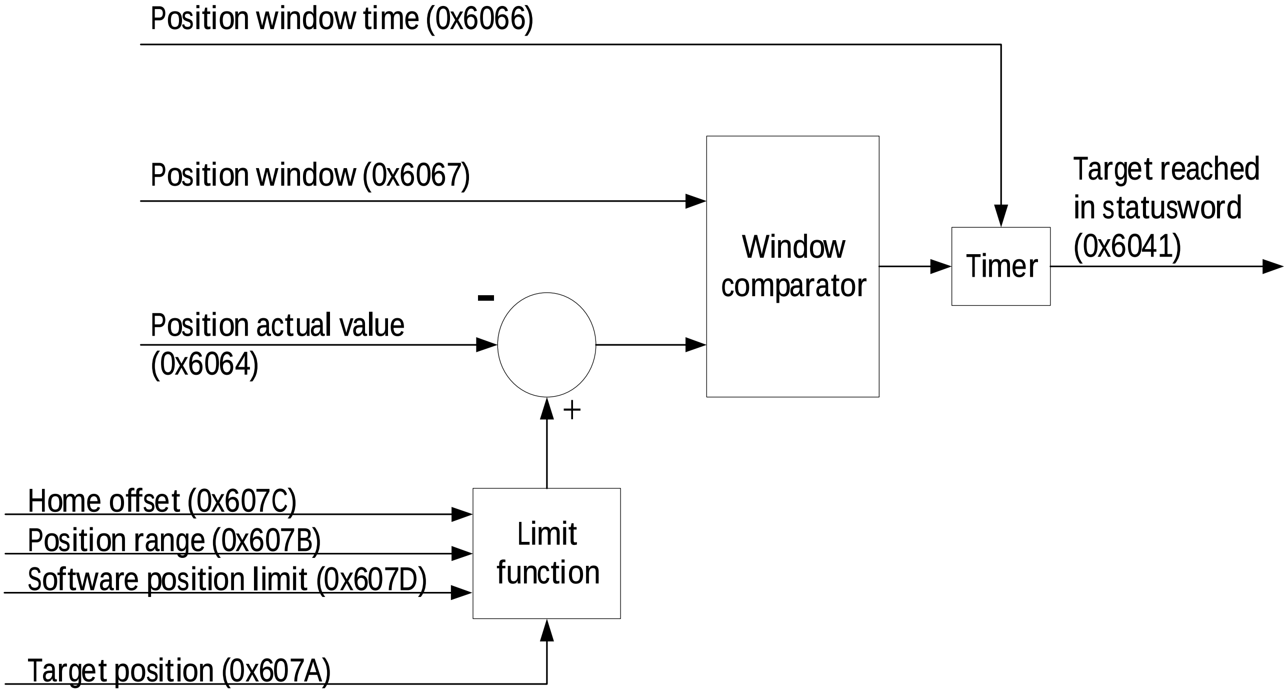
  When the position actual value is outside the following error window, which is symmetrically aligned to the position demand value, for over the configured time duration (i.e., following error timeout), the following error bit 13 in statusword is set. See the chart below.

  (see next page)


Following error time out (0x6066)
Following error window (0x6065)
Position demand value (0x6062)
+
Position actual value (0x6064)
-
error
Window comparator
Timer
Following error in statusword (0x6041)

- Target reached

  When the position actual value is within the position window, which is symmetrically aligned to the position demand value, over a period of configured time duration (i.e., position window time), the target reached bit 10 in statusword is set. See the chart below.



# 7.2 Objects

## Object 0x6062: Position demand value

This object gives the demand position value. It is given in count.

- Object description

| | |
|---|---|
| Index | 6062 |
| Name | Position demand value |
| Object code | Variable |
| Data type | INT32 |

- Entry description

| | |
|---|---|
| Sub-index | 0x00 |
| Access | RO |
| PDO mapping | No |
| Value range | INT32 |
| Default value | No |
| Units | count |

# Object 0x6063: Position actual internal value

This object is for internal algorithm. It gives the actual value of the position measurement device, and is one of the two inputs of the closed-loop position control. The value is given in counts.

- Object description

| | |
|---|---|
| Index | 6063 |
| Name | Position actual internal value |
| Object code | Variable |
| Data type | INT32 |

- Entry description

| | |
|---|---|
| Sub-index | 0x00 |
| Access | RO |
| PDO mapping | No |
| Value range | INT32 |
| Default value | No |
| Units | count |

## Object 0x6064: Position actual value

The position actual value object indicates the actual value of the position measurement device (for example, an encoder). The value is given in counts.

- Object description

| | |
|---|---|
| Index | 6063 |
| Name | Position actual value |
| Object code | Variable |
| Data type | INT32 |

- Entry description

| | |
|---|---|
| Sub-index | 0x00 |
| Access | RO |
| PDO mapping | No |
| Value range | INT32 |
| Default value | No |
| Units | count |

## Object 0x6065: Following error window

This object is a configured range of tolerated position values symmetrically to the position demand value. If the actual position value falls out of the following error window, a "following error" occurs. For instance, a following error may occur

- When the driver is blocked
- When the profile velocity is unreachable, and
- If the closed-loop coefficient, i.e., gain, is wrong.

If the value of this object is set to FFFF FFFF, this following control function will be switched off.

- Object description

| | |
|---|---|
| Index | 6065 |
| Name | Following error window |
| Object code | Variable |
| Data type | UINT32 |

- Entry description

| | |
|---|---|
| Sub-index | 0x00 |
| Access | RW |
| PDO mapping | Yes |
| Value range | UINT32 |
| Default value | 0 |
| Units | Count |

## Object 0x6066: Following error time out

This object is a configured time duration for a following error situation.

If a following error occurs longer than this configured time duration (given in ms),

- A fault event will rise, and
- The bit 13 of the status word (i.e., following error) will set to 1.

- Object description

| | |
|---|---|
| Index | 6066 |
| Name | Following error timeout |
| Object code | Variable |
| Data type | UINT16 |

- Entry description

| | |
|---|---|
| Sub-index | 0x00 |
| Access | RW |
| PDO mapping | Yes |
| Value range | UINT16 |
| Default value | 0 |
| Units | Count |

## Object 0x6067: Position window

The position window object defines a symmetrical range of allowed positions relative to the target position. If the actual position of the encoder is within this window, the target position is regarded as reached.

If the value of this object is set to FFFF FFFF, this position window control function will be switched off.

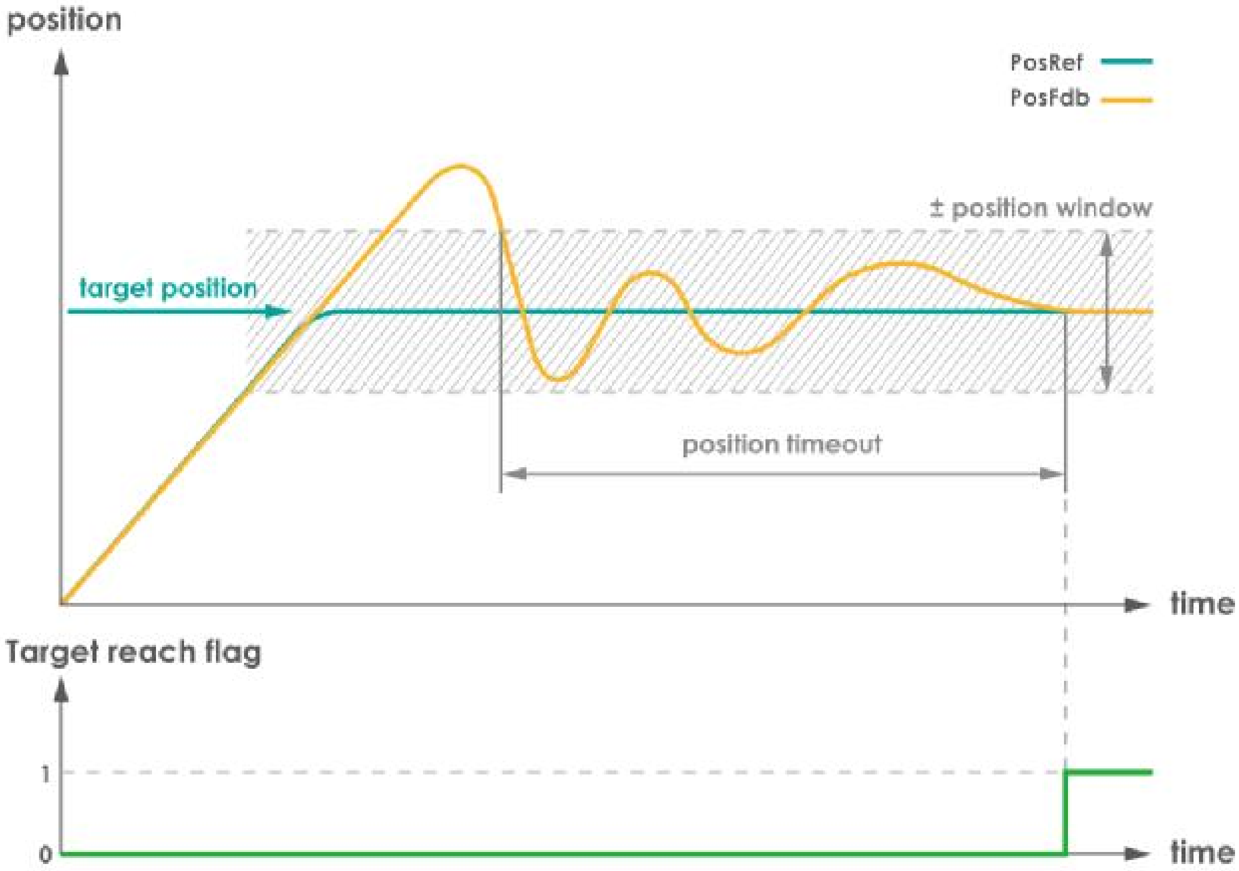
- Object description

| | |
|---|---|
| Index | 6067 |
| Name | Position window |
| Object code | Variable |
| Data type | UINT32 |

- Entry description

| | |
|---|---|
| Sub-index | 0x00 |
| Access | RW |
| PDO mapping | Yes |
| Value range | UINT32 |
| Default value | 0 |
| Units | Count |

See diagram on the next page.


position
PosRef
PosFdb
± position window
target position
position timeout
time
Target reach flag
1
0
time

## Object 0x6068: Position window time

When the actual position is within the position window (0x6067) for the configured position window time — given in ms — the bit 10 (target reach) in status word will set to 1.

- Object description

| | |
|---|---|
| Index | 6068 |
| Name | Position window time |
| Object code | Variable |
| Data type | UINT32 |

- Entry description

| | |
|---|---|
| Sub-index | 0x00 |
| Access | RW |
| PDO mapping | No |
| Value range | UINT32 |
| Default value | 0 |
| Units | ms |

# Object 0x60C2: Interpolation time period

This object will be used in cyclic synchronous position mode; it indicates the configured interpolation cycle time, and has 2 sub-indexes.

The interpolation time period (sub-index 0x01) value is given in $10^{(\text{interpolation time index})}$ second. The interpolation time index is dimensionless.

- Object description

| | |
|---|---|
| Index | 60C2 |
| Name | Interpolation time period |
| Object code | Record |
| Data type | Interpolation time period record (0x0080) |

- Entry description

| | |
|---|---|
| Sub-index | 0x00 |
| Description | Highest sub-index supported |
| Access | C |
| PDO mapping | No |
| Value range | 2 |
| Default value | 2 |
| Units | No |

| | |
|---|---|
| Sub-index | 0x01 |
| Description | Interpolation time period value |
| Access | RW |
| PDO mapping | No |
| Value range | 1 to 255 |
| Default value | 1 |
| Units | No |

| | |
|---|---|
| Sub-index | 0x02 |
| Description | Interpolation time index |
| Access | RW |
| PDO mapping | No |
| Value range | -3 to 1 |
| Default value | -3 |
| Units | Count |

## Object 0x60F4: Following error actual value

This object provides the actual value, given in counts, of the following error.

- Object description

| | |
|---|---|
| Index | 60F4 |
| Name | Following error actual value |
| Object code | Variable |
| Data type | INT32 |

- Entry description

| | |
|---|---|
| Sub-index | 0x00 |
| Access | RO |
| PDO mapping | Yes |
| Value range | INT32 |
| Default value | No |
| Units | Count |

# Object 0x60FC: Position demand internal value

This object provides the output of the trajectory generator in profile position mode. The value is given in counts.

- Object description

| | |
|---|---|
| Index | 60FC |
| Name | Position demand internal value |
| Object code | Variable |
| Data type | INT32 |

- Entry description

| | |
|---|---|
| Sub-index | 0x00 |
| Access | RO |
| PDO mapping | Yes |
| Value range | INT32 |
| Default value | No |
| Units | Count |

# 8. Homing Mode

## 8.1 General Information

To operate the drivers, an exact knowledge of absolute position is usually needed. Due to cost reasons, drivers often don't have an absolute encoder, a homing operation is necessary.

This chapter describes the methods by which drivers seek home position.

There are various methods achieving this by using

1. Limit switches at the end of travel, or
2. Home switch in mid-travel, or
3. The mostly used Index (zero) pulse train from an incremental encoder.

Input Data

The inputs to homing method are:

- Controlword
- Homing method
- Homing speed

  Users can define two kinds of speed. Usually, the faster one to find the home switch, and the slower one to find the index pulse.
- Homing acceleration
- Home offset

  The value of this object is used as the new position value of the home reference point (e.g., home switch, home index, mechanical hard stop), or mechanical home position.

In addition, the cpc specific:

- Hard stop current%, and
- Hard stop period

Output Data

There is no output data except for those bits in the statusword that return the status or result of the homing process and the demand to the position control loops.

# 8.2 Structure of Controlword and Statusword

## 8.2.1 Controlword of Homing Mode

| Bit | Function |
|---|---|
| 0~3 | *See chapter 4.3.1.1 - Device Control/Controlword structure |
| 4 | Homing operation start |
| 5~6 | Reserved |
| 7 | *See chapter 4.3.1.1 - Device Control/Controlword structure |
| 8 | Halt |
| 9~15 | *See chapter 4.3.1.1 - Device Control/Controlword structure |

- Definition of bit 4 and 8

| Bit | Name | Value | Description |
|---|---|---|---|
| 4 | Homing operation start | 0 | Homing mode inactive. |
| | | 1 | Start/continue homing procedure |
| 8 | Halt | 0 | Activate bit 4 |
| | | 1 | Stop the motor according to halt option code (0x605D). |

## 8.2.2 Statusword of Homing Mode

| Bit | Function |
|---|---|
| 0~9 | *See chapter 4.3.2.1 - Device Control/Statusword structure |
| 10 | Target reached |
| 11 | *See chapter 4.3.2.1 - Device Control/Statusword structure |
| 12 | Homing attained |
| 13 | Homing error |
| 14~15 | *See chapter 4.3.2.1 - Device Control/Statusword structure |

- Definition of bit 10, 12 and 13

| Bit 13 (homing error) | Bit 12 (Homing attained) | Bit 10 (Target reached) | Definition |
|---|---|---|---|
| 0 | 0 | 0 | Homing procedure is being carried out. |
| 0 | 0 | 1 | Velocity = 0, homing is not completed or not started. |
| 0 | 1 | 0 | Homing is completed, velocity is not 0. |
| 0 | 1 | 1 | Homing is completed, velocity is 0. |
| 1 | 0 | 0 | Homing error occurred, velocity is not 0. |
| 1 | 0 | 1 | Homing error occurred, velocity is 0. |
| 1 | 1 | X | Reserved. |

# 8.3 Object

## Object 0x607C: Home offset

This object defines the configured difference between

- The (new) zero position for application (which is finalized after homing is completed) AND
- The machine home position found during homing.

The default of home offset is 0.

During homing, the driver will seek and home on the home reference point (e.g., home switch, home index, or mechanical hard stop). When home offset object is applied, its value will be set as the position value of the machine home position. Please refer to the diagram on the next page.

- Object description

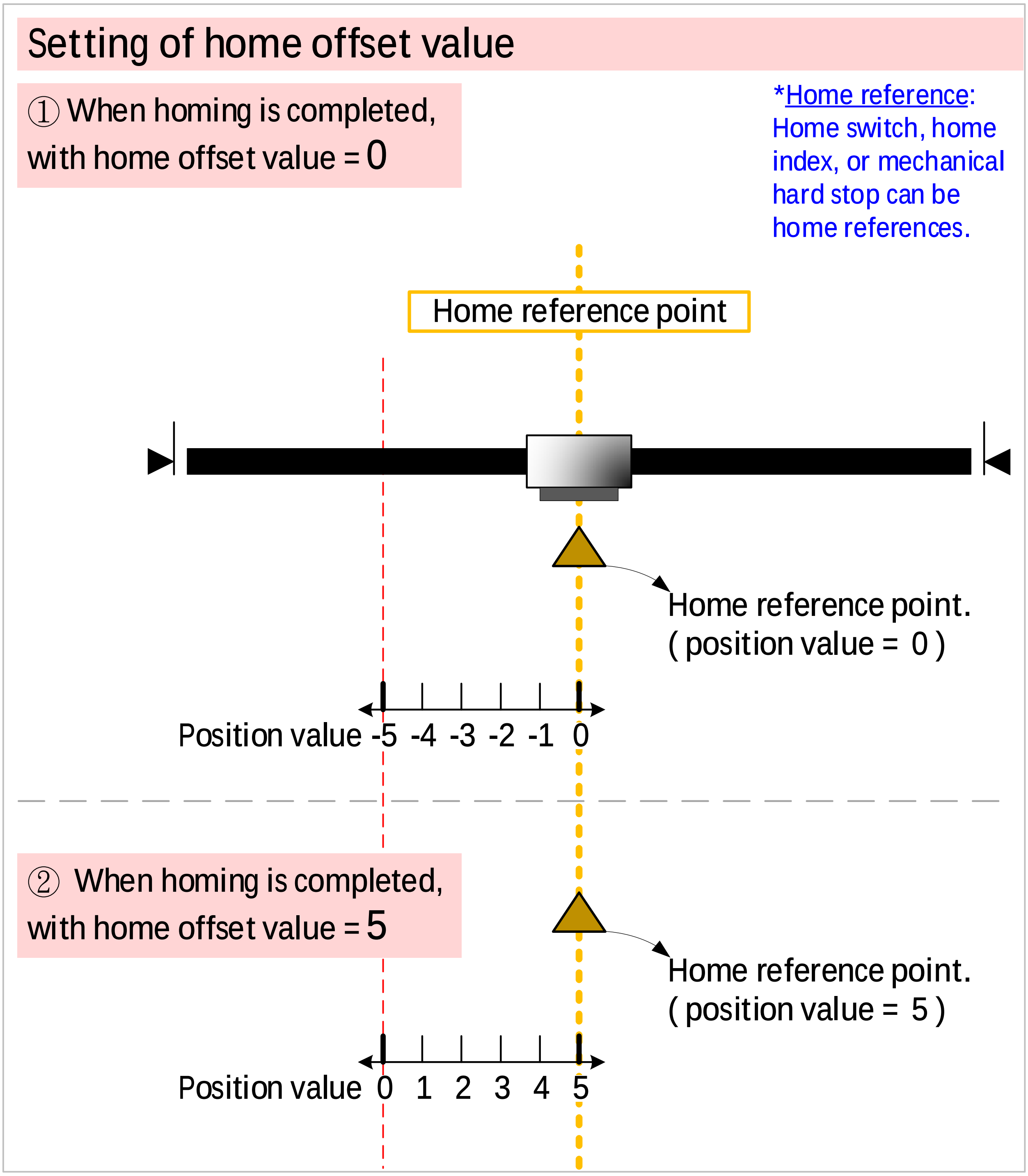
| Index | 607C |
|---|---|
| Name | Home offset |
| Object code | Variable |
| Data type | INT32 |

- Entry description

| Sub-index | 0x00 |
|---|---|
| Access | RW |
| PDO mapping | Yes |
| Value range | INT32 * |
| Default value | 0 |
| Units | count |

* When home offset object is applied, the new coordinate system (which is hence generated) needs to be within the modulo range.

< Result of setting “home offset value” >



## Setting of home offset value

① When homing is completed, with home offset value = 0

*Home reference: Home switch, home index, or mechanical hard stop can be home references.

② When homing is completed, with home offset value = 5

# Object 0x6098: Homing method

- Object description

| | |
|---|---|
| Index | 0x6098 |
| Name | Homing method |
| Object code | Variable |
| Data type | Integer8 |

- Entry description

| | |
|---|---|
| Sub-index | 0x00 |
| Access | RW |
| PDO mapping | Yes |
| Value range | -12 ~ 37 |
| Default value | 35 |
| Units | No |

- Data description

| | |
|---|---|
| -12 ~ -1 | cpc homing methods. See ch. 8.6 for details. |
| 0 | Do nothing |
| 1 ~ 37 | Method 1 ~ 37. See ch. 8.5 for details. |

# Object 0x60E3: Supported homing methods

It indicates the supported homing methods of the driver.

- Object description

| | |
|---|---|
| Index | 60E3 |
| Name | Supported homing methods |
| Object code | Array |
| Data type | Integer8 |

- Entry description

| | |
|---|---|
| Sub-index | 0x00 |
| Description | Highest sub-index supported |
| Access | C |
| PDO mapping | No |
| Value range | Manufacturer-specific -12 to -1, and all 402 standard methods. |
| Default value | 44 |
| Units | No |

The default value of other sub-indexes from 0x01 (1st supported homing method), 0x02 (2nd supported homing method) … till 0x44 (44th supported homing method) is "manufacturer-specific -12 to -1, and all 402 standard methods."

See the chart below:

| Sub-index # | Homing method # | Sub-index # | Homing method # | Sub-index # | Homing method # |
|---|---|---|---|---|---|
| 0x01 | -12 | 0x16 | 4 | 0x31 | 21 |
| 0x02 | -11 | 0x17 | 5 | 0x32 | 22 |
| 0x03 | -10 | 0x18 | 6 | 0x33 | 23 |
| 0x04 | -9 | 0x19 | 7 | 0x34 | 24 |
| 0x05 | -8 | 0x20 | 8 | 0x35 | 25 |
| 0x06 | -7 | 0x21 | 9 | 0x36 | 26 |
| 0x07 | -6 | 0x22 | 10 | 0x37 | 27 |
| 0x08 | -5 | 0x23 | 11 | 0x38 | 28 |
| 0x09 | -4 | 0x24 | 12 | 0x39 | 29 |
| 0x10 | -3 | 0x25 | 13 | 0x40 | 30 |
| 0x11 | -2 | 0x26 | 14 | 0x41 | 33 |
| 0x12 | -1 | 0x27 | 17 | 0x42 | 34 |
| 0x13 | 1 | 0x28 | 18 | 0x43 | 35 |
| 0x14 | 2 | 0x29 | 19 | 0x44 | 37 |
| 0x15 | 3 | 0x30 | 20 | -- | |

# Object 0x6099: Homing speeds

This object defines the configured speed used during homing, for searching the switch or encoder Index position.  The unit is count/s.

- Object description

| | |
|---|---|
| Index | 6099 |
| Name | Homing speeds |
| Object code | Array |
| Data type | UINT32 |

- Entry description

| | |
|---|---|
| Sub-index | 0x00 |
| Description | Highest sub-index supported |
| Access | C |
| PDO mapping | No |
| Value range | 2 |
| Default value | 2 |
| Units | No |

| | |
|---|---|
| Sub-index | 0x01 |
| Description | Speed during search for switch |
| Access | RW |
| PDO mapping | No |
| Value range | UINT32 |
| Default value | 20,000 |
| Units | count/s |

| | |
|---|---|
| Sub-index | 0x02 |
| Description | Speed during search for encoder Index |
| Access | RW |
| PDO mapping | No |
| Value range | UINT32 |
| Default value | 20,000 |
| Units | count/s |

## Object 0x609A: Homing acceleration

This object defines the configured acceleration and also the deceleration used during homing. The value is count/s$^{2.}$

- Object description

| | |
|---|---|
| Index | 609A |
| Name | Homing acceleration |
| Object code | Variable |
| Data type | UINT32 |

- Entry description

| | |
|---|---|
| Sub-index | 0x00 |
| Access | RW |
| PDO mapping | Yes |
| Value range | UINT32 |
| Default value | 20,000 |
| Units | count/s$^{2.}$ |

# Object 0x3293: Hard stop current

This object defines the current strength (% of the peak current) that the driver will consider as encountering a hard stop. The unit is percentage (%).

It also sets a limit on the current output of the drive during homing to prevent machine damage in the event of unexpected hard stop impact.

- Object description

| | |
|---|---|
| Index | 3293 |
| Name | Hard stop current |
| Object code | Variable |
| Data type | REAL32 |

- Entry description

| | |
|---|---|
| Sub-index | 0x00 |
| Access | RW |
| PDO mapping | No |
| Value range | 0.01 ~ 0.9 x peak current |
| Default value | 0.5 |
| Units | percentage (%) |

## Object 0x3294: Hard stop period

This object defines the time length of.
When the time duration that the drive current output exceeds hard stop current (0x3293) for a duration longer than defined by this “hard stop period” object, the driver will consider this situation as encountering a hard stop.

- Object description

| | |
|---|---|
| Index | 3294 |
| Name | Hard stop period |
| Object code | Variable |
| Data type | UINT16 |

- Entry description

| | |
|---|---|
| Sub-index | 0x00 |
| Access | RW |
| PDO mapping | No |
| Value range | UINT 16 |
| Default value | 250 |
| Units | UINT 16 |

# 8.4 Functional Description

By choosing a homing mode, the following aspects are defined:

- The homing signal
- The direction of activation, and
- Position of the index pulse, where appropriate.

The home position and zero position will be replaced due to home offset; see previous descriptions of how home offset is used.

In the diagrams shown in the following chapters 8.5 and 8.6, the encircled number represents the chosen homing mode. The direction of movement is also indicated.

