



Homing Procedures and Setup

Application Note 141

This document describes setups and usage for homing axes with *Triamec* drives. It covers motion based homing, working with absolute encoders and homing from *TwinCAT* systems.

Table of Contents

1	Introduction.....	2	7	TwinCAT Homing with EtherCAT.....	8
2	Homing Method Standard.....	2	8	TwinCAT Homing with Tria-Link.....	9
3	Absolute Encoder Homing.....	6	9	TwinCAT Absolute Encoders with CNC.....	9
4	Homing Method Immediate.....	6		References.....	9
5	Homing Method AtPosition.....	6		Revision History.....	10
6	Homing Method TamaProgram.....	6			

1 Introduction

There are two classes of homing methods. The motion based method *Standard* derives the homing position from a move sequence. The second class is for absolute Encoders. These can use their absolute position information for homing. The position of an absolute encoder is usually aligned to the motor phasing, not the position required in the machine coordinate system. The machine manufacturer defines the zero of each axis. This reference position of an axis is derived by a homing or calibration procedure during machine setup at the manufacturers place. Then the position offset between absolute encoder and machine coordinate system may be stored in various ways.

The zero of an axis may be derived in various ways, using `Parameters.Homing.Method`.

- `Standard` by a homing search move, started by the control system.
- `AbsoluteEncoderXOffsetEncoder` from encoder persistency data of encoder[X], see chapter 3.
- `AbsoluteEncoderXOffsetDrive` from drive persistency data and the encoder[X] position.
- `Immediate` see chapter 4.
- `At Position` see chapter 5.
- `TamaProgram` see chapter 6.
- `SetAbsoluteOffsetX` see chapter 3.

If two absolute encoders are used for one axis, only one can be the position reference. This is why the encoder number X has to be specified for absolute encoder based methods.

Homing commands of an axis are at `Axes[].Commands.Homing.Command`.

- `Start` Start a motion based homing sequence.
- `Stop` Stop any motion based homing ongoing and related moves.
- `Invalidate` Clears the axis homing state.
- `SaveEncoder` Save persistency data to an absolute encoder (axis must be disabled).
- `InvalidateEncoder` Delete the persistency data of an absolute encoder (axis must be disabled).
- `SetPosition` set the encoder positions to `ReferencePosition` and set `HomingDone`.
- `SetHomingDone` set `HomingDone`.

The homing signal of an axis is the homing state `Axes[].Signals.General.HomingState`. It indicates not only the phases of a homing procedure, but also homing errors. The main states, which are also detached from specific homing methods are:

- `Idle` No valid homing yet and no commanded actions.
- `HomingDone` Homing action processed and valid homing reference active.

2 Homing Method Standard

The motion based homing consists of four phases

- The first search phase is typically used to move into a marker, but various triggers can be chosen.
- Then a relocate move is used to get to the position, where the next search should start.
- The second search phase typically triggers on an encoder index, but various triggers can be chosen.
- The move to home phase moves the axis to its final position.

The homing parameters of an axis are at `Axes[].Parameters.Homing`.

- Method set to Standard
- FirstSearchMove Parameters for the first search
- RelocateMove Parameters for the relocate move
- SecondSearchMove Parameters for the second search
- MoveToHomePosition Parameters for the final move to the home position

Please note that the Home Position is not the same as the Reference Position (see Figure 1). The reference position is the position the encoder is set to at the found position of the second search move. The home position is the position where the axis will move to after successful homing. The home position is a parameter which will be the same for all machines of a series. The reference position is a command value received from the control system before homing and is typically calibrated during machine assembly.

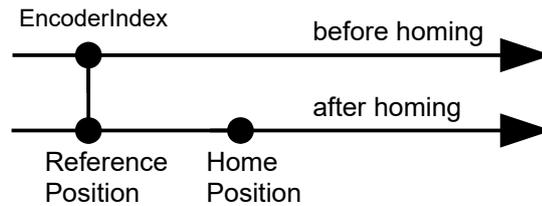


Figure 1: Difference between the reference position and the home position.

