BECKHOFF New Automation Technology

Manual | EN

TS5060

TwinCAT 2 | NC FIFO Axes

Supplement | Motion

2023-07-20 | Version: 1.1

Table of contents

1 Foreword

1.1 Notes on the documentation

This description is only intended for the use of trained specialists in control and automation engineering who are familiar with applicable national standards.
It is essential that the documentation and the following notes and explanations are followed when installing and commissioning the components.
It is the duty of the technical personnel to use the documentation published at the respective time of each installation and commissioning.

The responsible staff must ensure that the application or use of the products described satisfy all the requirements for safety, including all the relevant laws, regulations, guidelines and standards.

Disclaimer

The documentation has been prepared with care. The products described are, however, constantly under development.
We reserve the right to revise and change the documentation at any time and without prior announcement.
No claims for the modification of products that have already been supplied may be made on the basis of the data, diagrams and descriptions in this documentation.

Trademarks

Beckhoff®, TwinCAT®, TwinCAT/BSD®, TC/BSD®, EtherCAT®, EtherCAT G®, EtherCAT G10®, EtherCAT P®, Safety over EtherCAT®, TwinSAFE®, XFC®, XTS® and XPlanar® are registered trademarks of and licensed by Beckhoff Automation GmbH.
Other designations used in this publication may be trademarks whose use by third parties for their own purposes could violate the rights of the owners.

Patent Pending

The EtherCAT Technology is covered, including but not limited to the following patent applications and patents:
EP1590927, EP1789857, EP1456722, EP2137893, DE102015105702
with corresponding applications or registrations in various other countries.

EtherCAT® is a registered trademark and patented technology, licensed by Beckhoff Automation GmbH, Germany

Copyright

1.2 For your safety

Safety regulations

Read the following explanations for your safety.
Always observe and follow product-specific safety instructions, which you may find at the appropriate places in this document.

Exclusion of liability

All the components are supplied in particular hardware and software configurations which are appropriate for the application. Modifications to hardware or software configurations other than those described in the documentation are not permitted, and nullify the liability of Beckhoff Automation GmbH & Co. KG.

Personnel qualification

This description is only intended for trained specialists in control, automation, and drive technology who are familiar with the applicable national standards.

Signal words

The signal words used in the documentation are classified below. In order to prevent injury and damage to persons and property, read and follow the safety and warning notices.

Personal injury warnings

⚠ DANGER
Hazard with high risk of death or serious injury.

⚠ WARNING
Hazard with medium risk of death or serious injury.

⚠ CAUTION
There is a low-risk hazard that could result in medium or minor injury.

Warning of damage to property or environment

NOTICE
The environment, equipment, or data may be damaged.

Information on handling the product

ℹ This information includes, for example:
recommendations for action, assistance or further information on the product.

1.3 Notes on information security

The products of Beckhoff Automation GmbH & Co. KG (Beckhoff), insofar as they can be accessed online, are equipped with security functions that support the secure operation of plants, systems, machines and networks. Despite the security functions, the creation, implementation and constant updating of a holistic security concept for the operation are necessary to protect the respective plant, system, machine and networks against cyber threats. The products sold by Beckhoff are only part of the overall security concept. The customer is responsible for preventing unauthorized access by third parties to its equipment, systems, machines and networks. The latter should be connected to the corporate network or the Internet only if appropriate protective measures have been set up.

In addition, the recommendations from Beckhoff regarding appropriate protective measures should be observed. Further information regarding information security and industrial security can be found in our https://www.beckhoff.com/secguide.

Beckhoff products and solutions undergo continuous further development. This also applies to security functions. In light of this continuous further development, Beckhoff expressly recommends that the products are kept up to date at all times and that updates are installed for the products once they have been made available. Using outdated or unsupported product versions can increase the risk of cyber threats.

To stay informed about information security for Beckhoff products, subscribe to the RSS feed at https://www.beckhoff.com/secinfo.

2 Overview

The TwinCAT NC FIFO group provides the facility of supplying externally generated set values to a group of axes via the NC, and thus of driving them synchronously. The set values for the positions of the axes corresponding to a fixed, although selectable, time period (the FIFO cycle time) are present in the form of a FIFO which is kept topped up by the PLC. The FIFO allows **the position of a group of axes to be handled synchronously with reference to time**. The FIFO contains a fixed number of prescribed reference points, while the NC interpolates position between them and determines the speed. The FIFO's axis positions are

- either read out of a large table located in the PLC and calculated offline, or
- calculated by a function in the PLC

and continuously passed in each PLC cycle by a PLC program to the NC's FIFO.

Interfaces and Configuration

Interfaces and Configuration [▶ 10]

The user can parameterise the FIFO group via the **System Manager** or the PLC, and can manipulate the functionality of the FIFO group by means of **PLC blocks**. He also has available (in the System Manager and through the PLC) a cyclic NC-PLC **channel interface**. This contains the channel process image, and permits access to a variety of channel variables. Finally, he also has available (in the System Manager and through the PLC) a cyclic NC-PLC **axis interface**. This contains the axis process image for each FIFO axis, and permits access to a variety of axis variables.

FIFO Parameters and the FIFO

FIFO Parameters and the FIFO [▶ 11]

The FIFO group is parameterised by a number of data items. These must be set by the user:

- FIFO dimension *N* = number of FIFO axes,
- FIFO length *Z* = number of FIFO lines,
- FIFO cycle time *T_f* = cycle time for which the FIFO data was generated.

The FIFO is a table, *[Z][N]***T**, with *N* columns and *Z* lines. These lines are cyclically refilled after they have been used. At position *T[T][A]* the FIFO contains the set position value for axis A at time *T* × *T_f* (*T_f* FIFO cycle time).

Sequence and the Generation of Set Values

Sequence and the Generation of Set Values [▶ 13]

1. Preparation phase: Group FIFO and parameterise.
2. Fill FIFO and run FIFO group.
3. Prepare FIFO data and refill FIFO.
4. Stop and reconfigure group.

The generation of set values includes the set position, set speed, set acceleration and the set direction.

Override

Override [▶ 15]

The override is used to apply a synchronous online percentage change to the speeds of all the FIFO axes prescribed by the FIFO. There are two logical data channels:

- **Instantaneous override** changes the override instantaneously to the desired value. This means that it is up to the user to modify the override slowly enough that excessive following errors do not result from the change in override.
- The **PT-2 override**, which can be parameterised, and which uses a PT-2 transfer function to modify the override continuously towards the desired speed according to the difference from the current speed.

3 Interfaces and Configuration

The user can parameterise the FIFO group via the System Manager or the PLC, and can manipulate the functionality of the FIFO group by means of PLC blocks. He also has available (in the System Manager and through the PLC) a cyclic NC-PLC channel interface. This contains the channel process image, and permits access to a variety of channel variables. Finally, he also has available (in the System Manager and through the PLC) a cyclic NC-PLC axis interface. This contains the axis process image for each FIFO axis, and permits access to a variety of axis variables.

Configuration of the FIFO Group with the System Manager

Create the FIFO Channel

A FIFO channel must first be created in the System Manager. The method for inserting a channel is described in the System Manager documentation under NC Configuration.

Parameterise the FIFO Group

A description of the parameters can be found on the FIFO Parameters and the FIFO [▶ 11] page.

Group Axes

All the system's axes are always initialized as PTP axes. Once the steps described above have been carried out, and the parameters have been saved and entered into the registry, the axes can be moved to their base positions. It is important to note that the position must be compatible with the 1st line in the FIFO.

PLC blocks are then used to determine the FIFO group. In other words, selected PTP axes are included in the FIFO group.

PLC Blocks

The PLC library offers access to all the FIFO group's functionality:

- Configuration [▶ 19] and reconfiguration [▶ 19] between PTP axes and FIFO group.
- Filling the FIFO, running the group, refilling the FIFO.

