Positioning via pulse interface

service & SUPPORT

Micro Automation Positioning - SERVO

SIEMENS

FAQ

Note The application examples are not binding and do not claim to be complete regarding the circuits shown, equipping and any eventuality. The application examples do not represent customer-specific solutions. They are only intended to provide support for typical applications. You are responsible in ensuring that the described products are correctly used. These application examples do not relieve you of the responsibility in safely and professionally using, installing, operating and servicing equipment. When using these application examples, you recognize that Siemens cannot be made liable for any damage/claims beyond the liability clause described. We reserve the right to make changes to these application examples at any time without prior notice. If there are any deviations between the recommendations provided in these application examples and other Siemens publications - e.g. Catalogs - then the contents of the other documents have priority.

Warranty, liability and support

We do not accept any liability for the information contained in this document.

Any claims against us - based on whatever legal reason - resulting from the use of the examples, information, programs, engineering and performance data etc., described in this application example shall be excluded. Such an exclusion shall not apply in the case of mandatory liability, e.g. under the German Product Liability Act (“Produkthaftungsgesetz”), in case of intent, gross negligence, or injury of life, body or health, guarantee for the quality of a product, fraudulent concealment of a deficiency or breach of a condition which goes to the root of the contract (“wesentliche Vertragspflichten”). However, claims arising from a breach of a condition which goes to the root of the contract shall be limited to the foreseeable damage which is intrinsic to the contract, unless caused by intent or gross negligence or based on mandatory liability for injury of life, body or health The above provisions does not imply a change in the burden of proof to your detriment.

For questions about this document please use the following e-mail address:

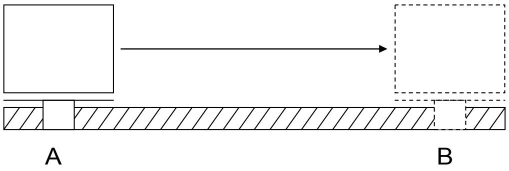
mailto:csweb@ad.siemens.de

Table of Contents

1 Objective

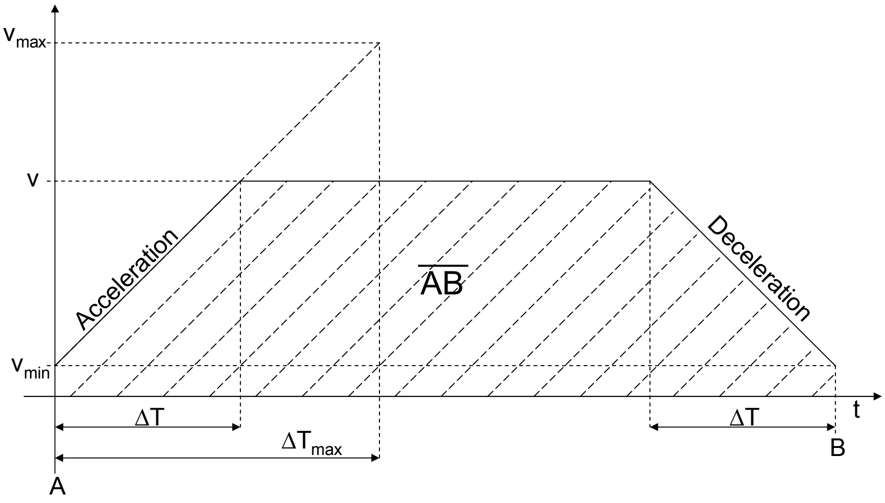
The aim is to move an object from point A to point B, e.g. by using a motor spindle.

Figure 1-1



To this end, the object is accelerated to a specified speed v within the time interval ΔT. The spindle moves with the speed v, and in order to reach the end position B, it is decelerated to zero within a second identical time interval. We obtain the distance from A to B from the area of the v-t diagram:

Figure 1-2



By specifying v_{min}, v_{max}, ΔT_{max} and the distance from A to B, we obtain the condition for the maximum achievable velocity v and the acceleration and deceleration time ΔT:

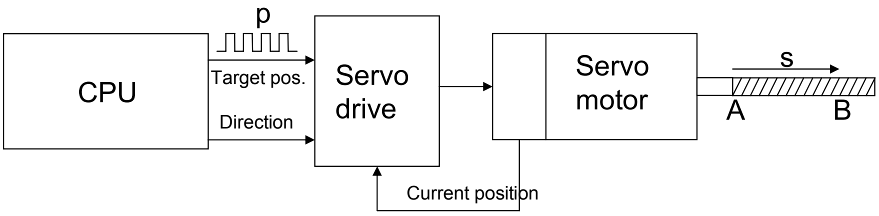
$$v \leq \sqrt{\frac{\overline{AB}}{\Delta T_{\max}} \cdot (v_{\max} - v_{\min}) + v_{\min}{}^{2}}$$

$$\Delta T = \frac{v - v_{\min}}{v_{\max} - v_{\min}} \cdot \Delta T_{\max}$$

2 Implementation

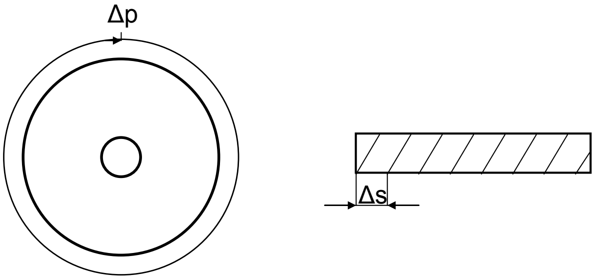
The positioning process is implemented using the pulse train output of the S7-200 CPUs. The CPU sends a pulse train (PTO) and the directional information (rotation direction of motor) to a servo drive. The servo drive converts the signal into the corresponding stator three-phase voltage depending on the frequency (proportional to the speed) and feeds the servo motor. The servo motor returns the actual value for the speed control to the servo drive.

Figure 2-1



The positioning conversion from the CPU pulse output into the distance traveled by the spindle is accomplished via the reference to one motor revolution.

Figure 2-2



Δp = Number of pulses required for one motor revolution

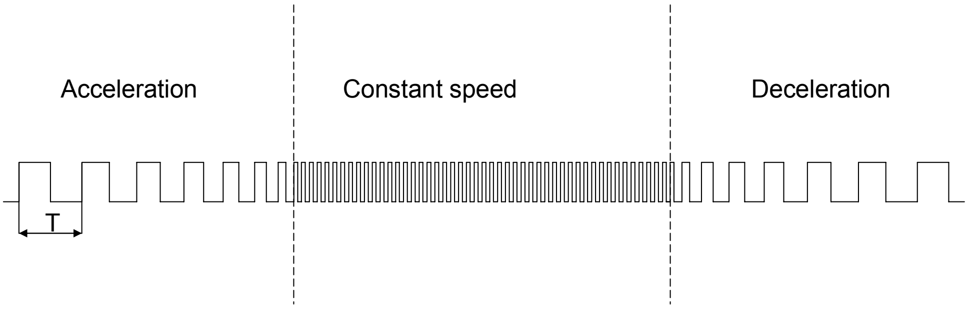
Δs = distance traveled by the spindle during one motor revolution

The required number of pulses for the distance from A to B can be calculated as follows:

$$p(\overline{AB}) = \frac{\Delta p}{\Delta s} \cdot \overline{AB}$$

Internally, the CPU calculates with the number of pulses instead of distance values. We thus obtain the following pulse train equivalent to the v-t diagram (Figure 1-2):

Figure 2-3



The smaller the cycle time T, the higher the travel speed.

The maximum travel speed v_{max} and the maximum motor speed n_{max} are thus dependent on the maximum pulse output frequency f_{max} of the CPU:

$$v_{\max} = \frac{\Delta s}{\Delta p} \cdot f_{\max}$$

$$n_{\max} = \frac{f_{\max}}{\Delta p} \cdot 60$$

The following table shows the maximum pulse output frequency of the S7-200 CPUs with 24 V DC output A0.0 and A0.1:

Table 2-1

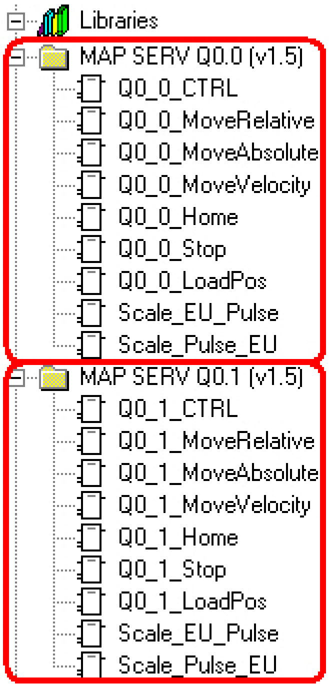
CPU	Pulse frequency (max.)
221	20 kHz
222	20 kHz
224	20 kHz
224XP	100 kHz
226	20 kHz

3 Positioning block library

The library “MAP SERV Q0.0”[1] provides the positioning blocks for the output A0.0 and “MAP SERV Q0.1” for A0.1.

Note Both pulse output libraries can be used in one project.

Figure 3-1



- Libraries
  - MAP SERV Q0.0 (v1.5)
    - Q0_0_CTRL
    - Q0_0_MoveRelative
    - Q0_0_MoveAbsolute
    - Q0_0_MoveVelocity
    - Q0_0_Home
    - Q0_0_Stop
    - Q0_0_LoadPos
    - Scale_EU_Pulse
    - Scale_Pulse_EU
  - MAP SERV Q0.1 (v1.5)
    - Q0_1_CTRL
    - Q0_1_MoveRelative
    - Q0_1_MoveAbsolute
    - Q0_1_MoveVelocity
    - Q0_1_Home
    - Q0_1_Stop
    - Q0_1_LoadPos
    - Scale_EU_Pulse
    - Scale_Pulse_EU

3.1 Overview

Table 3-1

Block	Function
Q0_x_CTRL	Parameter specification and control block
Q0_x_MoveRelative	Executing a relative positioning step
Q0_x_MoveAbsolute	Executing an absolute positioning step
Q0_x_MoveVelocity	Starting a move with preset velocity
Q0_x_Home	Search reference point position
Q0_x_Stop	Stopping the move
Q0_x_LoadPos	Load preset position marker
Scale_EU_Pulse	Converting distance values into pulse values
Scale_Pulse_EU	Converting pulse values into distance values

[1] **MAP SERV** = **M**icro **A**utomation **P**ositioning **SERV**o

3.2 General information

The block libraries enable a linear axis to be moved.

