Beckhoff TwinCAT

Using the Beckhoff Stepper Motor Controller

User Level: Intermediate

Revision: 2.11
Updated: 25 January 2008

Table Of Contents

Current Library Versions:

StepperMotor.lib Version 2.7
StepperMotorBX.lbx Version 2.7
StepperMotorBC.lb6 Version 2.7
StepperMotorNC.lib Version 2.0

Revision	Updated	Changes
V2.11	25/01/08	Libraries updated to V2.7. **SM_MoveVelo** modified so that to update velocity on every PLC cycle while **Execute** remains TRUE
V2.10	10/01/08	New StepperMotorNC library (V2.0) added for PC/CX platform containing new **SM_ConfigKL2531NC** and **SM_ConfigKL2541NC** function blocks that allow a KL2531 or KL2541 terminal linked to a TwinCAT NC axis to be configured from the PLC program
V2.9	8/01/08	Section 3.3 added to detail min and max parameter values and internal parameter resolution for **MinVelocity**, **MaxVelocity**, **MaxAccel** and **AccelThres**
V2.8	28/09/07	Libraries updated to V2.6. Error Code 6 added for **SM_MovePos** and **SM_MovePosAndVelo**
V2.7	22/02/07	Minor corrections to Done outputs for **SM_ConfigKL2531** and **SM_MicroStepsPerRev**
V2.6	08/12/06	Libraries updated to V2.5. Additional functionality and information added to **SM_MoveVelo**
V2.5	29/11/06	Libraries updated to V2.4. Additional information added to clarify the operation of **SM_ReadActPosition**
V2.4	16/11/06	Libraries updated to V2.3 with addition of new function block, **SM_MicroStepsPerRev**. Allows the number of micro-steps per rev for the motor to be read from the KL25*1 without needing to use **SM_ConfigKL2531** or **SM_ConfigKL2541**
V2.3	19/04/06	Error in maximum motor speed under NC controller corrected
V2.2	21/03/06	Automatic EEPROM parameter store removed from all FBs except **SM_ConfigKL2531** and **SM_ConfigKL2541**
V2.1	10/03/06	**SM_ReadActPosition** changed to allow continuous tracking of motor position while enabled
V2.0	1/2/06	Support for BX and BC platforms added to V2.0 libraries. Input output axis structures changed to 6-bytes to ensure word alignment on BC/BX platform. Timeout delay inputs added to FBs that use terminal register access
V1.3	23/1/06	Libraries for BC/BC removed. Word alignment causes a problem when linking the input and output axis structures to the terminal
V1.2	18/11/05	Libraries for BC/BX platforms added
V1.1	18/11/05	Additional info for KL2531/KL2541 configuration in KS2000, and TwinCAT NC axis configuration
V1.0	10/11/05	Renamed and restructured to include TwinCAT NC control of a stepper motor terminal and axis
V0.1		Original document for beta library V0.1

1. Introduction

The Beckhoff stepper motor controllers allow stepper motor axes to be integrated into a TwinCAT control system. The stepper motor controllers can be driven by a standard TwinCAT NC axis, or they can be controlled directly from a TwinCAT PLC program. The basic specifications for each terminal are shown in the table below

Terminal	Max Current per phase	Max Voltage	Max Output Frequency	Max Micro-step Frequency	Encoder Input
KL2531	1.5A	24Vdc	7.5kHz	125kHz	No
KL2541	5A	50Vdc	7.5kHz	125kHz	Yes

1.1. TwinCAT NC Axis Control

The Beckhoff stepper motor controllers can be linked to a standard TwinCAT NC axis, and configured and controlled in the same way as any other servo axis. The function blocks in library file StepperMotorNC.LIB can be used to configure the KL2531 or KL2541 terminal in this mode.

StepeprMotor.Lib **CANNOT** be used with a TwinCAT NC axis as the terminal process data is linked to a TwinCAT NC axis, and NOT to the TwinCAT PLC.

1.2. TwinCAT PLC Control

A Stepper Motor Controller library is available from Hayes Control Systems Ltd. This library contains function blocks that implement basic position and velocity control within a TwinCAT PLC program, but without using a TwinCAT NC axes. This allows basic position and velocity control of stepper motors to be implemented in the PC, CX, BX and BC range of controllers.

NOTE: Stepper Motor Controller library V2.0 is needed for the BX and BC controllers. V1.0 does NOT work on these platforms.

Function Block	Description
SM_ConfigKL2531	Allows the terminal to be configured from the PLC program at run-time
SM_ConfigKL2541	
SM_ConfigKL2531NC	Allows the terminal to be configured from the PLC program at run-time when linked to a TwinCAT NC axis
SM_ConfigKL2541NC	
SM_MicroStepsPerRev	Reads the configured number of microsteps per revolution
SM_Power	Enable the power output to the motor
SM_Stop	Cancel the move in progress and stop the motor
SM_Reset	Reset terminal from an error condition
SM_Status	Returns the terminal status information
SM_ReadActPosition	Reads the full 32-bit motor position
SM_HoldCurrent	Selects from 2 preset current values to be applied to the motor when the motor is at standstill
SM_MovePos	Move to absolute target position
SM_MovePosAndVelo	Move to absolute target position at specified velocity and acceleration ramps
SM_MoveVelo	Move at specified velocity
SM_SetActPosition	Sets the current motor position to a new value

2. TwinCAT NC Axis Control

Getting Started Note **"Basic Configuration of a TwinCAT NC Axis"** (available from Hayes Controls Systems Ltd) gives full details for implementing an axis using TwinCAT NC, and this document assumes that the user is familiar with this process. This chapter gives details that are different for implementing a stepper motor axis with TwinCAT NC, as opposed to implementing a standard servo axis with TwinCAT NC.

2.1. Axis Calculations

If an external encoder is not being used for an axis, position feedback is generated by the stepper motor terminal based on the number of output pulses. In this case, the maximum position accuracy and maximum velocity are dependant on each other, so consideration must be given to the axis requirements before the stepper motor controller is configured.

2.1.1. Positional accuracy

The number of position steps per motor revolution depends on the step angle of the motor and the configured number of micro-steps in the stepper controller.

$$\textbf{Position Steps} = \frac{\textbf{360° * Microsteps}}{\textbf{Motor Step Angle}}$$

For a motor with 1.8° step angle (200 steps/rev) running with 64-fold micro-stepping, the positional accuracy is 12800 increments per motor revolution. This value can be applied to the mechanical system to give the positional accuracy for the axis.

2.1.2. Speed Resolution

The KL2531 and KL2541 stepper motor controllers use a signed 16-bit velocity reference, so the maximum velocity is 32767 position steps per second. The table below shows the positional accuracy that can be obtained for a given number of micro-steps, and the corresponding maximum possible speed for the axis.

Micro-steps	Step positions/rev (for 1.8° step motor)	Revs/sec	Revs/min
64	12800	2.56	153.6
32	6400	5.12	307.2
16	3200	10.24	614.4

It can be seen that a higher number of micro-steps will give better positional accuracy from the motor, but at the cost of a lower maximum motor speed. A lower number of micro-steps will give a better top speed, but with lower positional accuracy.