There are 4 sources of homing signal, they are:

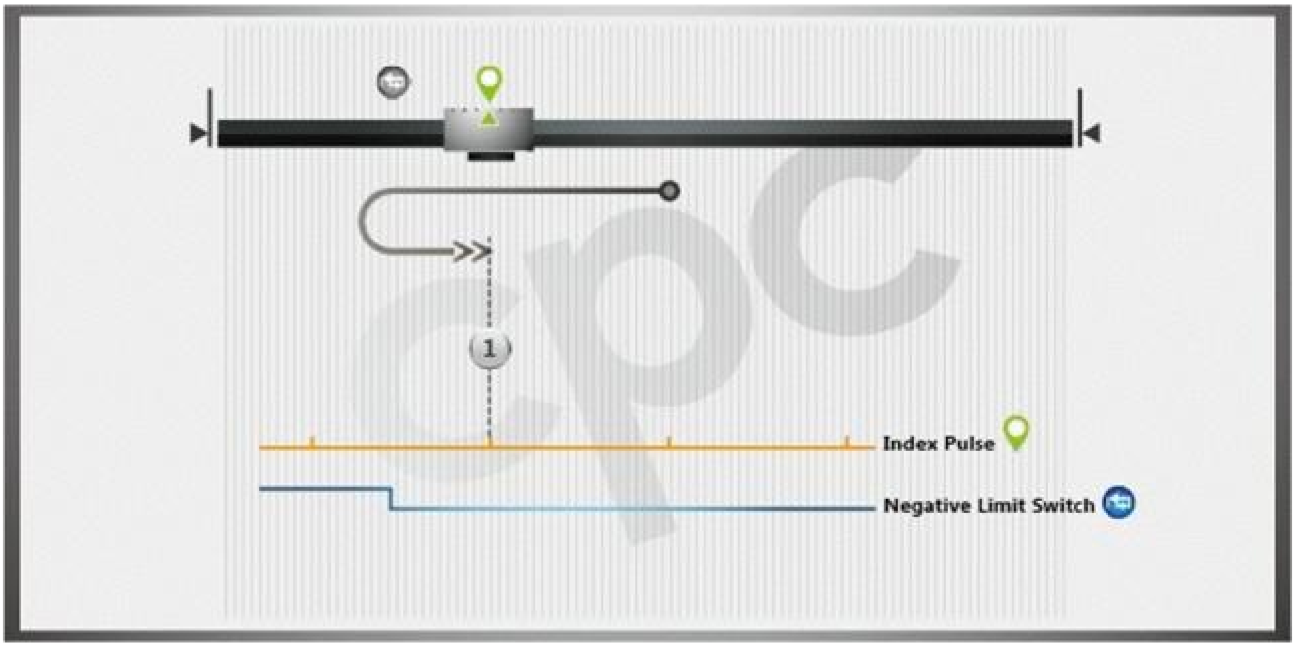
- Positive limit switch,
- Negative limit switch,
- Home switch, and
- Index pulse from an encoder

# 8.5 CiA 402 Homing Methods

## 8.5.1 By Limit Switch and Index Pulse

Method 1:

Home on the first index pulse after departing from the negative limit switch.

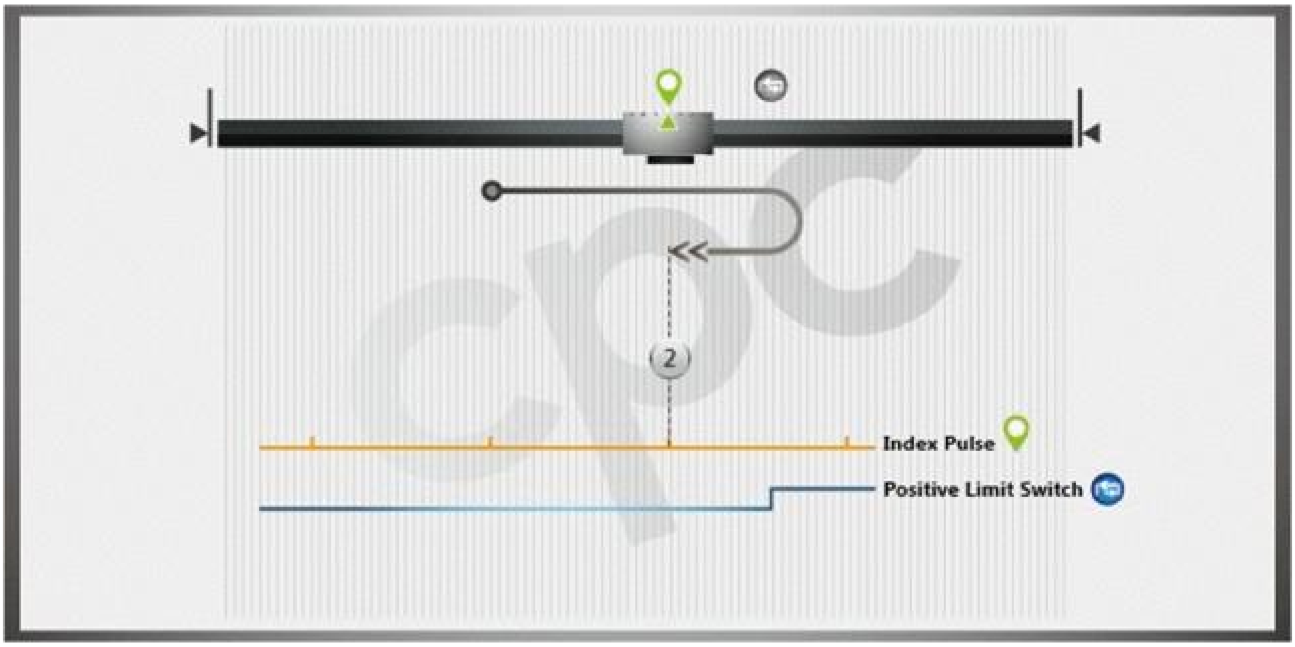


Homing process:

- Start with the negative motion unconditionally to the rising edge of the negative limit switch. Then, move in positive direction until the first index pulse is found.

Method 2:

Home on the first index pulse after departing from the positive limit switch.

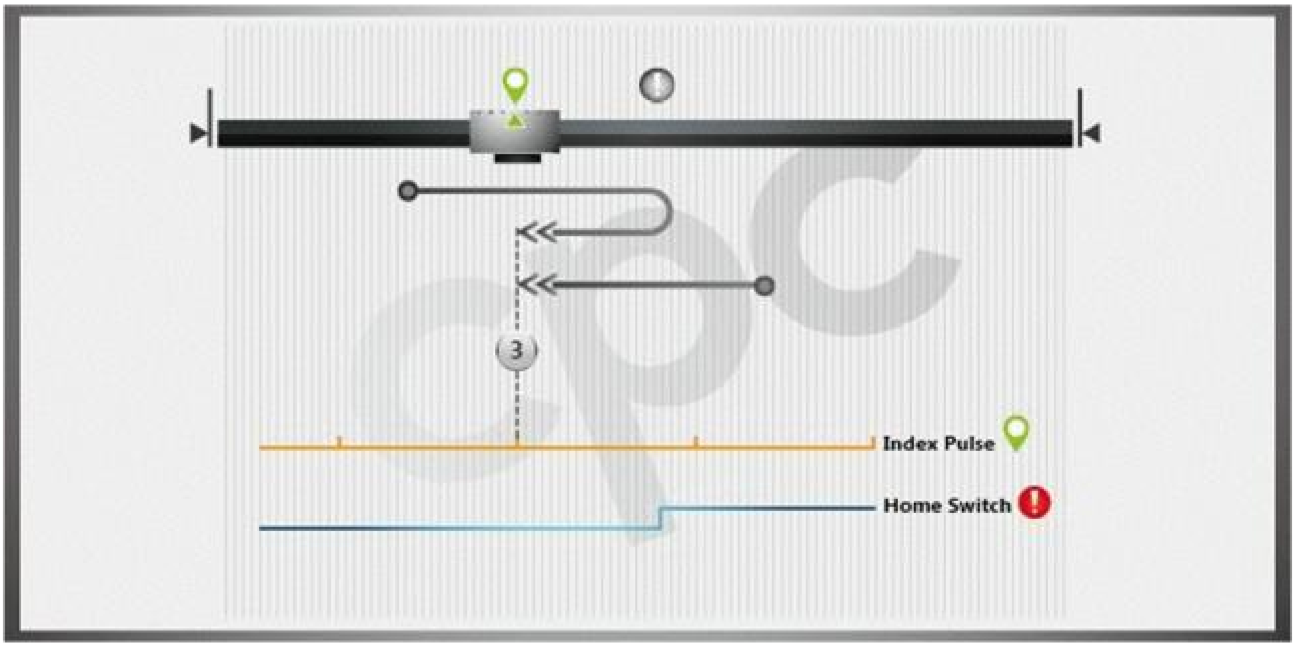


Homing process:

- Start with the positive direction unconditionally to the rising edge of the positive limit switch. Then, move in negative direction until the first index pulse is found.

Method 3:

Home on first index pulse after departing from home switch.



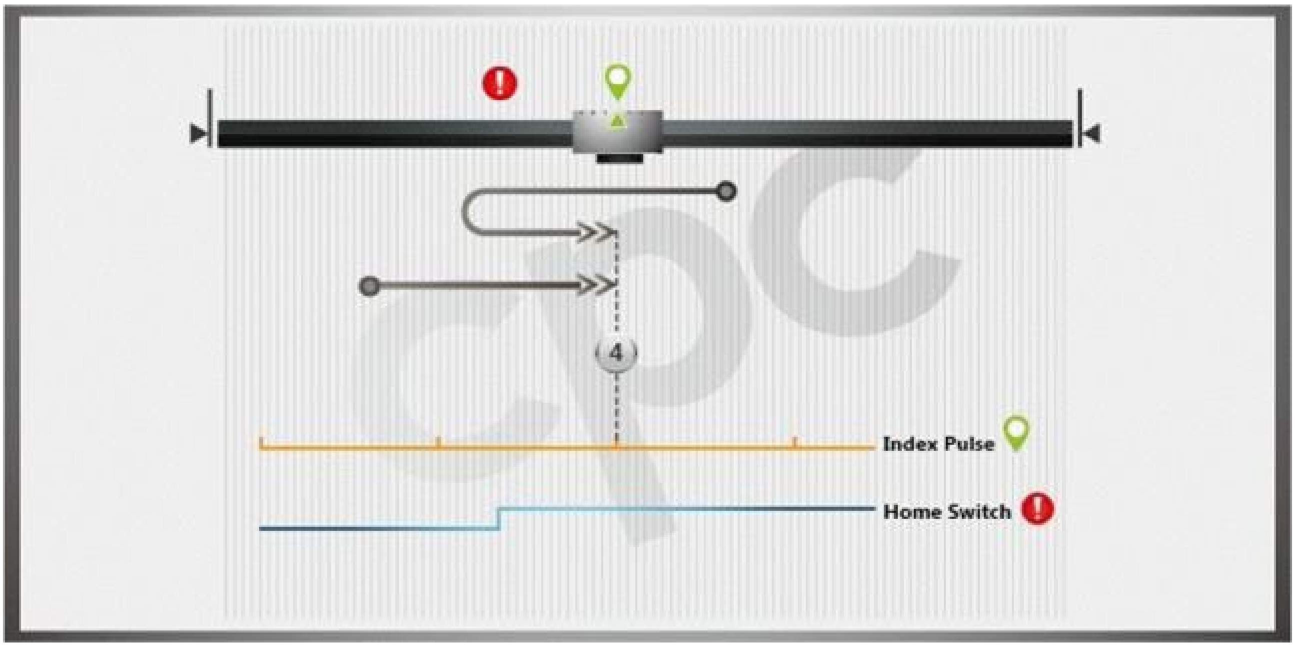
Homing process:

- If the home switch is inactive, start with the positive direction to the rising edge of the home switch. Then, move in negative direction until the first index pulse is found.
- If the home switch is active, start with the negative direction until the first index pulse is found.

## 8.5.2 By Home Switch and Index Pulse

Method 4:

Home on the first index pulse after engaging home switch.

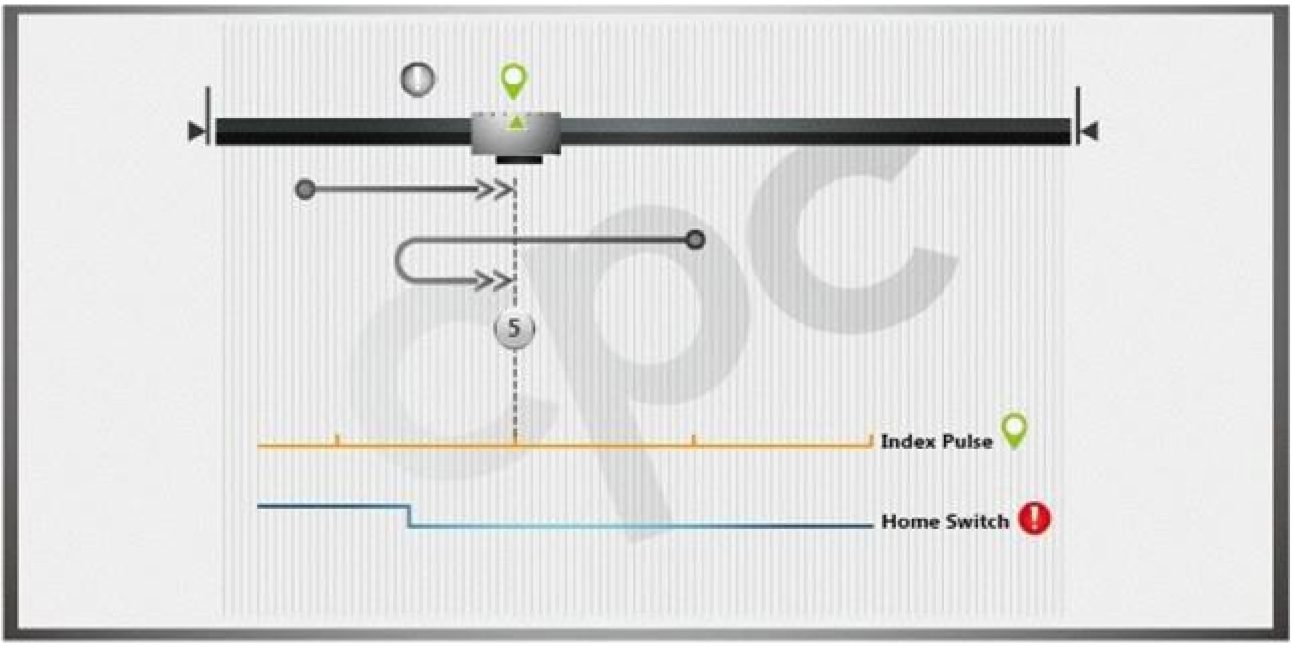


Homing process:

- If the home switch is active, start with the negative direction to the falling edge of the home switch. Then, move in positive direction until the first index pulse is found.
- If the home switch is inactive, start with the positive direction until the first index pulse is found.

Method 5:

Home on the first index pulse after departing from home switch.

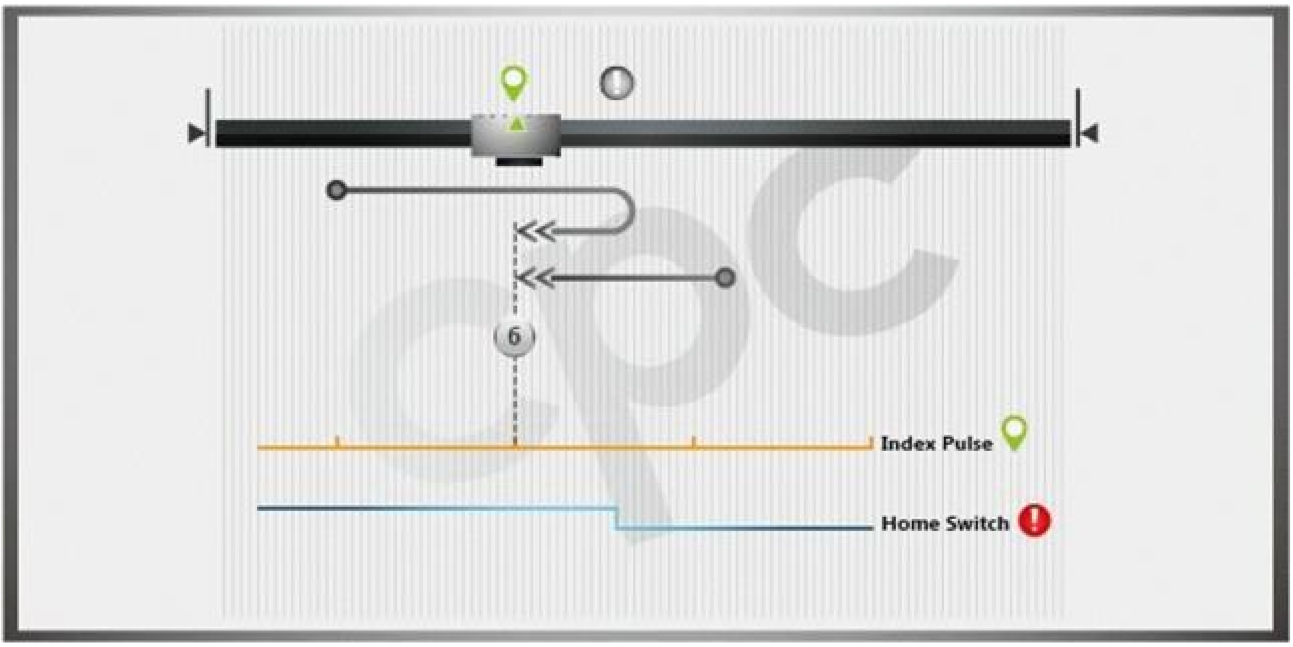


Homing process:

- If the home switch is active, start with the positive direction until the first index pulse is found.
- If the home switch is inactive, start with the negative direction to the rising edge of the home switch. Then, move in positive direction until the first index pulse is found.

Method 6:

Home on the first index pulse after engaging home switch.



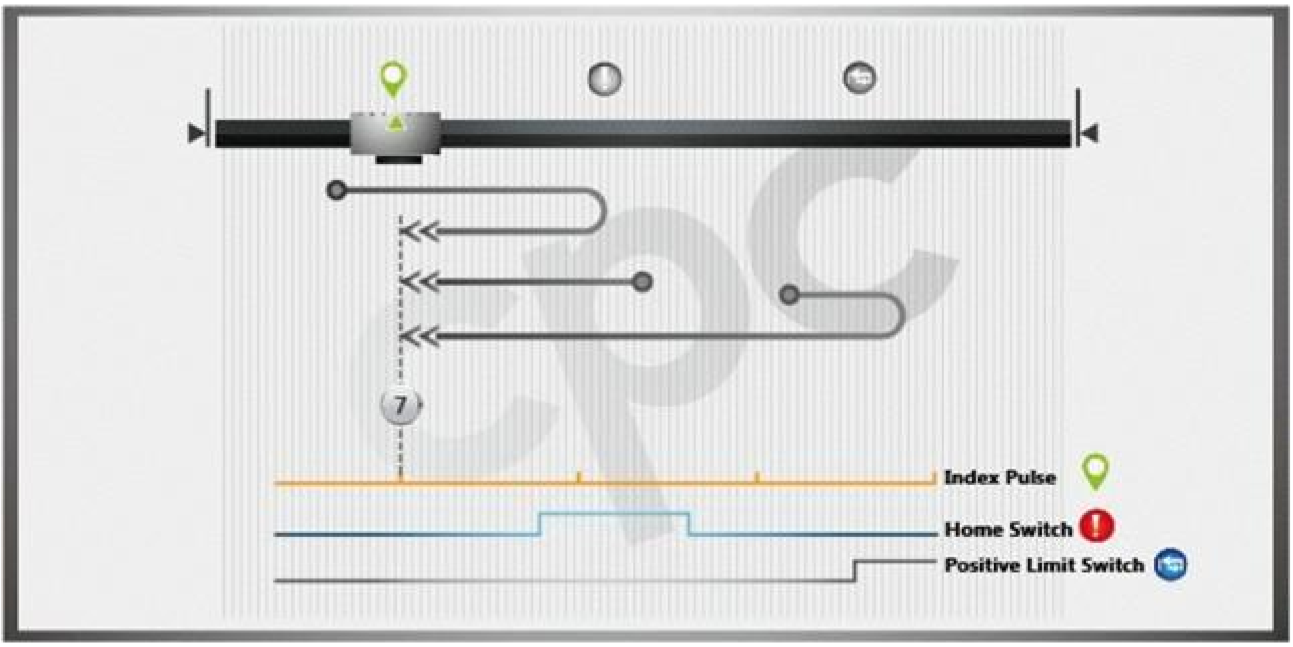
Homing process:

- If the home switch is active, start with the positive direction to the falling edge of the home switch. Then, move in negative direction until the first index pulse is found.
- If the home switch is inactive, start with the negative direction until the first index pulse is found.

### 8.5.3 By Home Switch, Index Pulse, and Limit Switch

Method 7:

Home on the first index pulse after departing from home switch while moving in negative direction.

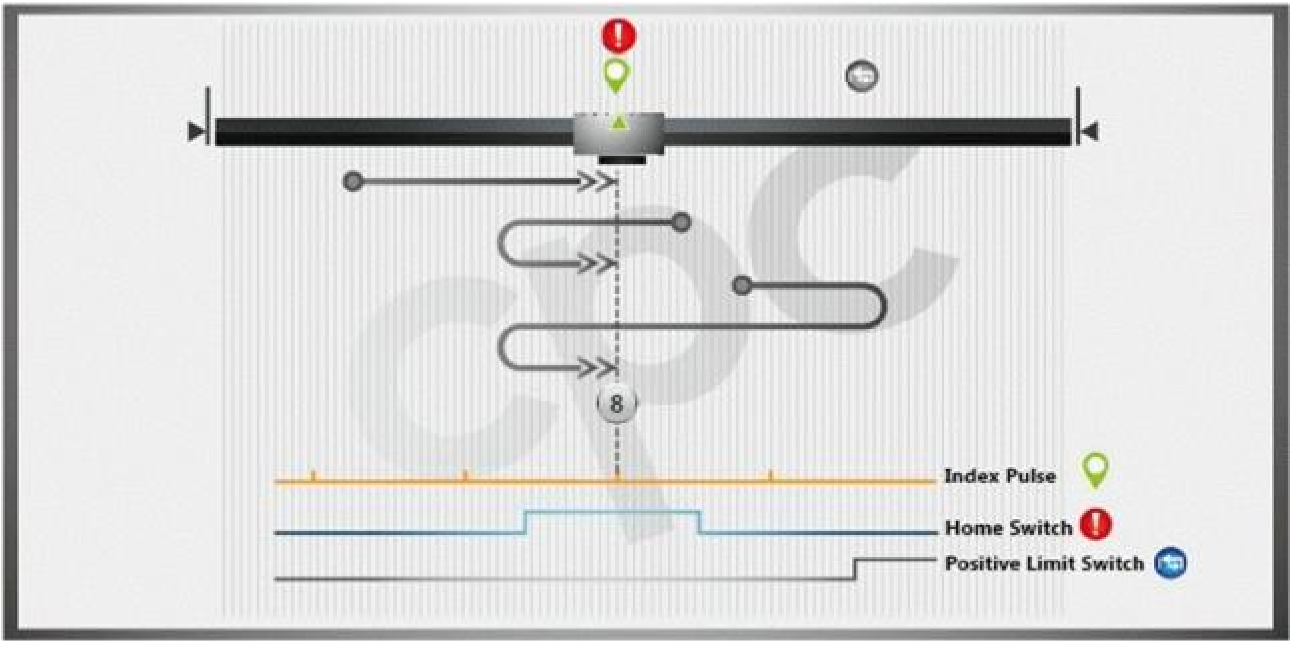


Homing process:

- If the home switch is inactive, start with the positive motion. If the home switch is engaged, move in negative direction until the home switch is disengaged, then find the first index pulse.
- If the home switch is active, start with the negative direction until the home switch is disengaged, then continue moving in negative direction until the first index pulse is found.
- If the home switch is inactive, start with the positive motion. If the positive limit switch is engaged, move in negative direction until the home switch is engaged and then disengaged, and then move in negative direction until the first index pulse is found.

Method 8:

Home on the first index pulse after engaging home switch while moving in positive direction.

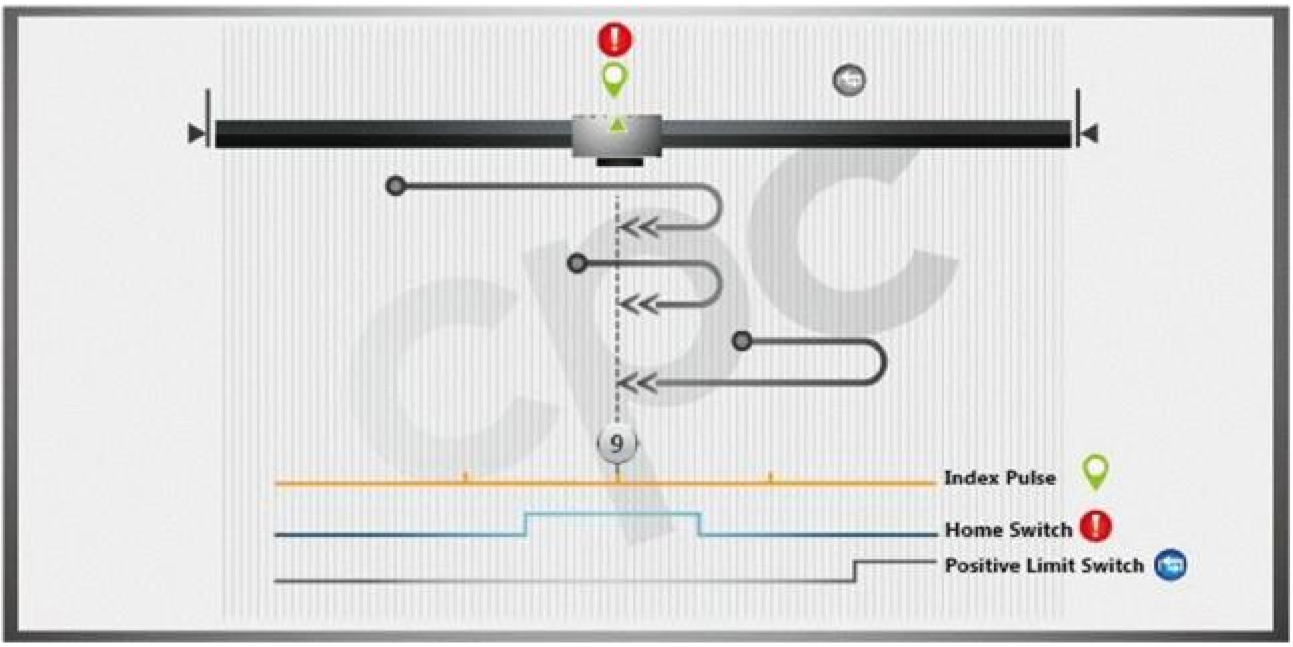


Homing process:

- If the home switch is inactive, start with positive direction until the home switch is met, then keep moving in positive direction until the first index pulse is found.
- If the home switch is active, start with negative direction until home switch is disengaged, then move in positive direction until home switch is engaged, and then find the first index pulse.
- If the home switch is inactive, start with positive direction; when the positive limit switch is engaged, move in negative direction until the home switch is engaged and disengaged, and then move in positive direction until the first index pulse is found.