3.2.1 Setup

To best use the positioning blocks, three limit switches are required along the spindle:

- one reference point switch (home) for the zero point of the pulse scale C_Pos
- one limit switch in positive count direction (Fwd_Limit) and one in negative count direction (Rev_Limit)

Note

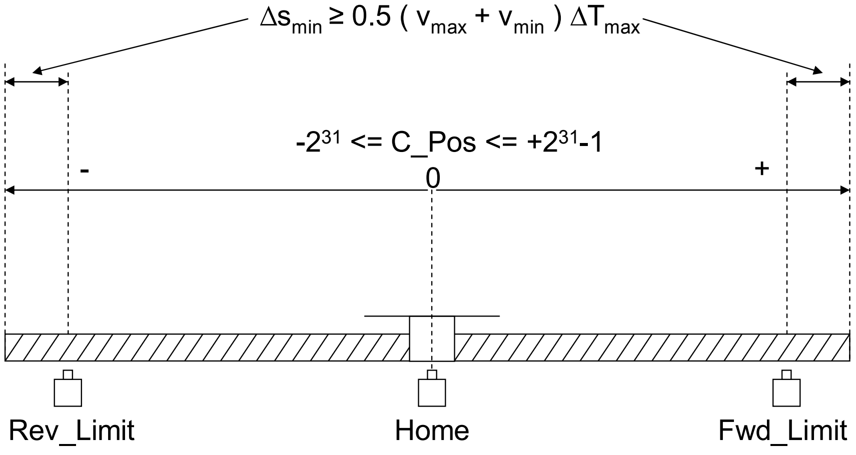
Definition of the direction:

forward: positive count direction

reverse: negative count direction

- The count value of the pulse scale C_Pos belongs to the data type DINT and thus comprises the integer value range of 32 bits (-2.147.483.648 to +2.147.483.647 when formatted with sign).
- If a limit switch is triggered by the object carrier, the object carrier will stop and it will be accelerated in the opposite direction. Therefore, the limit switches must be arranged so that there is still sufficient space Δs_{min} for the deceleration process towards the end of the travel path.

Figure 3-2


$\Delta s_{min} \geq 0.5 (v_{max} + v_{min}) \Delta T_{max}$
$-2^{31} <= C\_Pos <= +2^{31}-1$
-
0
+
Rev_Limit
Home
Fwd_Limit

3.2.2 Permanent assignment of I/Os

The following inputs and outputs are predefined when using one of the two positioning libraries:

Table 3-2

Name	MAP SERV Q0.0	MAP SERV Q0.1
Pulse output	A 0.0	A 0.1
Directional output	A 0.2	A 0.3
Reference point input	E 0.0	E 0.1
High-speed counters used	HC 0	HC 3
HSCx_Preset	SMD 42	SMD 142
PTOx_MAN_SPEED	SMD 172	SMD 182

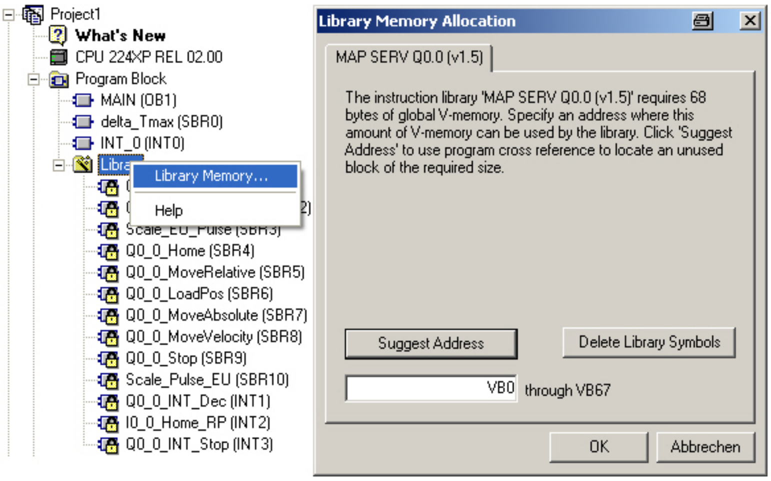
Note

- Status of the directional output: Default setting: 0 = reverse / 1= forward inversion possible using “Dir_Active_Low”=1 (see Table 3-3)
- The reference point limit switch functions to synchronize the high-speed counter with the current position of the spindle.
- The current position can be read out via the high-speed counter.
- HSCx_Preset indicates the next target position (delay start position, target position) where an interrupt is carried out.
- PTOx_MAN_SPEED indicates the target frequency.

3.2.3 Integrating the block library

In order to use the positioning blocks, 68 bytes of memory (per library) must be reserved for the global variables.

Figure 3-3



The instruction library 'MAP SERV Q0.0 (v1.5)' requires 68 bytes of global V-memory. Specify an address where this amount of V-memory can be used by the library. Click 'Suggest Address' to use program cross reference to locate an unused block of the required size.

VB0 through VB67

The following table shows the most important global variables with address offset.

Table 3-3

Symbol	Address	Comment
Disable_Auto_Stop	+V0.0	Default value 0 = The positioning process is stopped when the final position is reached even if the minimum pulse frequency has not been reached yet.
Dir_Active_Low	+V0.1	Default value 0 = The directional output shows 1 in forward direction
Final_Dir	+V0.2	Last direction during reference point search (default value 0)
Tune_Factor	+VD1	Tuning factor (default value 1.0)
Ramp_Time	+VD5	Ramp time = accel_dec_time
Max_Speed_DI	+VD9	Maximum pulse output frequency = Velocity_Max
SS_Speed_DI	+VD13	Minimum pulse output frequency = Velocity_SS
Homing_State	+VD18	Status of the reference point search (see table 3-8)
Homing_Slow_Spd	+VD19	Slow pulse frequency during reference point search (default value: Velocity_SS)
Homing_Fast_Spd	+VD23	Fast pulse frequency during reference point search (default value: Velocity_Max/2)
Fwd_Limit	+V27.1	Forward limit switch
Rev_Limit	+V27.2	Reverse limit switch
Homing_Active	+V27.3	Reference point search active
C_Dir	+V27.4	Current direction
Homing_Limit_Chk	+V27.5	Limit switch flag
Dec_Stop_Flag	+V27.6	Deceleration started
PTO0_LDPOS_Error	+ VB28	corresponds to Q0_x_LoadPos-Error (16#00 = no error, 16#FF = error)
Target_Location	+VD29	Target position
Deceleration_factor	+VD33	= (Velocity_SS – Velocity_Max) / accel_dec_time (format: REAL)
SS_Speed_real	+VD37	= Velocity_SS (format: REAL)
Est_Stopping_Dist	+VD41	Calculated deceleration path (format: DINT)

3.3 Positioning blocks

This programming is based on the integrated linear pulse train (PTO). This function is used to control the velocity. Additionally, the pulses are counted using the high-speed counter HSC. The deceleration starting point is calculated and triggered via the HSC interrupt.

3.3.1 Q0_x_CTRL

The control block Q0_x_CTRL transfers the global positioning parameters and must be started cyclically.

Figure 3-4

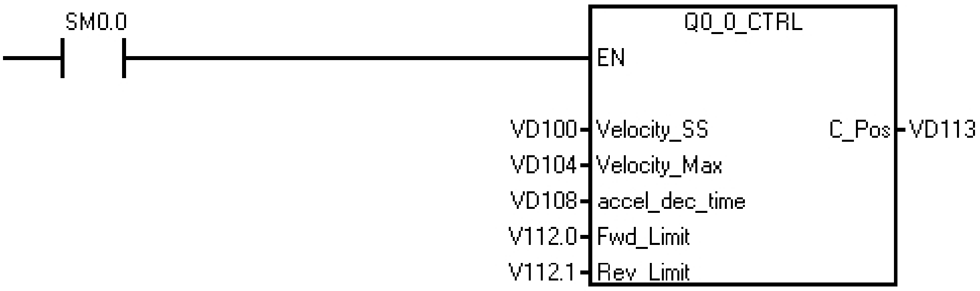


Table 3-4

Parameter	Type	Format	Unit	Meaning
Velocity_SS	IN	DINT	Pulse/ sec.	Start/stop frequency
Velocity_Max	IN	DINT	Pulse/ sec.	Maximum frequency
accel_dec_time	IN	REAL	Sec.	Maximum acceleration and deceleration time
Fwd_Limit	IN	BOOL		Limit switch for the forward travel
Rev_Limit	IN	BOOL		Limit switch for the reverse travel
C_Pos	OUT	DINT	Pulse	Current absolute position

Velocity_SS determines the minimum frequency at which the acceleration process starts or the deceleration ends. The value must always be > 0.

Velocity_Max provides the maximum frequency in pulses/second (important for motor speed limitation and dependent on the maximum pulse output frequency of the CPU -> Table 2-1).
accel_dec_time is limited to values between 0.02 and 32.0 seconds (should not be set below 0.5 seconds).

WARNING **Values outside the definition range of “accel_dec_time” are also accepted, but they can cause erroneous positioning.**

3.3.2 Scale_EU_Pulse / Scale_ Pulse_EU

Scale_EU_Pulse

The block Scale_EU_Pulse is used to convert a position quantity into the corresponding number of pulses. So this block allows converting a distance into the corresponding number of pulses or a velocity into a pulse frequency.

Figure 3-5

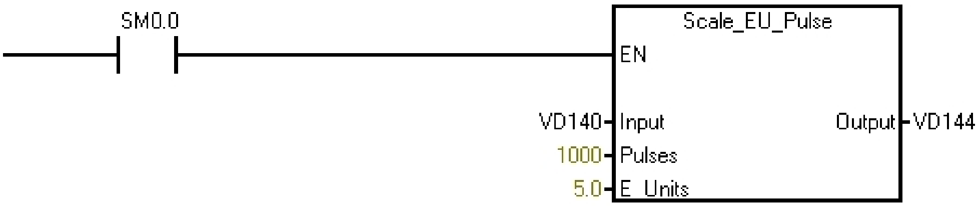


Table 3-5

Parameter	Type	Format	Unit	Meaning
Input	IN	REAL	mm or mm/s	Distance or velocity to be converted
Pulses	IN	DINT	Pulse / revol.	Number of pulses required for one revolution
E_Units	IN	REAL	mm / revol.	Distance traveled during one revolution
Output	OUT	DINT	Pulse or pulse/s	Equivalent pulse number or pulse frequency

The following equation is used with this function:

$$Output = \frac{Pulses}{E\_Units} \cdot Input$$

Scale_ Pulse_EU

The block Scale_Pulse_EU is used to convert a number of pulses into a distance value.