2.1.3. Maximum motor speed

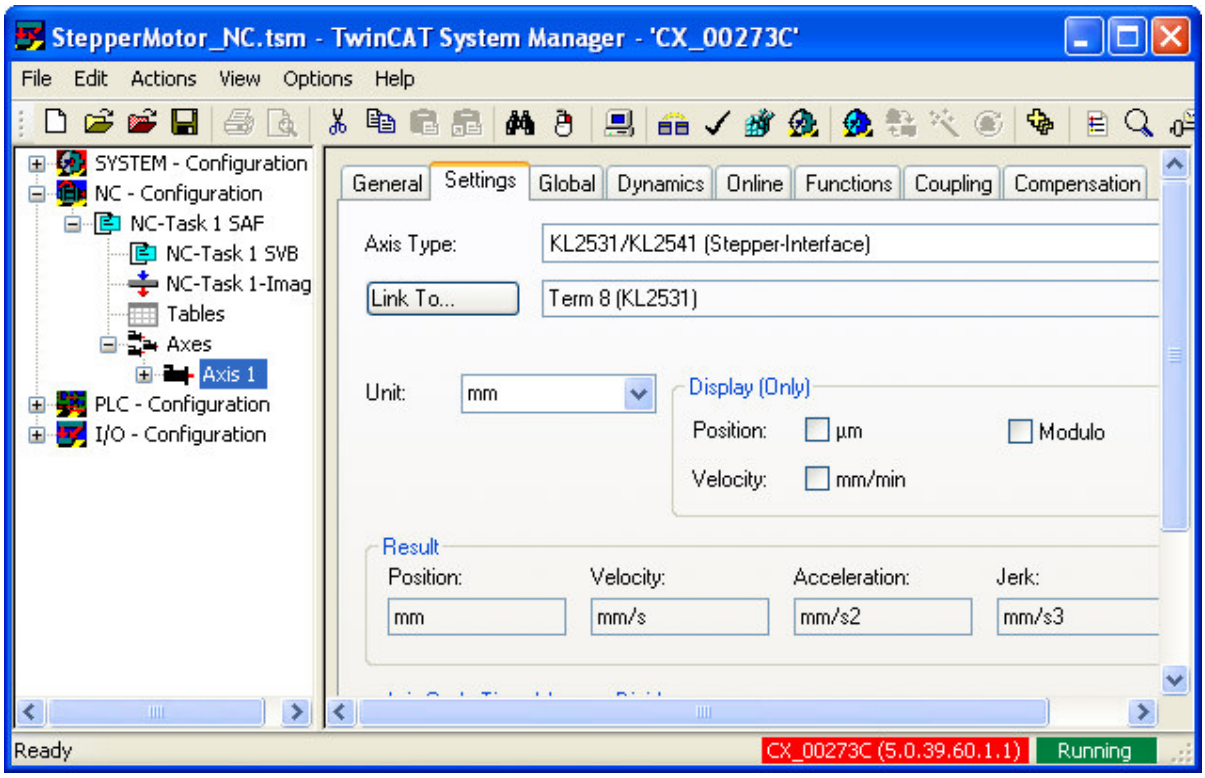
The KL2531 and KL2541 stepper motor controllers use a signed 16-bit velocity reference, where full scale output (32767) gives the maximum output speed of 125000 micro- steps per second.

Micro-steps	Step positions/rev (for 1.8° step motor)	Revs/sec (max)	Revs/min (max)
64	12800	9.77	586.2
32	6400	19.53	1171.8
16	3200	39.06	2343.6

2.2. Linking TwinCAT NC Axis to a Stepper Motor Terminal

To link a TwinCAT NC axis to a stepper motor terminal:

1. Setting the axis type to “KL2531/KL2541 (Stepper-Interface)”
2. Click “Link To…”
3. Select the required stepper motor controller terminal
4. Download and activate the TwinCAT System Manager configuration

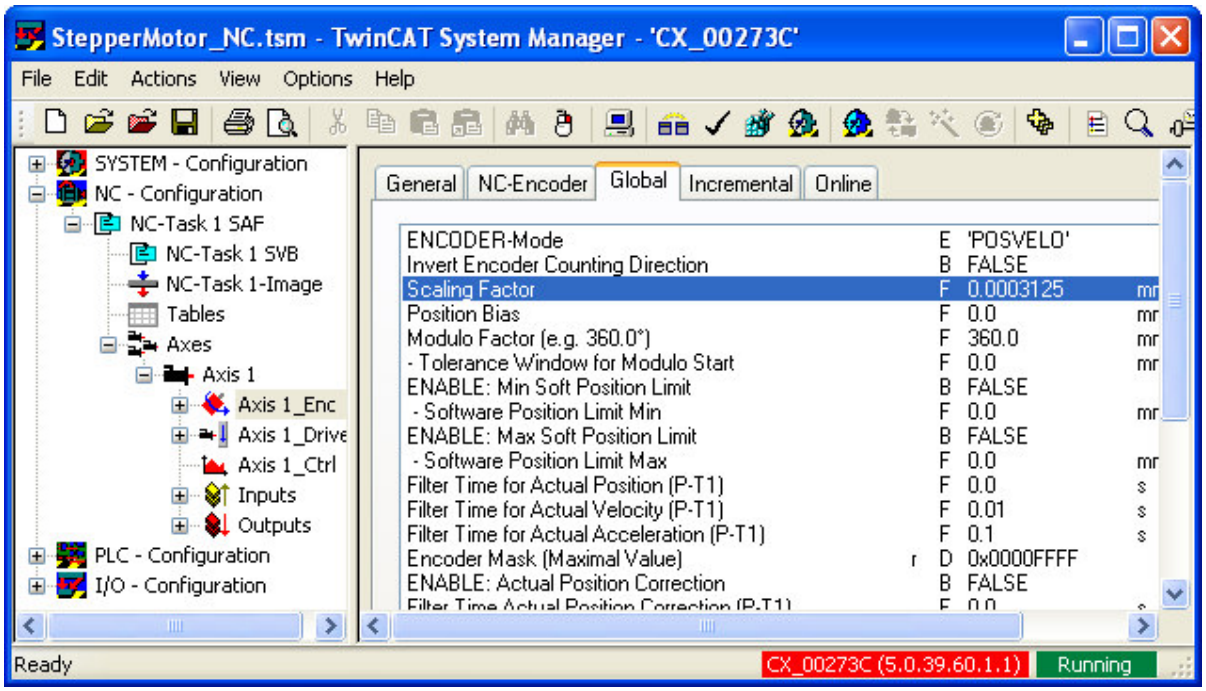


2.3. TwinCAT NC Axis Scaling Factor

The Axis Scaling Factor specifies how far the axis will move (in mm or °) for one position step change on the motor shaft. When the internal step count is being used for position feedback, **`SM_ConfigKL2531NC`** and **`SM_ConfigKL2541NC`** will return a value called **`MicroStepsPerRev`**. This is the number of increments that will be fed back to the TwinCAT NC axis for 1 revolution of the motor.

To set the Scaling Factor for the axis:

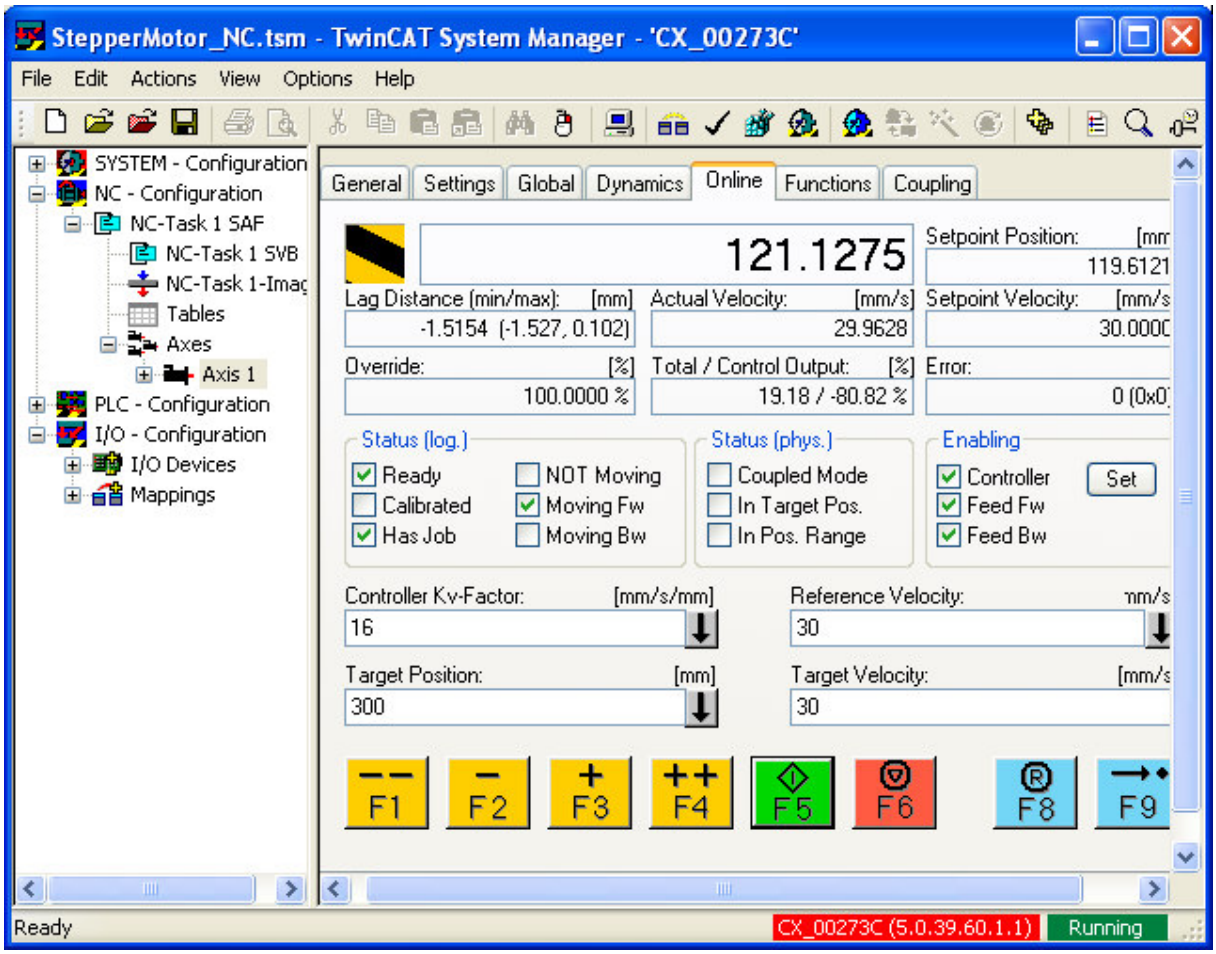
1. Set "Scaling Factor" to the required value
2. Press ENTER
3. Click "Download"
4. Select "Save Now"