Method 9:

Home on the first index pulse after engaging home switch while moving in negative direction.

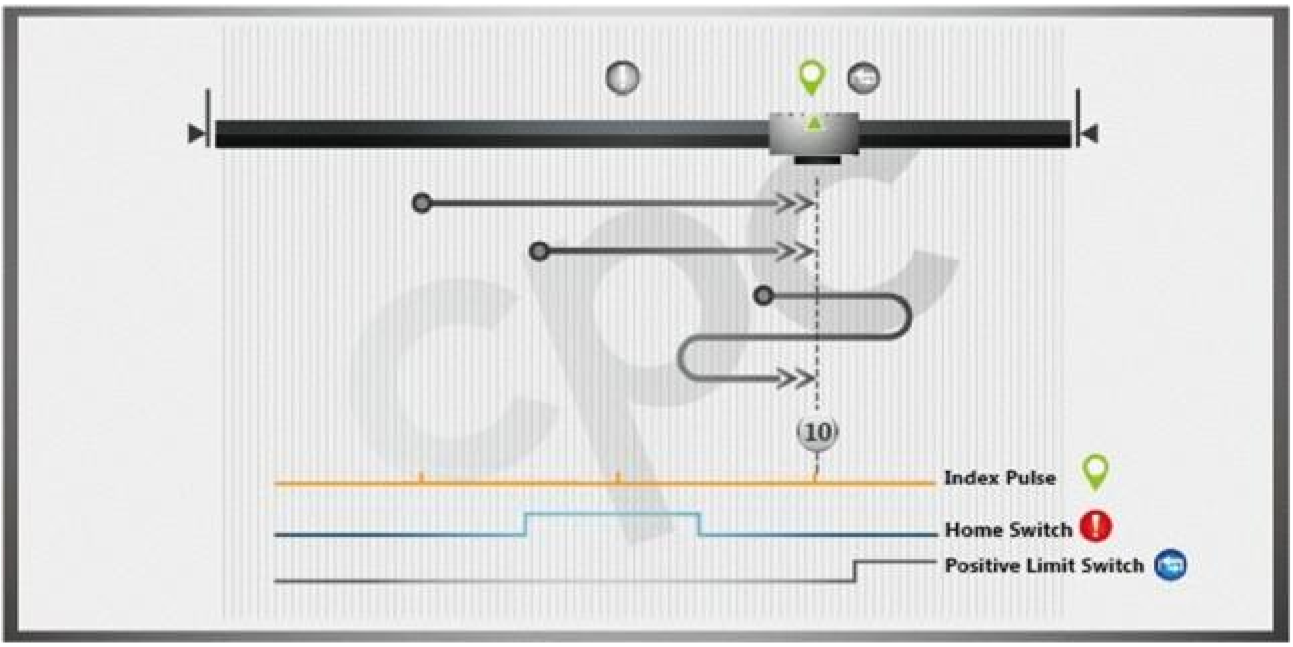


Homing process:

- Start with positive direction unconditionally. If home switch is engaged, keep moving in positive direction until home switch is disengaged, then move in negative direction until home switch is engaged, and then find the first index pulse.
- Start with positive direction unconditionally. If home switch is disengaged, move in negative direction until home switch is engaged, then find the first index.
- Start with the positive motion unconditionally. If the positive limit switch is engaged, move in negative direction until home switch is engaged, then find the first index pulse.

Method 10:

Home on the first index pulse after departing from home switch while moving in positive direction.

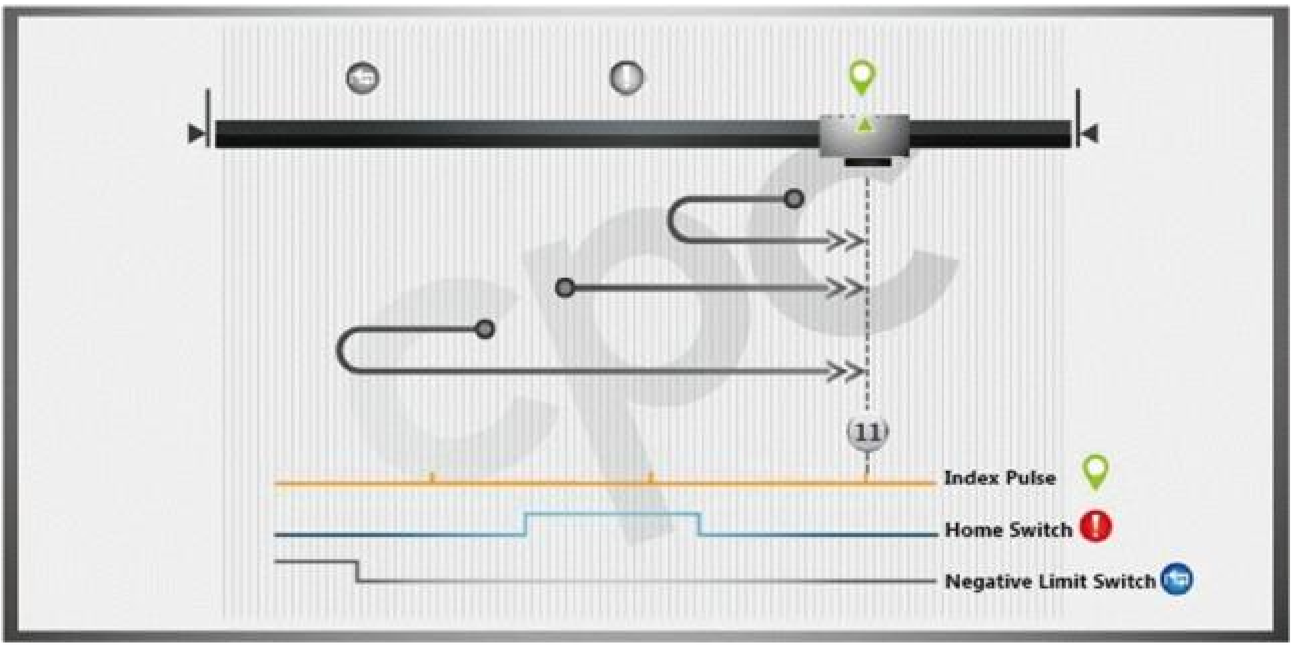


Homing process:

- Start with positive direction unconditionally. If home switch is then engaged, keeping moving in positive direction until home switch is disengaged, then find the first index pulse.
- Start with positive direction unconditionally. If home switch is active and then disengaged, keep moving in positive direction until the first index pulse is found.
- Start with position direction unconditionally. If positive limit switch is then engaged, move in negative direction. If home switch is engaged, move in positive direction until home switch is disengaged, then find the first index pulse.

Method 11:

Home on the first index pulse after departing from home switch while moving in positive direction.

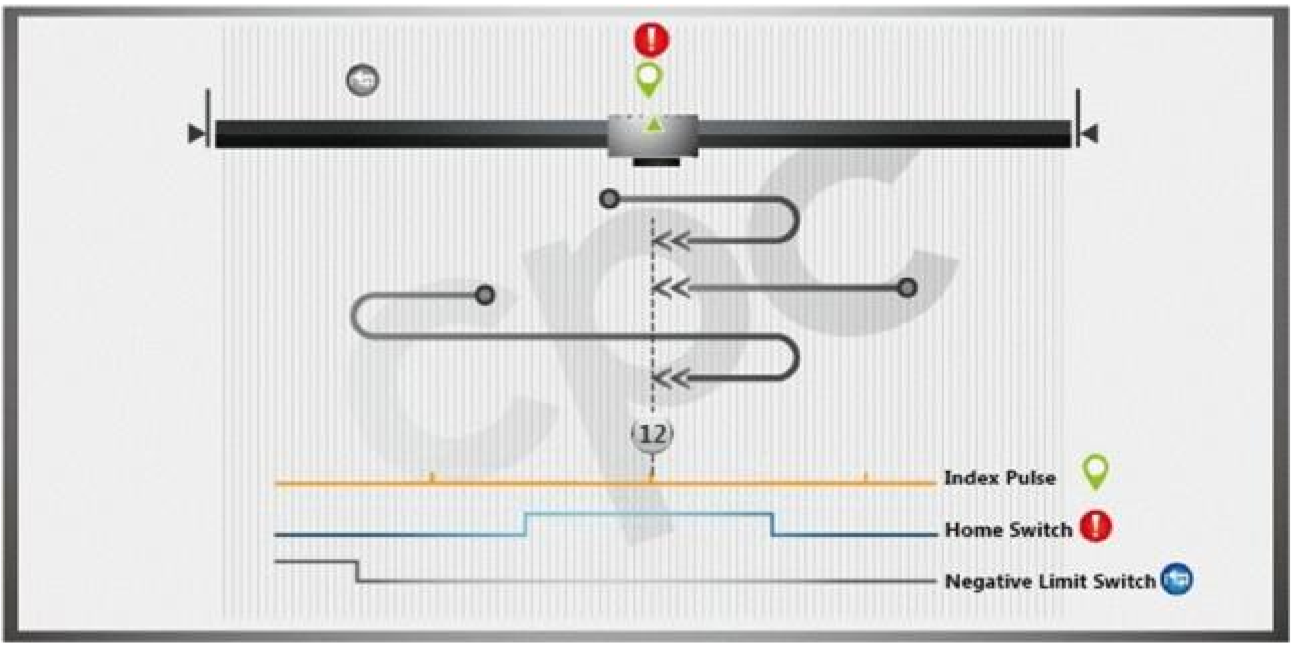


Homing process:

- If home switch is inactive, move in negative direction. If home switch is then engaged, move in positive direction until home switch is disengaged, then, find the first index pulse.
- If home switch is active, move in positive direction until home switch is disengaged, continue moving in positive direction until the first index pulse is found.
- If home switch is inactive, move in negative direction. If negative limit switch is then engaged, move in positive direction until home switch is engaged, continue moving in positive direction until home switch is disengaged, then, find the first index pulse.

Method 12:

Home on the first index pulse after engaging home switch while moving in negative direction.

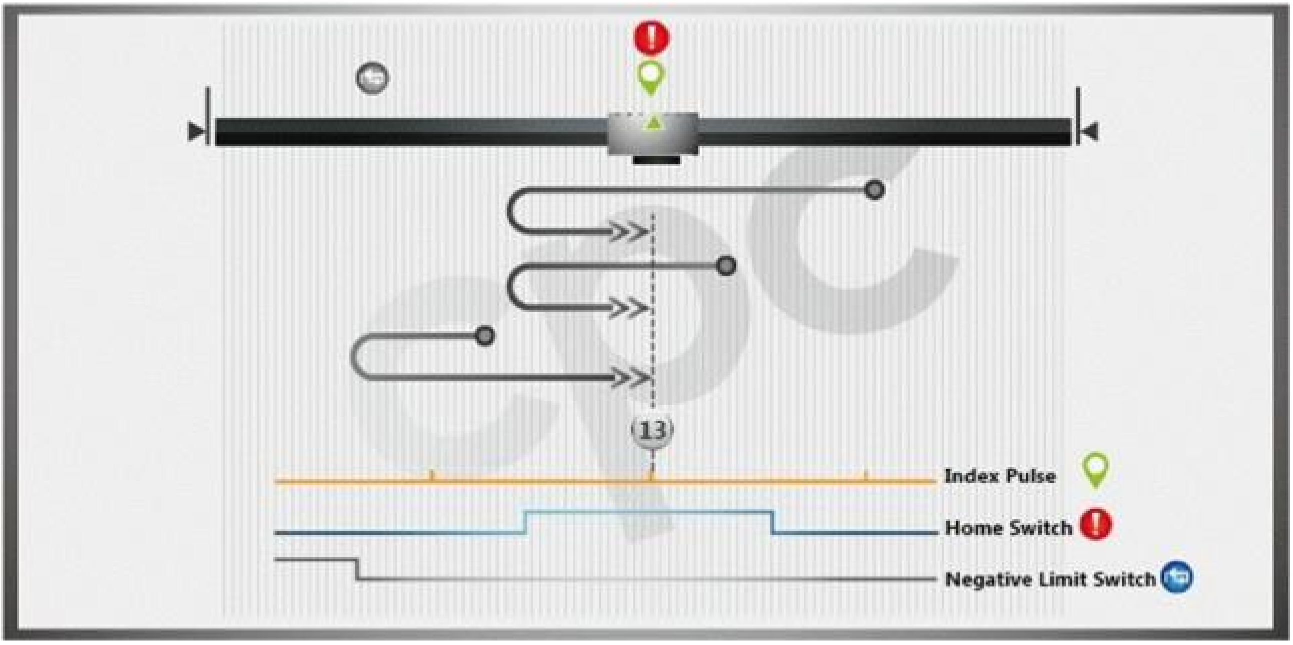


Homing process:

- If home switch is active, move in positive direction. If home switch is then disengaged, move in negative direction until home switch is engaged, then, find the first index pulse.
- If home switch is inactive, move in negative direction until home switch is engaged, then, continue moving in negative direction until the first index pulse is found.
- If home switch is inactive, move in negative direction. If negative limit switch is then engaged, move in positive direction until home switch is engaged. If home switch is then disengaged, move in negative direction until home switch is engaged, then, find the first index pulse.

Method 13:

Home on the first index pulse after engaging home switch while moving in positive direction.

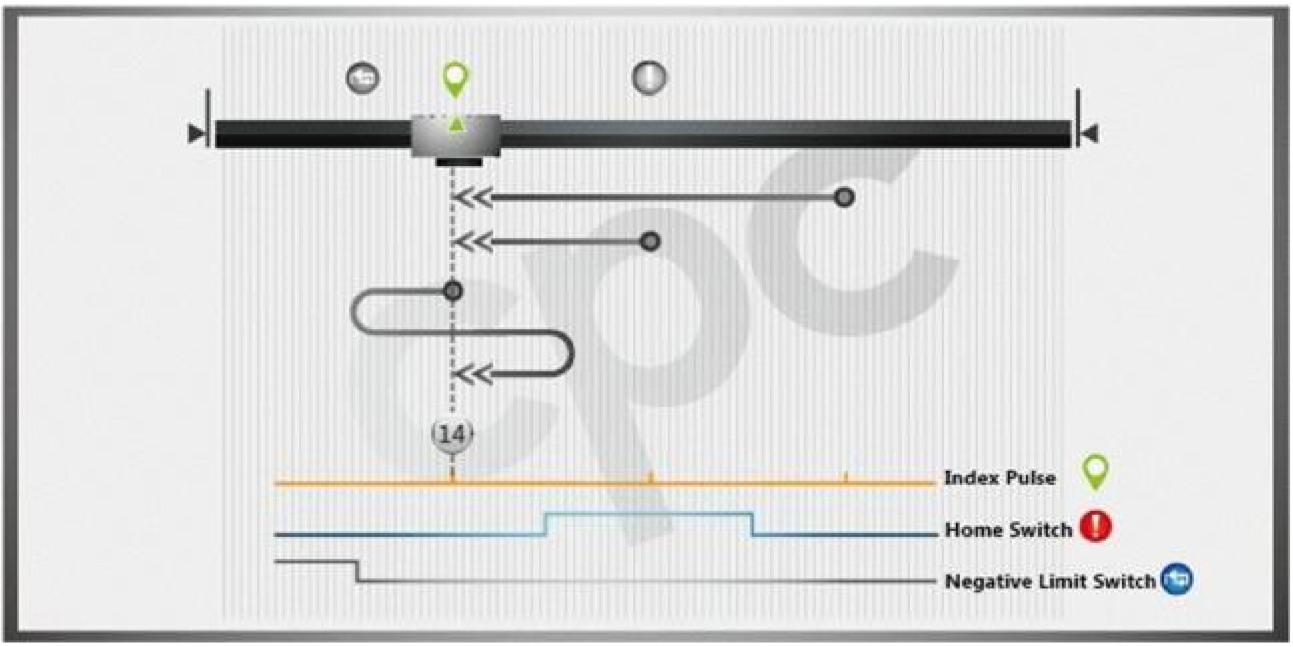


Homing process:

- Start with negative motion unconditionally. If home switch is then engaged, continue moving in negative direction. If home switch is then disengaged, move in positive direction until home switch is engaged, then, find the first index pulse.
- Start with negative motion unconditionally. If home switch is then disengaged, move in positive direction until home switch is engaged, then find the first index pulse.
- Start with negative motion unconditionally. If negative limit switch is then engaged, move in positive direction until home switch is engaged, then find the first index pulse.

Method 14:

Home on the first index pulse after departing home switch while moving in negative direction.



Homing process:

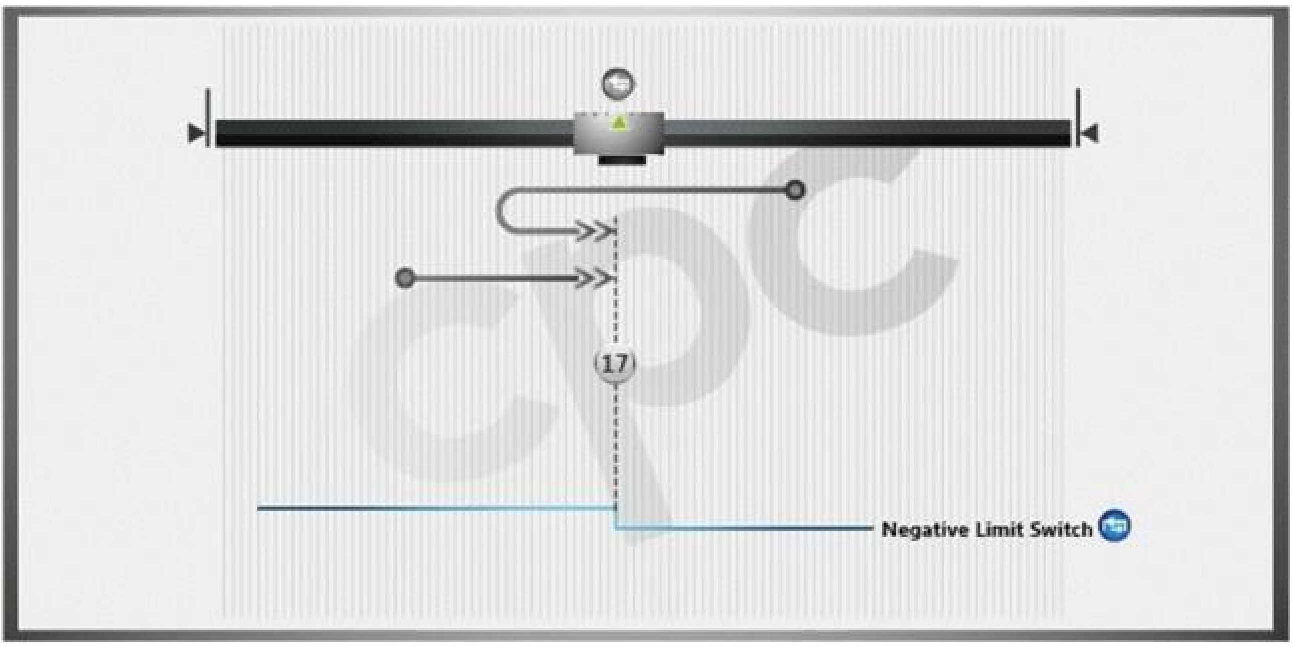
- Start with negative direction unconditionally. If home switch is inactive, move in negative direction. If home switch is then engaged, keep moving in negative direction until home switch is disengaged, then, find the first index pulse.
- Start with negative direction unconditionally. If home switch is then disengaged, keep moving in negative direction until the first index pulse is found.
- Start with negative direction unconditionally. If negative limit switch is then engaged, move in positive direction. If home switch is then engaged, move in negative direction until home switch is disengaged, then, find the first index pulse.

## 8.5.4 Method 15 to 16: Reserved.

## 8.5.5 By Limit Switch

Method 17:

Home on negative limit switch without index pulse.

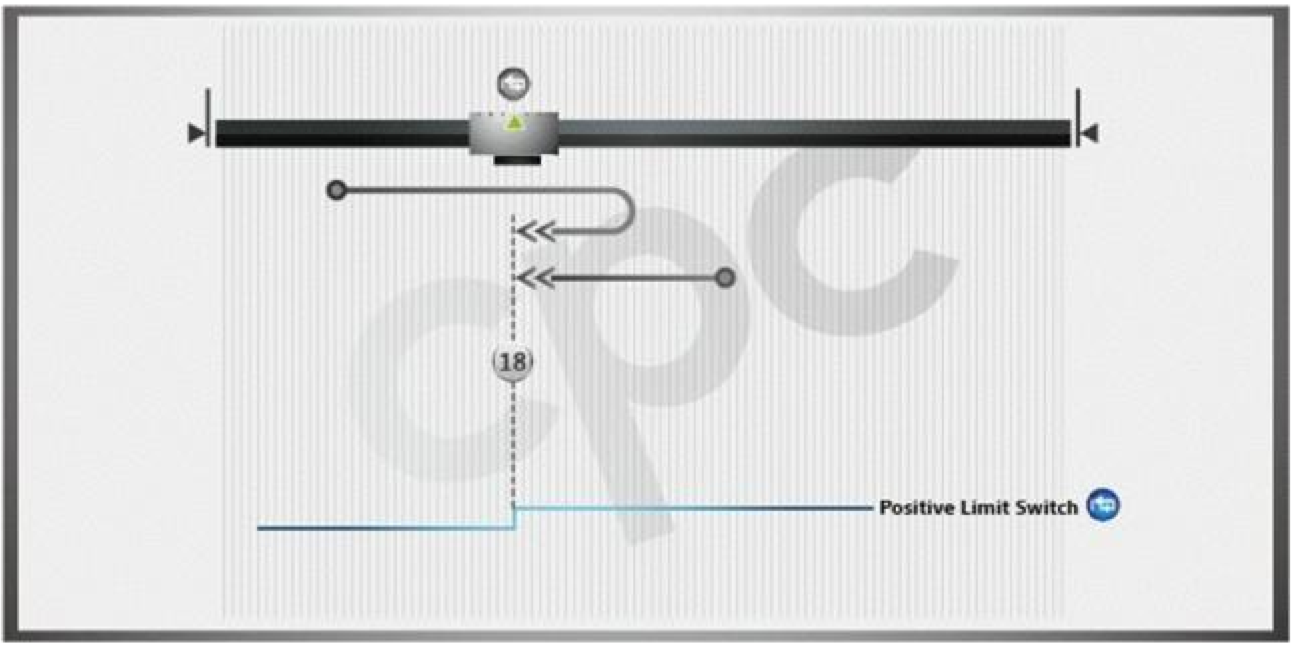


Homing process:

- If negative limit switch is inactive, move in negative direction. If negative limit switch is then engaged, move in positive direction to locate the falling edge of the negative limit switch.
- If negative limit switch is active, move in positive direction to locate the falling edge of the negative limit switch

Method 18:

Home on positive limit switch without index pulse.



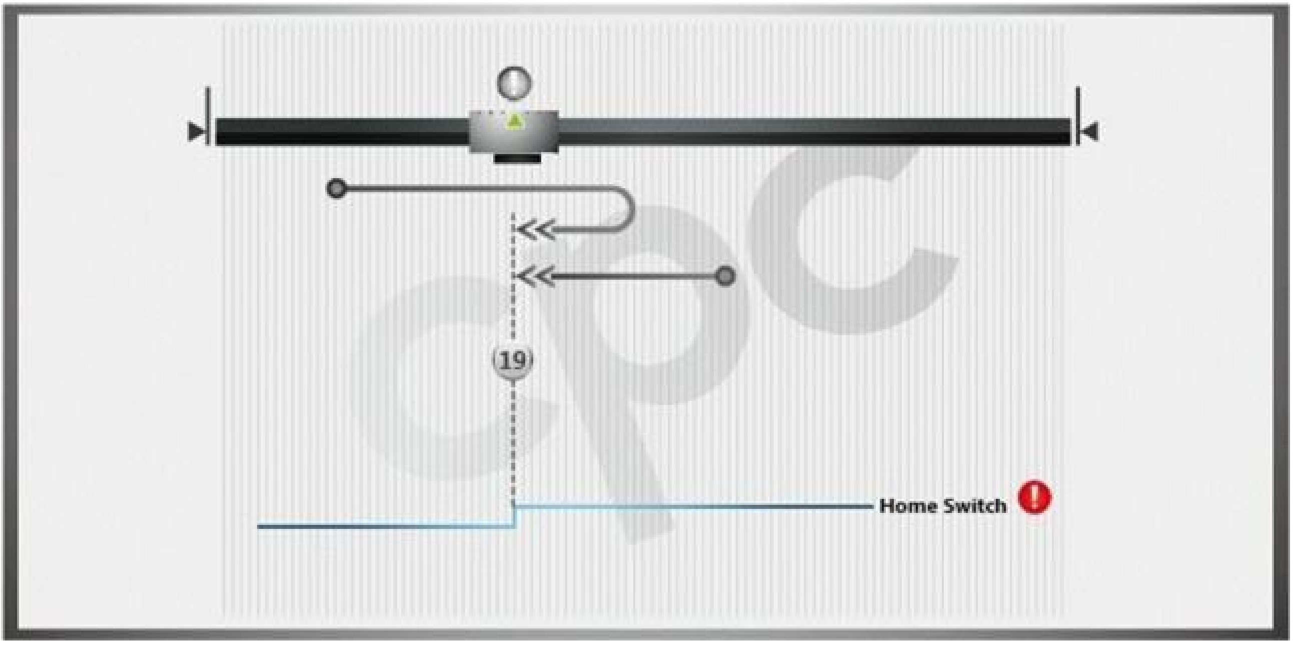
Homing process:

- If positive limit switch is inactive, move in positive direction. If positive limit switch is then engaged, move in negative direction to locate the falling edge of the positive limit switch.
- If positive limit switch is active, move in negative direction to locate the falling edge of the positive limit switch

## 8.5.6 By Rising/Falling Edge of Home Switch

Method 19:

Home on falling edge of home switch without index pulse.