Figure 3-6

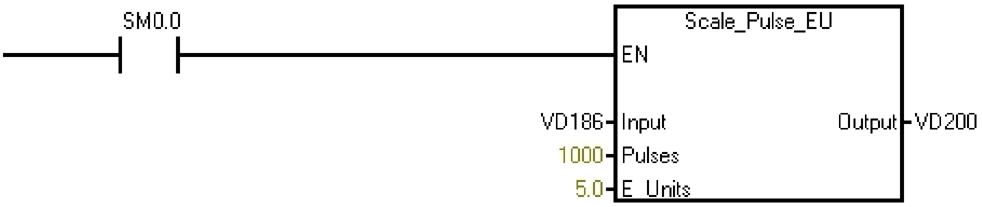


Table 3-6

Parameter	Type	Format	Unit	Meaning
Input	IN	REAL	Pulse or pulse/s	Pulse number or pulse frequency to be converted
Pulses	IN	DINT	Pulse / revol.	Number of pulses required for one revolution
E_Units	IN	REAL	mm / revol.	Distance traveled during one revolution
Output	OUT	DINT	mm or mm/s	Corresponding distance or velocity

The following equation is used with this function:

$$Output = \frac{E\_Units}{Pulses} \cdot Input$$

Note Millimeters are a good choice for the unit of the distance (any other selection is possible, too, as long as it is identical).

3.3.3 Q0_x_Home

Figure 3-7

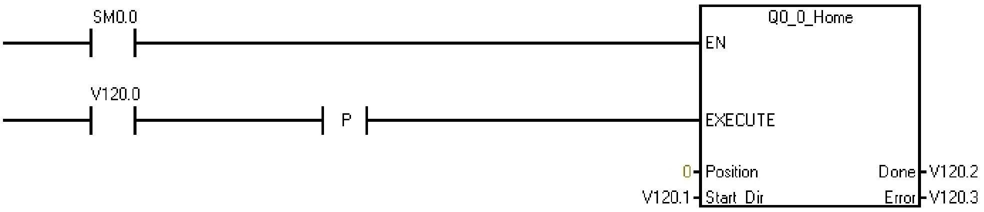


Table 3-7

Parameter	Type	Format	Unit	Meaning
EXECUTE	IN	BOOL		Execution bit for the reference point search
Position	IN	DINT	Pulse	Absolute position value in the reference point
Start_Dir	IN	BOOL	Sec.	Direction in which the reference point search starts (0=reverse, 1=forward)
Done	OUT	BOOL		Ready message (1 = done)
Error	OUT	BOOL		Error message (1 = error)

This function starts the reference point search in the preset direction “Start_Dir” and increases the frequency up to “Homing_Fast_Spd”. When a limit switch (“Fwd_Limit” or “Rev_Limit”) is reached, the spindle is decelerated and continues the search in the opposite direction. In case of a positive edge of the reference point proximity switch (input E0.0; with Q0_1_Home: E0.1), the spindle is decelerated to the frequency “Homing_Slow_Spd”. If the current direction is identical with “Final_Dir” (see Table 3-3), the spindle stops in case of a falling edge of the reference point proximity switch. The position counter HC0 is set to the absolute value at the input “Position”.

If the current direction is not identical with the “Final_Dir”, a change of direction is necessary. Thus it is ensured that the spindle (depending on the “Final_Dir”) always stops at the same end (falling edge) of the reference point proximity switch.

The reference point positioning processes and an explanation is given in chapter 4.2.7 and 5.

Note The reference point search is always finished by a falling edge.

The spindle thus stops in a negative final search direction “Final_Dir”=0 at the left end of the reference point proximity switch and if “Final_Dir”=1, at the right end of the reference point proximity switch.

WARNING **If the bit “Final_Dir” changes, the reference point will be shifted by the width of the proximity switch signal.**

The status of the reference point search can be monitored using the global variable Homing_State (see Table 3-3).

Table 3-8

Valu e	Meaning
0	Reference point found
2	Search started
4	Search is continued in the opposite direction with Homing_Fast_Spd (after triggering the limit switch or reference point limit switch)
6	Reference point found, deceleration process initiated
7	Search is continued in the direction “Final_Dir“ with Homing_Slow_Spd (after reference point was found with Homing_Fast_Spd)
10	Error (no reference point found between actuation of both limit switches)

3.3.4 Q0_x_MoveRelative

This function makes the spindle move by a specified pulse number in a specified direction with a specified frequency.

Figure 3-8

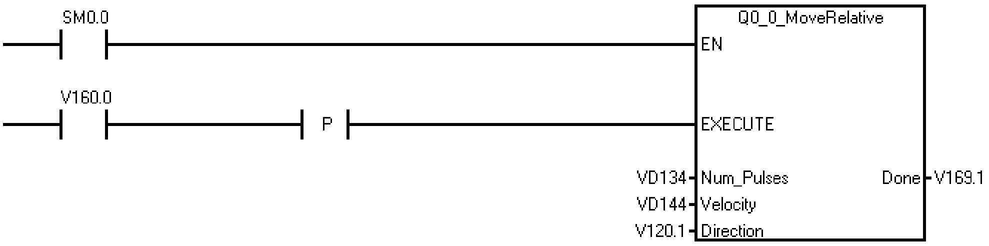
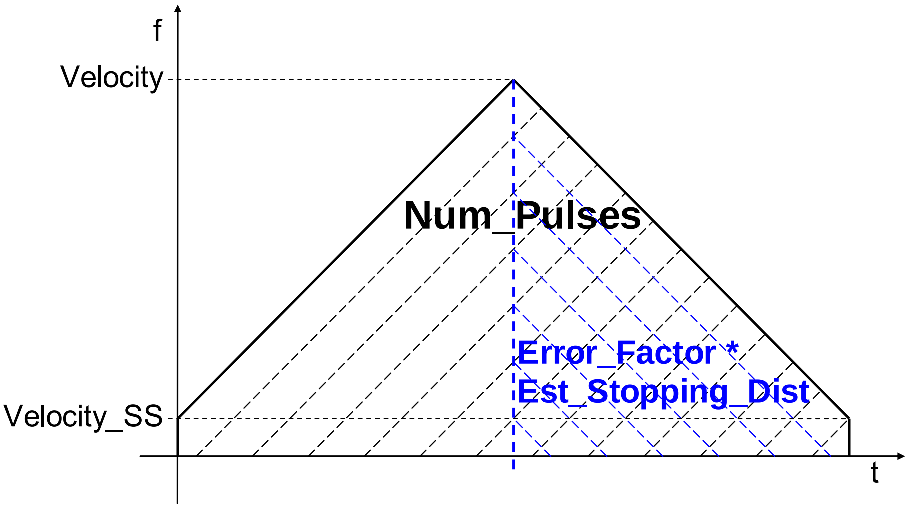


Table 3-9

Parameter	Type	Format	Unit	Meaning
EXECUTE	IN	BOOL		Execution bit to the relative positioning process
Num_Pulses	IN	DINT	Pulse	Pulse distance (must be greater than 1)
Velocity	IN	DINT	Pulse/ sec.	Preset frequency (Velocity_SS <= Velocity <= Velocity_Max)
Direction	IN	BOOL		Preset direction (0=reverse, 1=forward)
Done	OUT	BOOL		Ready message (true = done)

The parameter “Num_Pulses” is obtained as area under the frequency-time diagram.

Figure 3-9



Note

The positioning comes before reaching the specified frequency!

The definition of excess parameters (more parameters must be specified than required) can lead to a condition violation of the maximum attainable travel speed (frequency) (see chapter 1). Figure 3-9 shows the border case: The specified frequency “Velocity” is only just reached with the specified area “Num_Pulses”. A higher frequency value would not be fulfilled and would be reduced (controllable via the frequency index PTOx_MAN_SPEED).

The following conditions must be satisfied to reach the preset frequency:

- Num_Pulses >= Velocity/650
  Otherwise, the calculation time for the deceleration start position will be insufficient. In that case:
  Velocity = Velocity_SS
- Error_Factor * Est_Stopping_Dist <= Num_Pulses/2
  The corrected calculated deceleration distance must not be greater than half the total distance. Otherwise: Velocity = Velocity/2

3.3.5 Q0_x_MoveAbsolute

This function makes the spindle move to a preset position (pulse flag) with a preset frequency.

Figure 3-10

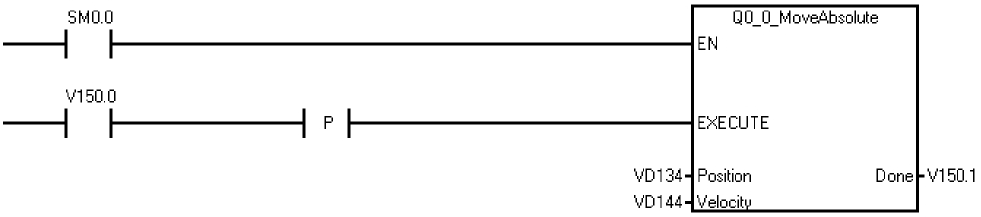
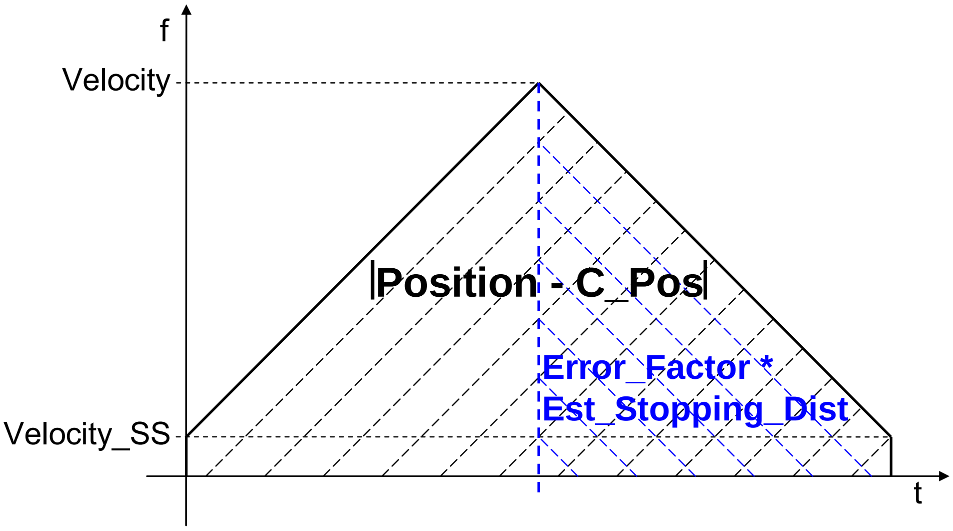


Table 3-10

Parameter	Type	Format	Unit	Meaning
EXECUTE	IN	BOOL		Execution bit to the absolute positioning process
Position	IN	DINT	Pulse	Absolute positioning
Velocity	IN	DINT	Pulse/ sec.	Preset frequency
Done	OUT	BOOL		Ready message (true = done)

With this function, the area under the frequency-time diagram is obtained from the difference of the absolute position specification and the current position.

Figure 3-11



Hinweis

The positioning comes before reaching the specified frequency!

The definition of excess parameters (more parameters must be specified than required) can lead to a condition violation of the maximum attainable travel speed (frequency) (see chapter 1). Figure 3-11 shows the border case: The specified frequency “Velocity” is only just reached with the specified area |“Position“ – „C_Pos“|. A higher frequency value would not be fulfilled and would be reduced (controllable via the frequency index PTOx_MAN_SPEED).