2.4. TwinCAT NC Axis Position Loop Gain

As TwinCAT NC uses velocity control to move a stepper motor axis with KL2531 and KL2541, the position loop proportional gain (Kv) will need to be adjusted to tune the performance on the axis and reduce the following error. A good starting point for Kv is to set it to 50% of the number of micro-steps being used on the axis.

1. Set “Controller Kv-Factor:” to the required value. A typical value when using the internal step count as feedback is 10.
2. Click on the “” arrow to update the value
3. Monitor the following error in the “Lag Distance” window when the axis is moving. The minimum and maximum following errors during the move are shown
4. Click the F8 RESET button to reset the minimum and maximum following error values when the axis at standstill

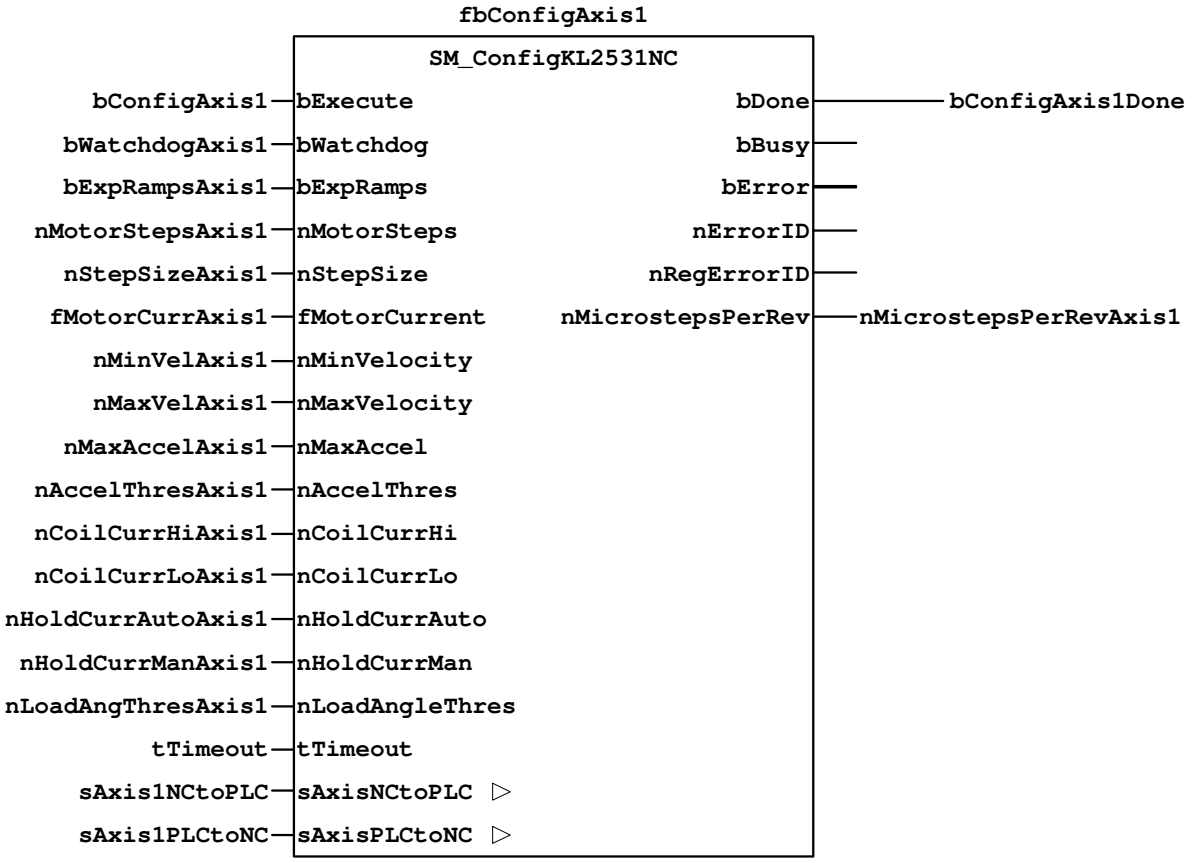


When the axis has been configured and can be controlled from the “Online” tab in TwinCAT System Manager, TwinCAT PLC control can be implemented in the way as for a normal servo axis. Full details are given in the application note “**PTP Motion Control**” and this document is available from Hayes Control Systems Ltd.

2.5. SM_ConfigKL2531NC

SM_ConfigKL2531NC is used configure the KL2531 parameter set from the PLC program when the KL2531 is linked to a TwinCAT NC axes. The TwinCAT NC axis **MUST** be disabled while the KL2531 terminal configuration is updated.

For full details of all inputs, refer to Section 3.4, "SM_ConfigKL2531" on page 12.