The following parameters are contained in the register nodes `FirstSearchMove` and `SecondSearchMove`.

- EventInput The trigger input to be searched for
- ActiveLow Choose True if the event is active low
- SignedMaxDistance The maximum distance a search will move if no trigger event is found.
- DynamicReduction Values smaller than 1.0 will reduce the path planner velocity settings.
- PositionErrorThreshold Threshold value for EventInput = PositionError.
- CurrentThreshold Threshold value for EventInput = Current.
- EventAxis Specifies which axis is used as the source of the event.

The sign of `SignedMaxDistance` has a special meaning. If the trigger is not active before the move, the axis will move into the direction indicated by the sign of this parameter. Otherwise, the direction will be reversed. Consider that `ActiveLow` does not affect the move direction. The following table illustrates the resulting move direction depending on the trigger state and the sign of `SignedMaxDistance`.

SignedMaxDistance	>0	>0	<0	<0
Initial trigger state	FALSE	TRUE	FALSE	TRUE
Resulting initial move direction	positive	negative	negative	positive

For the detection of a hard stop, the following two events can be used:

- In case the EventInput = PositionError, the event will be triggered in case the absolute value of the master position error exceeds the threshold configured in register `PositionErrorThreshold`.
- In case the EventInput = Current, the event will be triggered in case the absolute value of `Signals.PositionController.Controllers[0].Output + Signals.PositionController.Controllers[1].Output` exceeds the threshold configured in register `CurrentThreshold`.



If `EventInput = ExternalInputBit0`, the position latch is triggered by bit0 of the register `Axes[].Commands.General.ExternalInput`. This register can be written cyclically by the control system, or *Tama*, to allow latching or homing, based on external events.

Only `EventInputs` from encoders can be tapped from the other axis with the setting of `EventAxis`. Multiple axes of the same drive cannot source events with the same `EventAxis` setting. An error is thrown in this case.

INFO The first and second search checks for an early trigger. It assumes the trigger condition is not reached within the first 5ms of the search. If the trigger is found earlier, it throws an early trigger error.

The folder `RelocateMove` contains the same parameter `DynamicReduction` plus

- `Distance` The signed distance to move.

The folder `MoveToHomePosition` also contains the parameter `DynamicReduction` plus

- `Position` The absolute position for the final move.

The homing of an axis is started with the command `Start` of `Axes[].Commands.Homing.Command`. It can be interrupted with the command `Stop` of the same register. The register `TestNotEnabled` in the same folder allows testing the homing triggers by manually moving the axis. The states will behave as usual, but no motion commands are issued. `ReferencePosition` is the reference position as received from the control system. The first example (Figure 2) shows a move to the positive limit marker, followed by a reverse move. Then move in positive direction to search the encoder index. Please note that the end position of the `RelocateMove` is attained in the axis mode standstill.

The second example (Figure 3) is a move to the positive end marker, followed by a negative `RelocateMove` and continues in negative direction to search the encoder index. Since the `RelocateMove` and the `SecondSearchMove` have the same direction, the axis will not stop in between for optimal motion behavior. For technical reasons, the `RelocateMove` move is a continuous move, not a discrete move.