The following conditions must be satisfied to reach the preset frequency:

- |Position – C_Pos| >= Velocity/650
  Otherwise, the calculation time for the deceleration start position will be insufficient. In that case:
  Velocity = Velocity_SS
- Error_Factor * Est_Stopping_Dist <= |Position – C_Pos|/2
  The corrected calculated deceleration distance must not be greater than half the total distance. Otherwise:
  Velocity = Velocity/2

3.3.6 Q0_x_MoveVelocity

This function makes the spindle move in a preset direction with a preset frequency. A change of frequency (but no change of direction) is possible during the process via the EXECUTE bit.

Figure 3-12

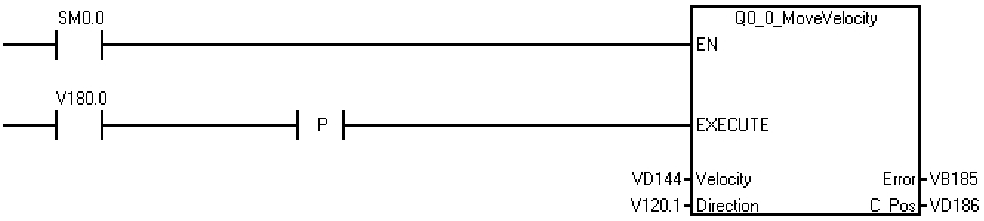
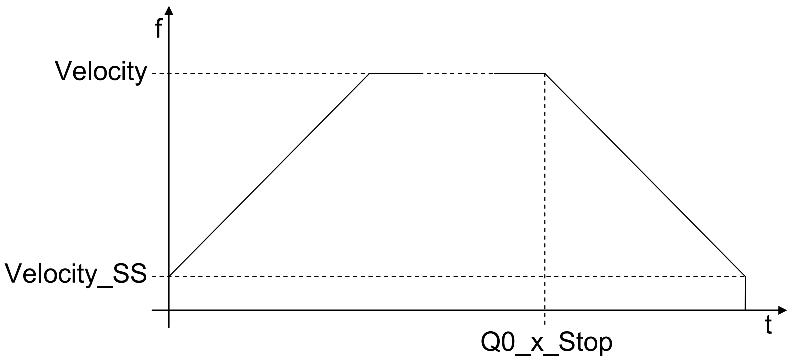


Table 3-11

Parameter	Type	Format	Unit	Meaning
EXECUTE	IN	BOOL		Execution bit
Velocity	IN	DINT	Pulse/ sec.	Preset frequency
Direction	IN	BOOL		Preset direction (0=reverse, 1=forward)
Error	OUT	BYTE		Error message (0 = no error) 1 = immediate stop without delay; 3 = execution error)
C_Pos	OUT	DINT	Pulse	Current absolute position

Note The Q0_x_MoveVelocity function can only be stopped using the Q0_x_Stop block.

Figure 3-13



3.3.7 Q0_x_Stop

This function decelerates the spindle until it stands still.

Figure 3-14

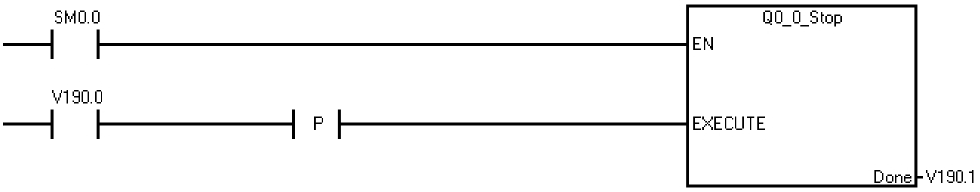


Table 3-12

Parameter	Type	Format	Unit	Meaning
EXECUTE	IN	BOOL		Execution bit
Done	OUT	BOOL		Ready message (true = done)

3.3.8 Q0_x_LoadPos

This function sets the absolute position counter to a preset value.

Figure 3-15

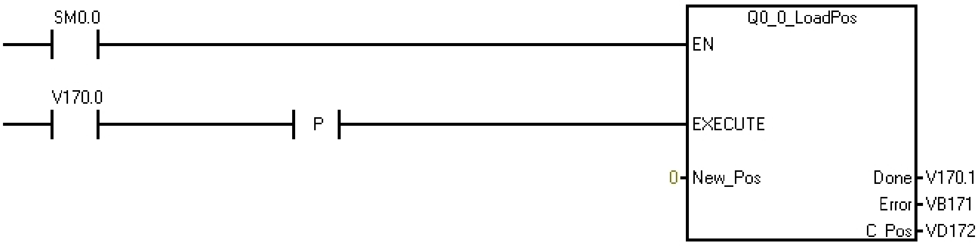


Table 3-13

Parameter	Type	Format	Unit	Meaning
EXECUTE	IN	BOOL		Execution bit for setting the absolute positioning counter
New_Pos	IN	DINT	Pulse	Preset count value
Done	OUT	BOOL	Pulse/ sec.	Ready message (true = done)
Error	OUT	BYTE		Error message (0 = no error)
C_Pos	OUT	DINT	Pulse	Current absolute position

Note This function invalidates the reference point. A new reference point must be searched in order to clearly determine the absolute position.

3.4 Calibration

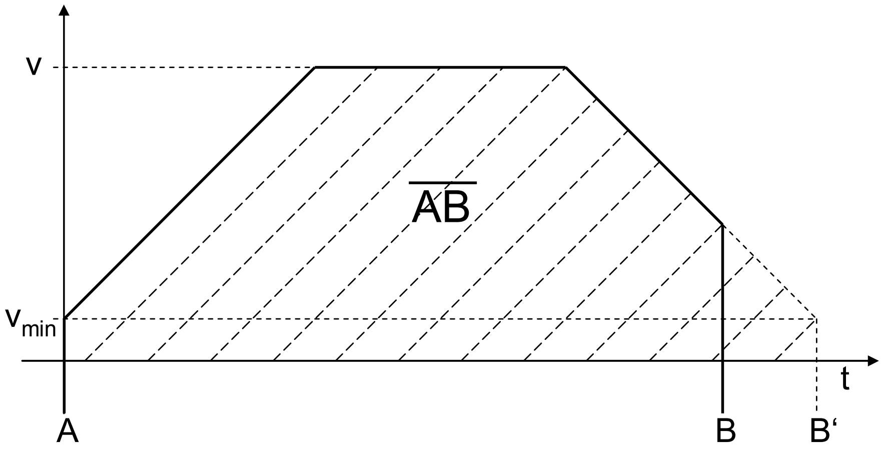
The algorithm that calculates the deceleration path (pulse number) and thus the starting point of the deceleration ramp does not correspond exactly to the real profile of the pulse train due to the step formation of the deceleration ramp. This fact may require a calibration under the circumstances described below.

3.4.1 Tuning factor

The deviation between the calculated and the real deceleration distance can be corrected by the parameter “Tune_Factor” (default value = 1).

The best value for this tuning factor depends on the minimum, maximum and target pulse frequency and on the maximum deceleration time (identical to the maximum acceleration time).

Figure 3-16

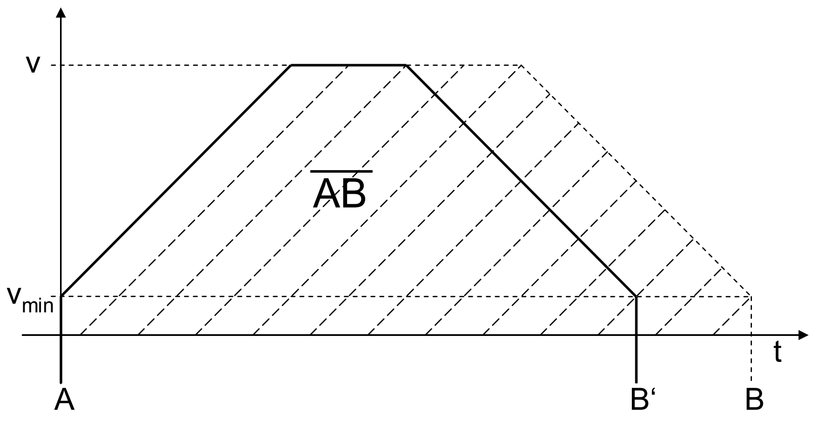


The bit „Disable_Auto_Stop“=“0“ (see Table 3-3) prevents that the target is passed over. The calculated target position B’ is after the real target position B. If B is reached before reaching the minimum pulse frequency, the deceleration process is aborted (default setting 0). Thus a too high (late) calculation of the deceleration start point is compensated for.

3.4.2 Application area

If, however, the calculation of the deceleration start point is too low (early) (the positioning process stops before reaching the target or the minimum pulse frequency), a recalibration must be carried out. This case, where the real target position B comes after the calculated target position B', is caused by a too high tuning factor.

Figure 3-17



Note If the calculation of the deceleration start position is not corrected (Tune_Factor=1), the target position will be passed over if "Disable_Auto_Stop"=1. The default setting "Disable_Auto_Stop"=0 will cause the deceleration ramp to be cut off upon reaching the target position.

3.4.3 Determining the tuning factor

Note A recalibration does not require the servo drive to be connected to the CPU.

1. Set the bit "Disable_Auto_Stop".
2. Set the "Tune_Factor" to the value 1.
3. Set the absolute position counter to 0 using the function Q0_x_LoadPos.
4. Carry out a relative positioning process with the desired parameters (minimum, maximum, target pulse frequency and maximum deceleration time) and with sufficient pulse steps to meet this specifications (Q0_x_MoveRelative).

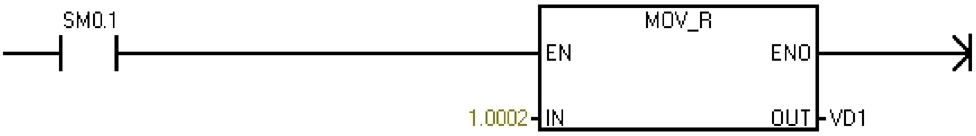
5. Check the achieved target position. The wanted tuning factor "Tune_Factor" is obtained from the target position of the high-speed counter "HC0", the target specification "Num_Pulses" and the estimated deceleration distance "Est_Stopping_Dist" (global parameter + VD41):

$$Tune\_Factor = \frac{HC0 - Num\_Pulses + Est\_Stopping\_Dist}{Est\_Stopping\_Dist}$$

$$Est\_Stopping\_Dist = \frac{Velocity^2 - Velocity\_SS^2}{Velocity\_Max - Velocity\_SS} \cdot \frac{accel\_dec\_time}{2}$$

6. Insert a network after the network that calls Q0_x_CTRL where the new determined value is passed to the symbol variable "Tune_Factor" (absolute address). (It must be the absolute address (+VD1) since you can not use the symbolic address.)