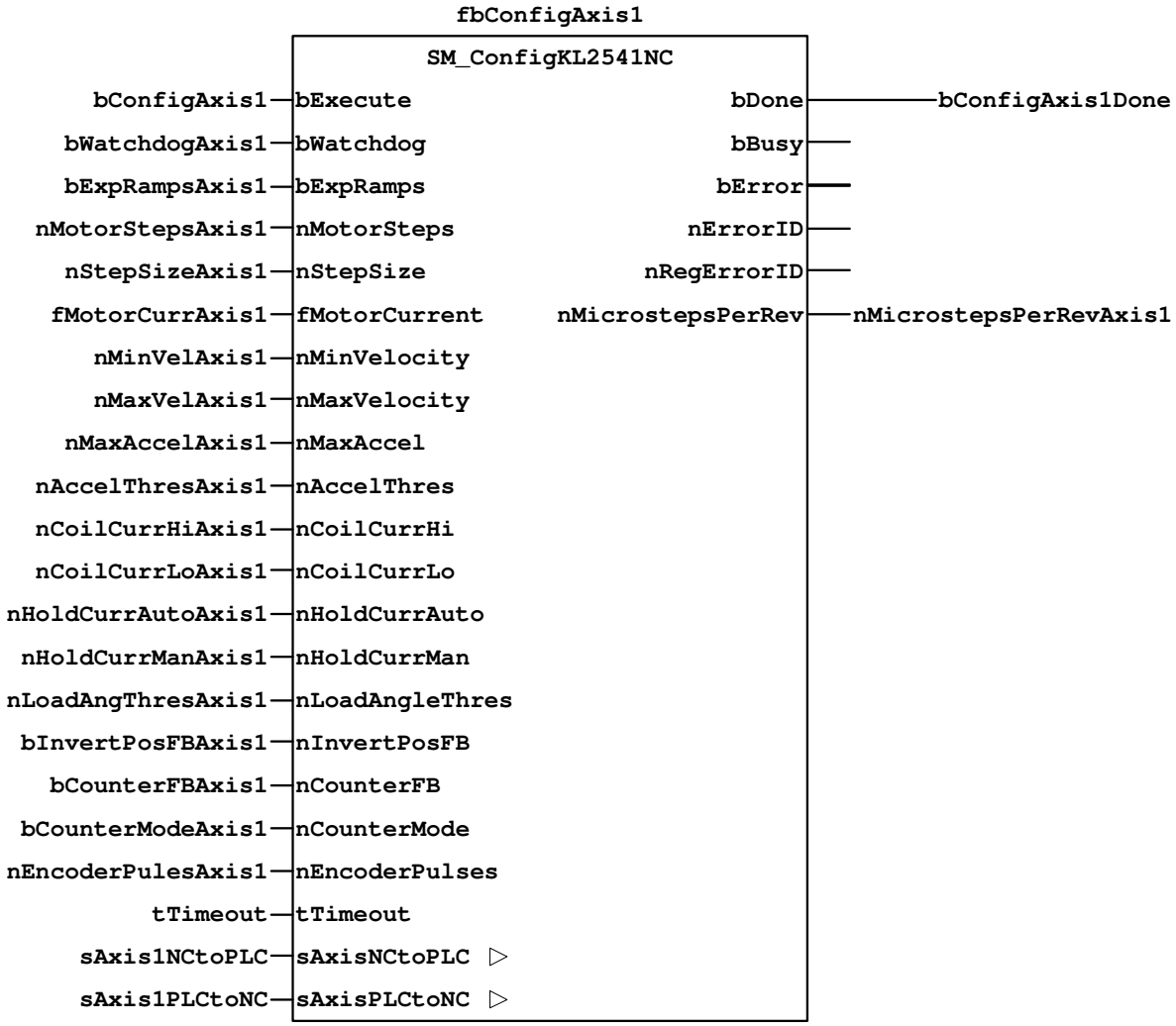


Input	Data type	Description
tTimeout	**TIME**	Timeout delay to prevent FB from locking up in the event of an unexpected error. In general, allow 5s per 1ms of the TwinCAT NC cycle time
sNCtoPLCAxis	**NCTOPLC_AXLESTRUCT**	TwinCAT NC axis input data structure
sPLCtoNCCAxis	**PLCTONC_AXLESTRUCT**	TwinCAT NC axis output data structure
Output	**Data type**	**Description**
bBusy	**BOOL**	Set TRUE while access to KL2531 is active
nRegErrorID	**UDINT**	Indicates error code is register access failed

2.6. SM_ConfigKL2541NC

SM_ConfigKL2541 is used configure the KL2541 parameter set from the PLC program, and write the register values to EEPROM memory. The TwinCAT NC axis **MUST** be disabled while the KL2531 terminal configuration is updated.

For full details of all inputs, refer to Section 3.5, "SM_ConfigKL2541" on page 14



Input	Data type	Description
tTimeout	**TIME**	Timeout delay to prevent FB from locking up in the event of an unexpected error. In general, allow 5s per 1ms of the TwinCAT NC cycle time
sNCtoPLCAxis	**NCTOPLC_AXLESTRUCT**	TwinCAT NC axis input data structure
sPLCtoNCCAxis	**PLCTONC_AXLESTRUCT**	TwinCAT NC axis output data structure
Output	**Data type**	**Description**
bBusy	**BOOL**	Set TRUE while access to KL2531 is active
nRegErrorID	**UDINT**	Indicates error code is register access failed

3. TwinCAT PLC Control

Basic position and velocity control can be implemented using the internal position controller in the KL2531 and KL2541 terminals, thus allowing stepper motors to be controlled directly from the TwinCAT PLC program. This allows basic stepper motor control to be implemented on BX controllers. The appropriate library file for the target TwinCAT platform must be included in the PLC project file.

2 function blocks are provided to allow basic configuration of the KL2531 and KL2541 terminals when running under direct TwinCAT PLC control. KS2000 is only required for advanced configuration of the KL2531 and KL2541 terminals.

3.1. TwinCAT PLC Configuration

The library file for the target platform must be added to the TwinCAT PLC project:

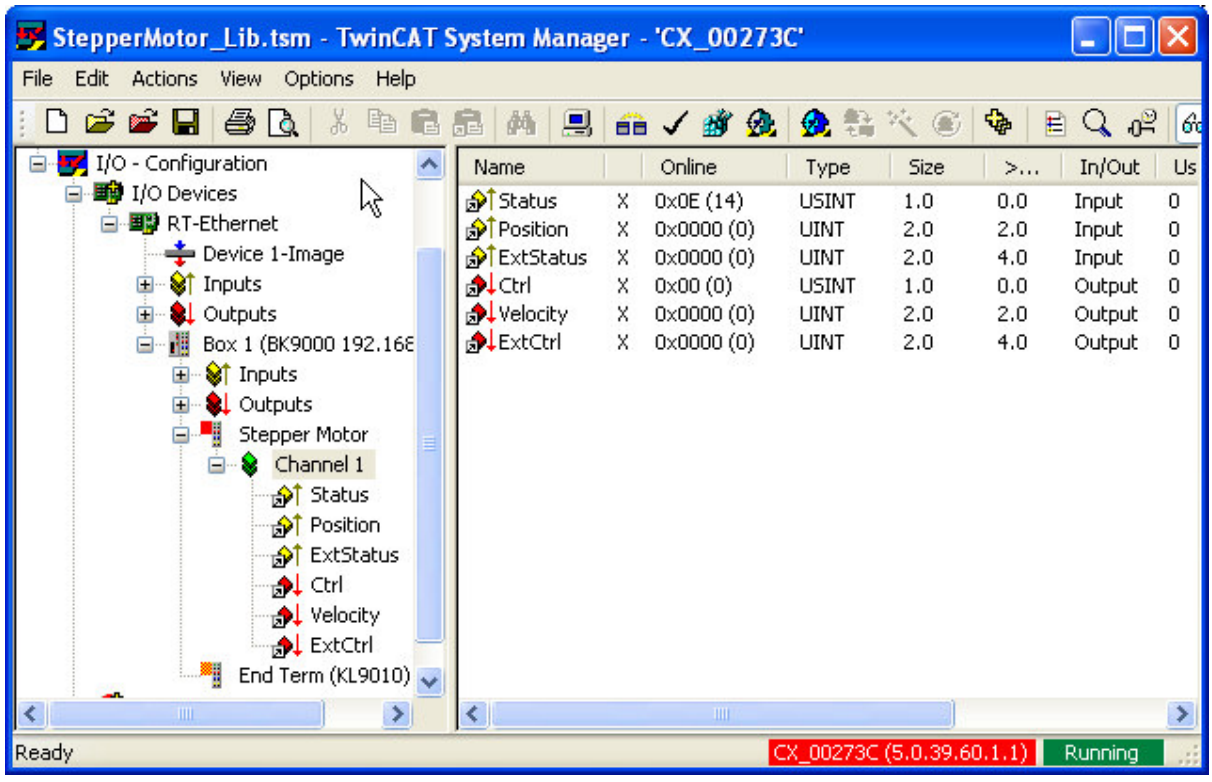
PC	StepperMotor.LIB
BX/BCxx50	StepperMotorBX.LBX
BCxx00	StepperMotorBC.LB6

An input (**S_SMDataIn** in **%I**) and output (**S_SMDataOut** in **%Q**) data structure must be declared for each KL2531 and KL2541 terminal used in the system. Each structure contains a total of 6 data bytes.

```
SMAxis1In    AT    %IB0  :  S_SMDataIn;
SMAxis1Out   AT    %QB0  :  S_SMDataOut;
SMAxis2In    AT    %IB6  :  S_SMDataIn;
SMAxis2Out   AT    %QB6  :  S_SMDataOut;
```

3.2. TwinCAT System Manager Configuration

The KL2531 and KL2541 stepper terminals must be linked to an input and output data structure in TwinCAT System Manager. The terminal variables and data structure elements have identical names, so the links for each terminal can be easily made.