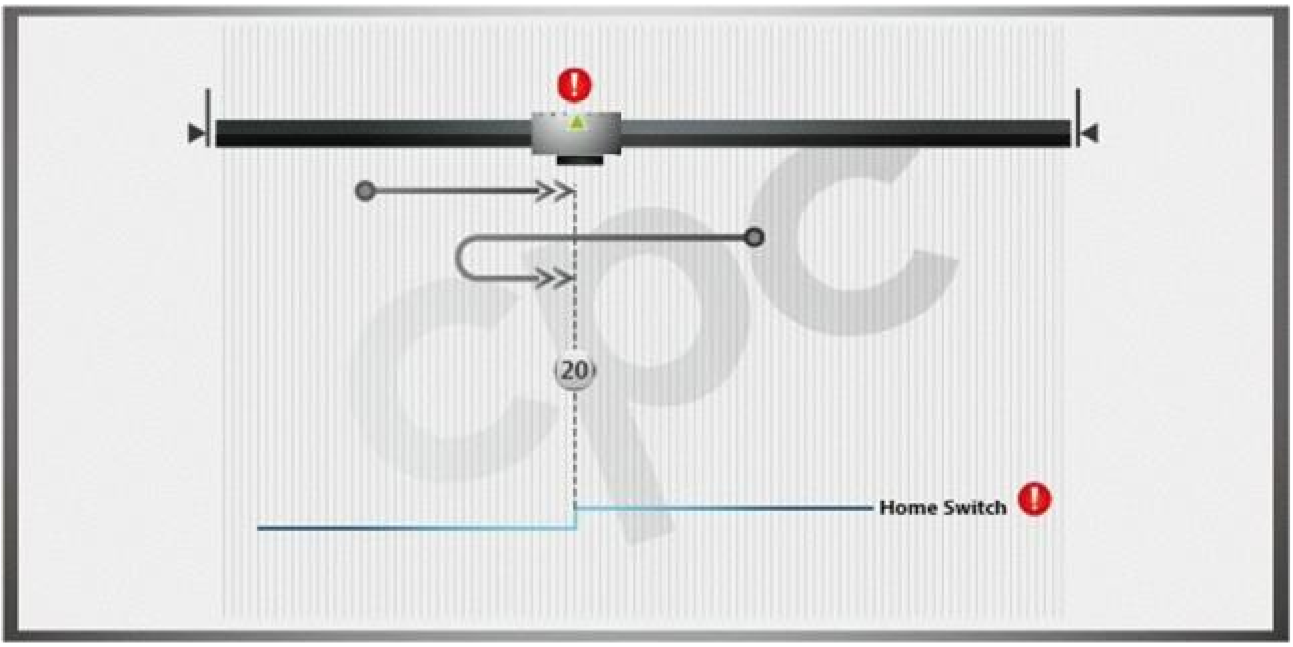
Homing process:

If home switch is inactive, move in positive direction. If home switch is then engaged, move in negative direction to locate the falling edge of the home switch.

If home switch is active, move in negative direction to locate the falling edge of the home switch.

Method 20:

Home on rising edge of home switch without index pulse.



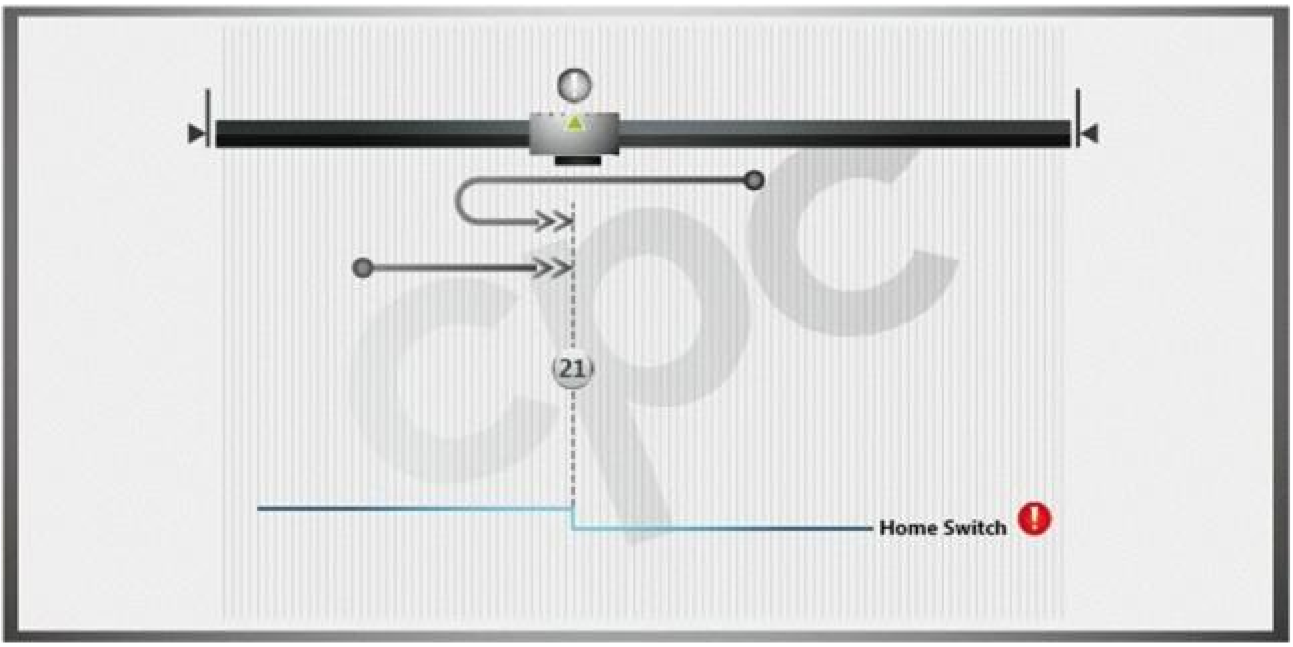
Homing process:

If home switch is inactive, move to rising edge of the home switch in positive direction.

If home switch is active, move in negative direction. If home switch is then disengaged, move to rising edge of the home switch in positive direction.

Method 21:

Home on falling edge of home switch without index pulse.

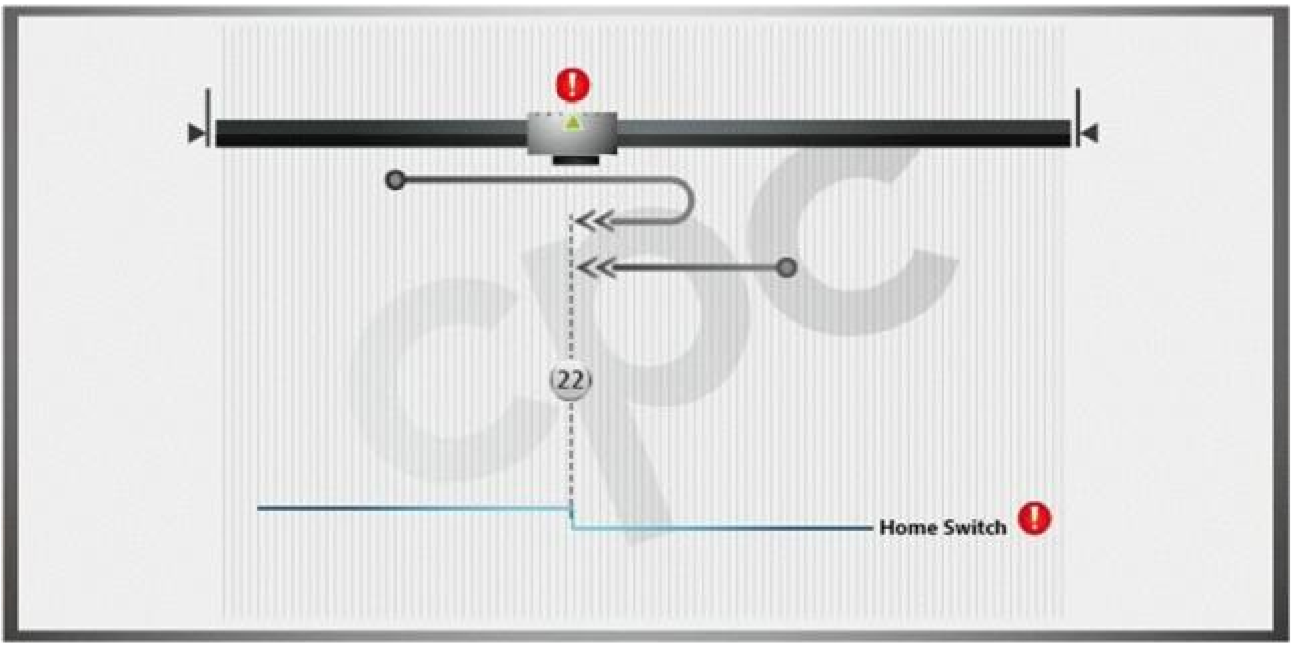


Homing process:

- If home switch is inactive, move in negative direction. If home switch is then engaged, move to falling edge of the home switch in positive direction.
- If home switch is active, move to falling edge of the home switch in positive direction.

Method 22:

Home on rising edge of home switch without index pulse.



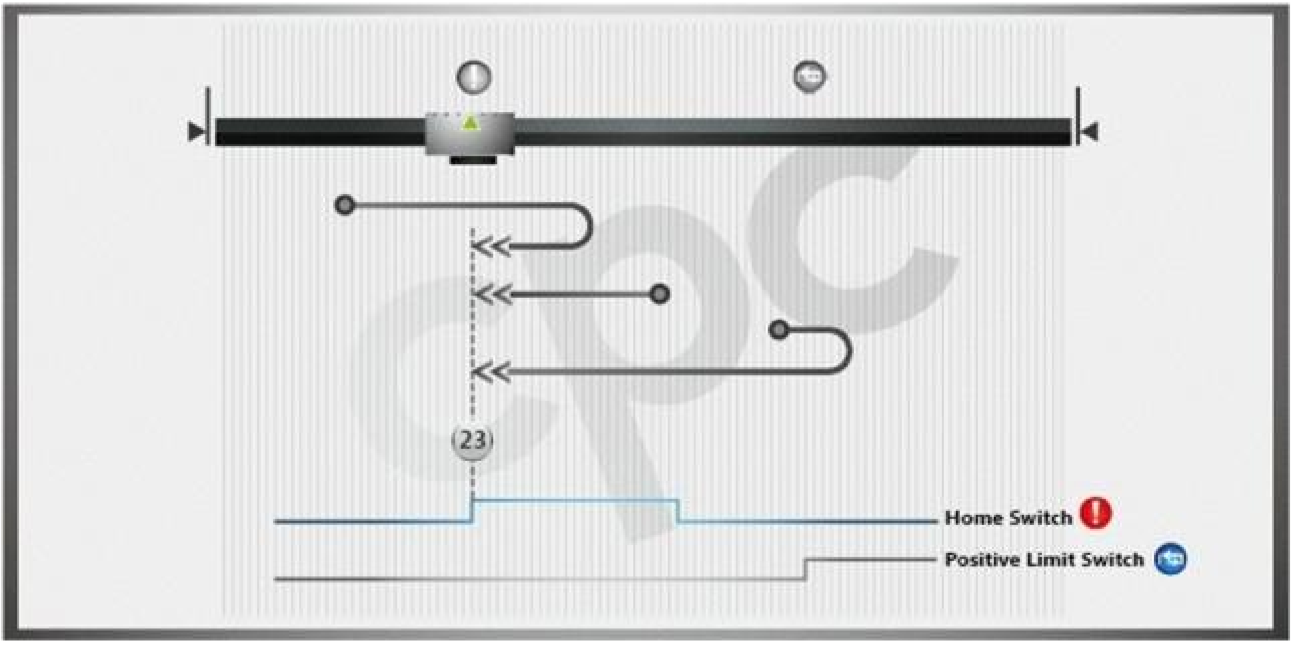
Homing process:

- If home switch is active, move in positive direction. If home switch is then disengaged, move to rising edge of the home switch in negative direction.
- If home switch is inactive, move to rising edge of the home switch in negative direction.

## 8.5.7 By Home Switch and Limit Switch

Method 23:

Home on falling edge of home switch while moving in negative direction.

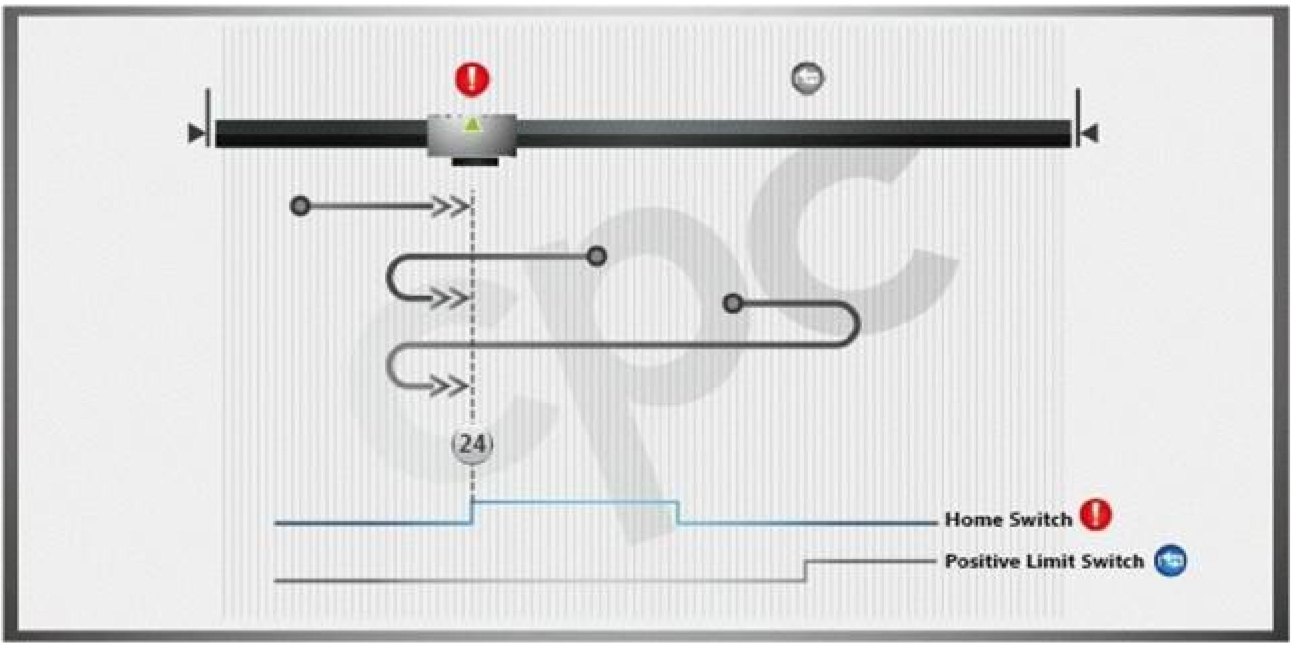


Homing process:

- If home switch is inactive, move in positive direction. If home switch is then engaged, move to falling edge of the home switch in negative direction.
- If home switch is active, move to falling edge of the home switch in negative direction.
- If home switch is inactive, move in positive direction. If positive limit switch is then engaged, move in negative direction. If home switch is then engaged, keeping moving to locate the falling edge of the home switch.

Method 24:

Home on rising edge of home switch while moving in positive direction.

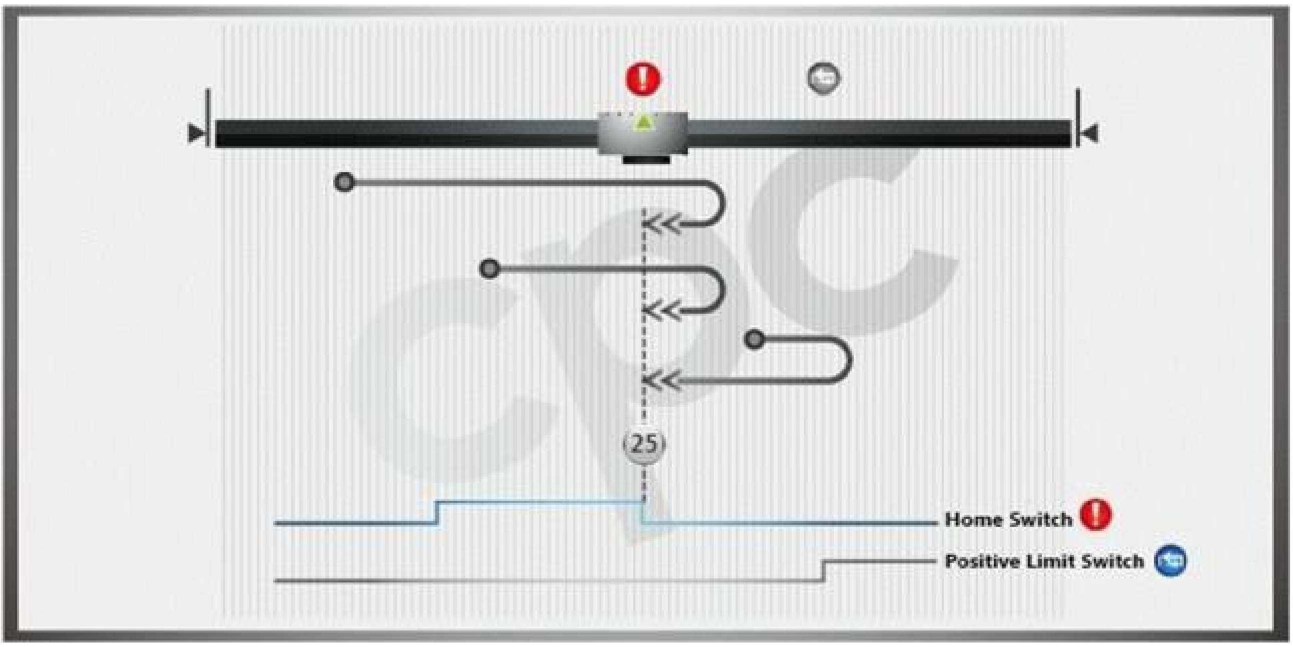


Homing process:

- If home switch is inactive, move to rising edge of the home switch in positive direction.
- If home switch is active, move in negative direction. If home switch is then disengaged, move to rising edge of the home switch in positive direction.
- If home switch is inactive, move in positive direction. If positive limit switch is then engaged, move in negative direction until home switch is engaged, continue moving in negative direction. If home switch is then disengaged, then, move to rising edge of the home switch in positive direction.

Method 25:

Home on rising edge of home switch while moving in negative direction.

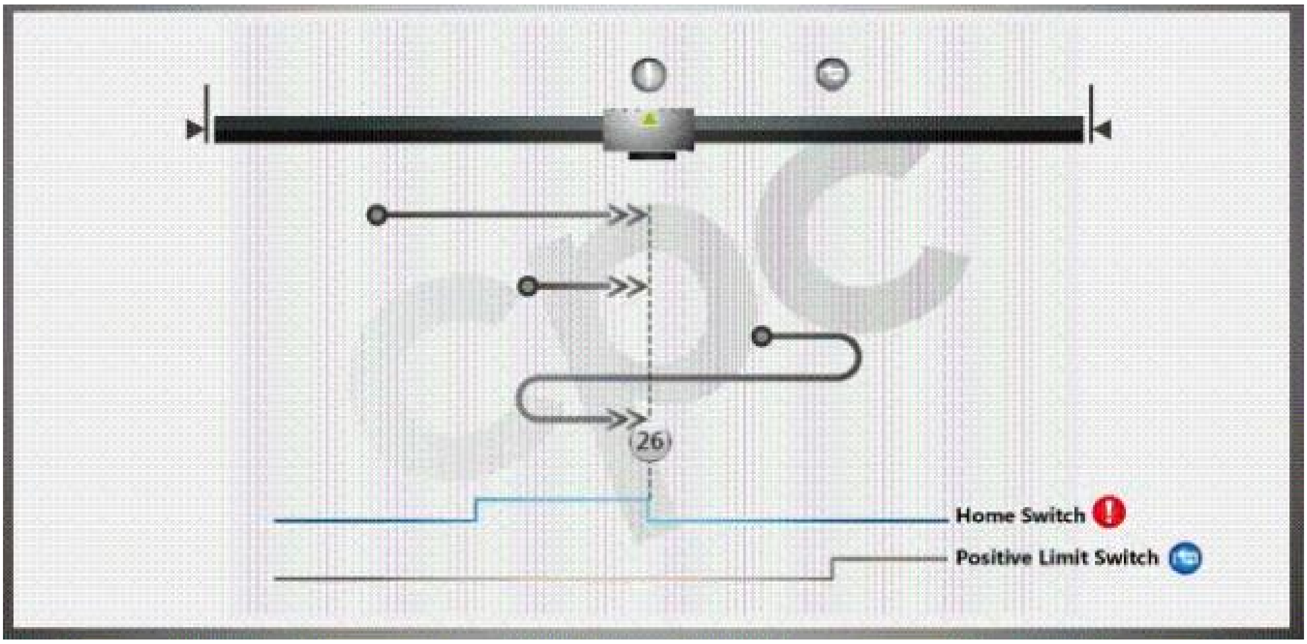


Homing process:

- If home switch is inactive, move in positive direction. If home switch is then engaged, continue moving in positive direction. If home switch is then disengaged, move to rising edge of the home switch in negative direction.
- If home switch is active, move in positive direction. If home switch is then disengaged, move to rising edge of the home switch in negative direction.
- If home switch is inactive, move in positive direction. If positive limit switch is then engaged, move to rising edge of the home switch in negative direction.

Method 26:

Home on falling edge of home switch while moving in positive direction.

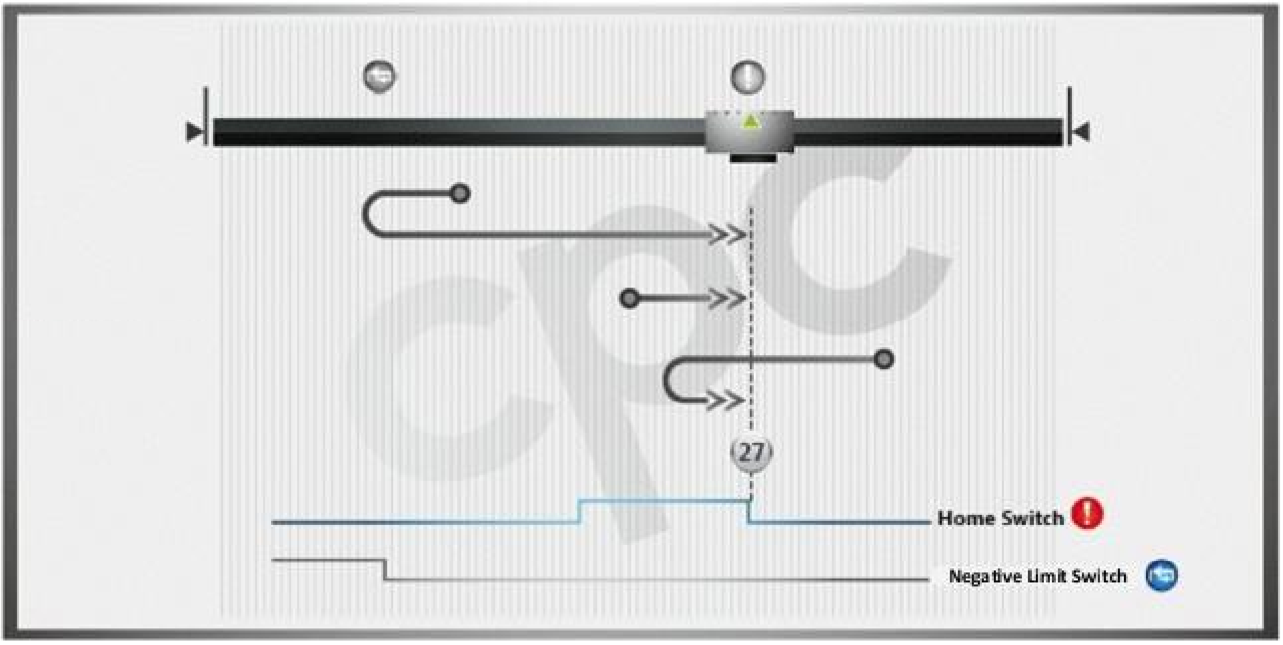


Homing process:

- If home switch is inactive, move in positive direction. If home switch is then engaged, move to falling edge of the home switch in positive direction.
- If home switch is active, move to falling edge of the home switch in positive direction.
- If home switch is inactive, move in positive direction. If positive limit switch is then engaged, move in negative direction. If home switch is then engaged, move to falling edge of the home switch in positive direction.

Method 27:

Home on falling edge of home switch while moving in positive direction.