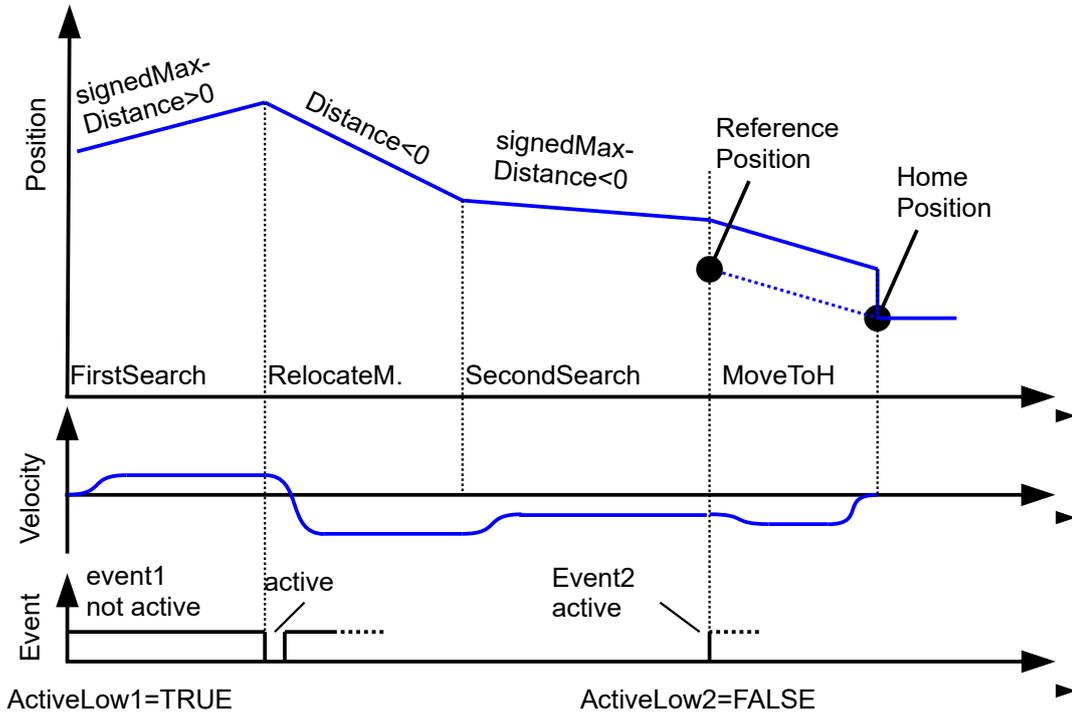


Figure 3: Second example with negative SecondSearch move.
 Figure 2: First example of a Standard homing search sequence.

The third example (Figure 4) shows the same parameters as the second example. This time the marker was already occupied when starting. Therefore the FirstSearchMove moves the opposite direction.

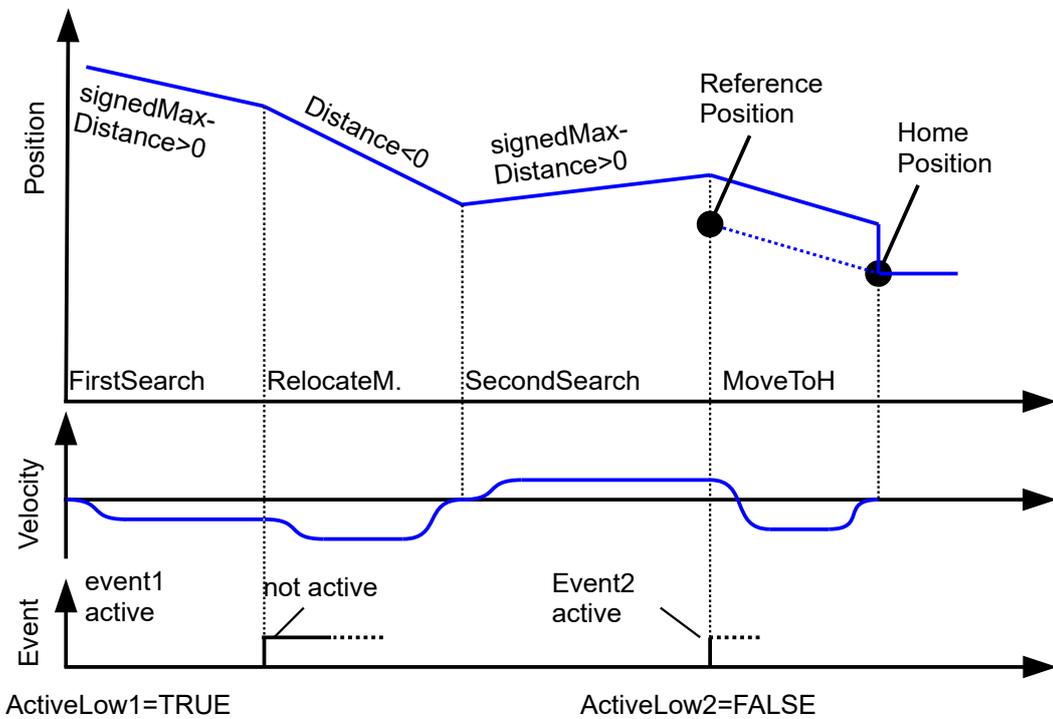


Figure 4: Third example - starting the sequence with the first event already active.