NC-PLC-Interface

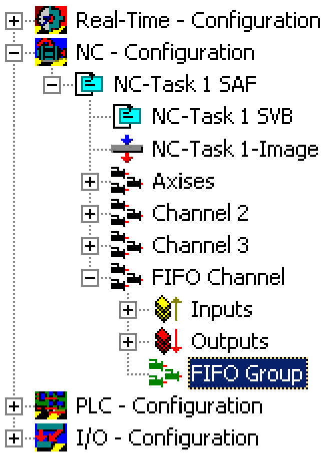
The cyclic NC-PLC channel interface contains the **channel process image** and, in particular, access to the channel variables. See TwinCAT NC I documentation

The cyclic NC-PLC axis interface contains the **axis process image** and access to the axis variables. See TwinCAT NC General documentation

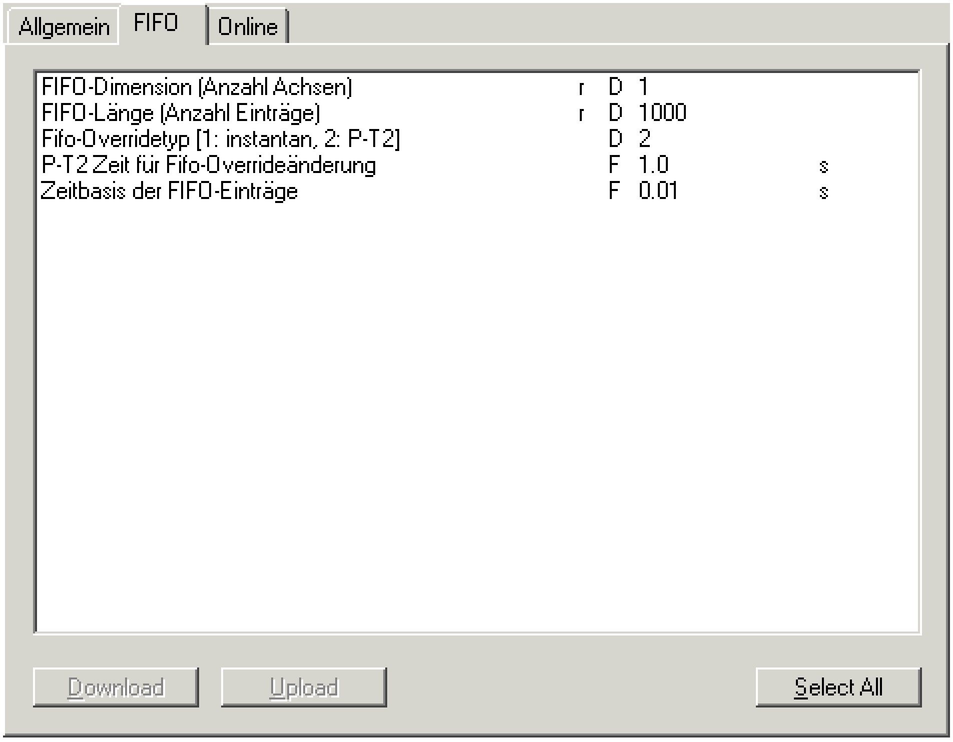
4 FIFO Parameters and the FIFO

The FIFO group is parameterised with the aid of the System Manager, although this can also be partially carried out via ADS, i.e. using the PLC.

Parameterisation of the FIFO Group



```
Real-Time - Configuration
NC - Configuration
   NC-Task 1 SAF
      NC-Task 1 SVB
      NC-Task 1-Image
      Axises
      Channel 2
      Channel 3
      FIFO Channel
         Inputs
         Outputs
         FIFO Group
PLC - Configuration
I/O - Configuration
```



FIFO-Dimension (Anzahl Achsen)	r	D	1	
FIFO-Länge (Anzahl Einträge)	r	D	1000	
Fifo-Overridetyp [1: instantan, 2: P-T2]		D	2	
P-T2 Zeit für Fifo-Overrideänderung		F	1.0	s
Zeitbasis der FIFO-Einträge		F	0.01	s

The FIFO group is parameterised by a number of data items. These must be set by the user:

- FIFO dimension *N* = number of FIFO axes,
- FIFO length *Z* = number of FIFO lines,
- FIFO cycle time *T_f* (the timebase of the FIFO entries) = cycle time for which the FIFO data was generated.
- FIFO-Override [▶ 15]

The FIFO is a table, *[Z][N]T*, with *N* columns and *Z* lines. These lines are cyclically refilled after they have been used. At position *T[T][A]* the FIFO contains the set position value for axis A at time *T* × *T_f* (*T_f* FIFO cycle time).

FIFO Timing and Size

The timing of the data generation in the PLC (the PLC cycle time), of the ADS transfer via PLC (the PLC cycle time and the size of the PLC's internal data buffer), the usage of the FIFO contents (SAF cycle time) and the length of the FIFO, *N* must be set up and adapted to one another in such a way that

- the FIFO does not run dry,
- the FIFO (as a rule) can accept the data offered by the PLC.

FIFO Cycle Time

The statement execution time constant (known as the SAF cycle time, *T_saf*) does not have to be identical to the time constant for which the FIFO was designed or generated (the FIFO cycle time, *T_f*).

T_f > T_saf:

If the FIFO cycle time is bigger than the SAF cycle time, then a **fine interpolation** is in operation, so that the necessary intermediate values for the position control may in some cases be linearly interpolated a number of times from the same pair of neighbouring reference points in the FIFO. The acceleration and the speed have correspondingly coarse resolutions, and should not be used in unfiltered form as input magnitudes.

T_f < T_saf:

If on the other hand the FIFO cycle time is smaller than the SAF cycle time, then **coarse interpolation** is in action. In such cases, a portion of the FIFO information remains unused.

Table 1: Table of FIFO Parameters

FIFO parameter	Meaning and boundary conditions
FIFO dimension *N*	Number of FIFO axes $1 \le N$
FIFO lines *Z*	Number of FIFO lines $2 \le Z$
FIFO cycle time *T_f*	FIFO cycle time 0.0 < *T_f*

5 Sequence and the Generation of Set Values

1. Preparation phase: Group FIFO and parameterize.
2. Fill FIFO and run FIFO group.
3. Prepare FIFO data and refill FIFO.
4. Stop and reconfigure group.

The generation of set values includes the set position, set speed, set acceleration and the set direction.

Preparation phase

1. All the axes are moved to a base position in PTP mode (which must be compatible with the first line in the FIFO).
2. Create the FIFO Channel: interpreter channel without interpreter.
3. Create the FIFO group: the axes are collected into the FIFO group.
4. Parameterize the FIFO group.

Start Phase

1. The FIFO is filled with data as far as possible.
2. Set the FIFO override.
3. Run the FIFO group.

Running Phase

1. Data for the FIFO is provided in the PLC by the user or is generated, and is collected in a buffer internal to the PLC.
2. PLC cycles transfer this data to the FIFO in the NC. If the FIFO is too full to accept the whole of the buffer's contents, the whole buffer is refused, and the PLC must offer it again. To assist optimization of the timing it is possible for the number of filled lines in the FIFO to be interrogated.
3. The NC works through the FIFO entries in SAF cycles. If the FIFO is empty the axes retain the set position values of the last FIFO line (which causes a following error if the axes were not at that stage stationary).
4. For all the axes, the FIFO generator must provide: the set position, set speed, set acceleration and set direction.

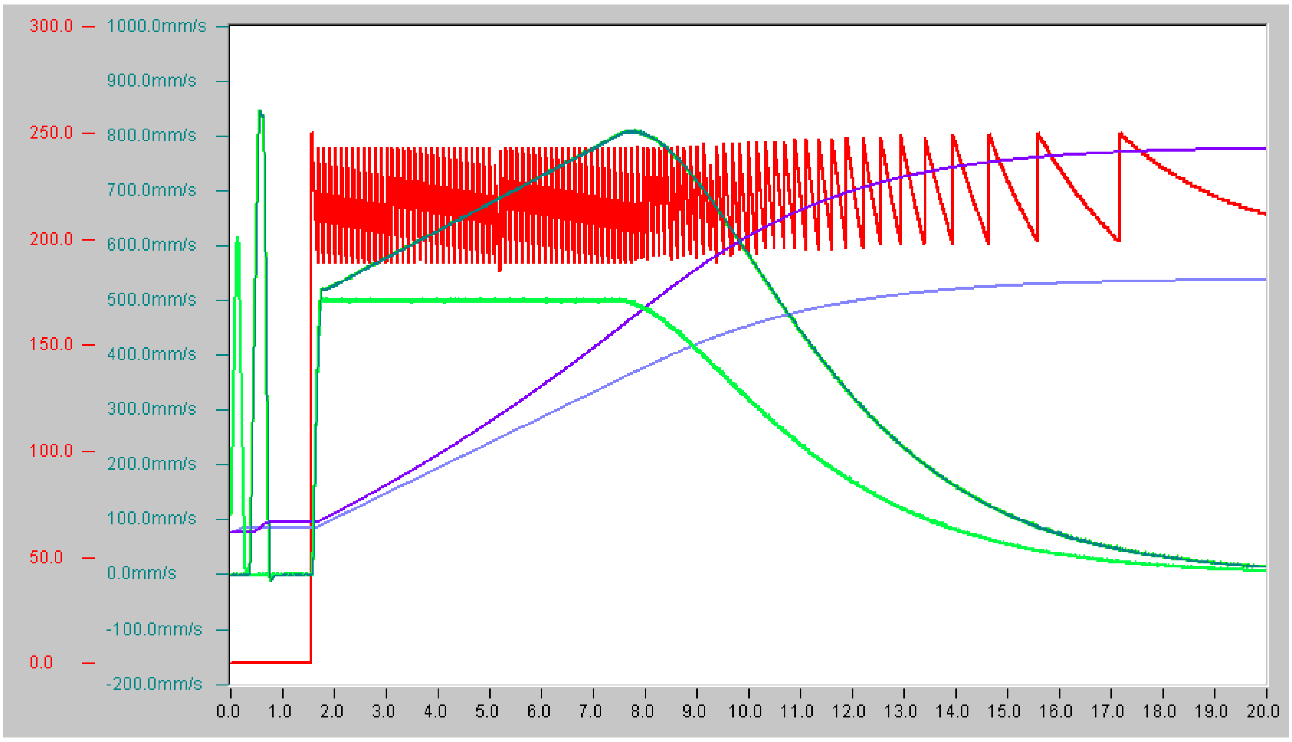
End Phase

1. The positioning process normally ends by the contents of the FIFO bringing the axes to a halt followed by a stationary interval.
2. The axes can be stopped by a FIFO override of 0.0.
3. The FIFO can run dry, bringing the axes to an instantaneous halt.
4. After being stopped, the axes in the FIFO group can be reconfigured as PTP axes.