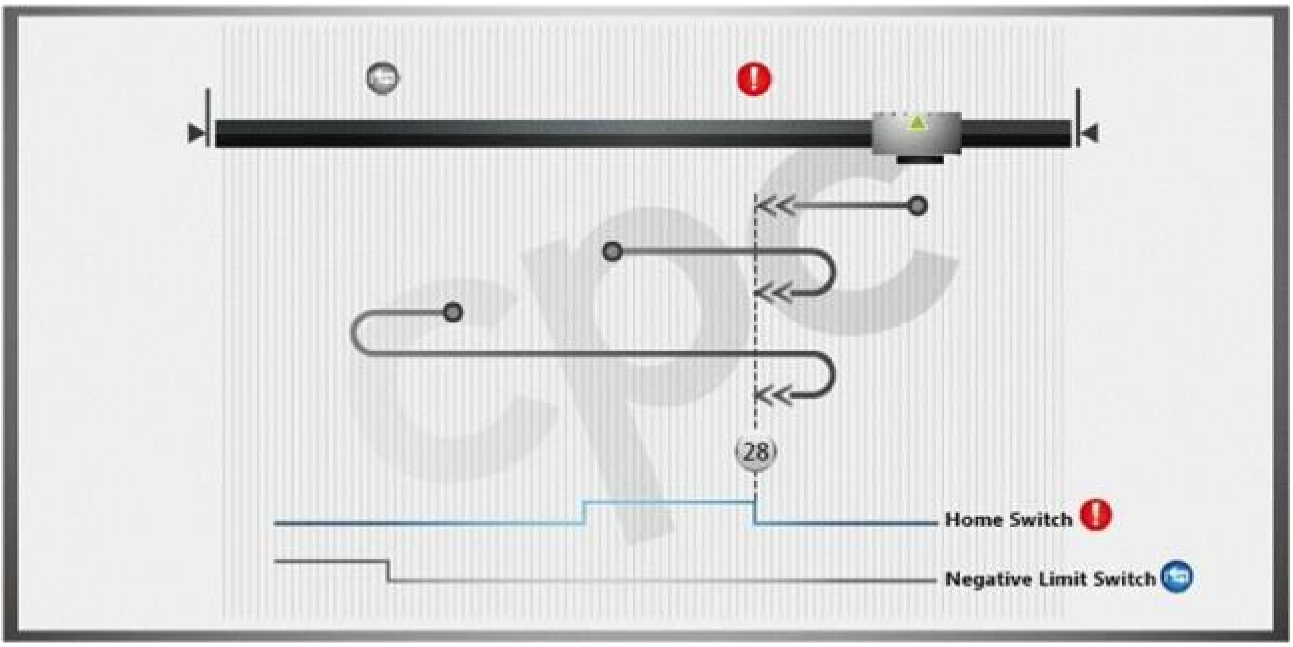


Homing process:

- If home switch is inactive, move in negative direction. If negative limit switch is then engaged, move in positive direction. If home switch is then engaged, move to falling edge of the home switch in positive direction.
- If home switch is active, move to falling edge of the home switch in positive direction.
- If home switch is inactive, move in negative direction. If home switch is then engaged, move to falling edge of the home switch in positive direction.

Method 28:

Home on rising edge of home switch while moving in negative direction.

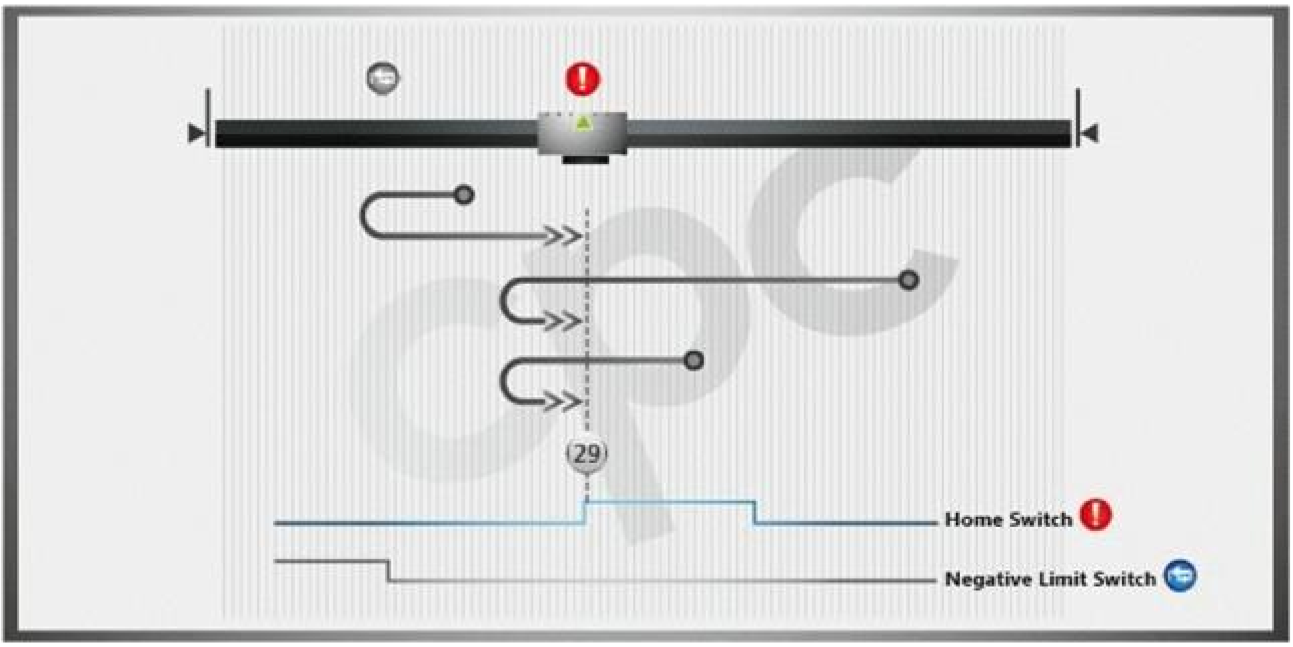


Homing process:

- If home switch is inactive, move to rising edge of the home switch in negative direction.
- If home switch is active, move in positive direction. If home switch is then disengaged, move to rising edge of the home switch in negative direction.
- If home switch is inactive, move in negative direction. If negative limit switch is then engaged, move in positive direction until home switch is engaged, continue moving in positive direction. If home switch is then disengaged, move to rising edge of the home switch in negative direction.

Method 29:

Home on rising edge of home switch while moving in positive direction.

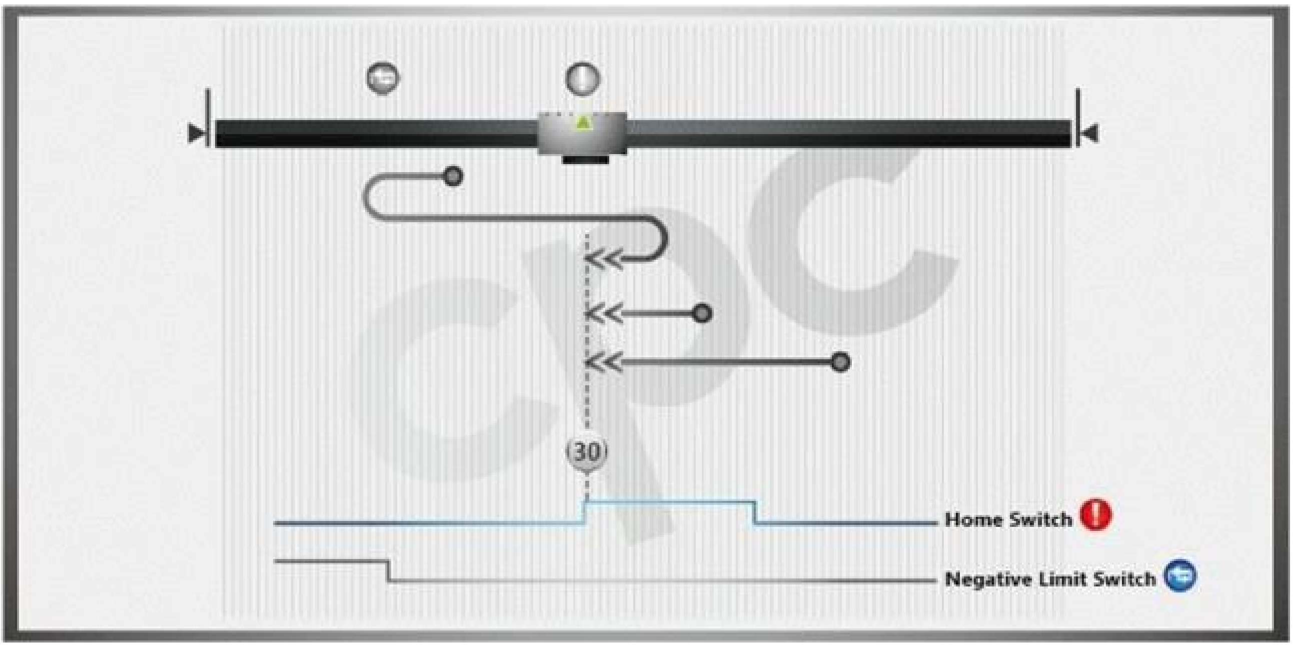


Homing process:

- If home switch is inactive, move in negative direction. If negative limit switch is then engaged, move to rising edge of the home switch in positive direction.
- If home switch is inactive, move in negative direction until home switch is engaged, continue moving in negative direction. If home switch is then disengaged, move to rising edge of the home switch in positive direction.
- If home switch is active, move in negative direction. If home switch is then disengaged, move to rising edge of the home switch in positive direction.

Method 30:

Home on falling edge of home switch while moving in negative direction.



Homing process:

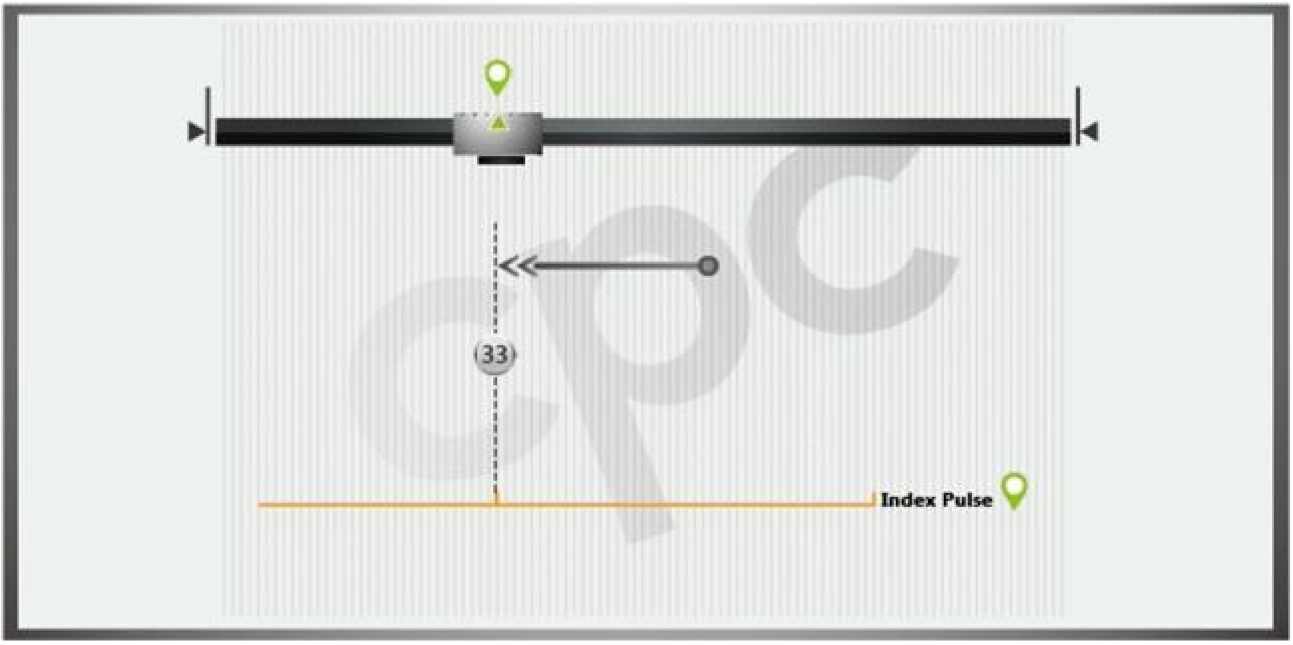
- If home switch is inactive, move in negative direction. If negative limit switch is then engaged, move in positive direction. If home switch is then engaged, move to falling edge of the home switch in negative direction.
- If home switch is active, move to the falling edge of the home switch in negative direction.
- If home switch is inactive, move in negative direction. If home switch is then engaged, move to falling edge of the home switch in negative direction.

## 8.5.8 Method 31 to 32: Reserved.

### 8.5.9 By First Pulse

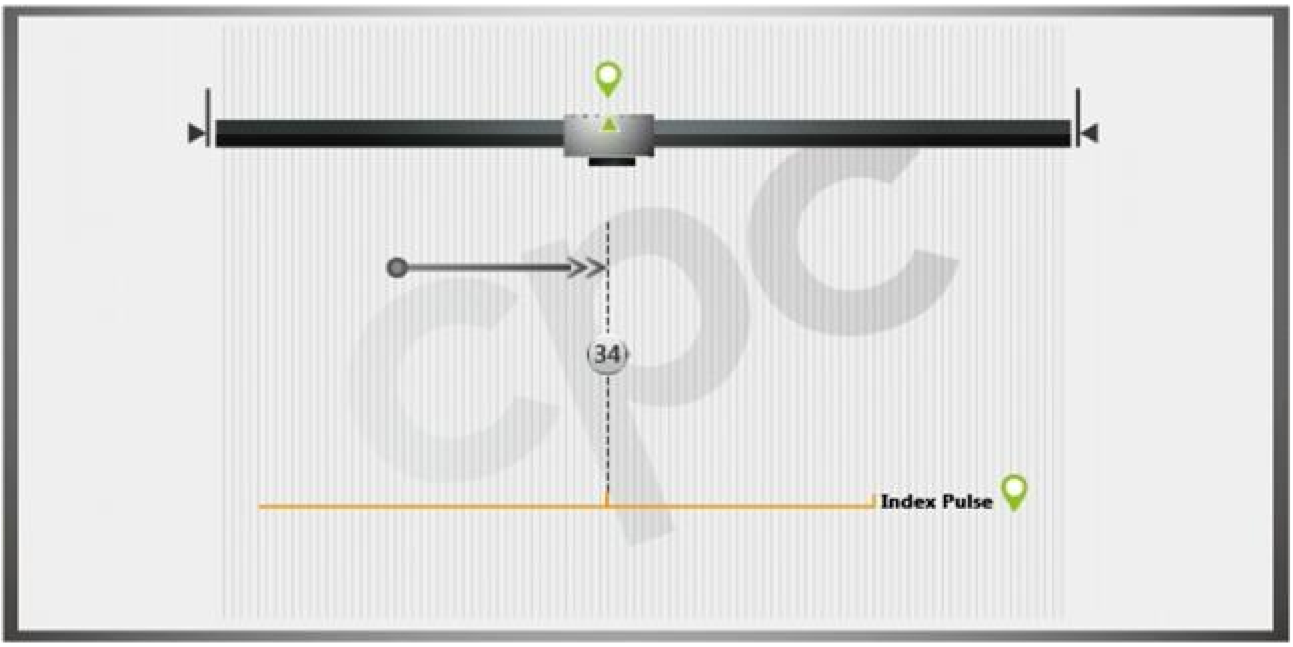
Method 33:

Home on the first pulse while moving in negative direction.



Method 34:

Home on the first pulse while moving in positive direction.



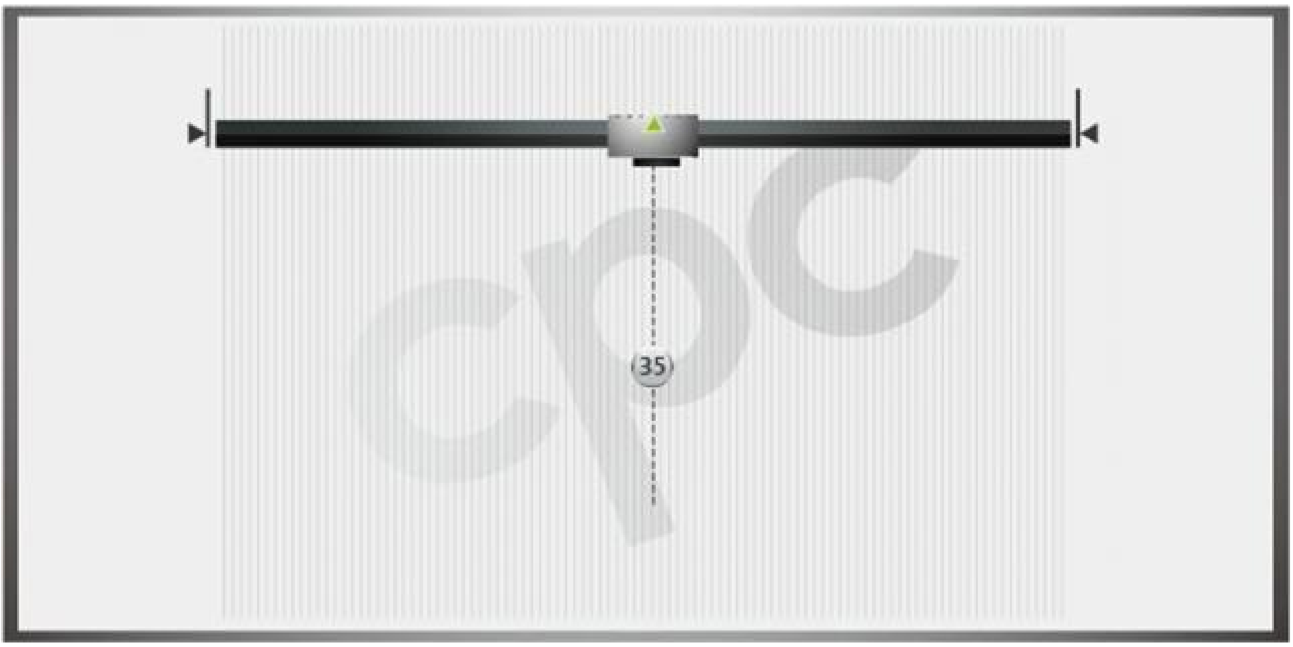
## 8.5.10 By Current Position

Method 35:

Home on the current position.

*Note

Method 37 = Method 35

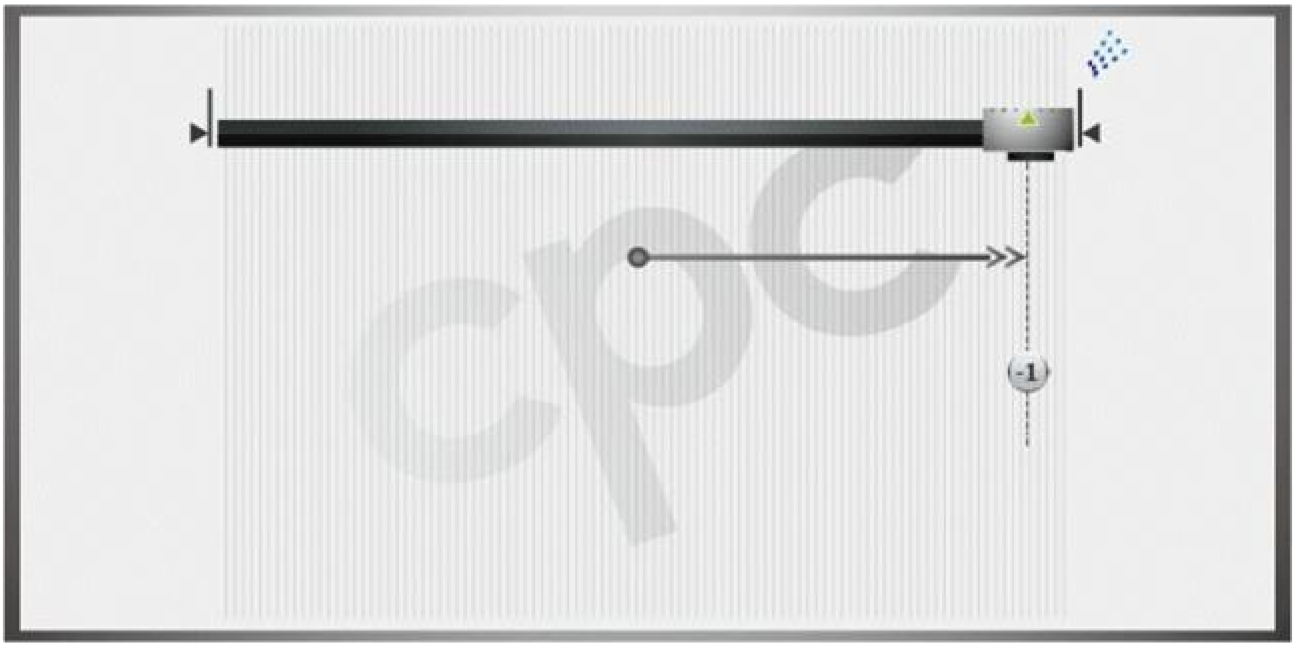


# 8.6 cpc Homing Methods

## 8.6.1 By Hard Stop

Method -1:

Home on the point of the positive hard stop.

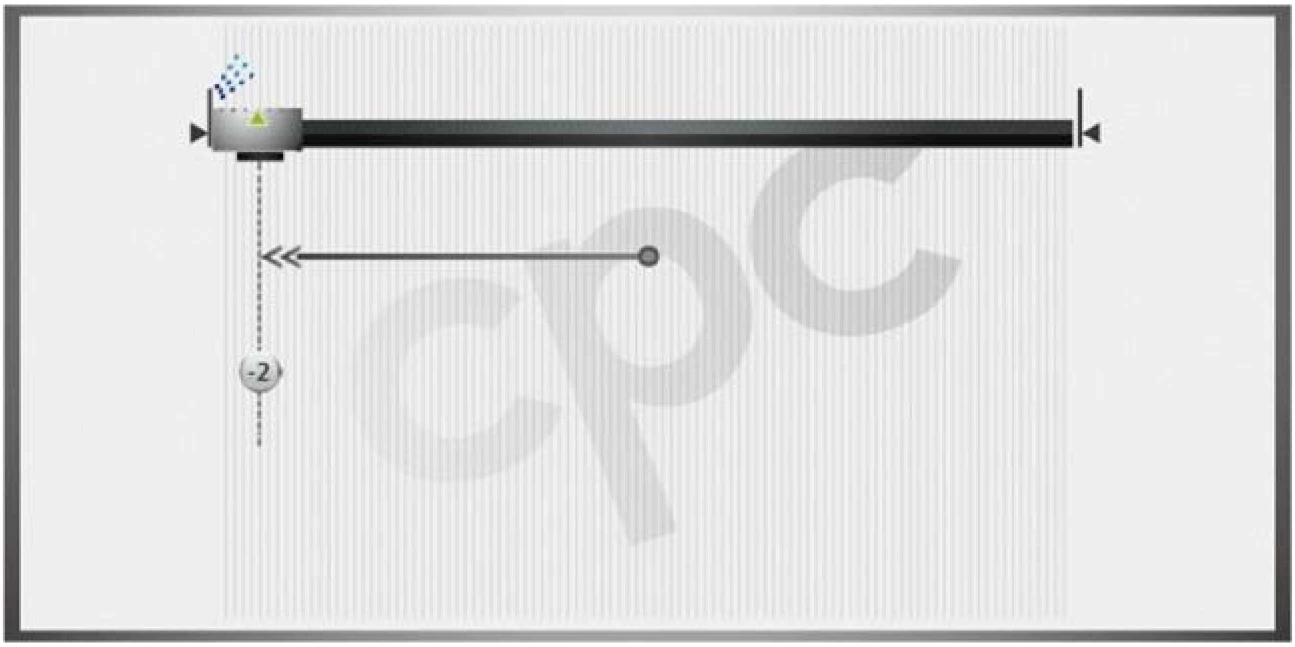


Homing process:

- Start with positive motion unconditionally until the positive hard stop is found.

Method -2:

Home on the point of the negative hard stop.



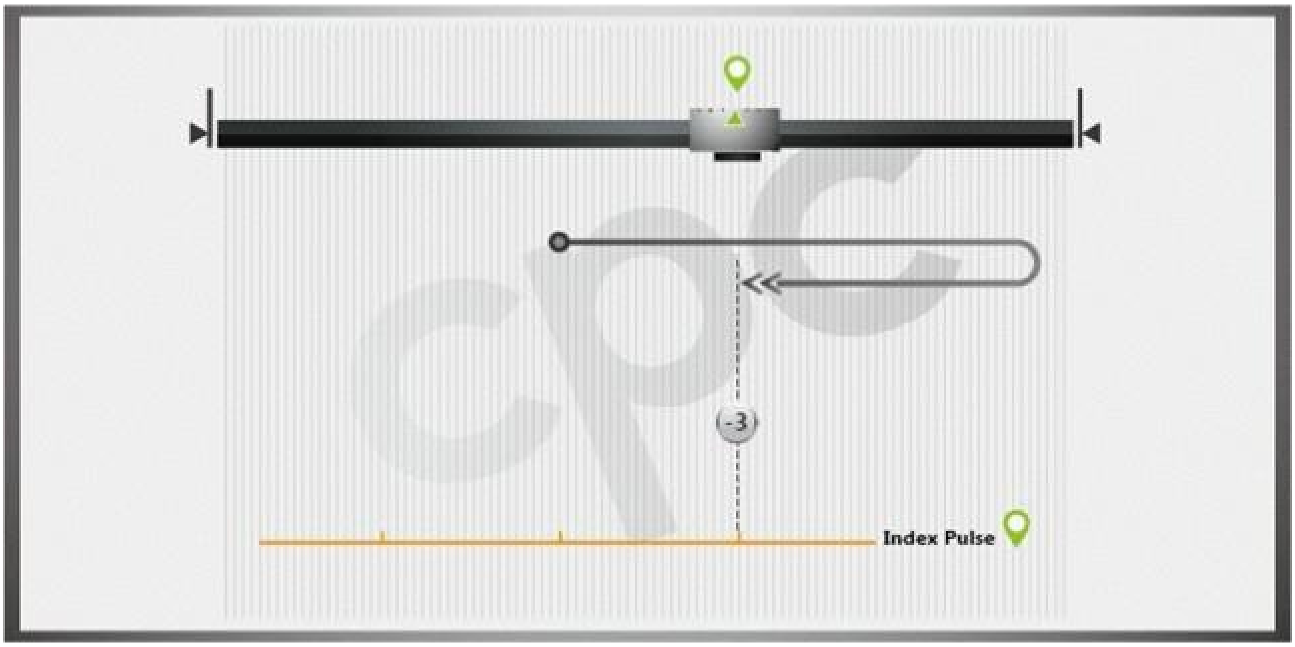
Homing process:

- Start with negative motion unconditionally until the negative hard stop is found.

### 8.6.2 By Hard Stop and Index

Method -3:

Home on the first index pulse after touching the positive hard stop.