3.3. Range and Resolution of Velocity and Acceleration values

KL2531 and KL2541 use integer values internally, so velocity and acceleration parameters have limited range and resolution, as shown in the table below. Velocity values are shown in microsteps/sec and acceleration values are shown in microsteps/sec^2.

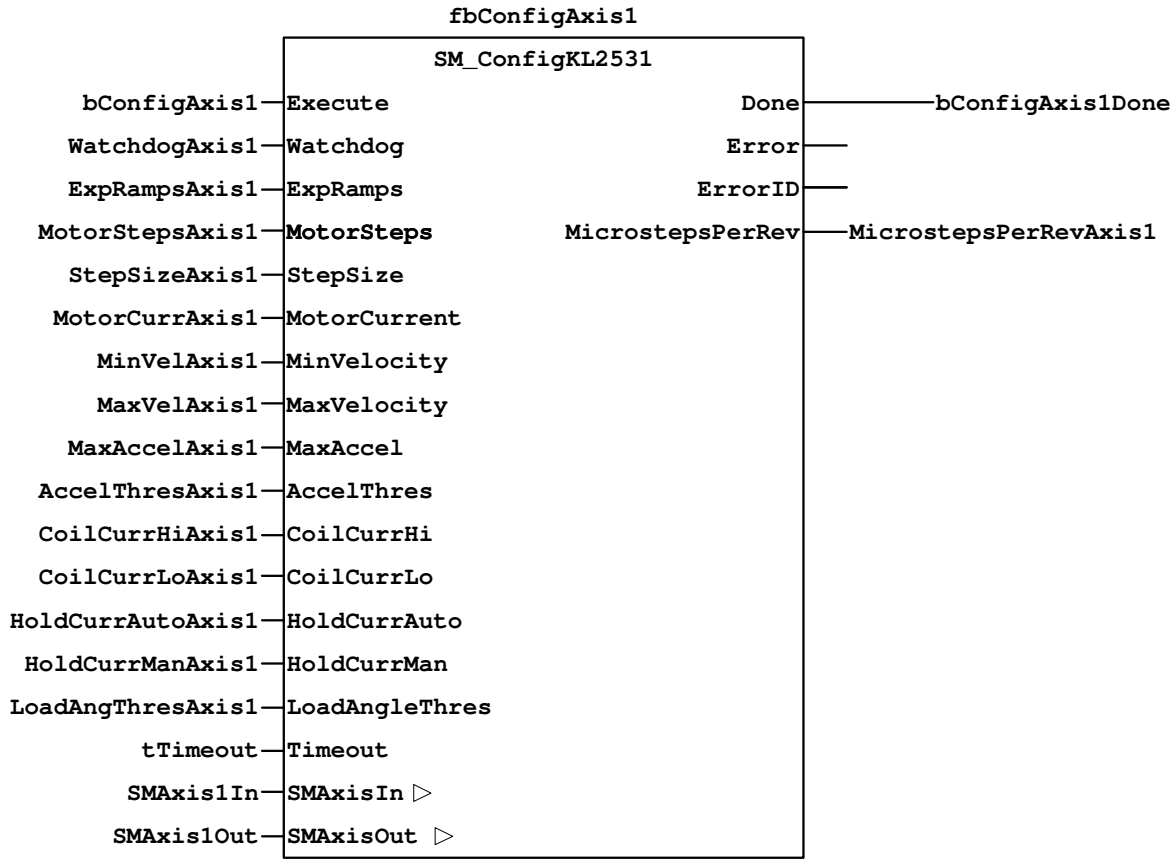
Parameter	Min Value	Max Value	Resolution
MinVelocity	0	125000	61.035
MaxVelocity	0	125000	61.035
MaxAccel	931	1906417	931
AccelThres	931	1906417	931

For a given input value for the above parameters, the effective value that will be produced by the terminal will the nearest multiple of the resolution value.

If a value of 25000 is set for **MaxVelocity**, the internal value will be 409.6 so this will be rounded to the nearest integer value of 410. The actual setting in the terminal will therefore be 410 * 61.035 = 25024 microsteps/sec.

3.4. SM_ConfigKL2531

SM_ConfigKL2531 is used configure the KL2531 parameter set from the PLC program, and write the register values to EEPROM memory. The motor should be disabled while the KL2531 terminal configuration is updated.