Figure 3-18



7. Reset the bit "Disable_Auto_Stop".

4 Software examples for positioning via pulse interface

Preliminary remark

If you need support for positioning via the pulse interface, you can download the software demo project “MAP SERV Q0.0.mwp” including test code and test parameters. The software example supports you during the first steps and tests with the positioning library “MAP SERV Q0.0.mwl”. It allows you to quickly test the function blocks and possible applications.

Download

The software examples are available on the HTML page from which you downloaded this document.

The download is a ZIP file which can be extracted using any unzip program. The zip.-file contains the following files:

Table 4-1

File name	Content
MAP SERV Q0.0.mwl	Positioning library for the output A0.0
MAP SERV Q0.1.mwl	Positioning library for the output A0.1
MAP SERV Q0.0.mwp	Demo project for the library “MAP SERV Q0.0.mwl”

Hardware

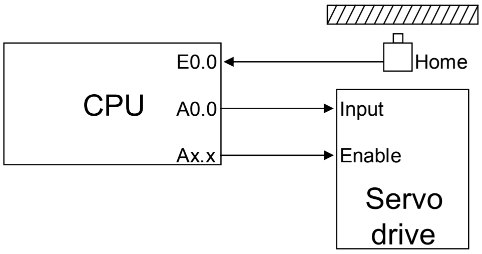
The example has been tested with a CPU 224XP (6ES7214-2AD23-0XB0).

Transmission rate of the interfaces:

Table 4-2

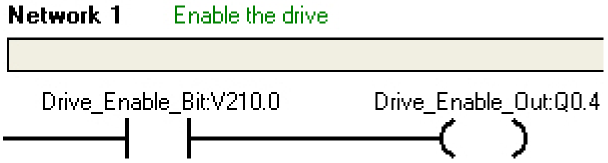
Interface	Transmission rate
Port 0	187.5 kbps
Port 1	19.2 kbps

Figure 4-1



You only have to connect the servo drive to the pulse train output A0.0 and the reference point input E0.0 must be connected. It may also be necessary to enable the drive (see network 1).

Figure 4-2



Operation

The sample project is operated via the status table.

Action	1. Open the sample project "MAP_SERV_Q0_0" in STEP7 Micro/WIN and load it into the CPU by clicking ⤓. 2. Open the "Operation" status table and activate the table status by clicking ▦. 3. Set the "Drive_Enable_Bit"="1" (V210.0) and apply the settings with ✎ to enable the drive (optional).
Response	The "Drive_Enable_Out" output is set ("A0.4" = "1") and enables the servo drive.
Note	• For a simple check you can also use a function without drive and motor. • The inputs of the limit switches "Limit_Fwd" and "Limit_Rev" are also simulated via the status table. • All EXECUTE bits are defined as push buttons and have to be reset before they can be actuated again. • All Done bits must be reset before they can generate a new Done message.

4.1 Task

A positioning process shall be carried out from point A (absolute position: 120 mm) to point B (in 1500 mm distance from A in positive direction).

Then start a move in negative direction in jog mode with a preset speed, stop after any time and determine the traveled distance.

The acceleration and deceleration time (to reach the target speed) shall be $\Delta T = 1$ s, the travel speed v = 100 mm/s and the minimum pulse frequency shall be 10 mm/s.

The nominal motor speed of $n_N = 3000$ 1/min must not be exceeded.

A CPU 224XP with a maximum pulse output frequency $f_{max} = 100$ kHz is used.

1000 pulses ($\Delta p = 1000$ pulses/revolutions) are required for one motor spindle revolution and the pitch of the spindle is $\Delta s = 5$ mm/revolution.

4.2 Description of the demo project “MAP SERV Q0.0.mwp”

The next chapter describes the individual networks and their functions.

4.2.1 Overview of the demo program “MAIN”

Table 4-3

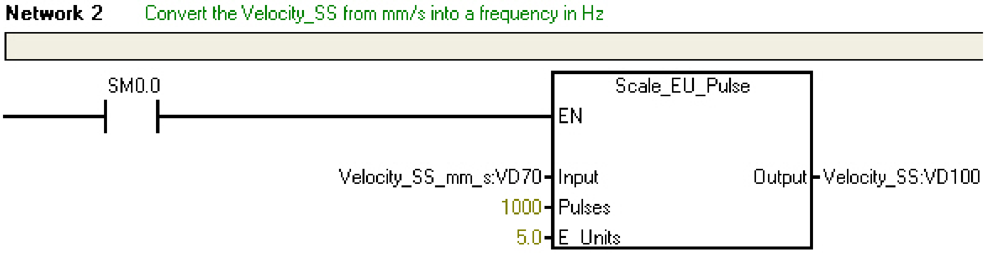
Network	Description
1	Enabling the servo drive
2	Converting the minimum speed into the minimum frequency
3	Calculating the maximum frequency in dependence on the nominal speed of the servo motor and output of the corresponding speed
4	Converting the target speed into the target frequency
5	Calculating the maximum acceleration and deceleration time
6	Calling the control block “Q0_0_CTRL”
7	Changing the default values
8	Calling the reference point search “Q0_0_Home”
9	Converting the distance into a pulse number
10	Calling the absolute positioning “Q0_0_MoveAbsolute”
11	Calling the relative positioning “Q0_0_MoveRelative”
12	Resetting the absolute position counter (“Q0_0_LoadPos”)
13	Calling the traversing operation with preset velocity
14	Stopping the traversing operation (“Q0_0_Stop”)

Network	Description
15	Converting the pulse number into a distance

4.2.2 Calculating the minimum frequency “Velocity_SS”

The minimum frequency is calculated using the block “Scale_EU_Pulse”.

Figure 4-3

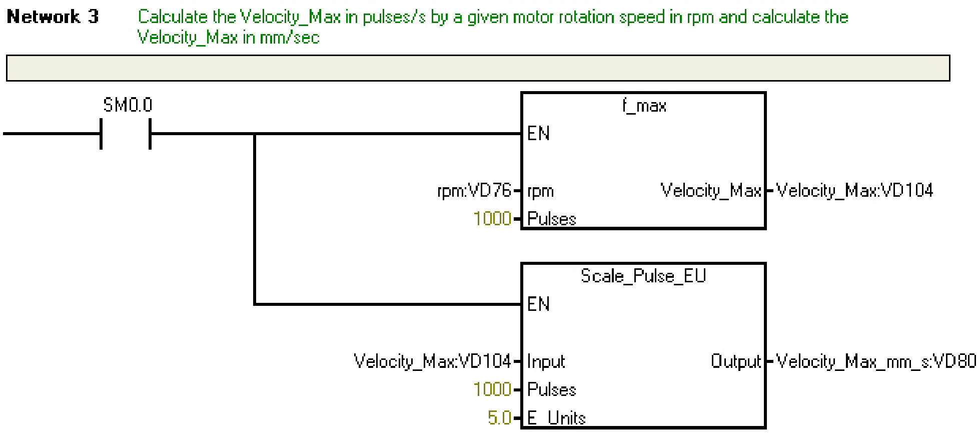


Network 2 Convert the Velocity_SS from mm/s into a frequency in Hz

Action	4. Enter the value “10.0” for the variable “Velocity_SS_mm_s” (VD70) (task: minimum speed of 10 mm/s) and apply the setting by clicking on .
Response	The converted minimum frequency is output as the variable “Velocity_SS” (VD100) with the value “2000” pulses per second.

4.2.3 Calculating the maximum frequency “Velocity_Max”

Figure 4-4



Network 3 Calculate the Velocity_Max in pulses/s by a given motor rotation speed in rpm and calculate the Velocity_Max in mm/sec

SM0.0
f_max
EN
rpm:VD76 - rpm
1000 - Pulses
Velocity_Max - Velocity_Max:VD104
Scale_Pulse_EU
EN
Velocity_Max:VD104 - Input
1000 - Pulses
5.0 - E_Units
Output - Velocity_Max_mm_s:VD80

fmax

The maximum frequency is calculated using the sub-program “f_max”. The nominal speed of the servo motor “rpm” (revolutions per minute) is converted into the corresponding frequency and multiplied with the number of pulses required for one revolution.

The following equation is behind this function:

$$Velocity\_Max = \frac{rpm}{60} \cdot Pulses$$

The parameter “Pulses” is already preset with the value “1000” from the task.

Scale_Pulses_EU

The function Scale_Pulses_EU is used to convert the maximum speed into the maximum frequency.

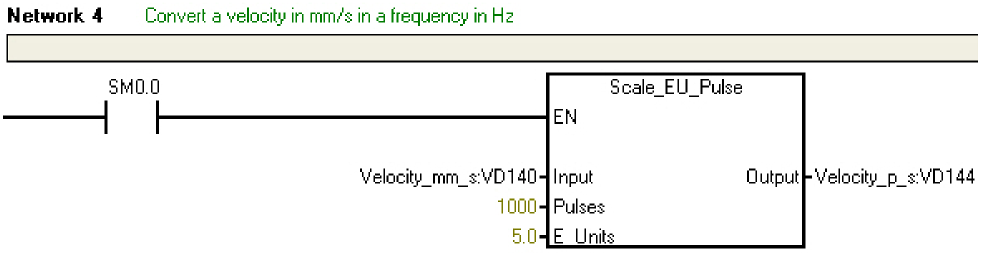
Action	5. Enter the value “3000” for the variable “rpm” (VD76) (task: The nominal motor speed is 3000 1/min) and apply the setting via .

Response	1. The calculated maximum frequency is output as the variable “Velocity_Max” (VD104) with the value “50000” pulses per second. 2. The converted maximum velocity is output as the variable “Velocity_Max_mm_s” (VD80) with the value “250.0” millimeters per second.

4.2.4 Calculating the target frequency “Velocity”

The target frequency is calculated using the block “Scale_EU_Pulse”.

Figure 4-5



Network 4 Convert a velocity in mm/s in a frequency in Hz

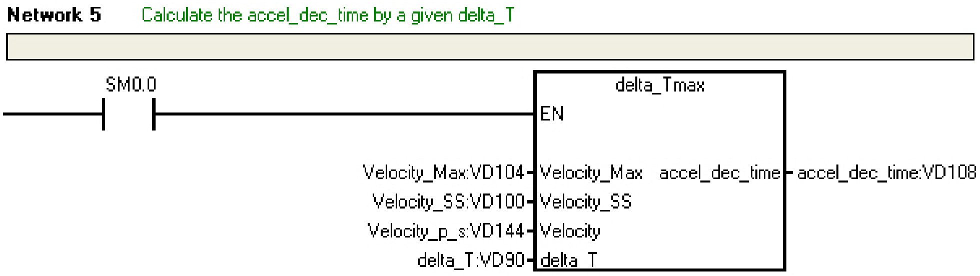
Action	6. Enter the value “100.0” for the variable “Velocity_mm_s” (VD140) (task: The target speed is 100.0 mm/s) and apply the setting via .