Set value generation

When working in sequence through the FIFO entries, the set value generation process determines not only the set position, but also the set speed and the set acceleration for an axis. The set speed is generally used as a speed pre-control, as in TwinCAT PTP.

With every SAF call, the FIFO generator seeks the nearest pair of lines according to the elapsed time (taking the override into account), performs a linear interpolation of the position, and calculates the speed and acceleration from the differences. There are known faults in this method of calculating the set values from differences: the magnitudes should not therefore be used as input values without some examination.



Process flow with two FIFO axes (positions and speeds, set and actual values) and the quantity of data in the FIFO (red).

6 Override

The override is used to apply a synchronous online percentage change to the speeds of all the FIFO axes prescribed by the FIFO. There are two logical data channels:

- **Instantaneous override** changes the override instantaneously to the desired value. This means that it is up to the user to modify the override slowly enough that excessive following errors do not result from the change in override.
- The **PT-2 override**, which can be parameterised, and which uses a PT-2 transfer function to modify the override continuously towards the desired speed according to the difference from the current speed.

General Notes

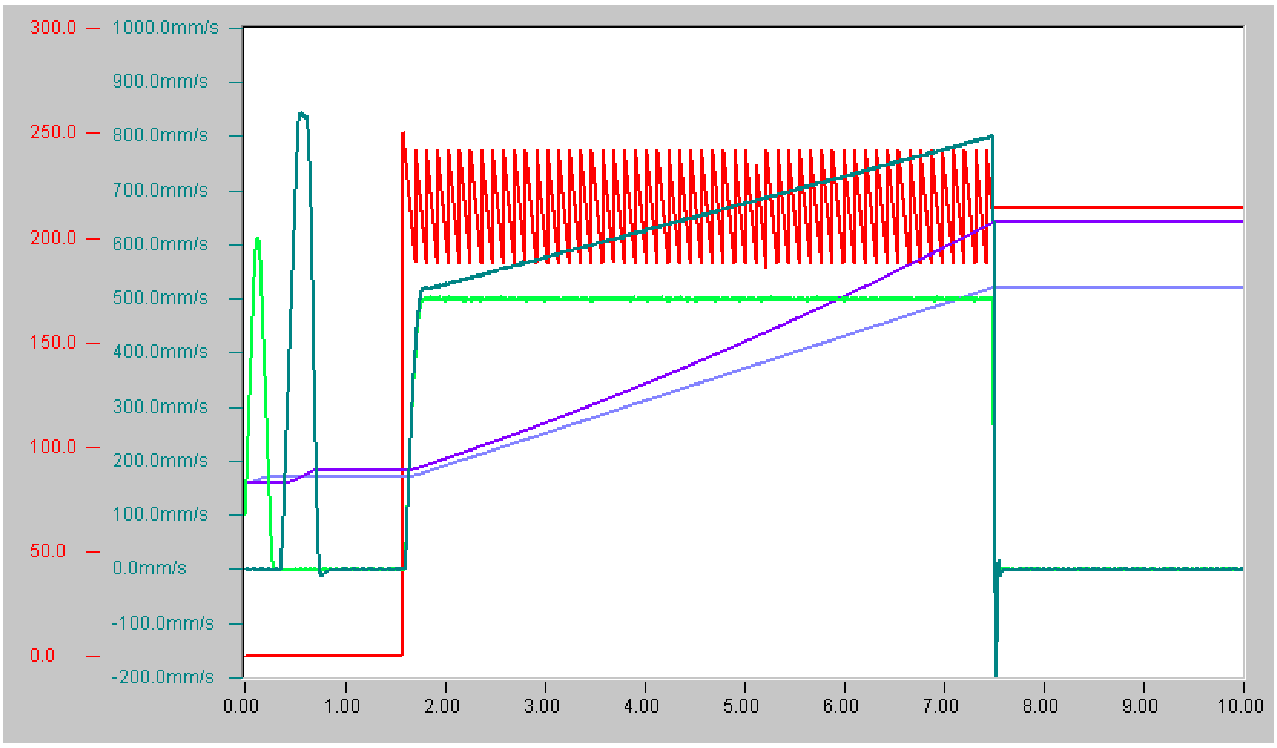
The override can be set right at the beginning.

The axes can be stopped through setting override *0.0*.

After such a stop, further changes to the override will no longer be accepted.

Instantaneous override

The instantaneous override alters the speed immediately to the desired value. Since a very sudden change in speed is normally undesirable, the user is responsible for ensuring that the override changes appropriately with time (controlled adjustment).

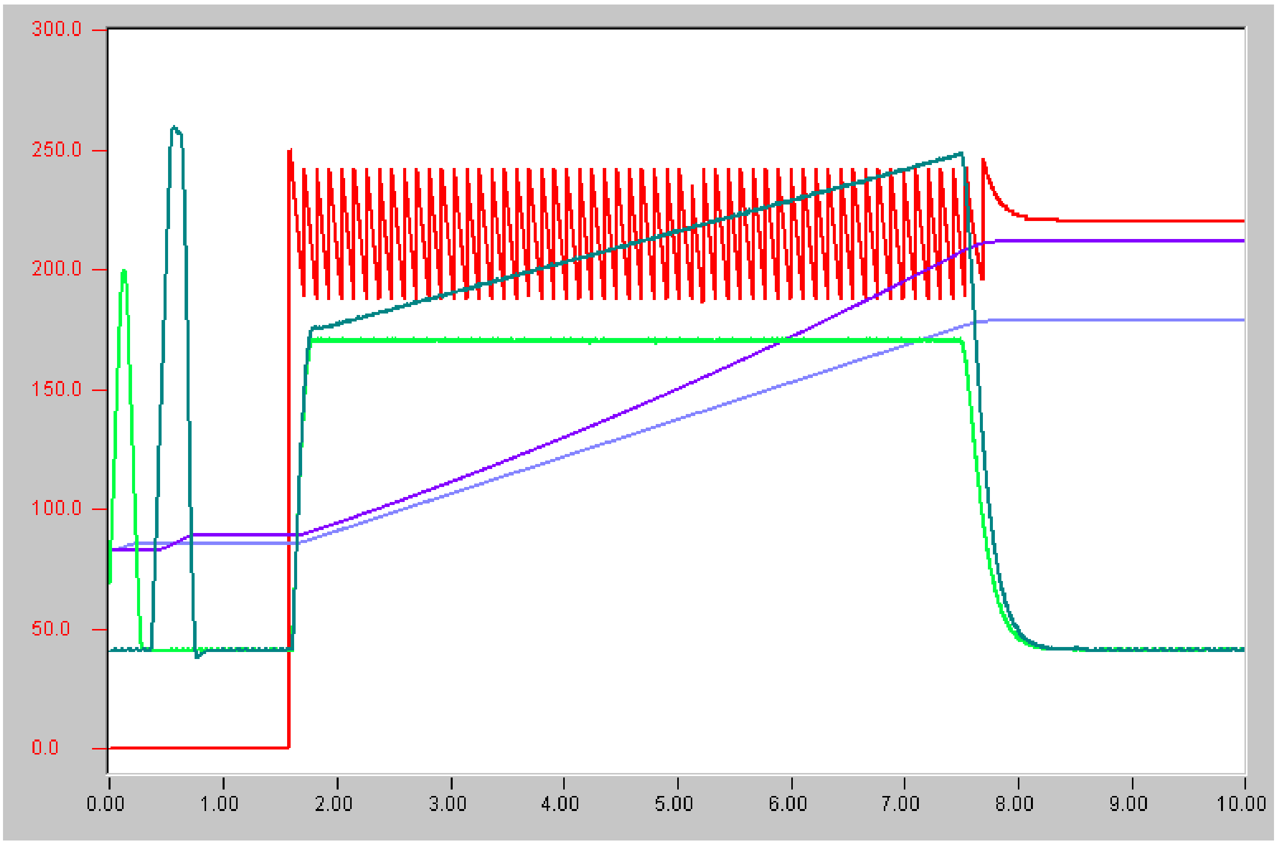


Instantaneous override

PT-2 Override

The PT2 override changes the speed continuously according to the difference between the current speed level and the desired speed level. The progress of this modification is specified by these external and internal parameters