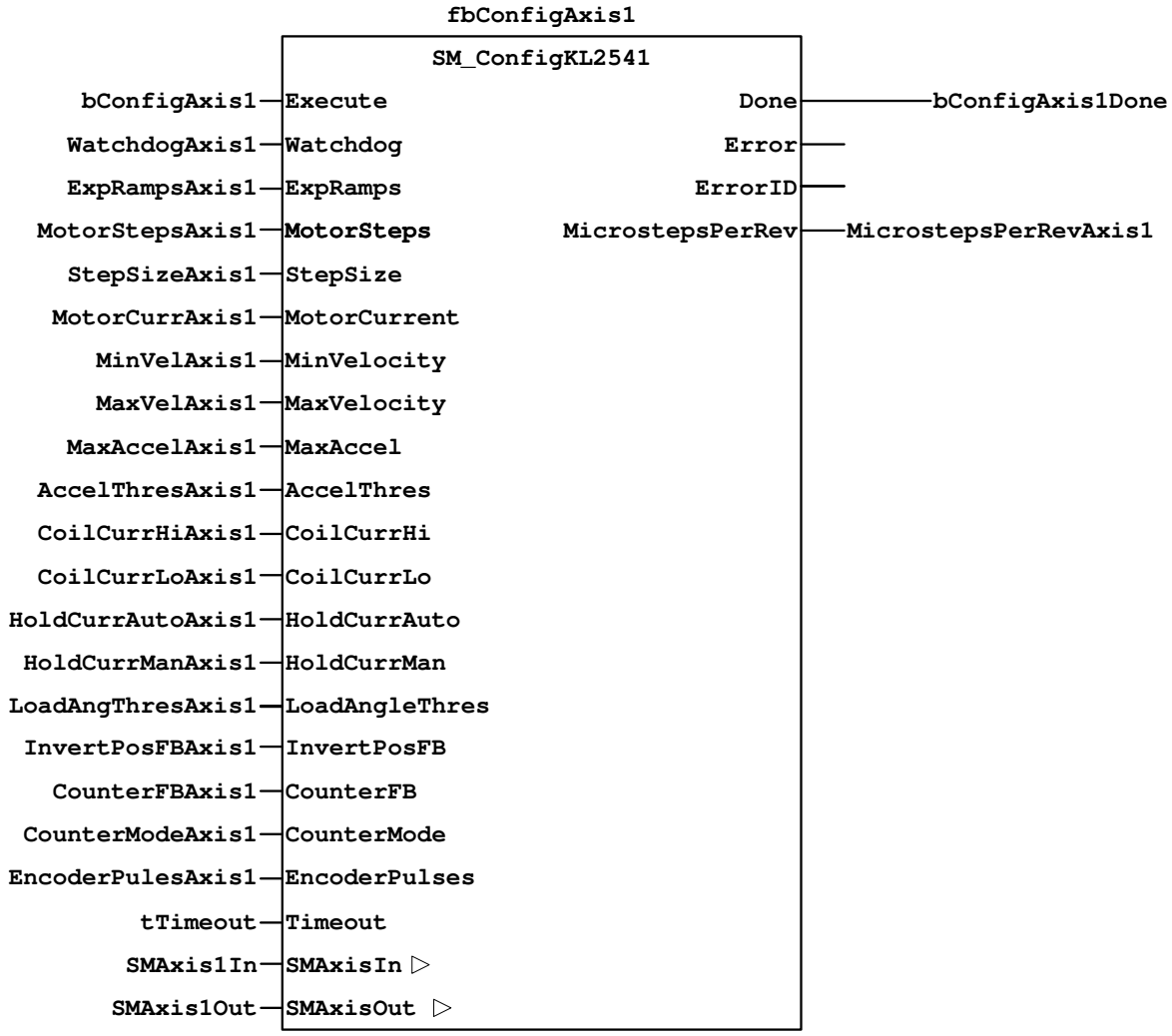


Input	Data type	Register	Description
Execute	**BOOL**		The command is activated on a rising edge
Watchdog	**BOOL**	R32.2	Set to TRUE to enable the internal 100ms watchdog
ExpRamps	**BOOL**	R32.5	Set to FALSE to use linear ramps Set to TRUE to use exponential ramps
MotorSteps	**UINT**	R33	Specifies the number of motor steps per rev
StepSize	**UINT**	R46	Selects the number of micro-steps per motor step Micro-steps = 2^{StepSize}, where **StepSize** = 0 to 6
MotorCurrent	**REAL**	R35, R36	Specifies the rated motor current in Amps KL2531 = 1.5A max, KL2541 = 5.0A max
MinVelocity	**UDINT**	R38	The maximum velocity from which the motor can stop directly without step errors in micro-steps/sec
MaxVelocity	**UDINT**	R39	Maximum velocity in micro-steps/sec
MaxAccel	**UDINT**	R40	Maximum accel/decel rate in microsteps/sec^2
AccelThres	**UDINT**	R41	Acceleration threshold in microsteps/sec^2
CoilCurrHi	**UINT**	R42	Specifies the coil current when accel rate > accel threshold. 100 = 100% of **MotorCurrent**
CoilCurrLo	**UINT**	R43	Specifies the coil current when accel rate <= accel threshold. 100 = 100% of **MotorCurrent**
HoldCurrAuto	**UINT**	R44	Automatic holding current applied to the motor at standstill. 100 = 100% of **MotorCurrent**
HoldCurrMan	**UINT**	R45	Manual holding current applied to the motor at standstill 100 = 100% of **MotorCurrent**
LoadAngleThres	**UINT**	R47	Specifies the load angle threshold from which the StallDetect LED starts flashing
Timeout	**TIME**		Timeout delay for terminal register access
SMAxisIn	**S_SMDataIn**		Stepper motor axis input data structure
SMAxisOut	**S_SMDataOut**		Stepper motor axis output data structure

Output	Data Type	Description
Done	**BOOL**	Set to TRUE when the function has completed
Error	**BOOL**	Set to TRUE if an error occurred
ErrorID	**UDINT**	Returns the error code if **Error** is TRUE
MicroSteps PerRev	**UDINT**	Returns the number of micro-steps per motor revolution for the current terminal configuration when **Done** is TRUE and **Error** is FALSE

3.5. SM_ConfigKL2541

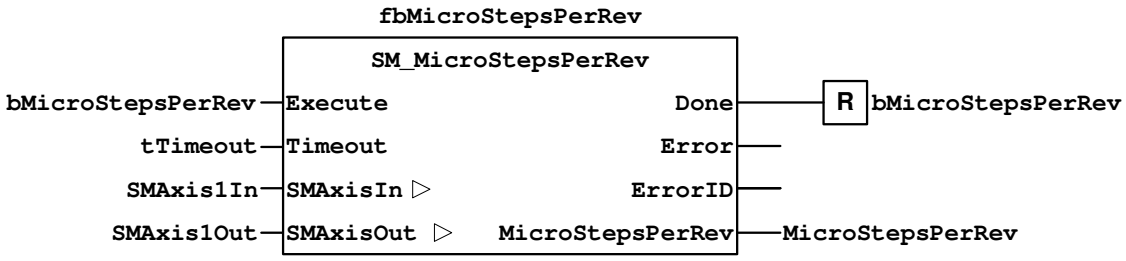
SM_ConfigKL2541 is used configure the KL2541 parameter set from the PLC program, and write the register values to EEPROM memory. The motor should be disabled while the KL2541 terminal configuration is updated. **SM_ConfigKL2541** has the same inputs as **SM_ConfigKL2531** plus the extra parameters shown in the table below.



Input	Data type	Register	Description
InvertPosFB	**BOOL**	R32.6	Set to TRUE to invert the encoder feedback
CounterFB	**BOOL**	R32.11	Set to TRUE to select internal counter feedback or FALSE to select encoder feedback
CounterMode	**BOOL**	R32.15	Set to TRUE to select Counter Mode or FALSE to select Encoder Mode
EncoderPulses	**UINT**	R34	Specifies the number of encoder pulses per rev

3.6. SM_MicroStepsPerRev

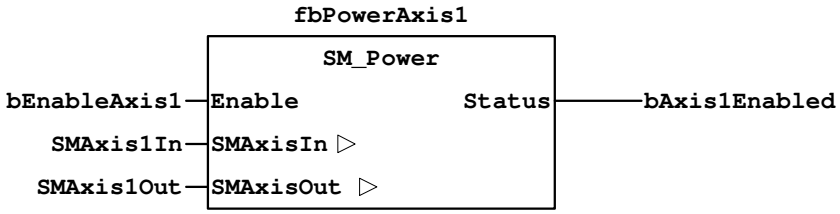
SM_MicroStepsPerRev allows the number of motor micro-steps per rev to be read from the KL25*1 terminal. It provides an alternative method to using **SM_ConfigKL2531** or **SM_ConfigKL2541** each time the system powers up.



Input	Data type	Description
Execute	**BOOL**	The command is activated on a rising edge
Timeout	**TIME**	Timeout delay for terminal register access
SMAxisIn	**S_SMDataIn**	Stepper motor axis input data structure
SMAxisOut	**S_SMDataOut**	Stepper motor axis output data structure
Output		
Done	**BOOL**	Set to TRUE when the function has completed
Error	**BOOL**	Set to TRUE if an error occurred
ErrorID	**UDINT**	Returns the error code if **Error** is TRUE
MicroStepsPerRev	**UDINT**	Returns the number of micro-steps per motor revolution for the current terminal configuration when **Done** is TRUE and **Error** is FALSE

3.7. SM_Power

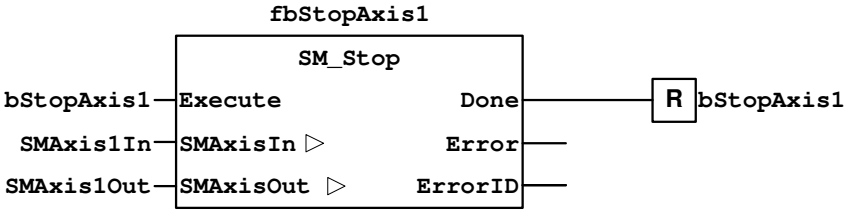
SM_Power is used to enable and disable the motor outputs. The KL25*1 terminals must be enabled before any motor move commands can be implemented. The present enable status of the terminal is also given.



Input	Data type	Description
Enable	**BOOL**	Enables and disables the motor outputs
SMAxisIn	**S_SMDataIn**	Stepper motor axis input data structure
SMAxisOut	**S_SMDataOut**	Stepper motor axis output data structure
Output		
Status	**BOOL**	Returns the enabled status of the terminal

3.8. SM_Stop

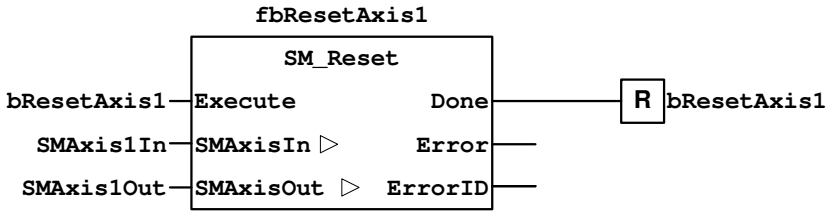
SM_Stop is used to cancel any existing move command and bring the motor to standstill. The **Done** output is set when the motor has reached zero speed.



Input	Data type	Description
Execute	**BOOL**	The command is activated on a rising edge
SMAxisIn	**S_SMDataIn**	Stepper motor axis input data structure
SMAxisOut	**S_SMDataOut**	Stepper motor axis output data structure
Output		
Done	**BOOL**	Set to TRUE when the motor has reached zero speed
Error	**BOOL**	Set to TRUE if an error occurred
ErrorID	**UDINT**	Returns the error code if **Error** is TRUE

3.9. SM_Reset

SM_Reset is used to clear any errors that may occur with the terminal. The current error status can be checked using **SM_Status**.