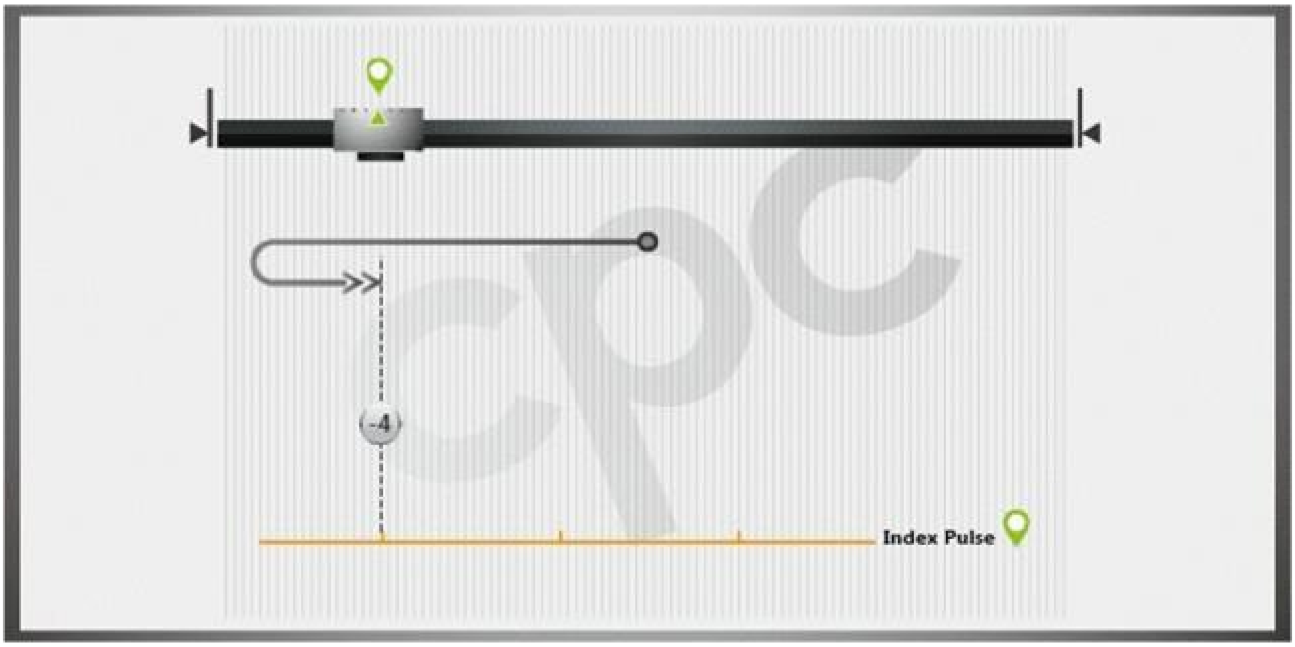


Homing process:

- Start with positive direction unconditionally. After touching the positive hard stop, move in negative direction until the first index is found.

Method -4:

Home on the first index pulse after touching the negative hard stop.



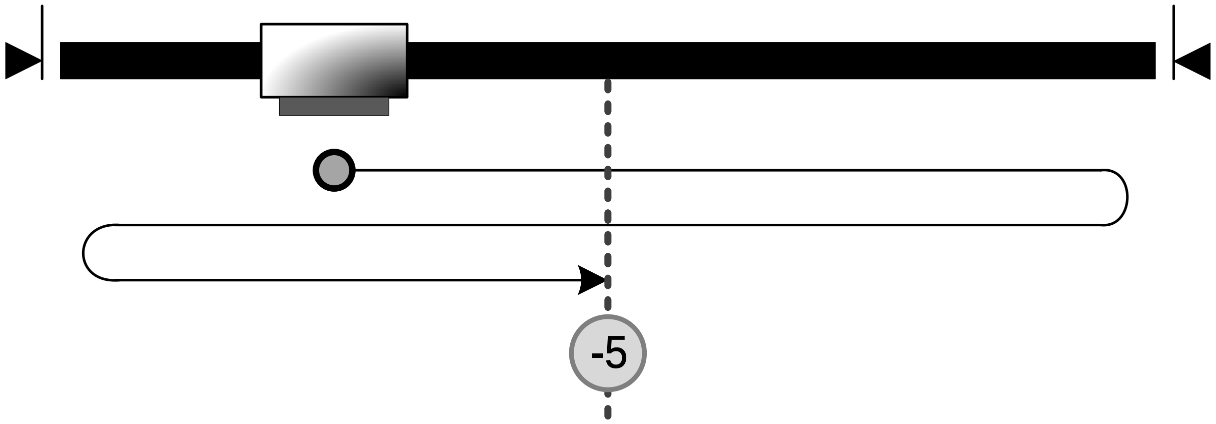
Homing process:

- Start with negative direction unconditionally. After touching the negative hard stop, move in positive direction until the first index is located.

## 8.6.3 By the middle of Hard Stop

Method -5:

Find middle between forward/backward hard stop, initial direction forward.

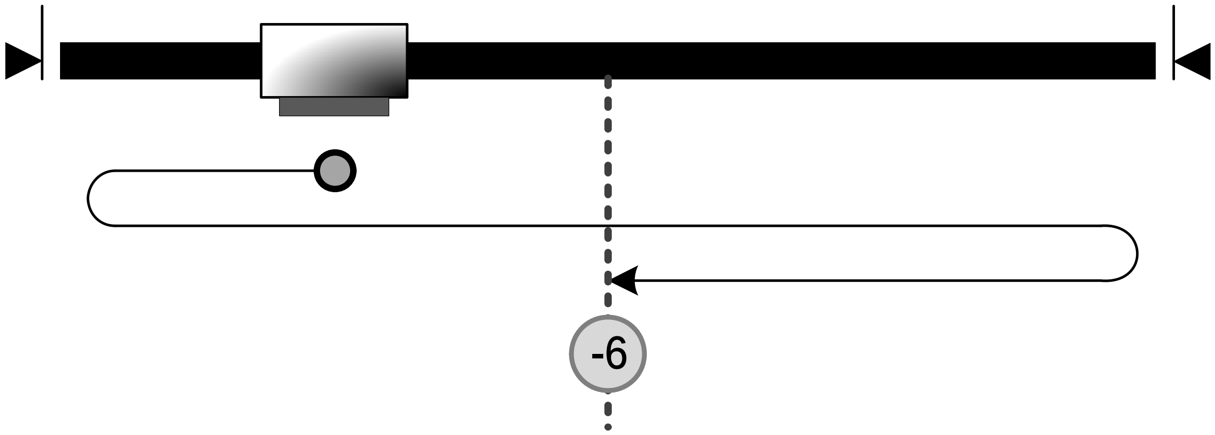


Homing process:

- Start with positive direction unconditionally. After touching the positive hard stop, move in negative direction until touching the negative hard stop, and then home on the middle of the two hard stops (found during homing).

Method -6:

Find middle between forward/backward hard stop, initial direction backward.



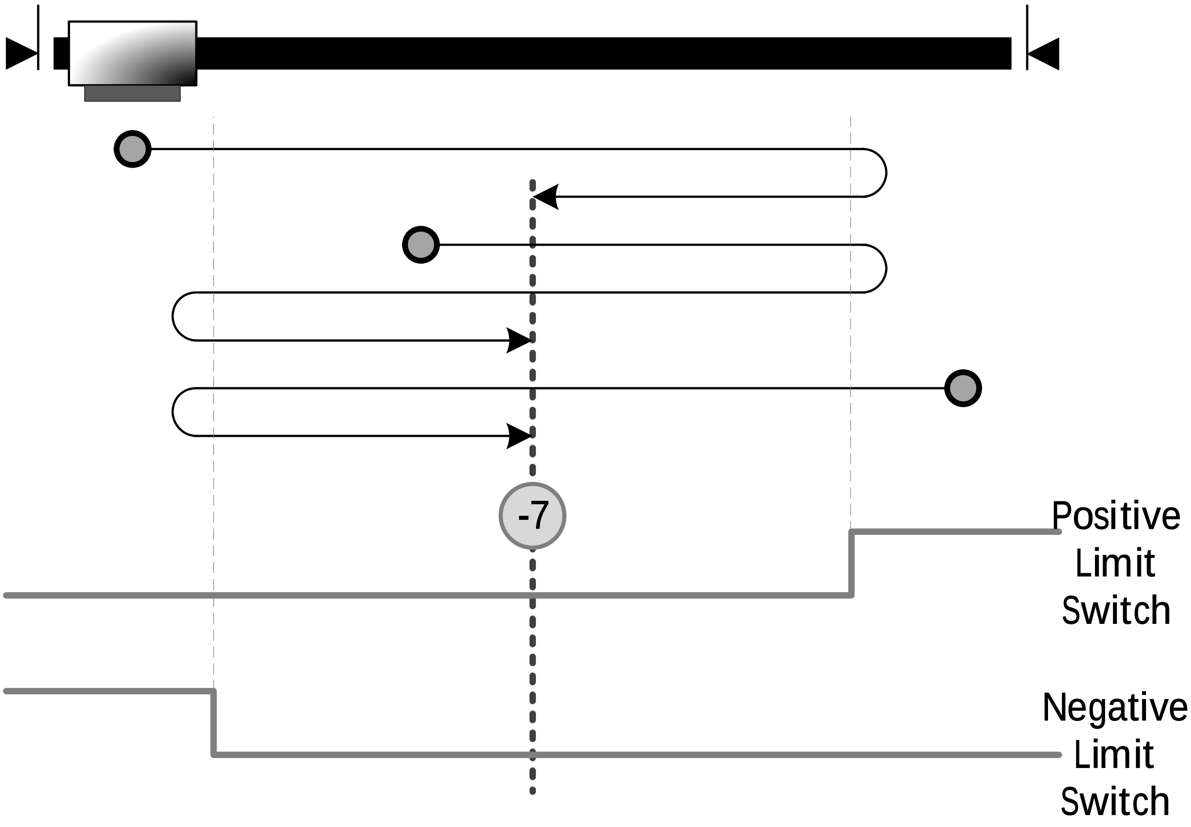
Homing process:

- Start with negative direction unconditionally. After touching the negative hard stop, move in positive direction until touching the negative hard stop, and then home on the middle of the two hard stops (found during homing).

## 8.6.4 By the middle of Limit Switch

Method -7:

Find middle between forward/backward limit switch falling edge, initial direction forward.

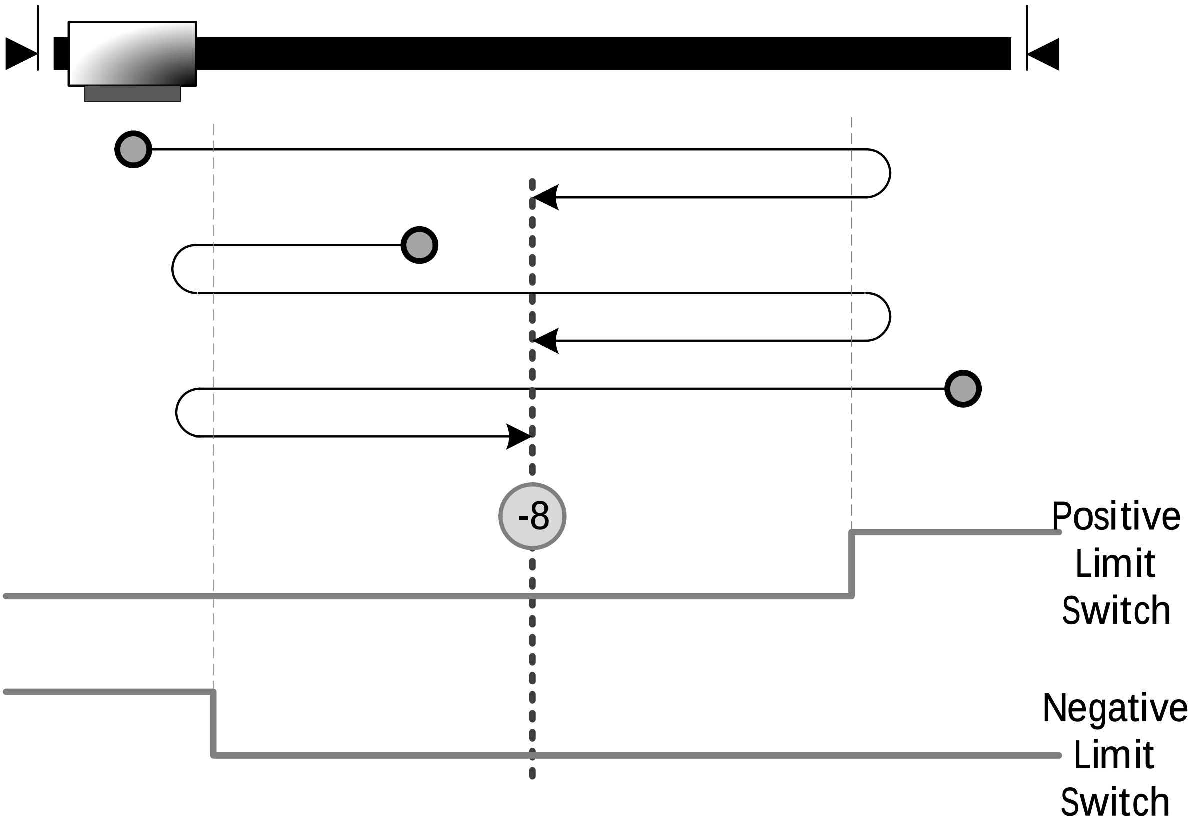


Homing process:

- If negative limit switch is active, start with positive direction until the positive limit switch is engaged; then, move in negative direction to find the middle of both switches.
- If negative limit switch is inactive, start with positive direction until the positive limit switch is engaged and then move in negative direction. If negative limit switch is engaged, move in positive direction until the middle of both switches is found.
- If the positive limit switch is active, start with negative direction until the negative limit switch is engaged; then, move in positive direction to find the middle of both switches.

Method -8:

Find middle between forward/backward limit switch falling edge, initial direction backward



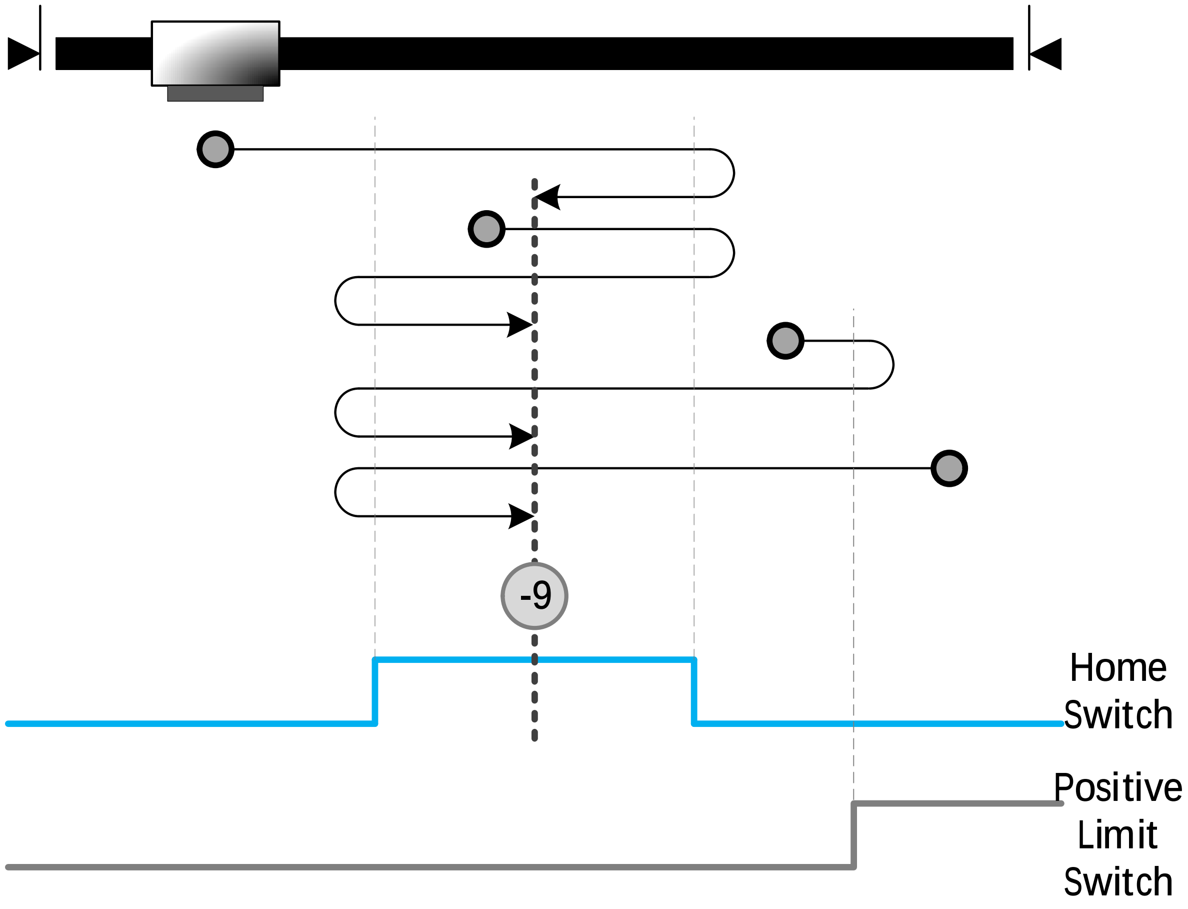
Homing process:

- If negative limit switch is active, start with positive direction until the positive limit switch is engaged; then, move in negative direction to find the middle of both switches.
- If negative limit switch is inactive, start with negative direction until the negative limit switch is engaged and then move in positive direction. If the positive limit switch is then engaged, move in negative direction until the middle of both switches is found.
- If the positive limit switch is active, start with negative direction until the negative limit switch is engaged; then, move in positive direction to find the middle of both switches.

### 8.6.5 By the middle of Home Switch

Method -9:

Find middle of home switch falling edge, initial direction forward, allow limit switch.



Homing process:

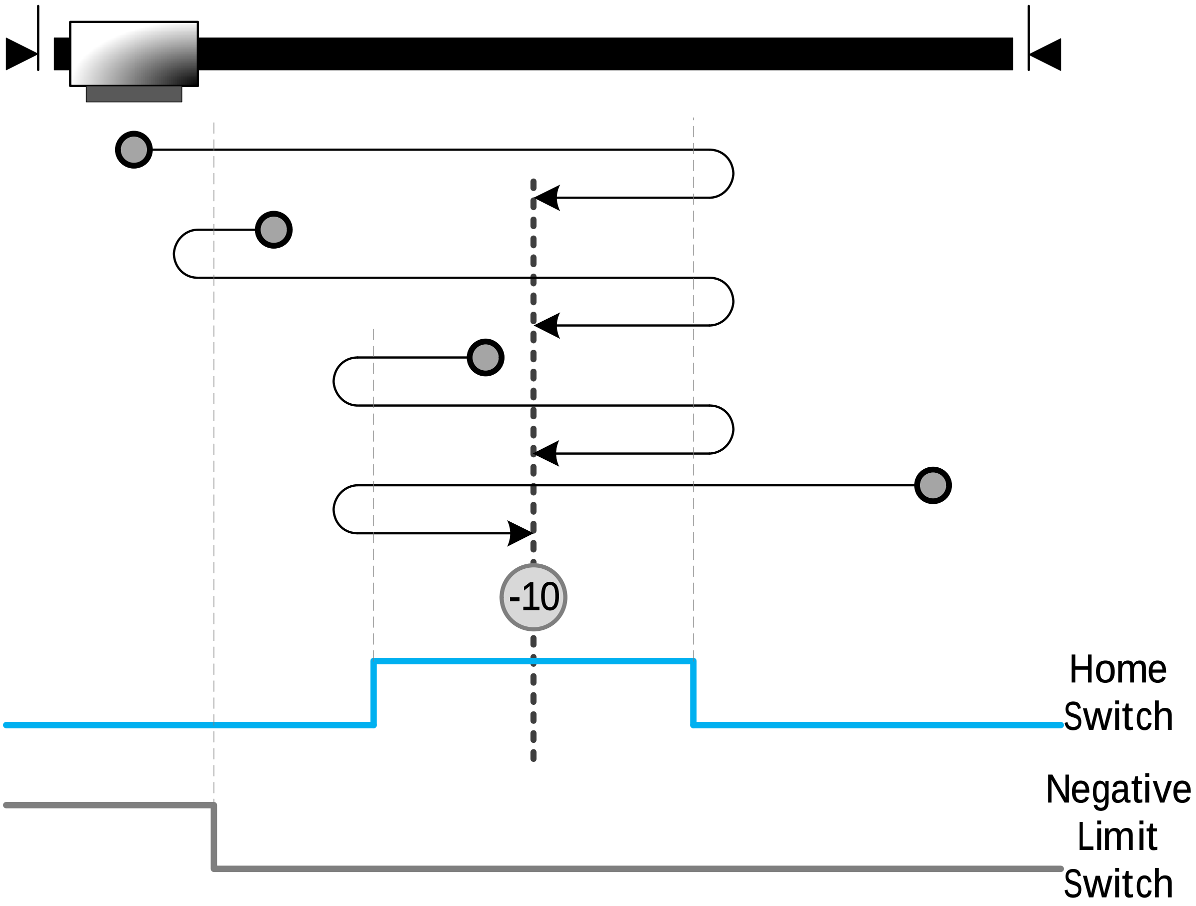
- If the home switch is inactive, start with positive direction until the home switch is engaged and disengaged, then move in negative direction until the middle of home switch is found.
- If the home switch is active, move in positive direction until the home switch is disengaged, then move in negative direction until the home switch is engaged and then disengaged. Reverse to move in positive direction and find the middle of home switch.
- If the home switch is inactive, move in positive direction. If the positive limit switch is then engaged, move in negative direction until the falling edge of the

home switch is engaged, then, move in positive direction until the middle of the home switch is found.

- If the positive limit switch is active, move in negative until the falling edge of home switch is met, then move in positive direction and find the middle of home switch.

Method -10:

Find middle of home switch falling edge, initial direction backward, allow limit switch



Homing process:

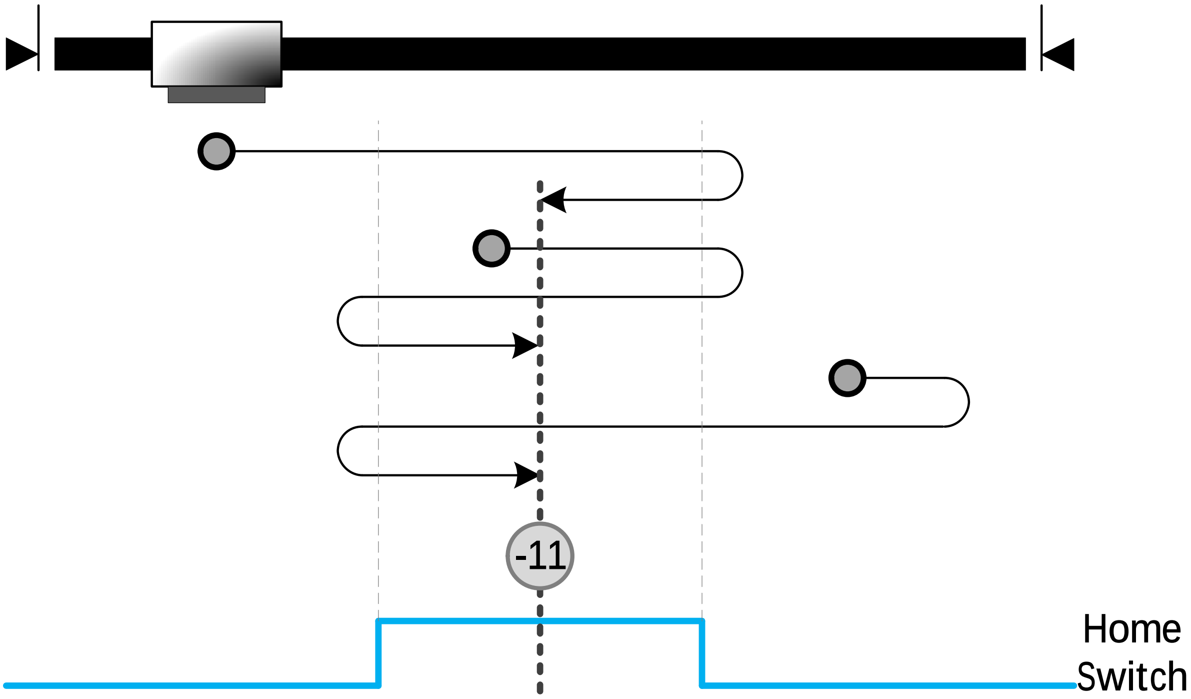
- If the negative limit switch is active, move in positive direction until the home switch is engaged and then disengaged. Then move in negative direction until the middle of the home switch is found.
- If the home switch is inactive, start with negative direction. If the negative limit switch is then engaged, move in right direction until the falling edge of

the home switch is met. Then move in negative direction to find the middle of the home switch.

- If home switch is active, move in negative direction. If the home switch is then disengaged, move in positive direction until the home switch is engaged and disengaged. Then, move in negative direction until the home switch is engaged again and find the middle of home switch.
- If the home switch is inactive, move in negative direction until the falling edge of home switch. If the home switch is then disengaged, move in positive direction until the middle of the home switch is found.

Method -11:

Find middle of home switch falling edge, initial direction forward, allow hard stop.