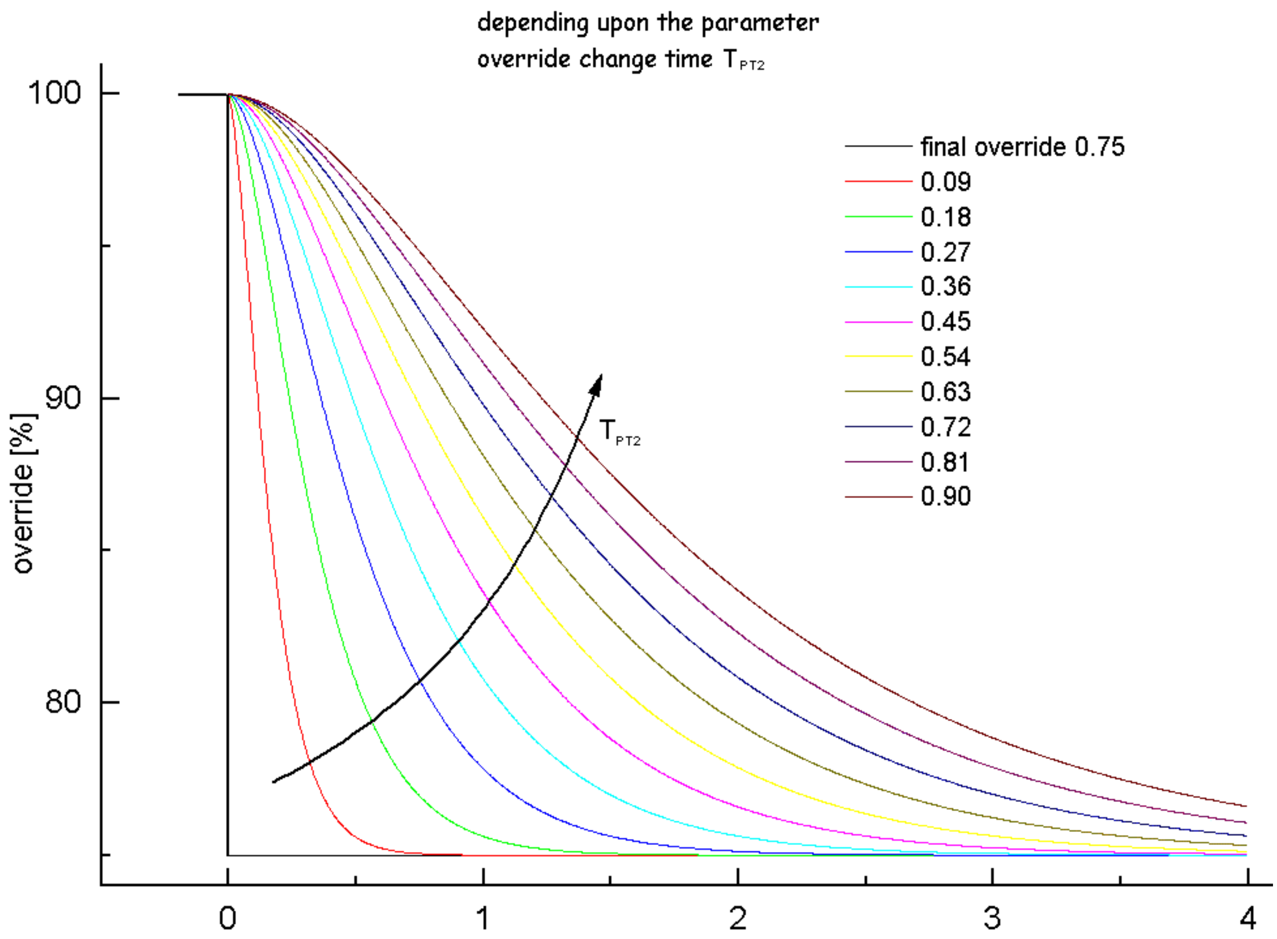
- Discriminate (damping) > 1.0 [is specified as 1.001]
- Time constant *T_0 0.0* < *T_0* < ∞ [freely selectable]
- Cutoff, which completes the exponentially gradual decay of an alteration in a finite time [is parameterised internally by means of the reference speeds of the FIFO axes].



PT2-Override

PT-2 Override Time Constant

The time constant *T_0* specifies the speed of the level change.



PT2 parameterisation: time constant *T_0*

7 PLC API

In many applications it is necessary to synchronise two or more axes. Axes can be coupled together in the TwinCAT NC PTP. A master axis is then actively controlled, and the position of one or more coupled slave axes is synchronously controlled by the NC.

The simplest type of coupling is linear coupling with a fixed ratio of transmission (an electronic gearbox).

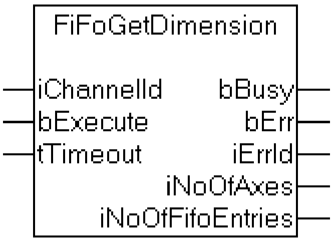
Some applications require a more complex coupling of master and slave, one which can not be described by a simple mathematical formula. Such a dependency can be described by means of a table that specifies an associated slave position for every master position.

The TwinCAT NC PTP offers the possibility of coupling a slave axis to a master axis by means of a table (electronic camshaft). Here the table contains a certain number of prescribed reference points, and the NC interpolates position and speed between them.

FIFO axes extend the table concept for any number of sequences that do not usually recur cyclically. In this case, rather than having one master-slave table prescribed in advance, the latest axis positions are constantly "topped up" by a PLC program. Because the FIFO axes are combined into a FIFO group, synchronised axis motions can be implemented.

The TwinCAT PLC library **TcNcFifoAxis.lib**, available as an additional product, provides easy management of FIFO axes. An example program using FIFO axes makes use of this library.

7.1 FiFoGetDimension



FiFoGetDimension determines the dimensioning, i.e. the number of the axes and the maximum number of FIFO entries in a FIFO channel.

Interface

```
VAR_INPUT
    iChannelId  : UDINT;
    bExecute    : BOOL;
    tTimeout    : TIME;
END_VAR
VAR_OUTPUT
    bBusy        : BOOL;
    bErr         : BOOL;
    iErrId       : UDINT;
    iNoOfAxes    : UDINT;
    iNoOfFifoEntries : UDINT;
END_VAR
```

iChannelId: Channel ID of the FIFO channel.

bExecute: Edge-triggered signal for execution of the command.

tTimeout: ADS timeout (about 1 second).

bBusy: Becomes TRUE with a rising edge at bExecute, and remains TRUE until the block has executed the command.

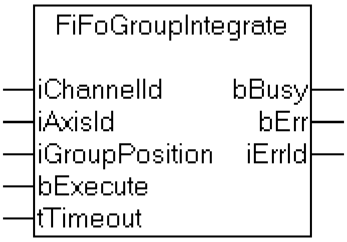
bErr: Becomes TRUE if an error occurs while executing the command.

bErrId: Error number (ADS or NC error number).

iNoOfAxes: Number of axes for which the FIFO channel has been parameterised.

iNoOfFifoEntries: Number of FIFO entries for which the FIFO channel has been parameterised.

7.2 FiFoGroupIntegrate



FiFoGroupIntegrate integrates an initially independent PTP axis into a FIFO channel. *iGroupPosition* specifies the logical sequence, and indicates the position within the channel at which the axis is to be inserted.

Interface

```
VAR_INPUT
    iChannelId      : UDINT;
    iAxisId     : UDINT; (* [1..n] *)
    iGroupPosition  : UDINT; (* [1..m] *)
    bExecute    : BOOL;
    tTimeout    : TIME;
END_VAR
VAR_OUTPUT
    bBusy    : BOOL;
    bErr     : BOOL;
    iErrId   : UDINT;
END_VAR
```

iChannelId: Channel ID of the FIFO channel.

iAxisId: Axis ID of an axis that is to be inserted into the FIFO channel.

iGroupPosition: Position of the axis within the FIFO channel (iGroupPosition>=1).

bExecute: Edge-triggered signal for execution of the command.

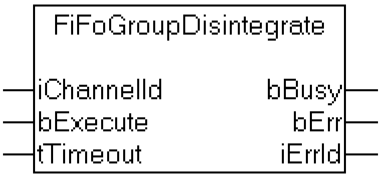
tTimeout: ADS timeout (about 1 second).

bBusy: Becomes TRUE with a rising edge at bExecute, and remains TRUE until the block has executed the command.

bErr: Becomes TRUE if an error occurs while executing the command.

bErrId: Error number (ADS or NC error number).

7.3 FiFoGroupDisintegrate



FiFoGroupDisintegrate dismantles a FIFO channel. This means that all the axes are removed from the FIFO channel and are placed in the standard PTP channel as independent PTP axes.

Interface

```
VAR_INPUT
    iChannelId  : UDINT;
    bExecute    : BOOL;
    tTimeout    : TIME;
END_VAR
VAR_OUTPUT
    bBusy       : BOOL;
    bErr    : BOOL;
    iErrId      : UDINT;
END_VAR
```

iChannelId: Channel ID of the FIFO channel.

bExecute: Edge-triggered signal for execution of the command.