Response	The calculated target frequency is output as the variable “Velocity_p_s” (VD144) with the value “20000” pulses per second.

4.2.5 Calculating the maximum acceleration and deceleration time “accel_dec_time”

You have to adjust the maximum acceleration and deceleration time “accel_dec_time” for a specified minimum velocity (network 2), maximum velocity (network 3) and target velocity (network 4) and the acceleration and deceleration time until reaching the target velocity of one second (task). This calculation is carried out by the sub-program “delta_Tmax”.

Figure 4-6



The equation for the calculation is derived from chapter 1 and is as follows:

$$accel\_dec\_time = \frac{Velocity\_Max - Velocity\_SS}{Velocity - Velocity\_SS} \cdot delta\_T$$

The maximum acceleration and deceleration time is limited additionally:

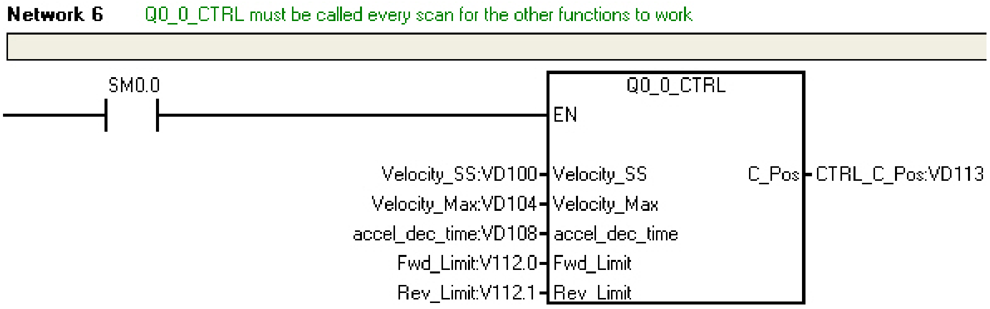
$$0{,}5s \le accel\_dec\_time \le 32{,}0s$$

Action	7. Enter the value “1.0” for the variable “delta_T” (VD90) (task: ΔT = 1s) and apply the setting via .
Response	The calculated maximum acceleration and deceleration time is output as the variable “accel_dec_time” (VD108) with the value “2.667” seconds.

4.2.6 Parameterizing the control block “Q0_0_CTRL”

The calculations in the networks 2 to 5 have returned all parameters for calling the control block “Q0_0_CTRL”.

Figure 4-7

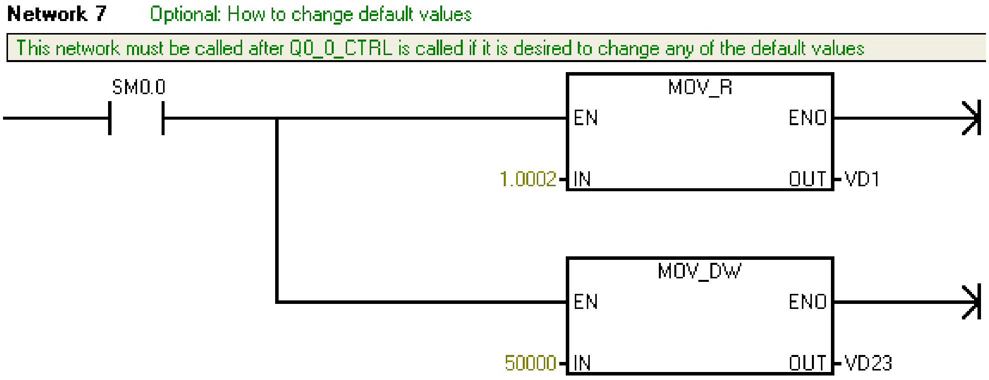


The actuation of the limit switches that limit the spindle is simulated via the variable bits V112.0 (“Fwd_Limit”) and V112.1 (“Rev_Limit”).

4.2.7 Changing the default values (optional)

If you want to change the preset parameters of the positioning library, this has to be done after Q0_x_CTRL has been called.

Figure 4-8



Network 7 Optional: How to change default values

This network must be called after Q0_0_CTRL is called if it is desired to change any of the default values

In the example, the parameter “Homing_Fast_Spd” (+VD23) in network 7 is set to the value “50000” pulses per second (equals the max. pulse frequency) and the “Tune_Factor” (+VD1) is increased to 1.0002 (the deceleration ramp is started earlier).

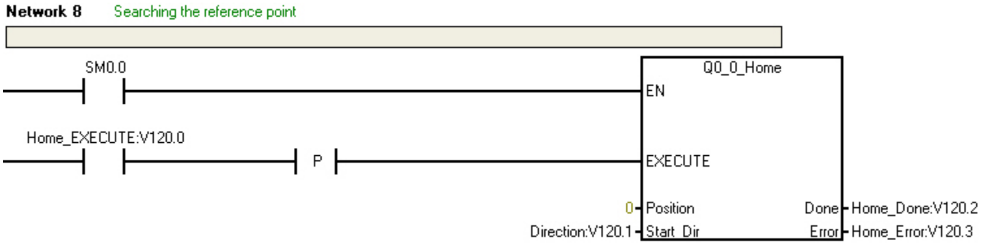
WARNING	**The absolute variable addresses must be used here (see table 3-3).**

Response	The variable table “Operation” allows you to verify that the values have been applied by calling the absolute addresses (“VD1” = “1.0002” and “VD23” = “50000”).

4.2.8 Reference point search

The reference point search is activated using the function “Q0_0_Home”.

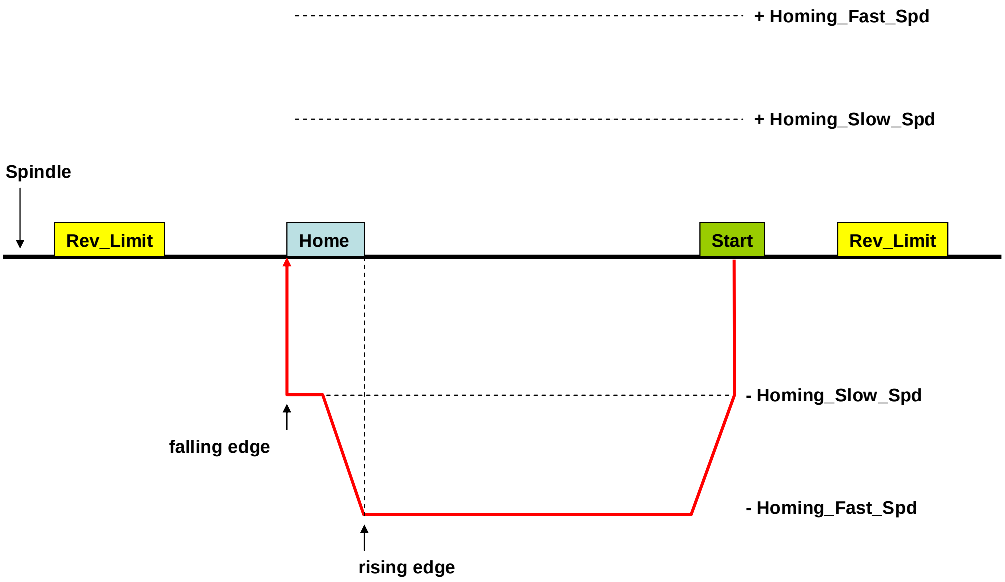
Figure 4-9



In order to explain the reference point search, the traversing diagram (Figure 4-10

) is used to simulate the following case: The start position “Start” is located to the right of the reference point “Home” The start direction “Start_Dir” and the final direction “Final_Dir” (+V0.2) of the reference point search are both set in negative count direction (“0”).

Figure 4-10



Action	8. Activate the reference point search by setting the bit “Home_EXECUTE” (V120.0) and apply the setting by clicking on .
Response	The servo motor starts with the frequency “Homing_Slow_Spd” (+VD19) and accelerates to the frequency “Homing_Fast_Spd” (+VD23). The negative sign in Figure 4-10 indicates the negative travel direction. The spindle will move with that frequency until it reaches the reference point.
Action	9. Trigger a positive edge via the circuit of the reference point input bit E0.0 and maintain this status (“1”).

Response The spindle is decelerated to the frequency “Homing_Slow_Spd” and moves with this frequency until it reaches the negative edge of the reference point input bit E0.0.

Action 10. Trigger a negative edge via the circuit of the reference point input bit E0.0 (status “0”).

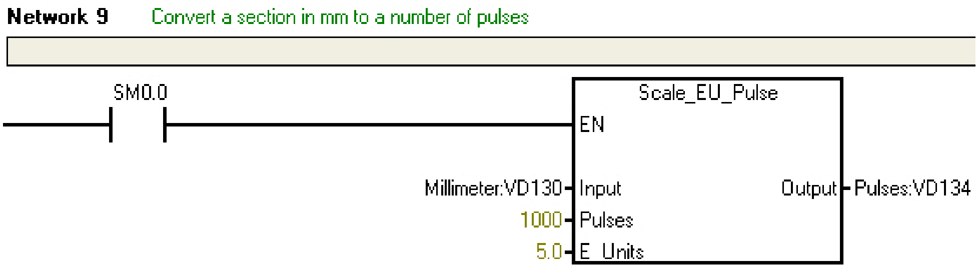
Response The spindle will stop immediately at the left end of the reference point Home“. The high-speed counter “HC0” is set to the “Position” parameter value (“0”) and the Done message bit “Home_Done” (V120.2) is set.

Note This traversing diagram (Figure 4-10) shows the case where deceleration to the frequency “Homing_Slow_Spd” is completed before reaching the negative edge of the reference point input bit E0.0. Additional traversing diagrams can be found in Chapter 5.

4.2.9 Converting the position data of point A (120 mm) into the corresponding pulse position

Network 9 is used to convert a distance into the corresponding pulse number by means of the function “Scale_EU_Pulse”.

Figure 4-11



Network 9 Convert a section in mm to a number of pulses

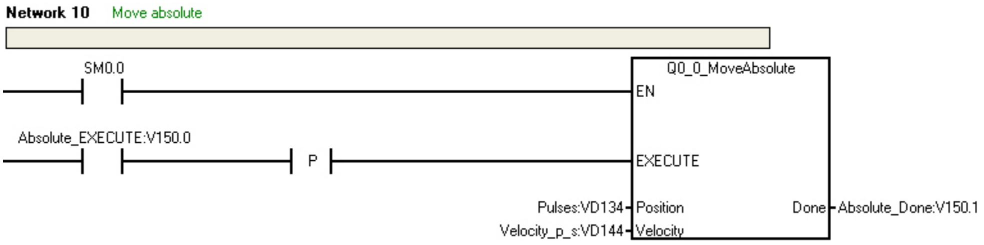
Action 11. Enter the value “120.0” mm into the status table for the variable VD130 “Millimeter” and apply the settings by clicking .