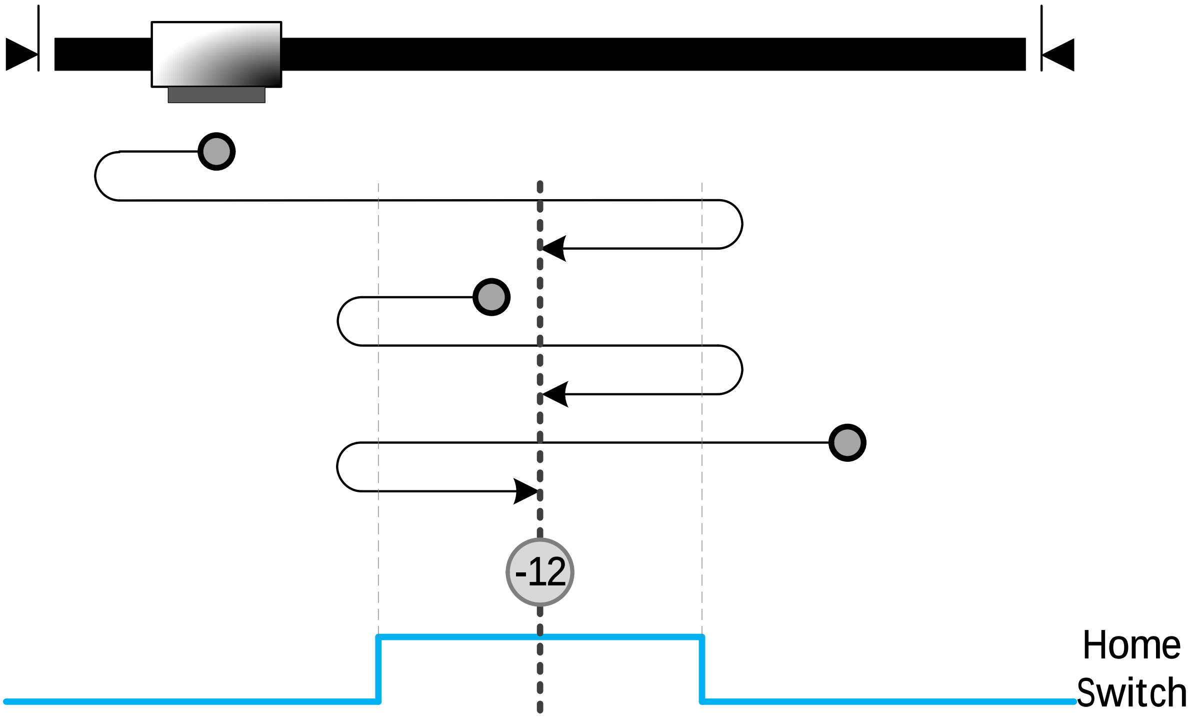


Homing process:

- Start with positive direction unconditionally. If home switch is inactive, keep going (and then if hard stop is met, reverse firstly) until meeting the rising and falling edge of the home switch, then reverse again to locate the middle of home switch.
- Start with positive direction unconditionally. If home switch is active, seek the positive side of home switch, then move in negative direction to meet the negative side of home switch, then reverse to find the middle of home switch.

Method -12:

Find middle of home switch falling edge, initial direction backward, allow hard stop.



Homing process:

- Start with negative direction unconditionally. If home switch is inactive, keep going (and then if hard stop is met, reverse firstly) until meeting the rising and falling edge of the home switch, then reverse again to locate the middle of home switch.
- Start with negative direction unconditionally. If home switch is active, go meet the negative side of home switch, then move in positive direction to meet the positive side of home switch, then reverse to find the middle of home switch.

# 9. Touch Probe Functionality

## 9.1 Object

### Object 0x60B8: Touch probe function

This object indicates the configured function of the touch probe.

- Object description

| | |
|---|---|
| Index | 60B8 |
| Name | Touch probe function |
| Object code | Variable |
| Data type | UINT16 |

- Entry description

| | |
|---|---|
| Sub-index | 0x00 |
| Access | RW |
| PDO mapping | Yes |
| Value range | UINT16 |
| Default value | 1 |
| Units | Count |

See value definition below.

- Value definition

| Bit | Value | Description |
|---|---|---|
| 0 | 0 | Switch off touch probe 1 |
| | 1 | Enable touch probe 1 |
| 1 | 0 | Trigger first event |
| | 1 | Continuous |
| 2, 3 | $00_b$ | Trigger with touch probe 1 input |
| | $01_b$ | Trigger with zero impulse signal or position encoder. |
| | $10_b$ | Reserved |
| | $11_b$ | Reserved. |
| 4 | 0 | Switch off sampling at positive edge of touch probe 1 |
| | 1 | Enable sampling at positive edge of touch probe 1. |
| 5 | 0 | Switch off sampling at negative edge of touch probe 1 |
| | 1 | Enable sampling at negative edge of touch probe 1. |
| 6, 7 | - | User-defined (e.g. for testing) |
| 8 | 0 | Switch off touch probe 2 |
| | 1 | Enable touch probe 2 |
| 9 | 0 | Trigger first event |
| | 1 | Continuous |
| 10, 11 | $00_b$ | Trigger with touch probe 2 input |
| | $01_b$ | Trigger with zero impulse signal or position encoder |
| | $10_b$ | Reserved |
| | $11_b$ | Reserved |
| 12 | 0 | Switch off sampling at positive edge of touch probe 2 |
| | 1 | Enable sampling at positive edge of touch probe 2 |
| 13 | 0 | Switch off sampling at negative edge of touch probe 2 |
| | 1 | Enable sampling at negative edge of touch probe 2 |
| 14, 15 | - | User-defined (e.g. for testing) |

## Object 0x60B9: Touch probe status

This object shows the status of the touch probe.

- Object description

| | |
|---|---|
| Index | 60B9 |
| Name | Touch probe status |
| Object code | Variable |
| Data type | UINT16 |

- Entry description

| | |
|---|---|
| Sub-index | 0x00 |
| Access | RO |
| PDO mapping | Yes |
| Value range | UINT16 |
| Default value | No |
| Units | No |

See value definition below.

- Value definition

| Bit | Value | Description |
|---|---|---|
| 0 | 0 | Touch probe 1 is switched off |
| | 1 | Touch probe 1 is enabled |
| 1 | 0 | Touch probe 1 no positive edge value stored |
| | 1 | Touch probe 1 positive edge position stored |
| 2 | 0 | Touch probe 1 no negative edge value stored |
| | 1 | Touch probe 1 negative edge position stored |
| 3~5 | 0 | Reserved |
| 6, 7 | - | User-defined (e.g. for testing) |
| 8 | 0 | Touch probe 2 is switched off |
| | 1 | Touch probe 2 is enabled |
| 9 | 0 | Touch probe 2 no positive edge value stored |
| | 1 | Touch probe 2 positive edge position stored |
| 10 | 0 | Touch probe 2 no negative edge value stored |
| | 1 | Touch probe 2 negative edge position stored |
| 11~13 | 0 | Reserved |

# Object 0x60BA: Touch probe 1 positive edge

This object shows the position value of the touch probe 1 at positive edge. The value is given in counts.

- Object description

| | |
|---|---|
| Index | 60BA |
| Name | Touch probe 1 positive edge |
| Object code | Variable |
| Data type | INT32 |

- Entry description

| | |
|---|---|
| Sub-index | 0x00 |
| Access | RO |
| PDO mapping | Yes |
| Value range | INT32 |
| Default value | No |
| Units | count |

## Object 0x60BB: Touch probe 1 negative edge

This object shows the position value of the touch probe 1 at negative edge. The value is given in counts.

- Object description

| | |
|---|---|
| Index | 60BB |
| Name | Touch probe 1 negative edge |
| Object code | Variable |
| Data type | INT32 |

- Entry description

| | |
|---|---|
| Sub-index | 0x00 |
| Access | RO |
| PDO mapping | Yes |
| Value range | INT32 |
| Default value | No |
| Units | count |

# Object 0x60BC: Touch probe 2 positive edge

This object shows the position value of the touch probe 2 at positive edge. The value is given in counts.

- Object description

| | |
|---|---|
| Index | 60BC |
| Name | Touch probe 2 positive edge |
| Object code | Variable |
| Data type | INT32 |

- Entry description

| | |
|---|---|
| Sub-index | 0x00 |
| Access | RO |
| PDO mapping | Yes |
| Value range | INT32 |
| Default value | No |
| Units | count |

## Object 0x60BD: Touch probe 2 negative edge

This object shows the position value of the touch probe 2 at negative edge. The value is given in counts.

- Object description

| | |
|---|---|
| Index | 60BD |
| Name | Touch probe 2 negative edge |
| Object code | Variable |
| Data type | INT32 |

- Entry description

| | |
|---|---|
| Sub-index | 0x00 |
| Access | RO |
| PDO mapping | Yes |
| Value range | INT32 |
| Default value | No |
| Units | count |

# 9.2 Touch Probe Edge Counter for Continuous Mode

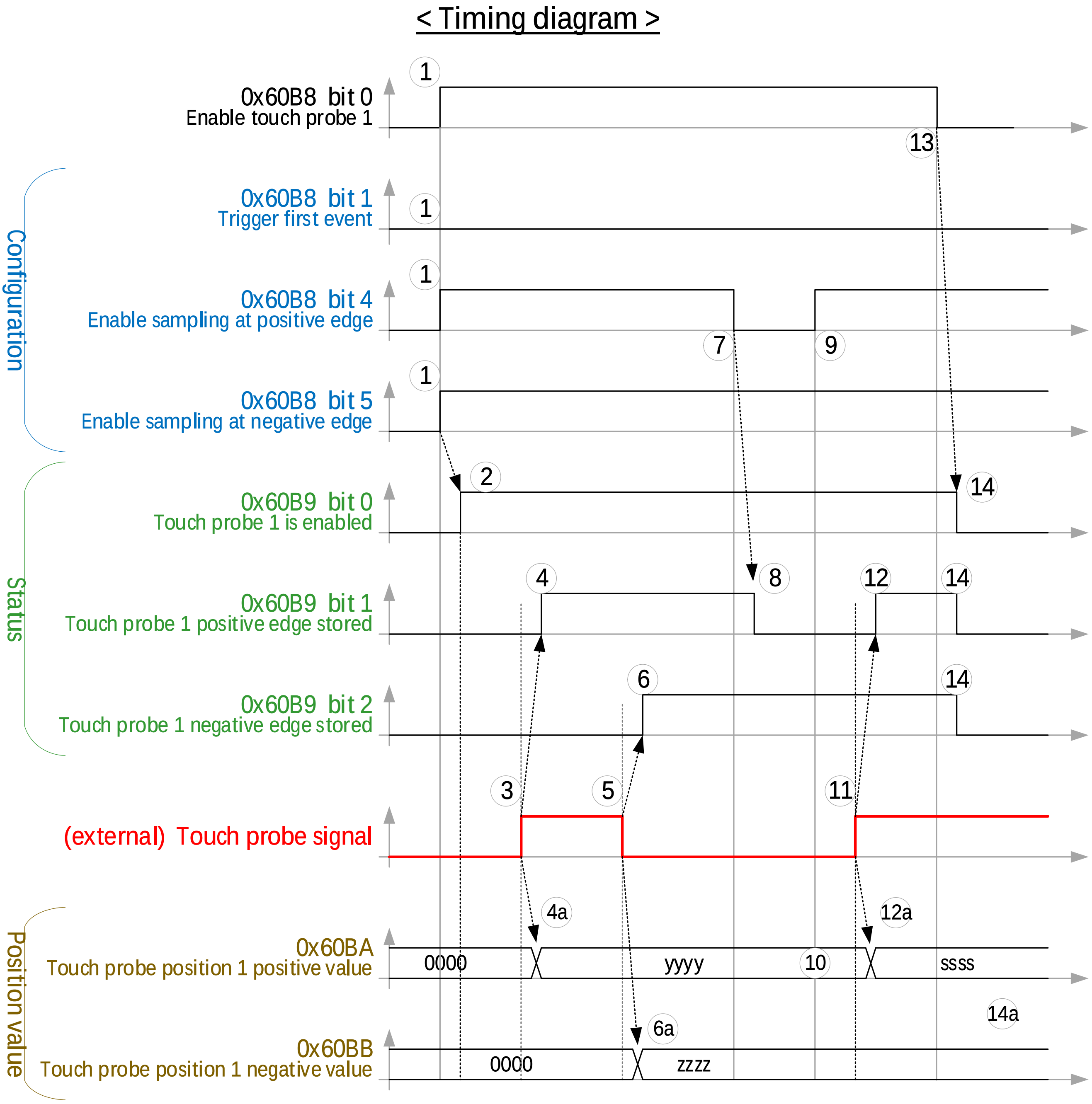
## 9.2.1 General information

For continuous touch probe mode [namely, object 0x60B8, bit 1 = 1 (continuous) or object 0x60B8, bit 9 = 1 (continuous)], a counter per touch probe channel is incremented on each touch probe event (i.e., rising or falling edge). Hence, the control device can check how many touch probe events have happened between the control cycles.

For each touch probe and each edge, a counter (namely, 0x60D5, 0x60D6, 0x60D7 and 0x60D8) is defined.

Please refer to the timing diagram example (ch. 9.2.2) for further explanation.

## 9.2.2 Timing Diagram—Example of Touch Probe Edge Counter for Continuous Mode



## < Explanation of the timing diagram >

Object 0x60B8: the configured function of the touch probe.

Object 0x60B9: the status of the touch probe.

| No. # | Touch probe behavior | |
|---|---|---|
| (1) | 0x60B8, bit 0 = 1 | Enable touch probe 1 |
| | 0x60B8, bit 1, 4, 5 | Configure and enable touch probe 1 positive and negative edge |
| (2) | è 0x60B9, bit 0 = 1 | Status “Touch probe 1 enabled” is set |
| (3) | External touch probe signal has positive edge | |
| (4) | è 0x60B9, bit 1 = 1 | Status “Touch probe 1 positive edge stored” is set |
| (4a) | è 0x60BA | Touch probe position 1 positive value is stored |
| (5) | External touch probe signal has negative edge | |
| (6) | è 0x60B9, bit 2 = 1 | Status “Touch probe 1 negative edge stored” is set |
| (6a) | è 0x60BB | Touch probe position 1 negative value is stored |
| (7) | 0x60B8, bit 4 = 0 | “Sampling positive edge” is disabled |
| (8) | è 0x60B9, bit 0 = 0 | Status “Touch probe 1 positive edge stored” is reset |
| (8a) | è 0x60BA | Touch probe position 1 positive value is not changed |
| (9) | 0x60B8, bit 4 = 1 | “Sampling positive edge” is enabled |
| (10) | è 0x60BA | Touch probe position 1 position value is still not changed |
| (11) | External touch probe signal has positive edge | |
| (12) | è 0x60B9, bit 1 = 1 | Status “Touch probe 1 positive edge stored” is set |
| (12a) | è 0x60BA | Touch probe position 1 positive value is stored |
| (13) | 0x60B8, bit 0 = 0 | Touch probe 1 is disabled |
| (14) | è 0x60B9, bit 0,1,2 = 0 | Status bits are reset |
| (14a) | è 0x60BA, 0x60BB | Touch probe position 1 positive/negative value are not changed |

## 9.2.3 Object

### Object 0x60D5: Touch probe 1 positive edge counter

- This object is a continuous counter which is incremented with each positive edge (rising edge) at touch probe 1.
- This counter is only valid when the “touch probe 1 input” is enabled (i.e., 0x60B8, bit 0 = 1).
- For single measuring, its range is 0 to 1.
  For continuous measuring, its range is an unsigned 16-bit value with overflow.

- Object description

| | |
|---|---|
| Index | 60D5 |
| Name | Touch probe 1 positive edge counter |
| Object code | Variable |
| Data type | UINT16 |

- Entry description

| | |
|---|---|
| Sub-index | 0x00 |
| Access | RO |
| PDO mapping | Yes |
| Value range | UINT16 |
| Default value | No |
| Units | No |

## Object 0x60D6: Touch probe 1 negative edge counter

- This object is a continuous counter which is incremented with each negative edge (falling edge) at touch probe 1.
- This counter is only valid when the "touch probe 1 input" is enabled (i.e., 0x60B8, bit 0 = 1).
- For single measuring, its range is 0 to 1.
  For continuous measuring, its range is an unsigned 16-bit value with overflow.

l Object description

| | |
|---|---|
| Index | 60D6 |
| Name | Touch probe 1 negative edge counter |
| Object code | Variable |
| Data type | UINT16 |

l Entry description

| | |
|---|---|
| Sub-index | 0x00 |
| Access | RO |
| PDO mapping | Yes |
| Value range | UINT16 |
| Default value | No |
| Units | No |

## Object 0x60D7: Touch probe 2 positive edge counter

- This object is a continuous counter which is incremented with each positive edge (rising edge) at touch probe 2.
- This counter is only valid when the “touch probe 1 input” is enabled (i.e., 0x60B8, bit 0 = 1).
- For single measuring, its range is 0 to 1.
  For continuous measuring, its range is an unsigned 16-bit value with overflow.

ı Object description

| | |
|---|---|
| Index | 60D7 |
| Name | Touch probe 2 positive edge counter |
| Object code | Variable |
| Data type | UINT16 |

ı Entry description

| | |
|---|---|
| Sub-index | 0x00 |
| Access | RO |
| PDO mapping | Yes |
| Value range | UINT16 |
| Default value | No |
| Units | No |

## Object 0x60D8: Touch probe 2 negative edge counter

- This object is a continuous counter which is incremented with each negative edge (falling edge) at touch probe 2.
- This counter is only valid when the “touch probe 1 input” is enabled (i.e., 0x60B8, bit 0 = 1).
- For single measuring, its range is 0 to 1.
  For continuous measuring, its range is an unsigned 16-bit value with overflow.

ı Object description

| | |
|---|---|
| Index | 60D8 |
| Name | Touch probe 2 negative edge counter |
| Object code | Variable |
| Data type | UINT16 |

ı Entry description

| | |
|---|---|
| Sub-index | 0x00 |
| Access | RO |
| PDO mapping | Yes |
| Value range | UINT16 |
| Default value | No |
| Units | No |

# 10. Profile Velocity Mode

## 10.1 General Information

The profile velocity mode includes the following sub-functions:

- Demand value input via trajectory generator
- Velocity capture using the position sensor or velocity sensor
- Velocity control function with the appropriate input and output signals
- Monitoring of the profile velocity using a window function
- Monitoring of the velocity actual value using a threshold

The input parameters are:

- Profile velocity
- Profile acceleration
- Profile deceleration
- Quick stop deceleration

# 10.2 Structure of Controlword and Statusword

## 10.2.1 Controlword of the Profile Velocity Mode.

| Bit | Function |
|---|---|
| 0~3 | *See chapter 4.3.1.1 - Device Control/Controlword structure |
| 4~6 | Reserved |
| 7 | *See chapter 4.3.1.1 - Device Control/Controlword structure |
| 8 | Halt |
| 9~15 | *See chapter 4.3.1.1 - Device Control/Controlword structure |

Definition of bit 8

| Bit | Name | Value | Description |
|---|---|---|---|
| 8 | Halt | 0 | Execute or continue the motion. |
| | | 1 | Stop the motor according to the halt option code (0x605D). |

## 10.2.2 Statusword of the Profile Velocity Mode.

| Bit | Function |
|---|---|
| 0~9 | *See chapter 4.3.2.1 - Device Control/Statusword structure |
| 10 | Target reached |
| 11 | *See chapter 4.3.2.1 - Device Control/Statusword structure |
| 12 | Zero speed |
| 13 | Reserved |
| 14~15 | *See chapter 4.3.2.1 - Device Control/Statusword structure |

- Definition of bit 10 and 12

| Bit | Name | Value | Description |
|---|---|---|---|
| 10 | Target reached | 0 | Halt = 0: Target not reached.<br>Halt = 1: the motor decelerates. |
| | | 1 | Halt = 0: Target is reached.<br>Halt = 1: the motor decelerates to 0 velocity. |
| 12 | Zero speed | 0 | Present speed is above the velocity threshold. |
| | | 1 | Present speed is less than or equal to the velocity threshold. |

# 10.3 Functional Description

- The actual velocity is acquired via the position encoder.

- Target reached (bit 10 in statusword)

  While the difference between the target velocity and the velocity actual value is within the velocity window longer than the velocity window time, the target reached bit (bit 10) will be set to 1.

- Velocity threshold (bit 12 in statusword)
  - = 0

    When the actual velocity is greater than the velocity threshold longer than the threshold time.
  - = 1

    When the actual velocity is less than or equal to the velocity threshold. This situation will be regarded as the motor is not moving.

# 10.4 Objects

## Object 0x606B: Velocity demand value

The value of velocity demand as generated by the trajectory generator.

- Object description

| | |
|---|---|
| Index | 606B |
| Name | Velocity demand value |
| Object code | Variable |
| Data type | INT32 |

- Entry description

| | |
|---|---|
| Sub-index | 0x00 |
| Access | RO |
| PDO mapping | No |
| Value range | INT32 |
| Default value | No |
| Units | count/s |

## Object 0x606C: Velocity actual value

This object gives the actual velocity value acquired from the velocity sensor or the position encoder.

- Object description

| | |
|---|---|
| Index | 606C |
| Name | Velocity actual value |
| Object code | Variable |
| Data type | INT32 |

- Entry description

| | |
|---|---|
| Sub-index | 0x00 |
| Access | RO |
| PDO mapping | Yes |
| Value range | INT32 |
| Default value | 0 |
| Units | count/s |

## Object 0x606D: Velocity window

This object shows the configured velocity window.

- Object description

| | |
|---|---|
| Index | 606C |
| Name | Velocity actual value |
| Object code | Variable |
| Data type | UINT16 |

- Entry description

| | |
|---|---|
| Sub-index | 0x00 |
| Access | RW |
| PDO mapping | Yes |
| Value range | UINT16 |
| Default value | 0 |
| Units | count/s |

# Object 0x606E: Velocity window time

This object shows the configured velocity window time in ms.

- Object description

| | |
|---|---|
| Index | 606E |
| Name | Velocity actual value |
| Object code | Variable |
| Data type | UINT16 |

- Entry description

| | |
|---|---|
| Sub-index | 0x00 |
| Access | RW |
| PDO mapping | Yes |
| Value range | UINT16 |
| Default value | 0 |
| Units | ms |

## Object 0x606F: Velocity threshold

This object shows the configured velocity threshold.

- Object description

| | |
|---|---|
| Index | 606F |
| Name | Velocity threshold |
| Object code | Variable |
| Data type | UINT16 |

- Entry description

| | |
|---|---|
| Sub-index | 0x00 |
| Access | RW |
| PDO mapping | Yes |
| Value range | UINT16 |
| Default value | 0 |
| Units | count/s |

## Object 0x6070: Velocity threshold time

This object shows the configured velocity threshold time.

When the present speed is less than or equal to the velocity threshold (0x606F) longer than the velocity threshold time (0x6070), the motor will be regarded as stationary.

- Object description

| | |
|---|---|
| Index | 6070 |
| Name | Velocity threshold time |
| Object code | Variable |
| Data type | UINT16 |

- Entry description

| | |
|---|---|
| Sub-index | 0x00 |
| Access | RW |
| PDO mapping | Yes |
| Value range | UINT16 |
| Default value | 0 |
| Units | ms |

# 11. Profile Torque Mode

## 11.1 General Information

The profile torque mode allows a host (external) control system (i.e., closed-loop speed controller, open-loop transmission force controller) to transmit the target torque value (0x6071), which is processed through the trajectory generator. The torque slope parameter is required.

When the torque command is changed, the motor torque ramps to the new value at the rate programmed in “torque slope (index 0x6087).” If the control device switches the controlword bit 8 (Halt) from 0 to 1, the driver ramps the effort output to zero; If bit 8 is set from 1 to 0, the driver ramps the effort output up to the target torque.

All the definitions refer to the rotary motor. For linear motors, all the “torque” objects refer to a “force” instead.

Please note that the current limits (including continuous current and peak current) need to suit the motor rated current.

The inputs to the torque control are as follows:

- Target torque (0x6071)
- Torque slope (0x6087)
- Controlword
- Max current (0x6073)
- Motor rated torque (0x6076)
- Motor rated current (0x6075)

# 11.2 Structure of Controlword and Statusword

## 11.2.1 Controlword of the Profile Torque Mode

| Bit | Function |
|---|---|
| 0~3 | *See chapter 4.3.1.1 - Device Control/Controlword structure |
| 4~6 | Reserved |
| 7 | *See chapter 4.3.1.1 - Device Control/Controlword structure |
| 8 | Halt |
| 9~15 | *See chapter 4.3.1.1 - Device Control/Controlword structure |

Definition of bit 8

| Bit | Name | Value | Description |
|---|---|---|---|
| 8 | Halt | 0 | Execute or continue the motion. |
| | | 1 | Stop the motor according to the halt option code (0x605D). |

## 11.2.2 Statusword of the Profile Torque Mode

| Bit | Function |
|---|---|
| 0~9 | *See chapter 4.3.2.1 - Device Control/Statusword structure |
| 10 | Target reached |
| 11 | *See chapter 4.3.2.1 - Device Control/Statusword structure |
| 12~13 | Reserved |
| 14~15 | *See chapter 4.3.2.1 - Device Control/Statusword structure |

- Definition of bit 10

| Bit | Name | Value | Description |
|---|---|---|---|
| 10 | Target reached | 0 | Halt = 0: Target torque not reached.<br>Halt = 1: the motor decelerates. |
| | | 1 | Halt = 0: Target torque is reached.<br>Halt = 1: the motor decelerates to 0 velocity. |

# 11.3 Objects

## Object 0x6071: Target torque

This parameter is the input value for the torque controller in profile torque mode.

- Object description

| | |
|---|---|
| Index | 6071 |
| Name | Target torque |
| Object code | Variable |
| Data type | UINT16 |

- Entry description

| | |
|---|---|
| Sub-index | 0x00 |
| Access | RW |
| PDO mapping | Yes |
| Value range | UINT16 |
| Default value | 0 |
| Units | 0.1% |

# Object 0x6073: Max current

The motor peak current.

This value shows the maximum permissible torque creating current in the motor.

- Object description

| | |
|---|---|
| Index | 6073 |
| Name | Max current |
| Object code | Variable |
| Data type | UINT16 |

- Entry description

| | |
|---|---|
| Sub-index | 0x00 |
| Access | RW |
| PDO mapping | Yes |
| Value range | UINT16 |
| Default value | 0 |
| Units | 0.1% |

## Object 0x6074: Torque demand

This object provides the output value of the trajectory generator.

- Object description

| | |
|---|---|
| Index | 6074 |
| Name | Torque demand |
| Object code | Variable |
| Data type | UINT16 |

- Entry description

| | |
|---|---|
| Sub-index | 0x00 |
| Access | RO |
| PDO mapping | Yes |
| Value range | INT16 |
| Default value | None |
| Units | 0.1% |

## Object 0x6075: Motor rated current

The motor continuous current.

This value is acquired from the motor name-plate. All relative current data refers to this value.

The unit is in mA.

- Object description

| | |
|---|---|
| Index | 6075 |
| Name | Motor rated current |
| Object code | Variable |
| Data type | UINT32 |

- Entry description

| | |
|---|---|
| Sub-index | 0x00 |
| Access | RW |
| PDO mapping | No |
| Value range | UINT32 |
| Default value | 0 |
| Units | mA |

# Object 0x6076: Motor rated torque

This value is acquired from the motor name-plate. All relative torque data refer to this value.

Note:

For linear motors, the name of this object is not changed, but the motor rated force is entered in multiples of mN (milli Newton).