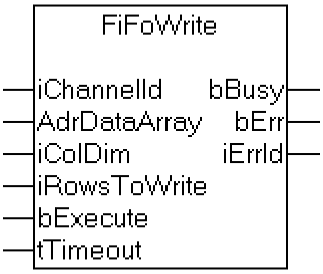
tTimeout: ADS timeout (about 1 second).

bBusy: Becomes TRUE with a rising edge at bExecute, and remains TRUE until the block has executed the command.

bErr: Becomes TRUE if an error occurs while executing the command.

bErrId: Error number (ADS or NC error number).

7.4 FiFoWrite



FiFoWrite reads position data for *iColDim* FIFO channel axes from *AdrDataArray* and transfers *iRowsToWrite* entries to the FIFO.

Interface

```
VAR_INPUT
    iChannelId    : UDINT;
    AdrDataArray  : POINTER TO LREAL; (* PLC: ARRAY[ ROWS, COLUMNS ] OF LREAL *)
    iColDim       : UDINT; (* second array dimension (COLUMNS) *)
    iRowsToWrite  : UDINT; (* number of rows to write <= RowDim *)
    bExecute      : BOOL;
    tTimeout      : TIME;
END_VAR
VAR_OUTPUT
    bBusy   : BOOL;
    bErr    : BOOL;
    iErrId  : UDINT;
END_VAR
```

iChannelId: Channel ID of the FIFO channel.

AdrDataArray: The address of a data field containing position data for the master and slave axes. The first of the field's dimensions describes the table lines, and the second dimension describes the columns.

iColDim: Number of columns in the data field. This value must correspond to the actual size of the second field dimension.

iRowsToWrite: Number of table lines. This value may be less than or equal to the size of the first field dimension.

bExecute: Edge-triggered signal for execution of the command.

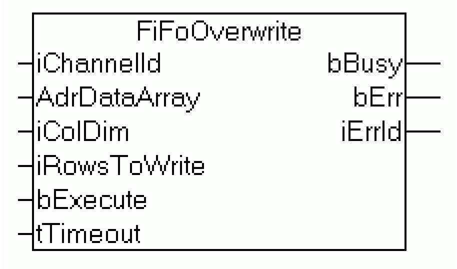
tTimeout: ADS timeout (about 1 second).

bBusy: Becomes TRUE with a rising edge at bExecute, and remains TRUE until the block has executed the command.

bErr: Becomes TRUE if an error occurs while executing the command.

bErrId: Error number (ADS or NC error number).

7.5 FiFoOverwrite



FiFoOverwrite reads position data for *iColDim* FIFO channel axes from *AdrDataArray* and transfers *iRowsToWrite* entries to the FIFO. Existing FIFO data will be overwritten.

Interface

```
VAR_INPUT
    iChannelId    : UDINT;
    AdrDataArray  : POINTER TO LREAL; (* PLC: ARRAY[ ROWS, COLUMNS ] OF LREAL *)
    iColDim       : UDINT; (* second array dimension (COLUMNS) *)
    iRowsToWrite  : UDINT; (* number of rows to write <= RowDim *)
    bExecute      : BOOL;
    tTimeout      : TIME;
END_VAR
VAR_OUTPUT
    bBusy   : BOOL;
    bErr    : BOOL;
    iErrId  : UDINT;
END_VAR
```

iChannelId: Channel ID of the FIFO channel.

AdrDataArray: The address of a data field containing position data for the master and slave axes. The first of the field's dimensions describes the table lines, and the second dimension describes the columns.

iColDim: Number of columns in the data field. This value must correspond to the actual size of the second field dimension.

iRowsToWrite: Number of table lines. This value may be less than or equal to the size of the first field dimension.

bExecute: Edge-triggered signal for execution of the command.

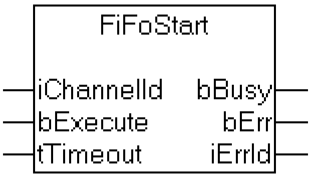
tTimeout: ADS timeout (about 1 second).

bBusy: Becomes TRUE with a rising edge at bExecute, and remains TRUE until the block has executed the command.

bErr: Becomes TRUE if an error occurs while executing the command.

bErrId: Error number (ADS or NC error number).

7.6 FiFoStart



FiFoStart initiates processing of the FIFO channel, so that all the axes in the FIFO channel are driven in accordance with the position data previously provided to the FIFO.

Interface

```
VAR_INPUT
    iChannelId  : UDINT;
    bExecute    : BOOL;
    tTimeout    : TIME;
END_VAR
VAR_OUTPUT
    bBusy   : BOOL;
    bErr    : BOOL;
    iErrId  : UDINT;
END_VAR
```

iChannelId: Channel ID of the FIFO channel.

bExecute: Edge-triggered signal for execution of the command.

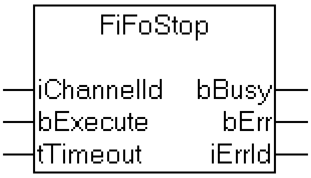
tTimeout: ADS timeout (about 1 second).

bBusy: Becomes TRUE with a rising edge at bExecute, and remains TRUE until the block has executed the command.

bErr: Becomes TRUE if an error occurs while executing the command.

bErrId: Error number (ADS or NC error number).

7.7 FiFoStop



FiFoStop halts a FIFO channel. Processing of the position data from the FIFO stops, and all the axes in the FIFO channel are halted.

Interface

```
VAR_INPUT
    iChannelId  : UDINT;
    bExecute    : BOOL;
    tTimeout    : TIME;
END_VAR
VAR_OUTPUT
    bBusy    : BOOL;
    bErr     : BOOL;
    iErrId   : UDINT;
END_VAR
```

iChannelId: Channel ID of the FIFO channel.

bExecute: Edge-triggered signal for execution of the command.

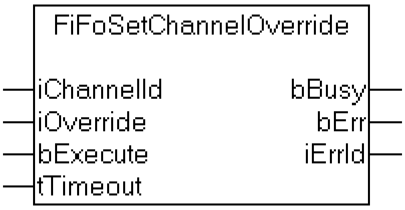
tTimeout: ADS timeout (about 1 second).

bBusy: Becomes TRUE with a rising edge at bExecute, and remains TRUE until the block has executed the command.

bErr: Becomes TRUE if an error occurs while executing the command.

bErrId: Error number (ADS or NC error number).

7.8 FiFoSetChannelOverride



FiFoSetChannelOverride sets the channel override by means of which the processing speed of the FIFO channel can be affected. The channel override should not be confused with the axis overrides for the individual axes.

Interface

```
VAR_INPUT
    iChannelId  : UDINT;
    iOverride   : UDINT;
    bExecute    : BOOL;
    tTimeout    : TIME;
END_VAR
VAR_OUTPUT
    bBusy   : BOOL;
    bErr    : BOOL;
    iErrId  : UDINT;
END_VAR
```

iChannelId: Channel ID of the FIFO channel.

iOverride: Channel override for the speed of the FIFO channel.

bExecute: Edge-triggered signal for execution of the command.

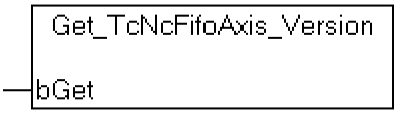
tTimeout: ADS timeout (about 1 second).

bBusy: Becomes TRUE with a rising edge at bExecute, and remains TRUE until the block has executed the command.

bErr: Becomes TRUE if an error occurs while executing the command.

bErrId: Error number (ADS or NC error number).

7.9 Get_TcNcFifoAxis_Version



Get_TcNcFifoAxis_Version determines the version number of the PLC library TcNcFifoAxis.lib. The function returns the version number in a string.

Interface

```
FUNCTION Get_TcNcFifoAxis_Version : STRING(20)
VAR_INPUT
    bGet : BOOL;
END_VAR
```

bGet: Signal for execution of the command.

8 Examples

8.1 FIFO Axes

In many applications it is necessary to synchronize two or more axes. Axes can be coupled together in the TwinCAT NC PTP. A master axis is then actively controlled, and the position of one or more coupled slave axes is synchronously controlled by the NC.
The simplest type of coupling is linear coupling with a fixed ratio of transmission (an electronic gearbox).

Some applications require a more complex coupling of master and slave, one which can not be described by a simple mathematical formula. Such a dependency can be described by means of a table that specifies an associated slave position for every master position.
The TwinCAT NC PTP offers the possibility of coupling a slave axis to a master axis by means of a table (electronic camshaft). Here the table contains a certain number of prescribed reference points, and the NC interpolates position and speed between them.

FIFO axes are here collected into a FIFO group. These extend the table concept for any number of sequences that do not usually recur cyclically. In this case, rather than having one master-slave table prescribed in advance, the latest axis positions are constantly "topped up" by a PLC program. Because the FIFO axes are combined into a FIFO group, synchronised axis motions can be implemented.

The present document describes the FIFO groups with the aid of a PLC program that is explained step by step.

NC FIFO axis group

If one or more axes are to be operated in a FIFO group, certain preparatory steps are necessary:

- All axes must be taken to a base position in PTP mode.
- The axes are collected into the FIFO group.
- The FIFO memory is fully or partially filled with axis position data.
- The FIFO group is started, and works through the positions that have so far been entered into to the table, taking them from the FIFO memory.
- While this is taking place the PLC is constantly topping up the FIFO memory with further FIFO data, until the positioning comes to a logical end.