Response The variable VD134 “Pulses” will then show the value of “24000” pulses.

4.2.10 Moving to point A

The move to point A is accomplished via the function “Q0_0_MoveAbsolute”.

Figure 4-12



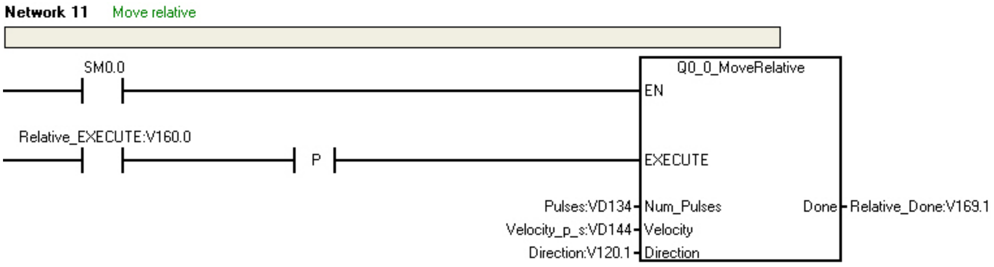
The values for the parameters “Position” and “Velocity” are taken from networks 4 (chapter 4.2.4) and 9 (4.2.9).

Action	12. Set the bit „Absolute_EXECUTE“ (V150.0) and apply the settings via .
Response	The spindle is started in positive direction (“A0.2”=”1”, “HC0” increases) and is accelerated to the frequency “Velocity_p_s”=”20000” (“SMD172”) per second. The spindle is decelerated before it reaches the target position and it stops with the value of the high-speed counter “HC0” at the target position “+24000” pulses at point A.

4.2.11 Moving to point B

Since the position data of point B refers to point A, we use the relative positioning function “Q0_0_MoveRelative” to move to point B.

Figure 4-13



Num_Pulses

The distance is converted into the pulse number “Num_Pulses” in network 9.

Action 13. Enter the value “1500.0” mm (B=A+1500mm) into the status table for the variable VD130 “Millimeter” and apply the settings via .

Response The variable VD134 “Pulses” will then show the value of “300000” pulses.

Velocity

The converted travel speed “Velocity” is taken over unchanged from network 4.

Direction

Point B lies in positive count direction from point A.

Action 14. Set the directional bit “Direction” (V120.1 = “1”) and apply the settings via .

EXECUTE

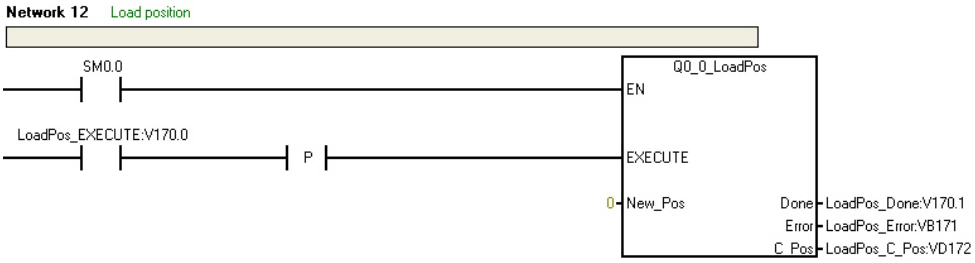
Action 15. Set the bit “Relative_EXECUTE” (V160.0 = „1“) and confirm the settings by clicking on .

Response The spindle is started in positive direction (“A0.2”=”1”, “HC0” increases) and is accelerated to the frequency “Velocity_p_s”=”20000” (“SMD172”) per second. The spindle is decelerated before it reaches the target position and it stops with the value of the high-speed counter “HC0” at the absolute position of point B “+324000” pulses (= +24000 + 30000).

4.2.12 Resetting the position counter

The position counter is reset in network 12 using the function “Q0_0_LoadPos”.

Figure 4-14

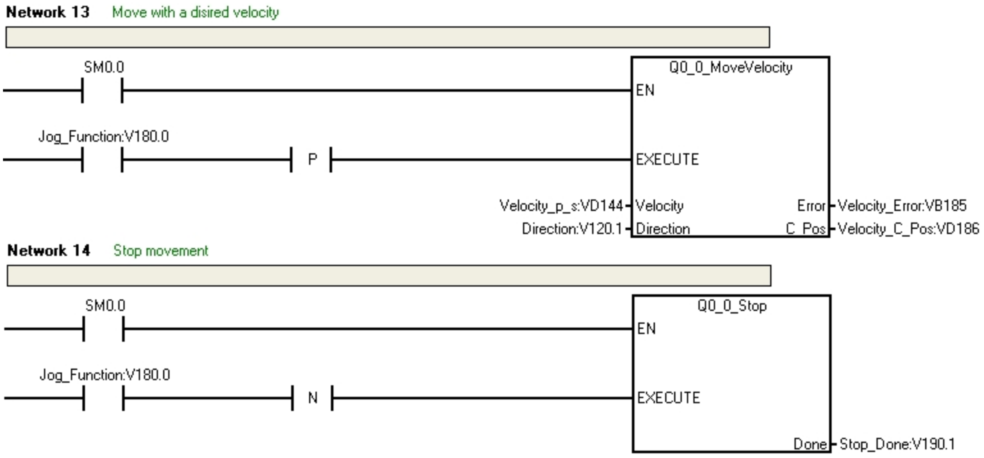


Action	16. Set the bit “LoadPos_EXECUTE” (V170.0 = “1”) and confirm the settings via .
Response	The position counter “LoadPos_C_Pos” (VD172) is set to the “New_Pos” parameter value (“0”) and the Done message bit “LoadPos_Done” is set (“V170.1” = “1”).
Note	The position counters “CTRL_C_Pos” (VD113), “LoadPos_C_Pos” (VD172) and “Velocity_C_Pos” (VD186) are identical to the high-speed counter HC0.

4.2.13 Realizing the jog mode

The “Jog_Function” bit (V180.0) calls the block “Q0_0_MoveVelocity” in network 13 with a rising edge and stops the positioning by calling the function “Q0_0_Stop” in network 14 with a falling edge.

Figure 4-15



Velocity

The converted travel speed “Velocity” is taken over unchanged from network 4.

Direction

Positioning should be carried out in negative count direction.

Action	17. Reset the directional bit “Direction” (V120.1 = “0”) and apply the settings via .

EXECUTE

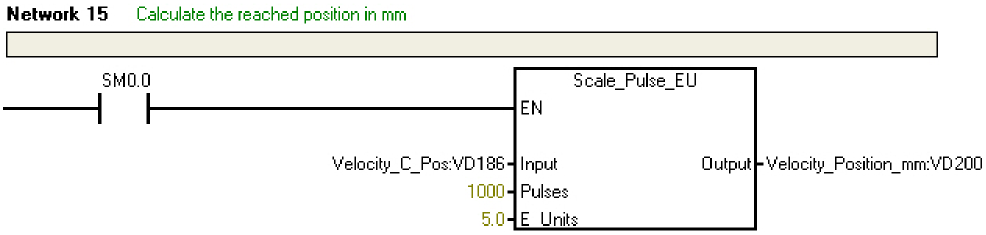
Action	18. Start the jog function using the “Jog_Function” bit (V180.0 = “1”) and apply the settings with .
Response	The spindle is started in negative direction (“A0.2”=”0”, the count value VD186 “Velocity_CPos” decreases) and it is accelerated to the frequency “Velocity_p_s”=”20000“ („SMD172“) per second.
Action	19. Finish the jog function using the “Jog_Function” bit (V180.0 = “0”) and apply the settings with .
Response	The spindle is decelerated until it stands still within the time “delta_T” (VD90) and the Done message bit “Stop_Done” is set (“V190.1” = “1”).

4.2.14 Determining the traveled distance

The pulse position “Velocity_C_Pos” (VD186) is converted into the path position “Velocity_Position_mm” (VD200) using the function “Scale_

Pulse_EU" in network 15. This value is identical to the traveled distance because the position counter HC0 was reset (chapter 4.2.12)

Figure 4-16



Network 15 Calculate the reached position in mm

Response The variable "Velocity_Position_mm" (VD200) indicates the traveled distance in millimeters. The sign is negative due to the specified direction.

5 Positioning diagrams for the function "Q0_x_Home"

When the reference point has been found, the spindle will always stop due to the negative edge of the reference point signal. This causes the object carrier to stop at the right or the left end of the reference point depending on the direction "Final_Dir".

Furthermore, a case discrimination is made every time the deceleration process is triggered by the reference point limit switch in the target direction "Final_Dir". This case discrimination verifies whether the deceleration to the frequency "Homing_Slow_Spd" is completed before reaching the negative edge of the reference point limit switch signal. The spindle will either stop immediately if the edge of the reference point signal is negative or it is accelerated in the opposite direction.

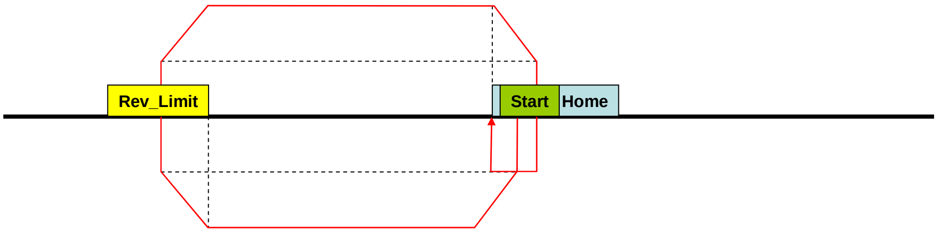
Overview

Table 5-1

Start_Dir	Final_Dir	Start (x from Home)	Stop (x from Home)	Case discrimination
0	0	right	left	yes
0	0	left	left	no
0	1	right	right	no
0	1	left	right	yes
1	0	left	left	no
1	0	right	left	yes
1	1	left	right	yes
1	1	right	right	no

The case that the start position is located within the reference point limit switch signal is covered by the mentioned cases, since a response will only take place to the positive edge of the reference point limit switch (see Figure 5-1).

Figure 5-1



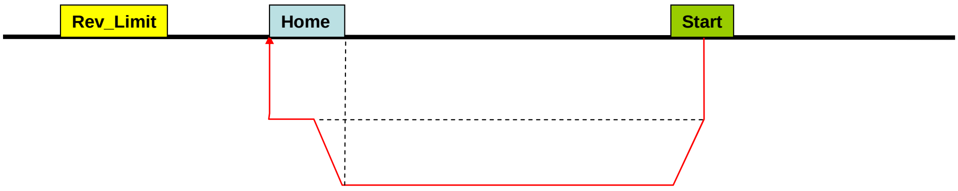
This case corresponds to the same positioning diagram as in Figure 5-4.