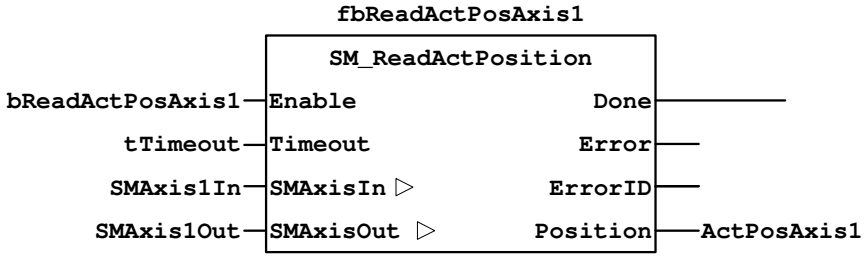
Input	Data type	Description
Execute	**BOOL**	The command is activated on a rising edge
SMAxisIn	**S_SMDataIn**	Stepper motor axis input data structure
SMAxisOut	**S_SMDataOut**	Stepper motor axis output data structure
Output		
Done	**BOOL**	Set to TRUE when the function has completed
Error	**BOOL**	Set to TRUE if an error occurred
ErrorID	**UDINT**	Returns the error code if **Error** is TRUE

3.10. SM_ReadActPosition

SM_ReadActPosition is used to track the absolute motor position. The **Position** output is valid while the **Done** output is set to TRUE.

The full 32-bit motor position is read from the KL2531/KL2541 internal registers on a rising edge of **Enable**, and the motor position is tracked using process data while the **Enable** input remains TRUE.

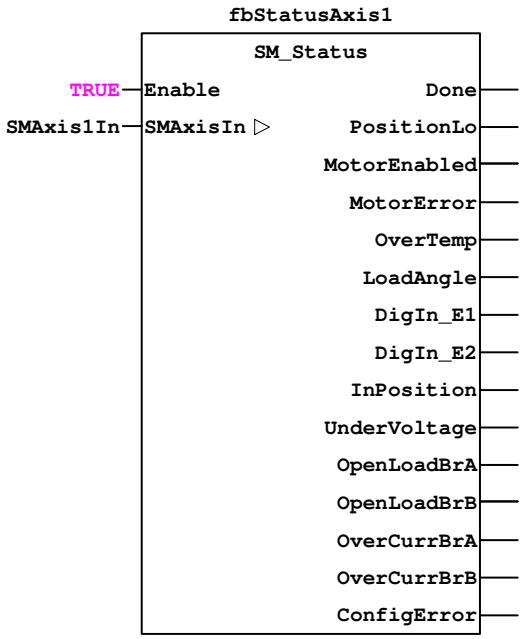
If **SM_SetActPosition** is used to change the axis position, **SM_ReadActPosition** will automatically re-read the full 32-bit position from the KL2531/KL2541internal registers.



Input	Data type	Description
Enable	**BOOL**	Full 32-bit motor position is read on rising edge. Motor position is tracked and updated while **Enable** is set to TRUE
Timeout	**TIME**	Timeout delay for terminal register access
SMAxisIn	**S_SMDataIn**	Stepper motor axis input data structure
SMAxisOut	**S_SMDataOut**	Stepper motor axis output data structure
Output		
Done	**BOOL**	Set to TRUE when the function has completed
Error	**BOOL**	Set to TRUE if an error occurred
ErrorID	**UDINT**	Returns the error code if **Error** is TRUE
ActPosition	**DINT**	Returns the current motor position when **Done** is TRUE and **Error** is FALSE

3.11. SM_Status

SM_Status returns the current operating status of the terminal. The status outputs are updated on each cycle if the **Execute** input is set to TRUE.



Input	Data type	Description
Enable	**BOOL**	Set to TRUE to monitor the status outputs
SMAxisIn	**S_SMDataIn**	Stepper motor axis input data structure
Output		
Done	**BOOL**	Set to TRUE when the status outputs are valid
PositionLo	**UINT**	Returns the low 16-bits of the absolute motor position
MotorEnabled	**BOOL**	Set to TRUE when the motor outputs are active
MotorError	**BOOL**	Set to TRUE when a status bits (marked '*' below) is TRUE
OverTemp	**BOOL**	Set to TRUE when motor over-temperature is detected
LoadAngle	**USINT**	Load angle gives an indication of the motor's mechanical load, and depends on the **MotorCurrent** and the current velocity
DigIn_E1	**BOOL**	Status of digital input E1
DigIn_E2	**BOOL**	Status of digital input E2
InPosition	**BOOL**	Set to TRUE when the motor has reached the target position
UnderVoltage	**BOOL**	* Supply voltage too low
OpenLoadBrA	**BOOL**	* Open circuit on Bridge A
OpenLoadBrB	**BOOL**	* Open circuit on Bridge B
OverCurrBrA	**BOOL**	* Over-current in Bridge A
OverCurrBrB	**BOOL**	* Over-current in Bridge B
ConfigError	**BOOL**	Configuration error

A configuration error can be caused by the following register values being set to 0:

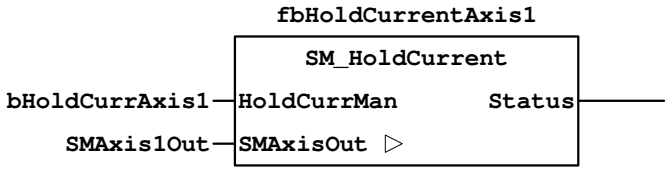
R33 **MotorSteps** (KL2541 only) R34 **EncoderPulses** (KL2541 only)

R38 **MinVelocity** R39 **MaxVelocity** R40 **MaxAccel**

R42 **CoilCurrHi** R43 **CoilCurrLo**

3.12. SM_HoldCurrent

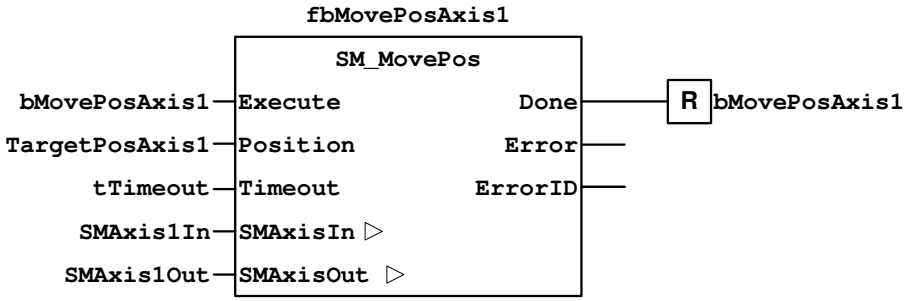
SM_HoldCurrent selects the current setting (**HoldCurrMan** or **HoldCurrAuto**) to be applied when the motor is at standstill. A higher current setting will give higher holding torque at zero speed, but will cause increase the self-heating effects within the motor. The **HoldCurrentAuto** setting is used by default.



Input	Data type	Description
HoldCurrentMan	**BOOL**	TRUE = **HoldCurrMan** FALSE = **HoldCurrAuto**
SMAxisIn	**S_SMDataIn**	Stepper motor axis input data structure
SMAxisOut	**S_SMDataOut**	Stepper motor axis output data structure
Output		
Status	**BOOL**	Indicates the holding current setting selected

3.13. SM_MovePos

SM_MovePos is used to move the motor to a specified absolute target position. The motor will use the existing **MaxVelocity** and **MaxAccel** settings to generate the velocity and acceleration profile for the position move.