- Object description

| | |
|---|---|
| Index | 6076 |
| Name | Motor rated torque |
| Object code | Variable |
| Data type | UINT32 |

- Entry description

| | |
|---|---|
| Sub-index | 0x00 |
| Access | RW |
| PDO mapping | No |
| Value range | UINT32 |
| Default value | 0 |
| Units | mNm (milli Newton metre) |

# Object 0x6077: Torque actual value

This value shows the actual value of the torque. It corresponds to the instantaneous torque in the motor.

Note – regarding object 0x6077 (Torque actual value ) and 0x6078 (Current actual value)

1. The cpc drive assumes 100% current = 100% torque.
2. The cpc drive uses object 0x6078. Nevertheless, the value of 0x6077 and 0x6078 is the same.

- Object description

| | |
|---|---|
| Index | 6077 |
| Name | Torque actual value |
| Object code | Variable |
| Data type | INT16 |

- Entry description

| | |
|---|---|
| Sub-index | 0x00 |
| Access | RO |
| PDO mapping | Yes |
| Value range | INT16 |
| Default value | 0 |
| Units | 0.1% |

# Object 0x6078: Current actual value

This object shows the actual value of the current. It corresponds to the current in the motor.

Note – regarding object 0x6077 (Torque actual value ) and 0x6078 (Current actual value)

1. The cpc drive assumes 100% current = 100% torque.
2. The cpc drive uses object 0x6078. Nevertheless, the value of 0x6077 and 0x6078 is the same.

- Object description

| | |
|---|---|
| Index | 6078 |
| Name | Current actual value |
| Object code | Variable |
| Data type | INT16 |

- Entry description

| | |
|---|---|
| Sub-index | 0x00 |
| Access | RO |
| PDO mapping | Yes |
| Value range | INT16 |
| Default value | 0 |
| Units | 0.1% |

# Object 0x6079: DC link circuit voltage

This object shows the DC link circuit voltage at the driver immediately.

- Object description

| | |
|---|---|
| Index | 6079 |
| Name | DC link circuit voltage |
| Object code | Variable |
| Data type | UINT32 |

- Entry description

| | |
|---|---|
| Sub-index | 0x00 |
| Access | RO |
| PDO mapping | No |
| Value range | UINT32 |
| Default value | 0 |
| Units | mV |

## Object 0x6087: Torque slope

This object describes the rate of change of torque in units of per thousand of rated torque per second.

- Object description

| | |
|---|---|
| Index | 6087 |
| Name | Torque slope |
| Object code | Variable |
| Data type | UINT32 |

- Entry description

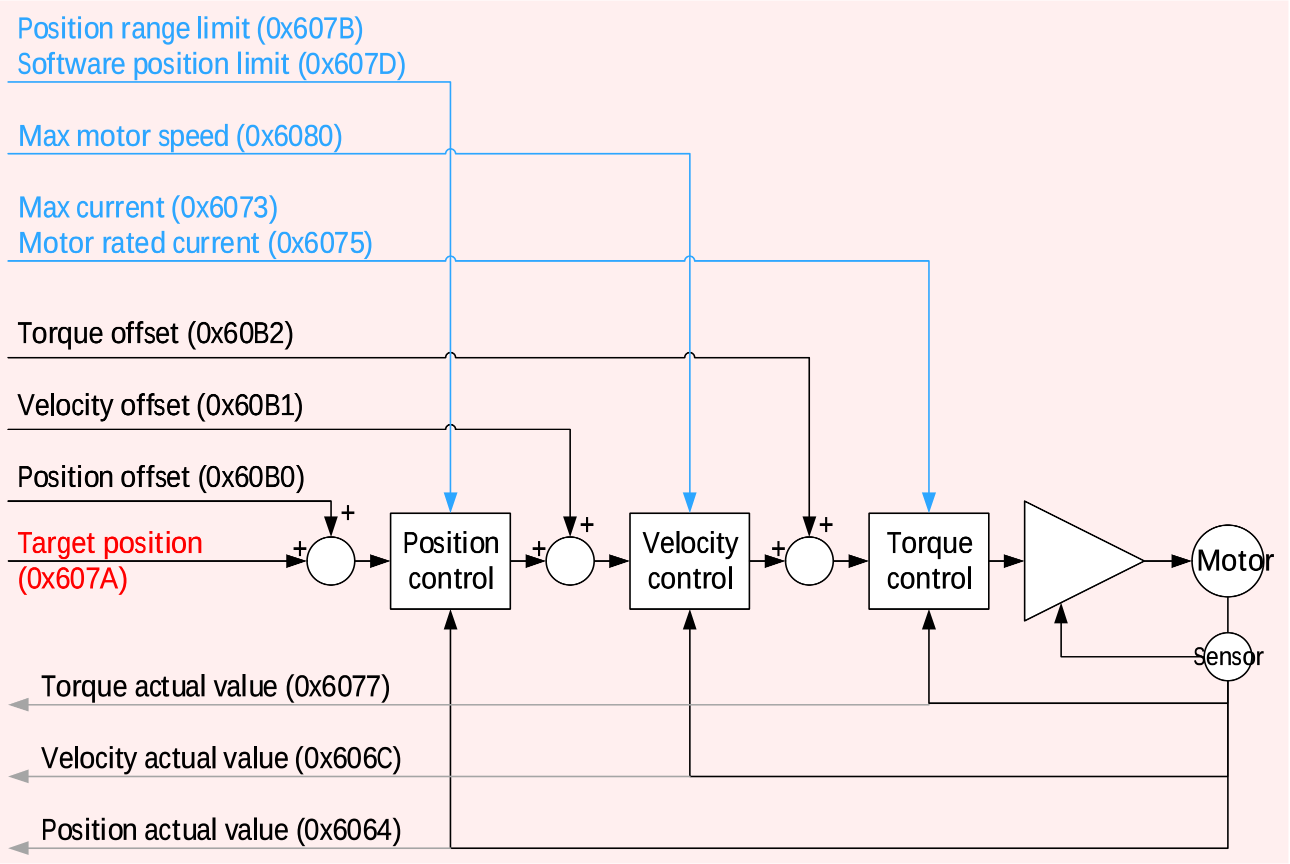
| | |
|---|---|
| Sub-index | 0x00 |
| Access | RW |
| PDO mapping | Yes |
| Value range | UINT32 |
| Default value | 100,000 |
| Units | 0.1%/s |

# 12. Cyclic Synchronous Position Mode

## 12.1 General Information

The feature of cyclic synchronous position mode (csp mode) is that the trajectory generator is located in the control device, not in the driver.
The overall structure of csp mode is as follows:

<Overview of cyclic synchronous position mode>



- The control device provides the target position (0x607A, which is interpreted as absolute value in this mode) to the driver and optionally, additive velocity offset as well as torque offset to allow for velocity and/or torque feedforward control.

- The driver executes the control functions (i.e., position control, velocity control, and torque control) and gives the control device the actual values (i.e., actual values of position, velocity, and torque) measured by sensors.
  - The driver external control function:

    The performance of the control function is influenced by control parameters (e.g., limit functions) which are externally applicable.

## 12.2 Functional Description

Inputs to the drive's control function:

- Target position (0x607A)
- Position offset (0x60B0)

  (optional; to be added to the target position to allow two examples to set up the position).
- Velocity offset (6x60B1)

  (optional; used for feedforward control)
- Torque offset (0x60B2)

  (optional; used for feedforward control)
- (In cascaded structure where position control is followed by a velocity or torque control) The output of the position control loop is used as an input for a further calculation in the driver.
- Position Limit functions, such as Position range limit (0X607B) and Software position limit (0X607D).
- Following error window (0x6065)
- Following error time out (0x6066)
- Quick stop deceleration (0x6085)
- Quick stop option code (0x605A)
- Interpolation time period (0x60C2)

  Defines the time duration between two updates of the target position and/or additive position and is used for intercycle interpolation.
- Position range limit (0x607B)

- Software position limit (0x607D)
- Max motor speed (0x6080)
- Max current (0x6073)
- Motor rated current (0x6075)

Outputs of the drive's control function:

- Position actual value (0x6064)

  It is used as a compulsory output to the control device.
- Velocity actual value (0x606C)
- Torque actual value (0x6077)
- Following error actual value (0x60F4)

  It can be used as an extra parameter.

# 12.3 Structure of Controlword and Statusword

The csp mode does not use mode-specific controlword bit.
It uses 2 bits in statusword for mode-specific purpose.

## 12.3.1 Statusword of the Cyclic Synchronous Position Mode

| Bit | Function |
|---|---|
| 0~9 | *See chapter 4.3.2.1 - Device Control/Statusword structure |
| 10 | Reserved |
| 11 | *See chapter 4.3.2.1 - Device Control/Statusword structure |
| 12 | Drive follows the command value. |
| 13 | Following error |
| 14~15 | *See chapter 4.3.2.1 - Device Control/Statusword structure |

- Definition of bit 12, and 13

| Bit | Name | Value | Description |
|---|---|---|---|
| 12 | Drive follows the command value. | 0 | The driver doesn't follow the command value – "Target position" is ignored. |
|  |  | 1 | The driver follows the command value – "Target position" is used as input to the position control loop. |
| 13 | Following error | 0 | No following error. |
|  |  | 1 | Following error occurs. |

# 12.4 Object

## Object 0x60B0: Position offset

This object shows the value of the position offset.

- Object description

| | |
|---|---|
| Index | 60B0 |
| Name | Position offset |
| Object code | Variable |
| Data type | Integer32 |

- Entry description

| | |
|---|---|
| Sub-index | 0x00 |
| Access | RW |
| PDO mapping | Yes |
| Value range | INT32 |
| Default value | 0 |
| Units | count |

## Object 0x60B1: Velocity offset

This object shows the offset of the target position.

- Object description

| | |
|---|---|
| Index | 60B1 |
| Name | Velocity offset |
| Object code | Variable |
| Data type | INT32 |

- Entry description

| | |
|---|---|
| Sub-index | 0x00 |
| Access | RW |
| PDO mapping | Yes |
| Value range | Integer32 |
| Default value | 0 |
| Units | count |

# Object 0x60B2: Torque offset

This object provides the offset for torque value.

- Object description

| | |
|---|---|
| Index | 60B2 |
| Name | Torque offset |
| Object code | Variable |
| Data type | Integer16 |

- Entry description

| | |
|---|---|
| Sub-index | 0x00 |
| Access | RW |
| PDO mapping | Yes |
| Value range | INT16 |
| Default value | 0 |
| Units | 0.1% |

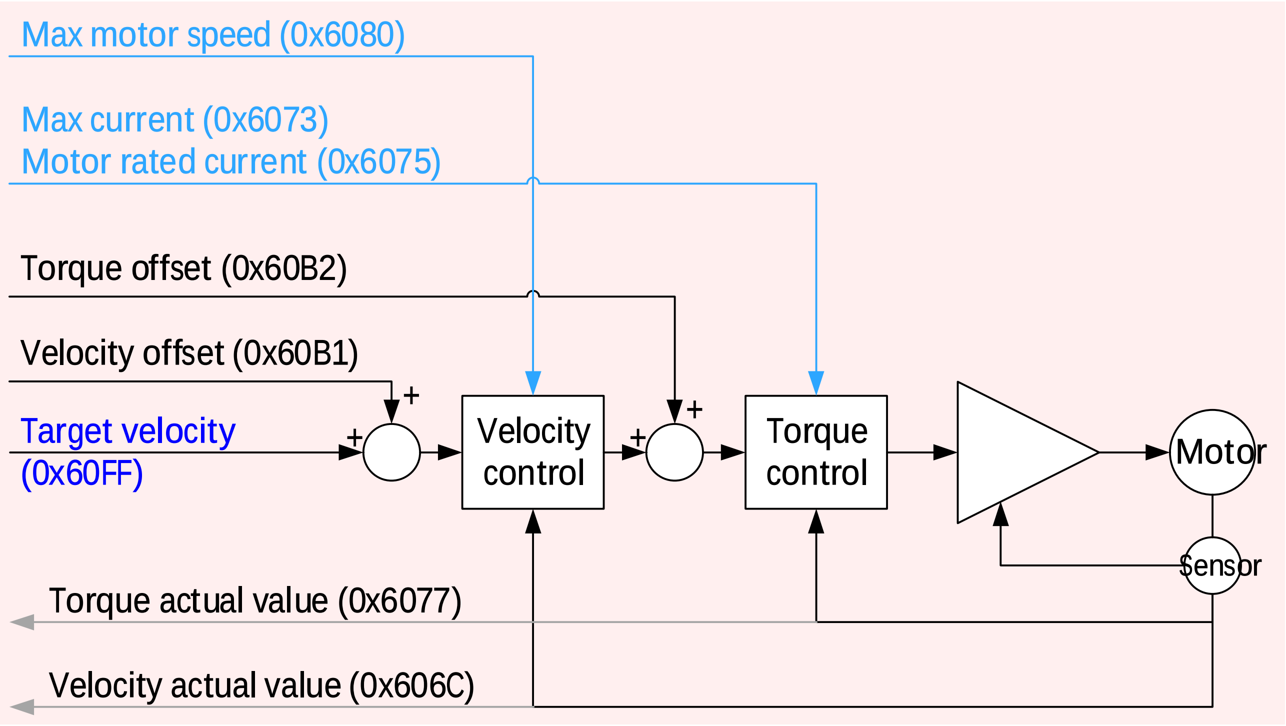
# 13. Cyclic Synchronous Velocity Mode

## 13.1 General Information

The feature of cyclic synchronous velocity mode (csv mode) is that the trajectory generator is in the control device, not in the driver.

The overall structure of csv mode is as follows:

<Overview of cyclic synchronous velocity mode>



- The control device provides the target velocity (0x60FF) to the driver and optionally, additive velocity offset as well as torque offset to allow a second source for velocity and/or torque feedforward.

- The driver executes the control functions (i.e., velocity control and torque control) and gives the control device the actual values (i.e., actual values of position, velocity, and torque) measured by sensors.
  - Users can close the position control loop over the communication system if they wish.

- The driver external control function:

  The performance of the control function is influenced by control parameters (e.g., limit functions) which are externally applicable.

- The csv mode includes the following sub-functions:
  - Demand value input;
  - Velocity capture using position sensor or velocity sensor;
  - Velocity control function with appropriate input and output signals;
  - Limitation of torque demand – based on the user-configured max motor speed, rated current and peak current.

# 13.2 Functional Description

Inputs to the drive's control function

- Target velocity (0x60FF)
- Velocity offset (6x60B1)

  (optional; to be added to the target velocity to allow two examples to set up the velocity).
- Torque offset (0x60B2)

  (optional; used for feedforward control)
- (In cascaded structure where a velocity control is followed by a torque control) The output of the velocity control loop is used as an input for a further calculation in the driver
- Quick stop deceleration (0x6085)
- Quick stop option code (0x605A)
- Interpolation time period (0x60C2)

  Defines the time duration between two updates of the target velocity and/or additive velocity and is used for intercycle interpolation.
- Max motor speed (0x6080)
- Max current (0x6073)
- Motor rated current (0x6075)

Outputs of the drive's control function

- Velocity actual value (0x606C)

  It is used as a compulsory output to the control device.
- Torque actual value (0x6077)

# 13.3 Structure of Controlword and Statusword

The csv mode doesn’t use mode-specific controlword bit.

It uses some bits in statusword for mode-specific purpose.

## 13.3.1 Statusword of the Cyclic Synchronous Velocity Mode

| Bit | Function |
|---|---|
| 0~9 | *See chapter 4.3.2.1 - Device Control/Statusword structure |
| 10 | Reserved |
| 11 | *See chapter 4.3.2.1 - Device Control/Statusword structure |
| 12 | Drive follows the command value. |
| 13 | Reserved |
| 14~15 | *See chapter 4.3.2.1 - Device Control/Statusword structure |

- Definition of bit 12

| Bit | Name | Value | Description |
|---|---|---|---|
| 12 | Drive follows the command value. | 0 | The driver doesn’t follow the command value – “Target velocity” is ignored. |
| | | 1 | The driver follows the command value – “Target velocity” is used as input to the velocity control loop. |

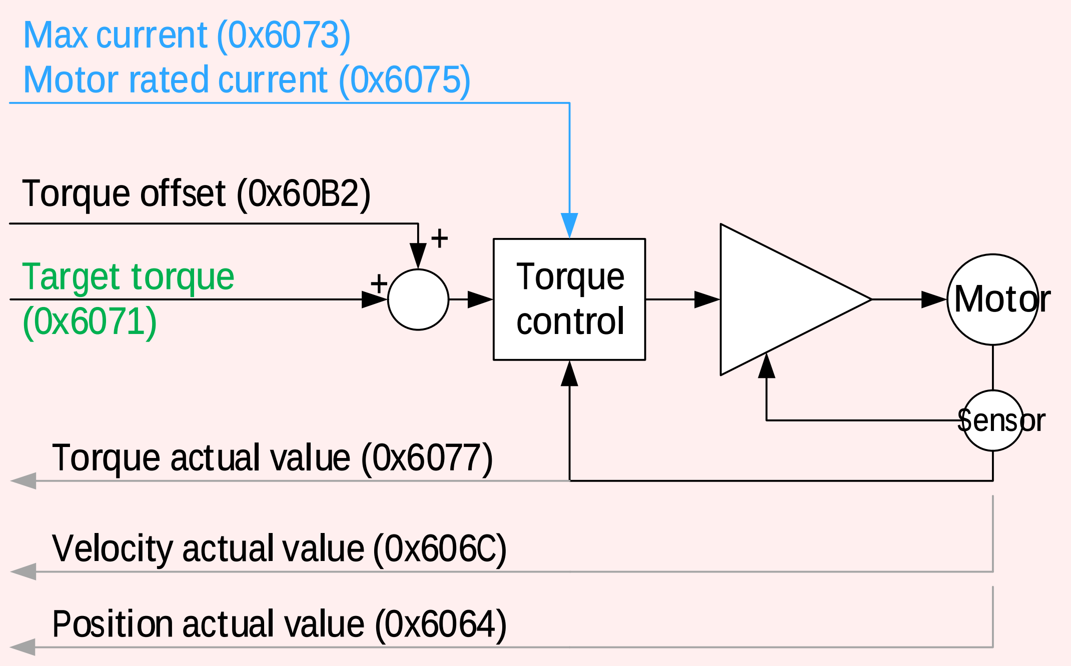
# 14. Cyclic Synchronous Torque Mode

## 14.1 General Information

The feature of cyclic synchronous torque mode (cst mode) is that the trajectory generator is in the control device, not in the driver.

The overall structure of cst mode is as follows:

<Overview of cyclic synchronous torque mode>


Max current (0x6073)
Motor rated current (0x6075)
Torque offset (0x60B2)
+
Target torque
(0x6071)
+
Torque
control
Motor
Sensor
Torque actual value (0x6077)
Velocity actual value (0x606C)
Position actual value (0x6064)

- The control device provides the target torque (0x6071) to the driver and optionally, additive torque offset to allow two examples to set up the torque.

- The driver executes the control functions (i.e., torque control) and gives the control device the actual values (i.e., actual values of position, velocity, and torque) measured by sensors.

- The cst mode includes the following sub-functions:
  - Demand value input;
  - Torque capture;
  - Torque control function with appropriate input and output signals;
  - Limitation of torque demand – based on the user-configured rated current and peak current.

## 14.2 Functional Description

Inputs to the drive's control function

- Target torque (0x6071)
- Torque offset (6x60B2)

  (optional; to be added to the target torque to allow two examples to set up the torque).
- Interpolation time period (0x60C2)

  Defines the time duration between two updates of the target velocity and/or additive velocity and is used for intercycle interpolation.
- Max current (0x6073)
- Motor rated current (0x6075)

Outputs of the drive's control function

- Torque actual value (0x6077)

  It is used as a compulsory output to the control device.

# 14.3 Structure of Controlword and Statusword

The csv mode doesn’t use mode-specific controlword bit.
It uses some bits in statusword for mode-specific purpose.

## 14.3.1 Statusword of the Cyclic Synchronous Torque Mode

| Bit | Function |
|---|---|
| 0~9 | *See chapter 4.3.2.1 - Device Control/Statusword structure |
| 10 | Reserved |
| 11 | *See chapter 4.3.2.1 - Device Control/Statusword structure |
| 12 | Drive follows the command value. |
| 13 | Reserved |
| 14~15 | *See chapter 4.3.2.1 - Device Control/Statusword structure |

- Definition of bit 12

| Bit | Name | Value | Description |
|---|---|---|---|
| 12 | Drive follows the command value. | 0 | The driver doesn’t follow the command value – “Target velocity” is ignored. |
| | | 1 | The driver follows the command value – “Target velocity” is used as input to the velocity control loop. |

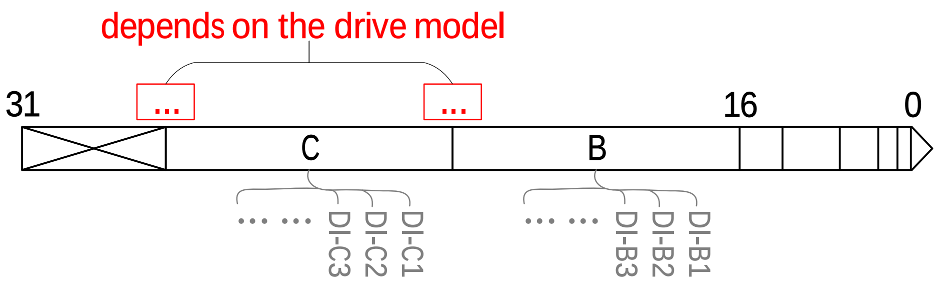
# 15. Optional application FE

The objects described in this chapter are used for the optional generic input/output functional elements.

## Object 0x60FD: Digital inputs

This object shows the logical input levels.

- Bit definition

| Bit | Function |
|---|---|
| 0 | Negative limit switch |
| 1 | Positive limit switch |
| 2 | Home switch |
| 3* | Interlock |
| 4 ~ 15 | Reserved |
| 16 ~ 31 | (Manufacturer-specific)<br>Digital-In port B + port C. See illustration below.<br><br> depends on the drive model<br>31<br>...<br>...<br>16<br>0<br>C<br>B<br>…… DI-C3 DI-C2 DI-C1<br>…… DI-B3 DI-B2 DI-B1 |

*: Bit 3 (interlock) shows the status of the digital input pin which is defined as “Quick Stop”; if quick stop signal is high, then bit 3 (interlock) signal is high.

- Object description

| | |
|---|---|
| Index | 60FD |
| Name | Digital inputs |
| Object code | Variable |
| Data type | UINT32 |

- Entry description

| | |
|---|---|
| Sub-index | 0x00 |
| Access | RO |
| PDO mapping | Yes |
| Value range | UINT32 |
| Default value | No |
| Units | No |

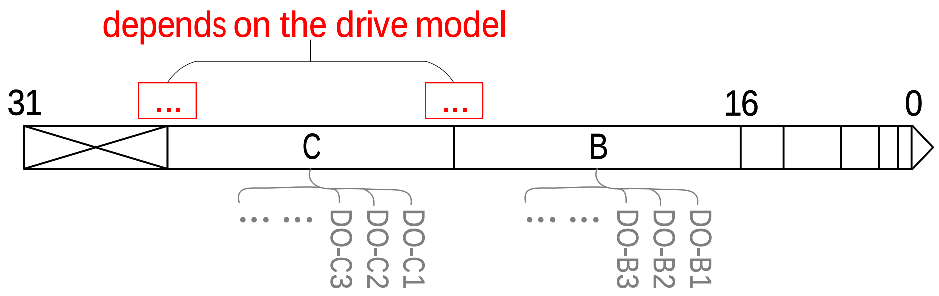
- Value definition

| Field | Value | Definition |
|---|---|---|
| Negative limit switch | 0 | Negative limit switch is not reached |
| | 1 | Negative limit switch is reached |
| Positive limit switch | 0 | Positive limit switch is not reached |
| | 1 | Positive limit switch is reached |
| Home switch | 0 | Home switch is not reached |
| | 1 | Home switch is reached |
| Interlock | 0 | Interlock is not activated |
| | 1 | Interlock is activated |
| Reserved | 0 | Reserved |
| Manufacturer-specific | 0 | Function is not activated |
| | 1 | Function is activated |

# Object 0x60FE: Digital outputs

This object shows the logical output levels.

- Bit definition

| Bit | Function |
|---|---|
| 0 | Set brake |
| 1 ~ 15 | Reserved |
| 16 ~ 31 | (Manufacturer-specific)<br>Digital-Out port B + port C. See illustration below.<br> |

- Object description

| Index | 60FE |
|---|---|
| Name | Digital output |
| Object code | Array |
| Data type | UINT32 |

- Entry description

| | |
|---|---|
| Sub-index | 0x00 |
| Description | Highest sub-index supported |
| Access | C |
| PDO mapping | No |
| Value range | 0x01 |
| Default value | 1 |
| Units | No |

| | |
|---|---|
| Sub-index | 0x01 |
| Description | Physical outputs |
| Access | RW |
| PDO mapping | No |
| Value range | UINT32 |
| Default value | 0000 0000h |
| Units | No |

- Value definition

| Field | Value | Definition |
|---|---|---|
| Set brake* | 0 | Brake is deactivated |
| | 1 | Brake is activated. |
| Reserved | 0x00 | Reserved |
| Manufacturer-specific | 0 | Function is not activated |
| | 1 | Function is activated |

*: This bit is read-only, it shows the status of the brake.

# 16. Device information

## Object 0x67FE: Version number

This object indicates the version number of the CiA® 402 DSP–Part 2 that is carried out in the device.

Currently cpc applies the version number 3.0.0.

- Bit definition

| Bit | Function |
|---|---|
| 0 ~ 7 | Sub version number |
| 8 ~ 15 | Minor version no. |
| 16 ~ 23 | Major version no. |
| 24 ~31 | Reserved |

.

- Object description

| | |
|---|---|
| Index | 67FE |
| Name | Version number |
| Object code | Variable |
| Data type | UINT32 |

- Entry description

| | |
|---|---|
| Sub-index | 0x00 |
| Access | C |
| PDO mapping | No |
| Value range | No |
| Default value | 3 |
| Units | No |

- Value definition

| Field | Value | Definition |
|---|---|---|
| Major version no. | 3 | Major version number for this profile specification version. |
| Minor version no. | 0 | Minor version number for this profile specification version. |
| Sub version no. | 0 | Sub-version number for this profile specification version. |
| Reserved | 0 | Reserved |

End of Page