3 Absolute Encoder Homing

The persistent position offset between absolute encoder and machine coordinate system can be stored in two ways. In the encoder itself or in the drive:

If the homing method `AbsoluteEncoderXOffsetEncoder` is selected, the position offset is loaded from the encoder EEPROM when the encoder is activated (after loading a configuration, resetting from an encoder error or during startup). If there is no such information in the encoder, the error `NoPersistencyData` is shown. The position offset is saved to the encoder with the command `SaveEncoder` of the register `Commands.Homing.Command`. Please note that this is only possible in *Disabled* state.

With the command `InvalidateEncoder` the persistent data of the encoder may be cleared.

If the homing method `AbsoluteEncoderXOffsetDrive` is selected, the position offset is loaded from the drive persistent data when the encoder is activated (after loading a configuration, resetting from an encoder error or during startup). The position offset is saved to the drive persistently by saving the drive configuration persistently as described in [1].

While the origin of the offset data with the last two methods is saved in the encoder or drive, there is a third method, which allows to set the offset for absolute encoders from the control system side: If the homing method `SetAbsoluteOffsetX` is selected, the encoder[X] position offset is directly taken from the register `Axes[].Commands.Homing.ReferencePosition` as soon as the homing command `Start` is set in `Axes[].Commands.Homing.Command` or by the control system. A larger value of `ReferencePosition` yields a larger position value and its unit is the same as the axis unit.

4 Homing Method Immediate

This homing method sets `Homing Done` without any action. There is no set position taking place. Final state is `HomingDone`.

5 Homing Method AtPosition

This homing method sets the actual position to the commanded value `ReferencePosition` without any move. Final state is `HomingDone`.

6 Homing Method TamaProgram

In more complex machines or axes setups, it may be necessary to customize the homing procedure. With the method `TamaProgram` it is possible to start a homing procedure which is implemented in a controller level, *Tama* program. Possible use cases are: manipulating homing data, run custom homing procedures, or solve multi axis homing interference.

The *Tama* program can be used for pre- or postprocessing of the *Standard* method, as well as a stand-alone homing method.

The `TamaProgram` method has its own state machine underlying. In Figure 5 the states and transitions are visualized. The states are reflected in the register `Axes[].Signals.Homing.State`.

- Idle and HomingDone
- TamaProgramRequest
- TamaProgramRunning
- *Homing Standard Routine*

See chapter 1.

Waits for handshake from Tama program. Times out in 100ms.

Main state for this method, accepting Tama actions.

The *Standard* method state machine..

Returns automatically to TamaProgramRunning. The *Standard* method can only be used as a whole, altering specific states of it is not possible.