5.1 Start_Dir=0, Final_Dir=0

5.1.1 Start position: to the right of the reference point

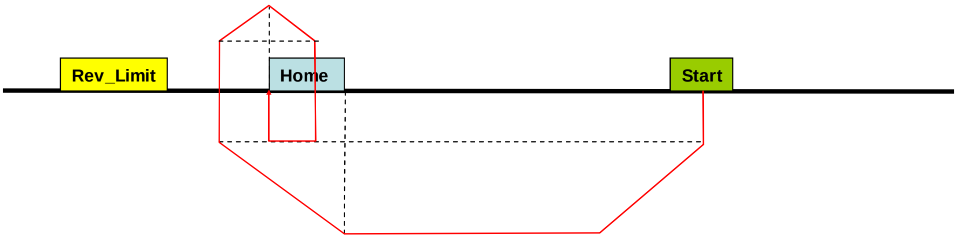
Case A: Deceleration to “Homing_Slow_Spd” has been completed before reaching the negative edge of the reference point input

Figure 5-2



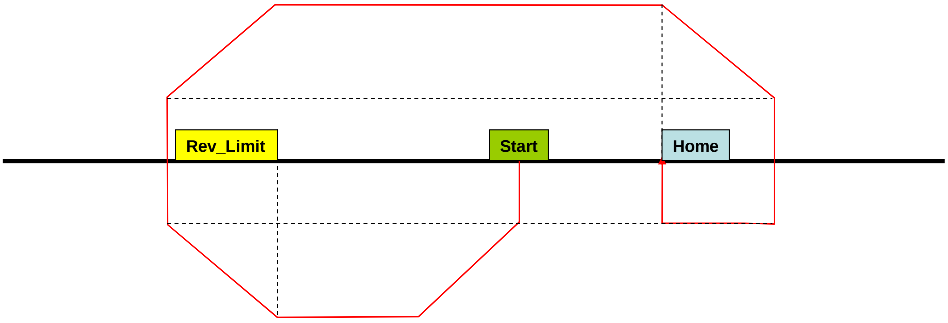
Case B: Deceleration to “Homing_Slow_Spd” has not been completed before reaching the negative edge of the reference point input

Figure 5-3



5.1.2 Start position: to the left of the reference point

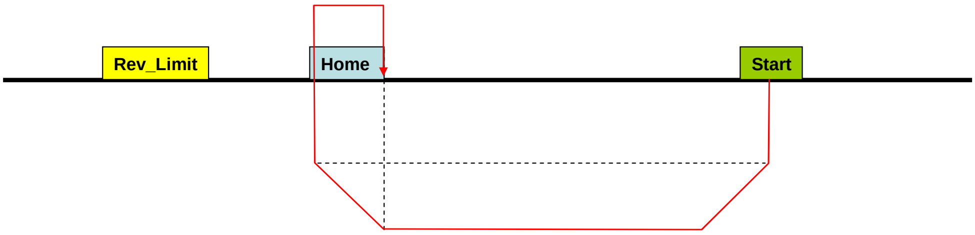
Figure 5-4



5.2 Start_Dir=0, Final_Dir=1

5.2.1 Start position: to the right of the reference point

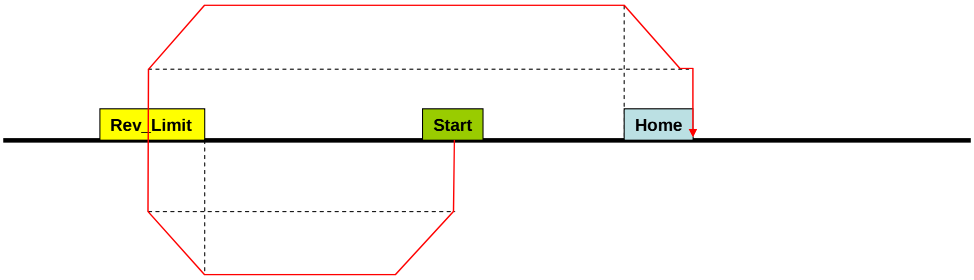
Figure 5-5



5.2.2 Start position: to the left of the reference point

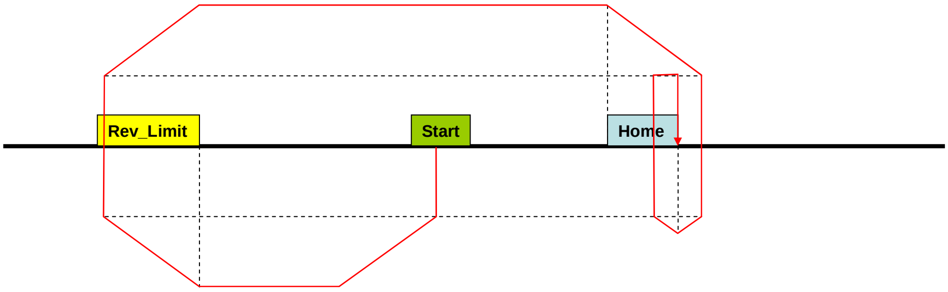
Case A: Deceleration to “Homing_Slow_Spd” has been completed before reaching the negative edge of the reference point input

Figure 5-6



Case B: Deceleration to “Homing_Slow_Spd” has not been completed before reaching the negative edge of the reference point input

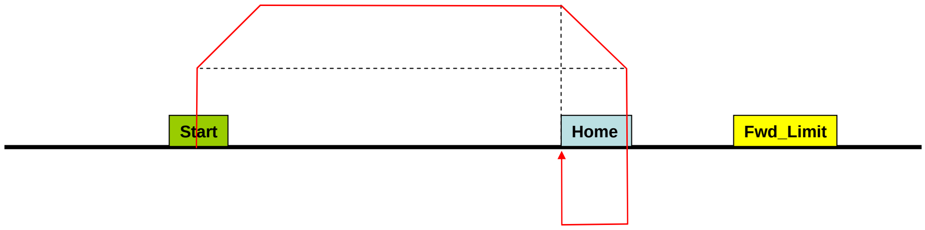
Figure 5-7



5.3 Start_Dir=1, Final_Dir=0

5.3.1 Start position: to the left of the reference point

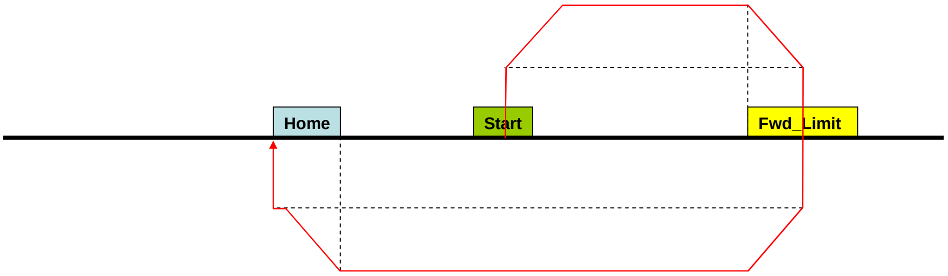
Figure 5-8



5.3.2 Start position: to the right of the reference point

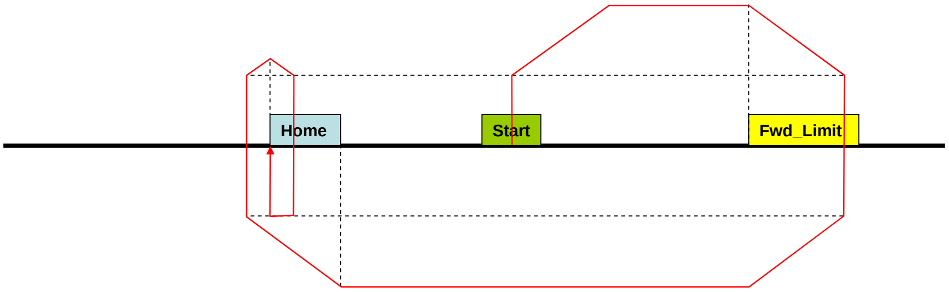
Case A: Deceleration to “Homing_Slow_Spd” has been completed before reaching the negative edge of the reference point input

Figure 5-9



Case B: Deceleration to “Homing_Slow_Spd” has not been completed before reaching the negative edge of the reference point input

Figure 5-10

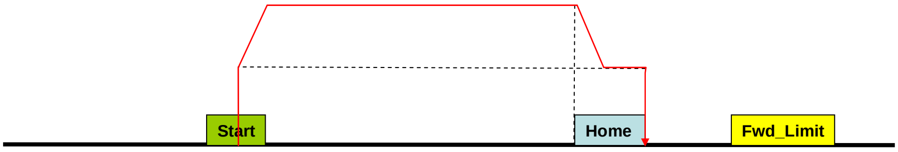


5.4 Start_Dir=1, Final_Dir=1

5.4.1 Start position: to the left of the reference point

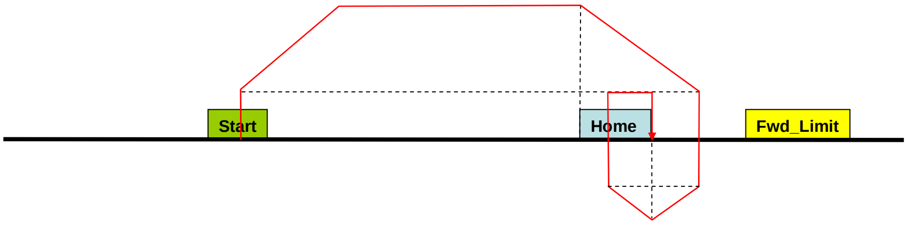
Case A: Deceleration to “Homing_Slow_Spd” has been completed before reaching the negative edge of the reference point input

Figure 5-11



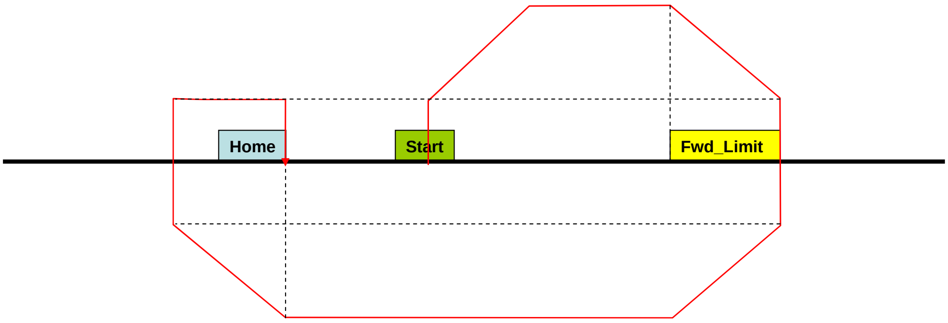
Case B: Deceleration to “Homing_Slow_Spd” has not been completed before reaching the negative edge of the reference point input

Figure 5-12



5.4.2 Start position: to the right of the reference point

Figure 5-13


Home
Start
Fwd_Limit