There is more information about this in the TwinCAT FIFO [▸ 8] chapter.

8.1.1 Installing and Running

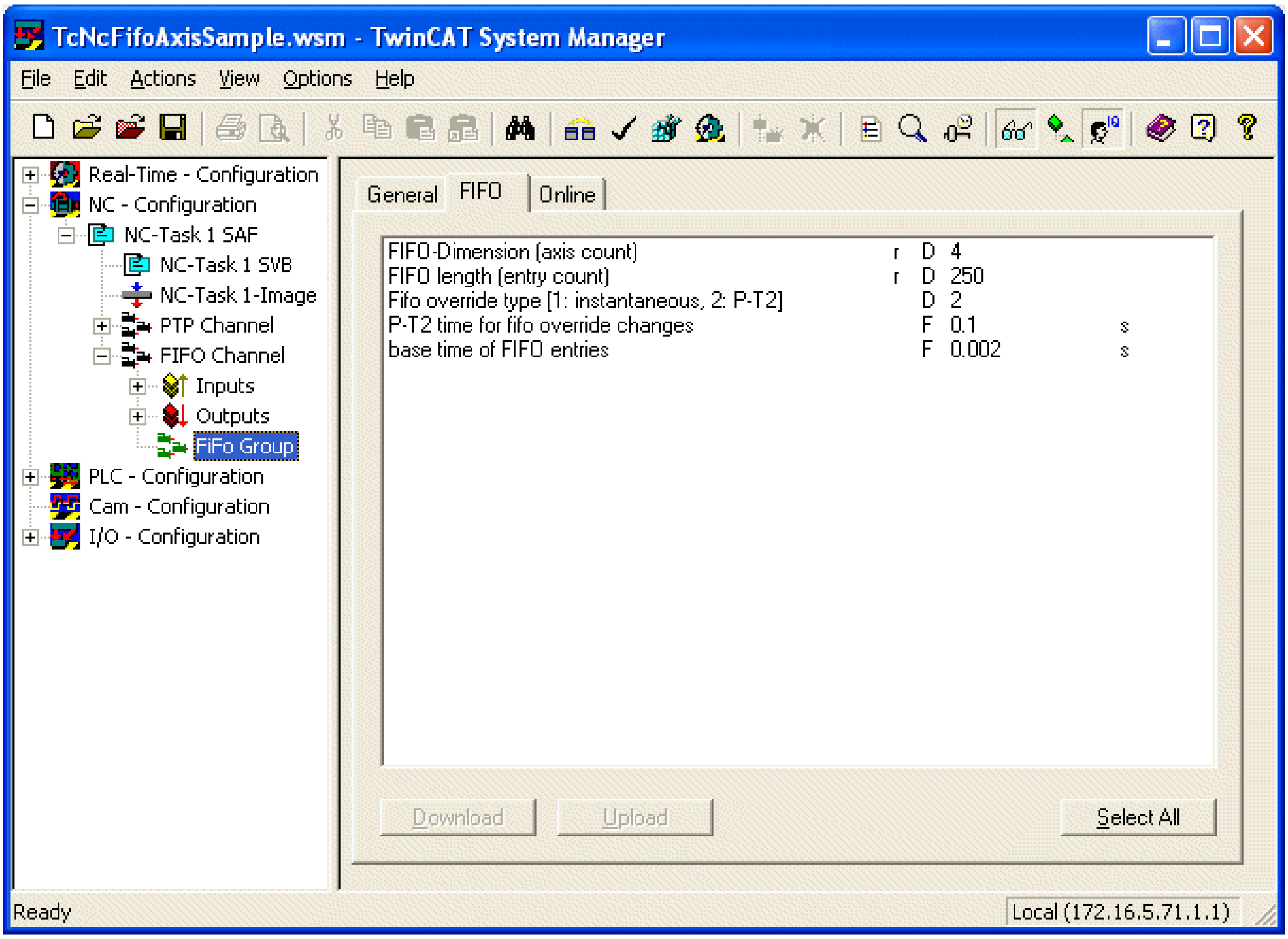
The FIFO Axes example program reads in the position data for a FIFO group from a file, feeding it to the group’s FIFO memory. The FIFO group is started, continuing until the data stream comes to an end. The data stream can only come to an end if the logical end of the positioning process has been reached, or if the "topping up" data rate is not adequate.

- Download FIFO Axes example program: https://infosys.beckhoff.com/content/1033/ts5060_tcnc_fifoaxes/Resources/3456013195/.exe

The example program uses the library TcNcFifoAxis.lib [▸ 18] which can be ordered as an additional product. It uses simulation axes (axes with simulated encoders) and can thus be run on any Windows NT PC without additional hardware. The file *TcNcFifoAxisSample.wsm* is loaded into the TwinCAT System Manager, and the system is started with this setting.

The NC configuration contains a PTP channel with four PTP axes and a FIFO channel that does not initially contain any axes. The FIFO group is configured under the FIFO channel. The FIFO memory has room for 250 position entries for four axes.

The FIFO store’s timebase is 2 ms. In this case the timebase is synchronous with the NC timebase, but this does not have to be the case. The timebase for the FIFO entries determines the interval at which a new axis positioning value is taken out of the FIFO memory. There is more information about this in the TwinCAT FIFO [▸ 8] chapter.



FIFO-Dimension (axis count)	r	D	4	
FIFO length (entry count)	r	D	250	
Fifo override type [1: instantaneous, 2: P-T2]		D	2	
P-T2 time for fifo override changes		F	0.1	s
base time of FIFO entries		F	0.002	s

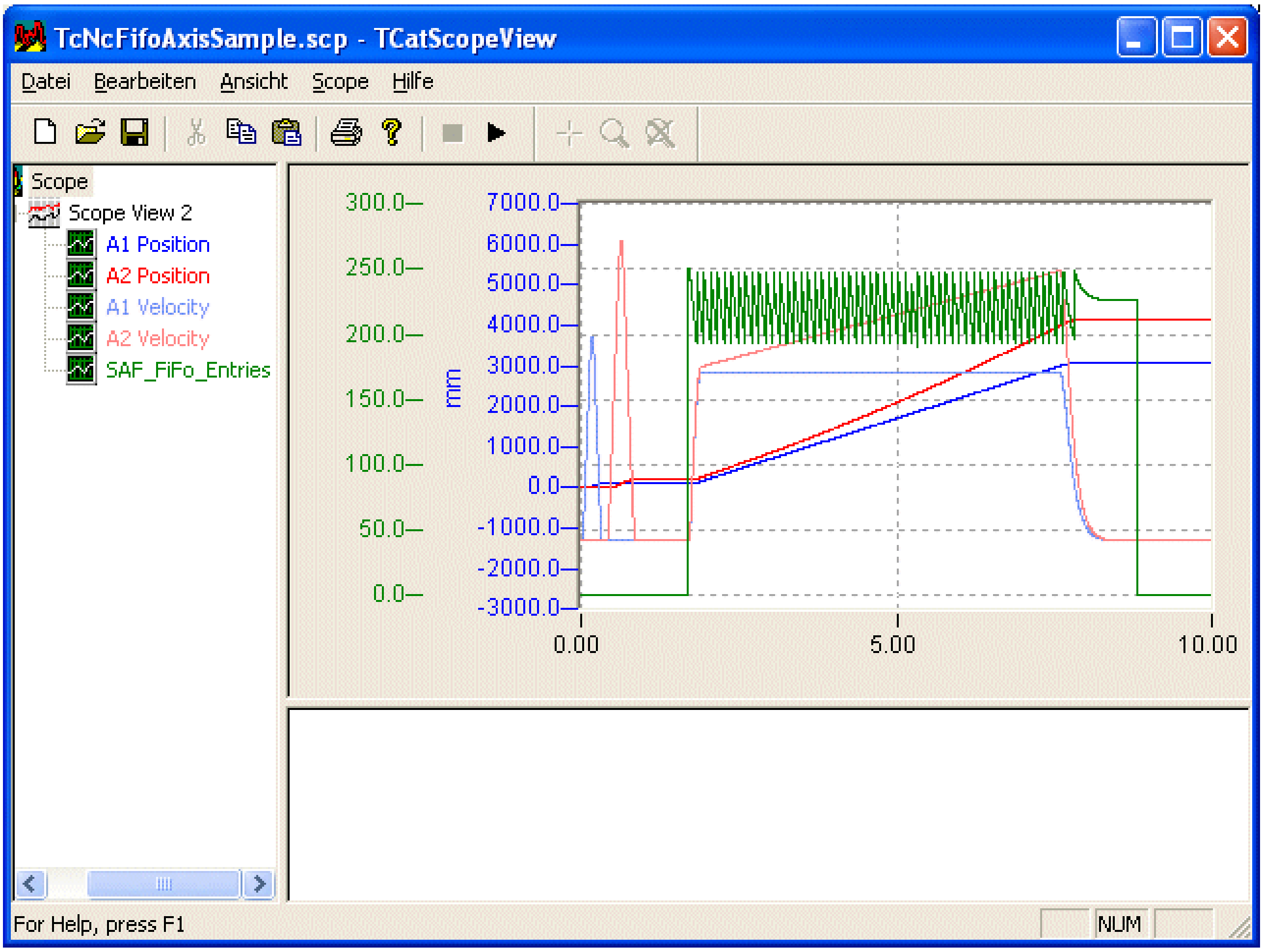
The axis speed results from the difference in position of two neighbouring FIFO entries divided by the FIFO timebase. This speed can be affected by altering the channel override. The override can be varied between 0% and 100% during the positioning. A value of 0% brings the axis to a stop, yet without halting the positioning task. The override types are described in the "Override" section.