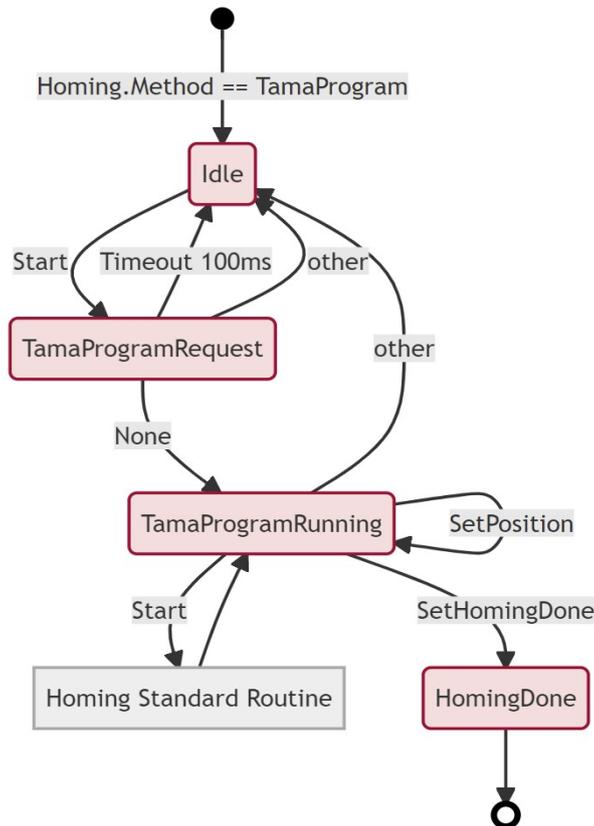


Figure 5: State machine in TamaProgram homing method.

Transitions are commanded in Axes[].Commands.Homing.Command. Any command that doesn't match a valid state transition, forces the state back to Idle. This transitions are named *other* in Figure 5.

7 TwinCAT Homing with EtherCAT

The reference position can be setup as shown in Figure 6.

Note For correct use of *EtherCAT* Startup variables with 64 bit Floating point precision, use ESI descriptions *Triamec1.8.xml* or newer.

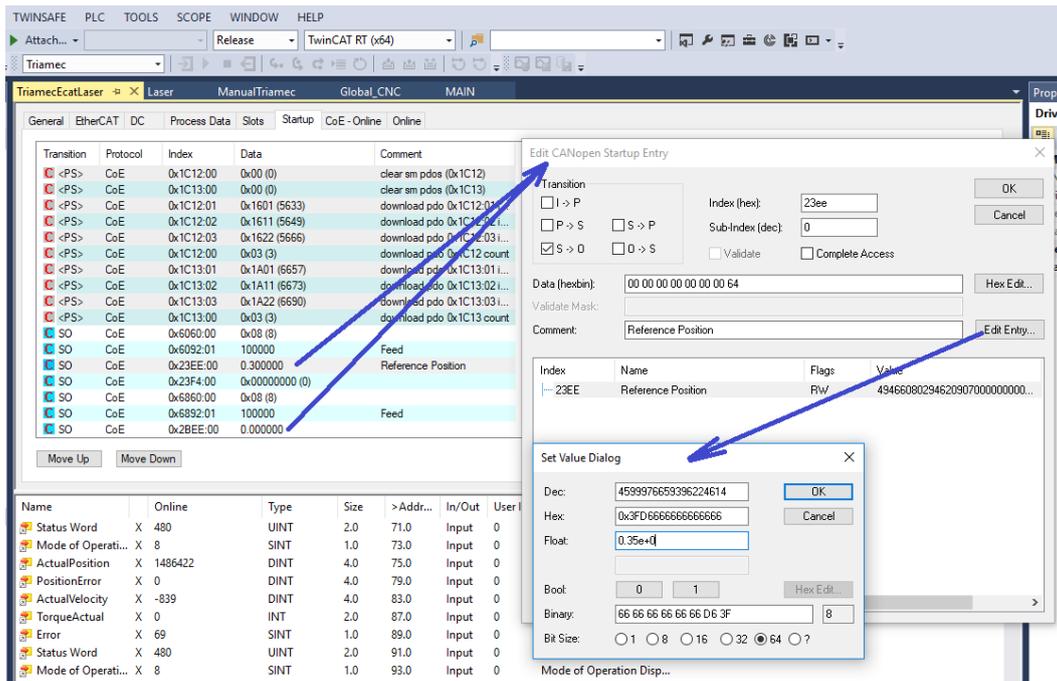


Figure 6: Setup Homing registers in TwinCAT

Homing from the CNC Module

To setup homing with the *TwinCAT CNC* module, ensure the cyclic data of the drive contains the objects 0x6060 and 0x6061 (mode of operation and mode of operation display). This is standard with the ESI file *Triamec1.1.xml* and newer.