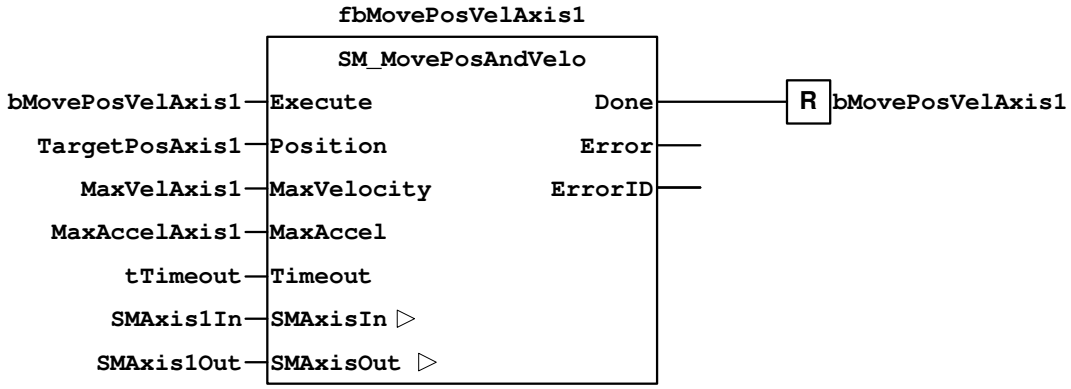


Input	Data type	Description
Execute	**BOOL**	The command is activated on a rising edge
Position	**DINT**	Specifies the absolute target motor position in micro-steps
Timeout	**TIME**	Timeout delay for terminal register access
SMAxisIn	**S_SMDataIn**	Stepper motor axis input data structure
SMAxisOut	**S_SMDataOut**	Stepper motor axis output data structure
Output		
Done	**BOOL**	Set to TRUE when the motor has reached the target position
Error	**BOOL**	Set to TRUE if an error occurred or move was interrupted
ErrorID	**UDINT**	Returns the error code if **Error** is TRUE

NOTE: the Timeout period only applies to terminal register access. It is NOT affected by the time taken for the motor to reach the target position.

3.14. SM_MovePosAndVelo

SM_MovePosAndVelo is used to move the motor to a specified absolute target position with the specified **MaxVelocity** and **MaxAccel** settings. These values are updated in the terminal prior to the position move, and will be retained in the terminal until over-written with new values.



Input	Data type	Description
Execute	**BOOL**	The command is activated on a rising edge
Position	**DINT**	Specifies the absolute target motor position in micro-steps
MaxVelocity	**UDINT**	Specifies the maximum velocity in micro-steps/sec
MaxAccel	**UDINT**	Specifies the maximum acceleration in micro-steps/sec^2
Timeout	**TIME**	Timeout delay for terminal register access
SMAxisIn	**S_SMDataIn**	Stepper motor axis input data structure
SMAxisOut	**S_SMDataOut**	Stepper motor axis output data structure
Output		
Done	**BOOL**	Set to TRUE when the motor has reached the target position
Error	**BOOL**	Set to TRUE if an error occurred or move was interrupted
ErrorID	**UDINT**	Returns the error code if **Error** is TRUE

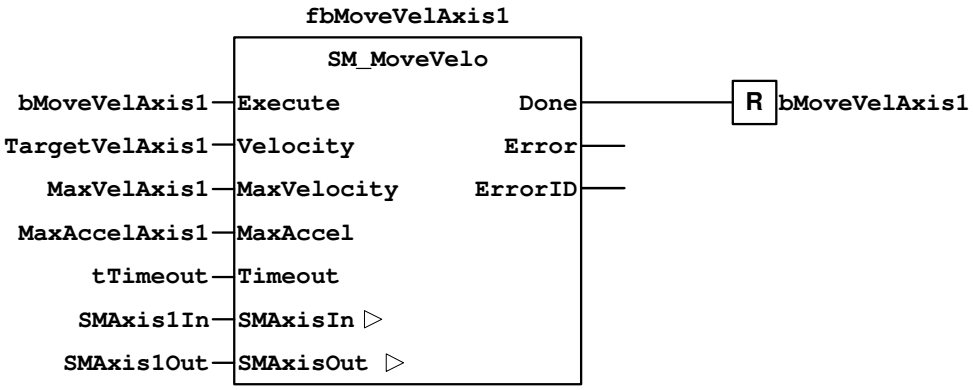
NOTE: the Timeout period only applies to terminal register access. It is NOT affected by the time taken for the motor to reach the target position.

3.15. SM_MoveVelo

SM_MoveVelo is used to run the motor at the specified velocity. **MaxVelocity** and **MaxAccel** can also be updated prior to the velocity. These values will be retained in the terminal until over-written with new values.

If **MaxVelocity** and/or **MaxAccel** input value is 0, the corresponding internal register values will NOT be updated.

When **Done** goes TRUE, **SM_MoveVelo** will update the velocity on every PLC cycle while **Execute** remains TRUE. Velocity will not be updated once **Execute** has been reset to FALSE.

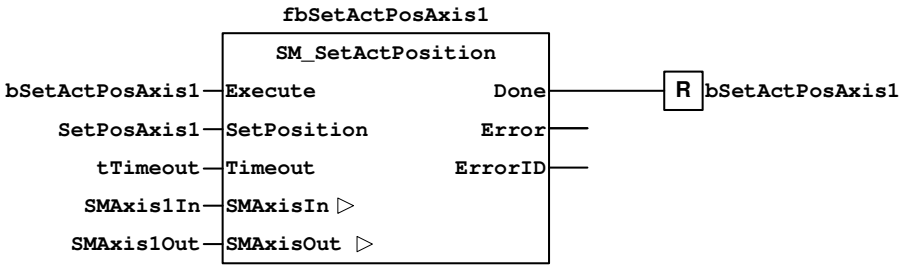


Input	Data type	Description
Execute	**BOOL**	The command is activated on a rising edge
Velocity	**DINT**	Specifies the target velocity in micro-steps/sec
MaxVelocity	**UDINT**	Specifies the maximum velocity in micro-steps/sec
MaxAccel	**UDINT**	Specifies the maximum acceleration in micro-steps/sec^2
Timeout	**TIME**	Timeout delay for terminal register access
SMAxisIn	**S_SMDataIn**	Stepper motor axis input data structure
SMAxisOut	**S_SMDataOut**	Stepper motor axis output data structure
Output		
Done	**BOOL**	Set to TRUE when the velocity command has been given
Error	**BOOL**	Set to TRUE if an error occurred
ErrorID	**UDINT**	Returns the error code if **Error** is TRUE

NOTE: the Timeout period only applies to terminal register access. It is NOT affected by the time taken for the motor to reach the target velocity.

3.16. SM_SetActPosition

SM_SetActPosition is used to set the current motor position to a specified value, e.g. as part of a homing routine. The motor must be disabled before the motor position can be updated.



Input	Data type	Description
Execute	**BOOL**	The command is activated on a rising edge
SetPosition	**DINT**	Specifies the new absolute position (in micro-teps)to over-write the actual motor position
Timeout	**TIME**	Timeout delay for terminal register access
SMAxisIn	**S_SMDataIn**	Stepper motor axis input data structure
SMAxisOut	**S_SMDataOut**	Stepper motor axis output data structure
Output		
Done	**BOOL**	Set to TRUE when the function has completed
Error	**BOOL**	Set to TRUE if an error occurred
ErrorID	**UDINT**	Returns the error code if **Error** is TRUE

4. Error Codes for Stepper Motor Function Blocks

The **ErrorID** outputs give a 32-bit error code in the format 16#00NNVVVV

NN	Error	Description
16#00	General error	See table below
16#01 to 16#3F	Register Write error	Returns the register number that did not verify correctly. 16#VVVV returns the attempted write value for the register
16#40 to 16#FF	N/A	Not used

The following table lists the General Error codes that may be seen when NN = 16#00.

CCCC	Error	Description
16#0001	Motor Enabled	Function can only be used when the motor is disabled
16#0002	Motor Disabled	Function can only be used when the motor is enabled
16#0003	Register Access Time-out	Function encountered a problem during register access and did not complete successfully
16#0004	Process Data Not Active	Another function was using register access when the function attempted to update the process data
16#0005	Position Move In Progress	A position move is already in progress. Use **SM_Stop** to cancel the current move
16#0006	Position Move Aborted	A position move in progress using **SM_MovePos** or **SM_MovePosAndVelo** was cancelled by **SM_Stop**