The example program assumes that a file with position data exists in the directory *C:\CamDesign*, although the directory can nevertheless be changed as described further below.

- FifoTableBig5.Bin

The example program, *TcNcFifoAxisSample.pro*, is loaded into TwinCAT PLC Control, compiled and run.

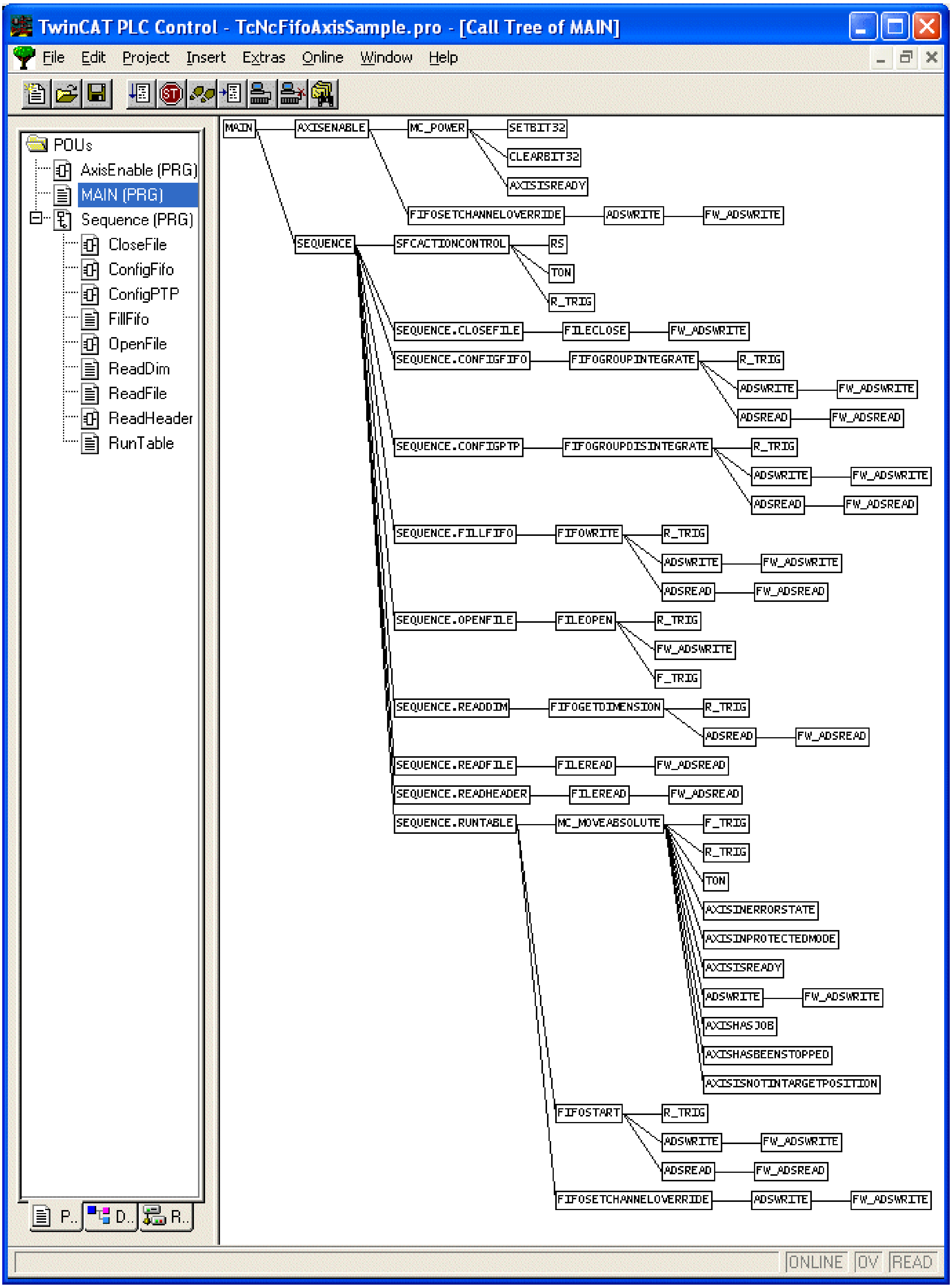
TwinCAT Scope is used in order to be able to examine the result. The supplied setting, *TcNcFifoAxisSample.scp*, allows the axis movements to be recorded.



8.1.2 Program Structure

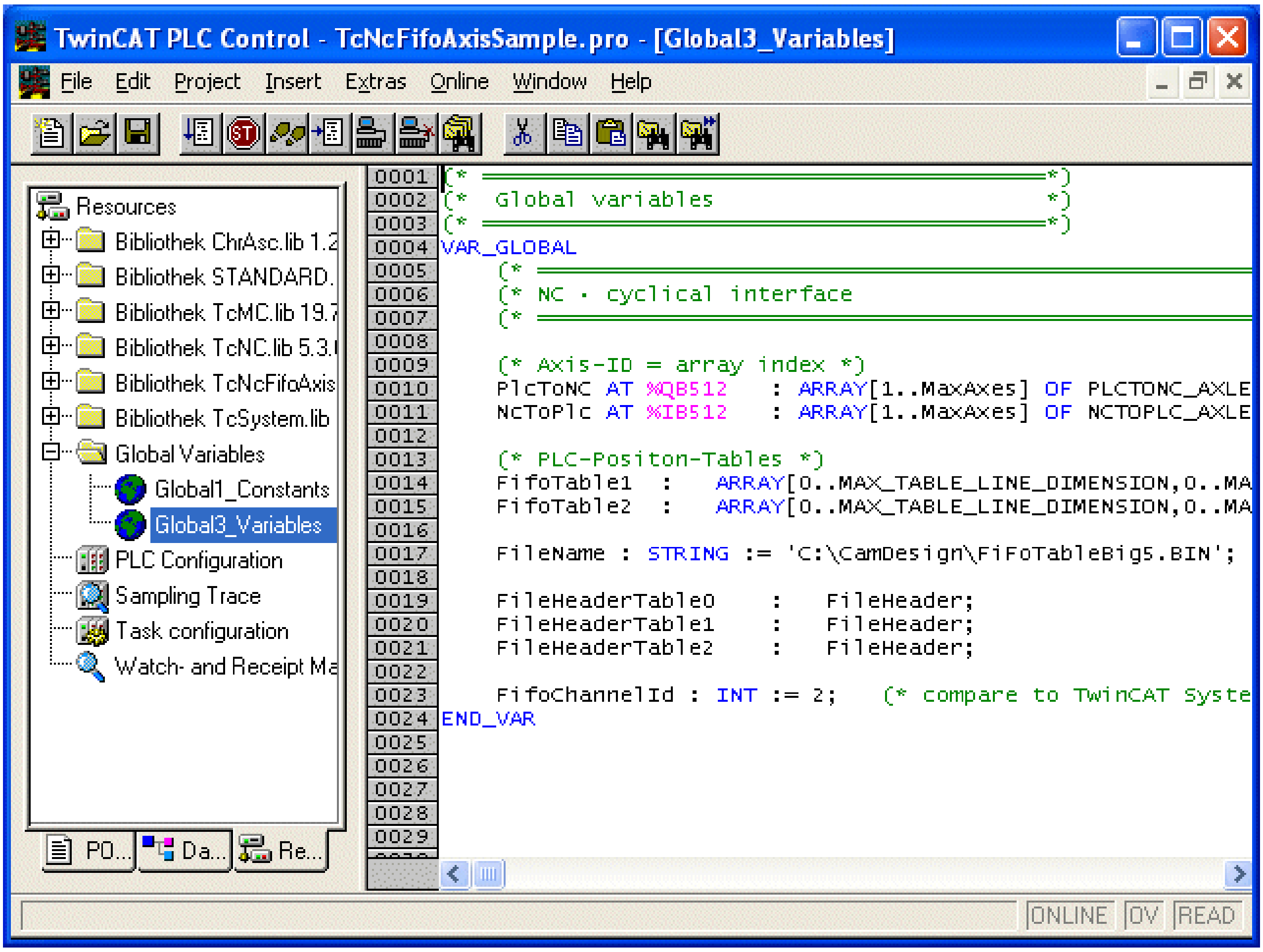
The *Main* module calls the *AxisEnable* and *Sequence* blocks. The *AxisEnable* block activates all the necessary enable signals for the axes, and will not be described in any more detail at this point.