The parameter list of a CNC axis must contain the following entries for correct homing behavior.

kenngr.device_id	8	(turn off the target reached check before homing)
kenngr.set_refpos_mode	ABSOLUTE	(P-AXIS-00278 : Modes for setting the homing pos)
kenngr.set_refpos_offset	0	(P-AXIS-00279 : [0.1um] or [10-4degree] Offset)
kenngr.homing_type	DRIVE_CONTROLLED	(P-AXIS-00299 : Homing type)
kenngr.vendor_id	100	(compatibility with G74 for CNC versions >3079.10)

The homing sequence is then started using the G-code G74. The following sample code starts homing of the Z axis first. When Z finished homing, the X and Y axes start homing simultaneously.

```
G74 Z1 X2 Y2
```

The position of the reference marker 0x23EE (axis0) and 0x2BEE (axis1) may vary from machine to machine after calibration. In the sample codes, this position is set to 0.0 in the startup list. The type is



double with metric units (m or rad) as used during drive setup. It might be overwritten by COE register write commands as described in [2].

8 TwinCAT Homing with Tria-Link

While the HMI is in Automatic or MDI mode, the homing procedure of an axis is started by setting its `referenceStart` flag to `TRUE`.

```
gAxis[].referenceStart := TRUE;
```

The HMI function **Reference** is linked with the following code in the `PRG_CNCChannelHandler` before the call to `CNCChannel`:

```
IF PLCMachineMode[nChan].Homing THEN
  PLCMachineMode[nChan].Homing := FALSE;
  AxisGroup.referenceStart := TRUE;
END_IF
```

Note In contrast to using the *EtherCAT* fieldbus, the standard CNC command G74 is not functional with a Tria-Link setup.

Homing from the CNC Module

The parameter list of a CNC axis must contain the following entries for correct homing behavior.

<code>kenngr.device_id</code>	8	(turn off the target reached check before homing)
<code>kenngr.set_refpos_mode</code>	ABSOLUTE	(P-AXIS-00278 : Modes for setting the homing pos)
<code>kenngr.set_refpos_offset</code>	0	(P-AXIS-00279 : [0.1um] or [10-4degree] Offset)
<code>kenngr.homing_type</code>	DRIVE_CONTROLLED	(P-AXIS-00299 : Homing type)
<code>kenngr.vendor_id</code>	100	(compatibility with G74 for CNC versions >3079.10)

9 TwinCAT Absolute Encoders with CNC

With the following entry in the axis parameters, the CNC immediately sets reference done without the need for homing.

<code>kenngr.abs_pos_guelzig</code>	1	(P-AXIS-00014 : Absolute measurement system)
-------------------------------------	---	--

References

- [1] "Servo Drive Setup Guide", ServoDrive-SetupGuide_EP013.pdf, Triamec Motion AG, 2022.
- [2] "Triamec TwinCat Ethercat", SWTC_TwinCAT-UserGuideEcat_EP009.pdf, Triamec Motion AG, 2022.

Revision History

Version	Date	Editor	Comment
001	2021-12-14	sm	Homing descriptions extracted from AN107
002	2022-03-10	sm	Add newly introduced registers to doc, update template, update TwinCAT sections.
003	2022-05-19	sm	Introduce TamaProgram homing method and new registers, minor style updates.
004	2023-09-20	ns, mvx	Homing with ExternalInputBit0, update explanation Tama Homing
005	2024-03-12	dg	Description for homing event current added

Copyright © 2024
Triamec Motion AG
All rights reserved.

Triamec Motion AG
Lindenstrasse 16
6340 Baar / Switzerland

Phone +41 41 747 4040
Email info@triamec.com
Web www.triamec.com

Disclaimer

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

- This document contains proprietary information belonging to Triamec Motion AG. Such information is supplied solely for the purpose of assisting users of Triamec products.
- The text and graphics included in this manual are for the purpose of illustration and reference only. The specifications on which they are based are subject to change without notice.
- Information in this document is subject to change without notice.