The *Sequence* program module controls the procedure for configuration of the FIFO group, topping up of the FIFO data and driving of the axes.


TwinCAT PLC Control - TcNcFifoAxisSample.pro - [Call Tree of MAIN]
File Edit Project Insert Extras Online Window Help
POUs
AxisEnable (PRG)
MAIN (PRG)
Sequence (PRG)
CloseFile
ConfigFifo
ConfigPTP
FillFifo
OpenFile
ReadDim
ReadFile
ReadHeader
RunTable
P..
D..
R..
MAIN
AXISENABLE
MC_POWER
SETBIT32
CLEARBIT32
AXISISREADY
FIFOSETCHANNELOVERRIDE
ADSWRITE
FW_ADSWRITE
SEQUENCE
SFCACTIONCONTROL
RS
TON
R_TRIG
SEQUENCE.CLOSEFILE
FILECLOSE
FW_ADSWRITE
SEQUENCE.CONFIGFIFO
FIFOGROUPINTEGRATE
R_TRIG
ADSWRITE
FW_ADSWRITE
ADSREAD
FW_ADSREAD
SEQUENCE.CONFIGPTP
FIFOGROUPDISINTEGRATE
R_TRIG
ADSWRITE
FW_ADSWRITE
ADSREAD
FW_ADSREAD
SEQUENCE.FILLFIFO
FIFOWRITE
R_TRIG
ADSWRITE
FW_ADSWRITE
ADSREAD
FW_ADSREAD
SEQUENCE.OPENFILE
FILEOPEN
R_TRIG
FW_ADSWRITE
F_TRIG
SEQUENCE.READDIM
FIFOGETDIMENSION
R_TRIG
ADSREAD
FW_ADSREAD
SEQUENCE.READFILE
FILEREAD
FW_ADSREAD
SEQUENCE.READHEADER
FILEREAD
FW_ADSREAD
SEQUENCE.RUNTABLE
MC_MOVEABSOLUTE
F_TRIG
R_TRIG
TON
AXISINERRORSTATE
AXISINPROTECTEDMODE
AXISISREADY
ADSWRITE
FW_ADSWRITE
AXISHASJOB
AXISHASBEENSTOPPED
AXISISNOTINTARGETPOSITION
FIFOSTART
R_TRIG
ADSWRITE
FW_ADSWRITE
ADSREAD
FW_ADSREAD
FIFOSETCHANNELOVERRIDE
ADSWRITE
FW_ADSWRITE
ONLINE
OV
READ

8.1.3 Global Variables

The position data file that is to be used is specified in the global variables. The filename, including the path to the file, can be altered here. The axis and channel interfaces are also declared here, and data buffers for reading the FIFO data out of a file are created.



```
(* ==========================================*)
(*  Global variables                          *)
(* ==========================================*)
VAR_GLOBAL
     (* ==============================================
     (* NC · cyclical interface
     (* ==============================================

     (* Axis-ID = array index *)
     PlcToNC AT %QB512    : ARRAY[1..MaxAxes] OF PLCTONC_AXLE
     NcToPlc AT %IB512    : ARRAY[1..MaxAxes] OF NCTOPLC_AXLE

     (* PLC-Positon-Tables *)
     FifoTable1  :    ARRAY[0..MAX_TABLE_LINE_DIMENSION,0..MA
     FifoTable2  :    ARRAY[0..MAX_TABLE_LINE_DIMENSION,0..MA

     FileName : STRING := 'C:\CamDesign\FiFoTableBig5.BIN';

     FileHeaderTable0     :    FileHeader;
     FileHeaderTable1     :    FileHeader;
     FileHeaderTable2     :    FileHeader;

     FifoChannelId : INT := 2;    (* compare to TwinCAT Syste
END_VAR
```

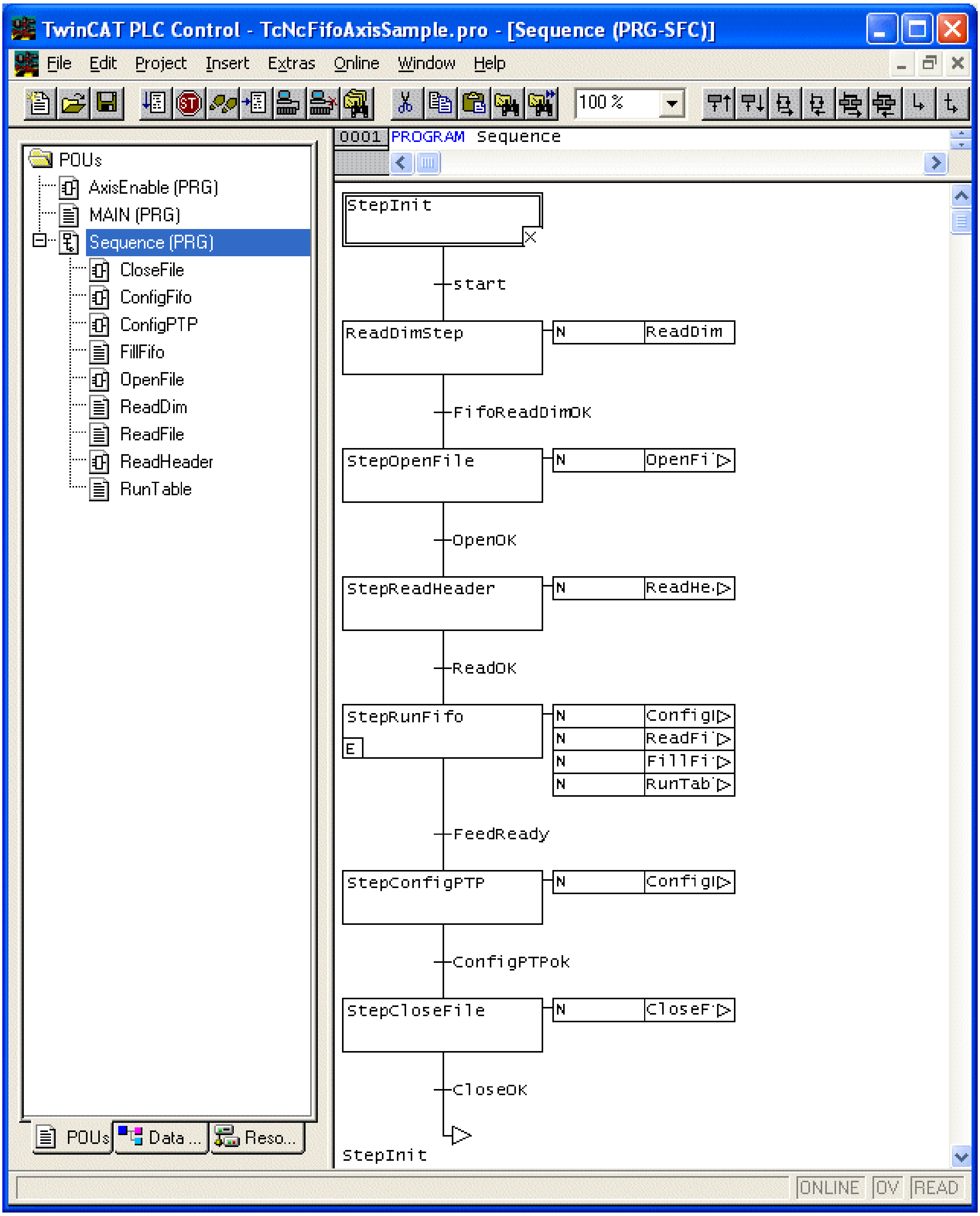
8.1.4 Function Block Sequence

The *Sequence* function block controls the sequence of events.

- In the step *ReadDimStep*, the dimensions configured in the TwinCAT System Manager, i.e. the number of axes and the number of lines in the FIFO memory, are read.
- After this, the position file is opened, and the file header read. The file header contains the number of columns and rows in the file.
- The most important step is *StepReadTable*. Here the data is read from the file into two alternately written buffers, each of about 64 kbytes in size (action *ReadFile*). The FIFO memory is continuously refilled from this intermediate buffer, keeping it as full as possible (action *FillFifo*). When the FIFO memory has been filled for the first time, the axes are started (action *RunTable*).
- After all the positions have been worked through, the FIFO group stops, and the group is dissolved by *StepConfigPTP*.
- Finally, the file is closed.

For the purposes of demonstration, this program reads the axis positions out of a file. The position data can of course be made available by quite different methods. They can, for example, also be directly calculated at run-time, or acquired by a measurement system.

The running of the program should not be interrupted at unsuitable times, for instance in order to load a new program version into the TwinCAT run-time system. This can lead to files remaining open, with a following inability to re-open them when the PLC next starts. In such cases the sequence will stop with an error message in one of the steps. If that happens, start the TwinCAT System again by means of the TwinCAT icon in task bar.



```
PROGRAM Sequence
```

More Information:
www.beckhoff.de/ts5060

Beckhoff Automation GmbH & Co. KG
Hülshorstweg 20
33415 Verl
Germany
Phone: +49 5246 9630
info@beckhoff.com
www.beckhoff.com