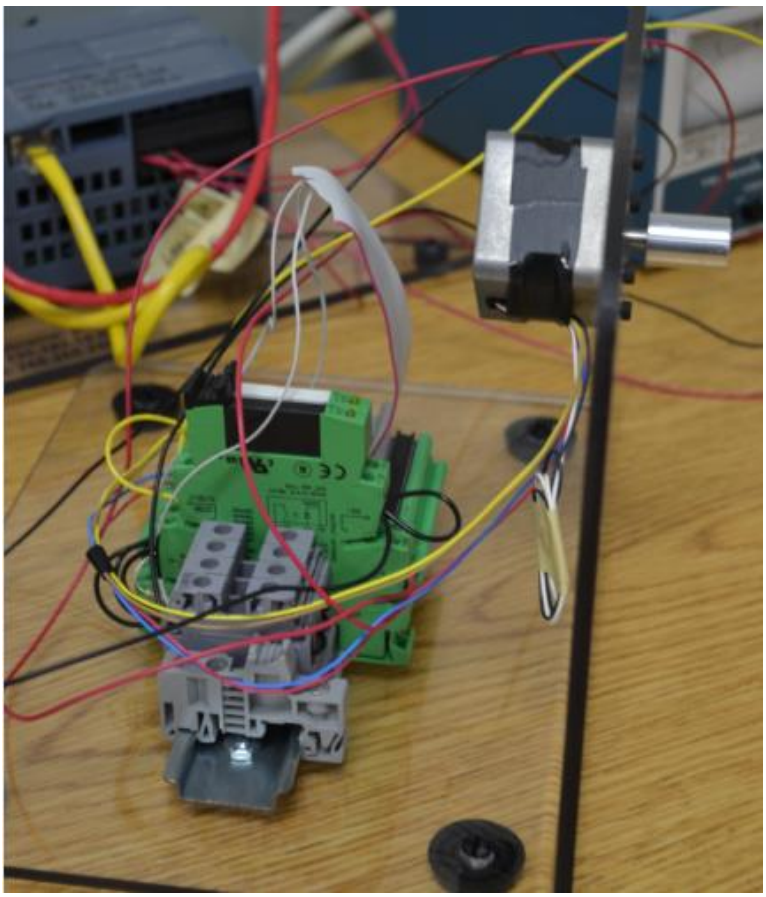
Chapter 18 Single Axis Stepper Control

Stepper Lab

While this stepper lab is are important, a later lab using multiple stepper drives is preferred. This single axis stepper control was first designed to interface with an Etch-a-Sketch. That design didn't work well due to limitations of the Etch-a-Sketch.

The present design has similar requirements to the servo in that a starter program was given with the student required to enhance the program with a HMI program as well as control of the motor in auto and manual.

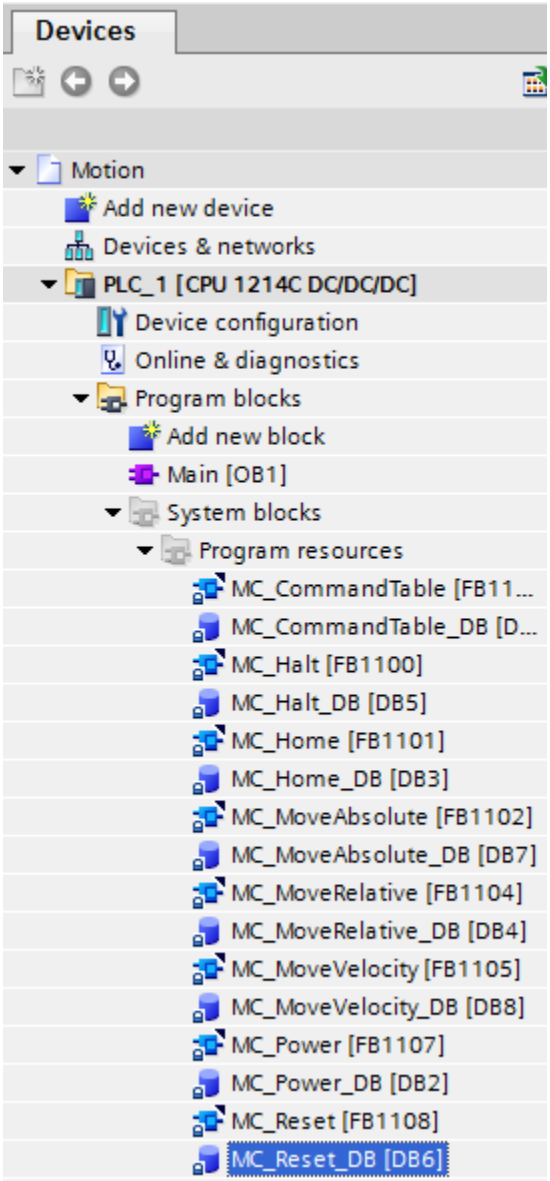


Stepper Motor and Controller Hybrid system using custom boards and Siemens PLC

One of the design components of this lab was the green Phoenix Contact interface devices. It was felt that students should be introduced to common devices used by industry rather than trying to build the interface on a breadboard. Cost does increase with this design with the cost of each of the interface devices >$100. The concept of off-the-shelf parts is an important one for the student to learn, however.

Programming the Stepper

The Siemens PLC is used to generate the pulses controlling the stepper motor through the EVAL6208N unit. Software configuration is described in this section. Use this application to successfully start and control the stepper motor.



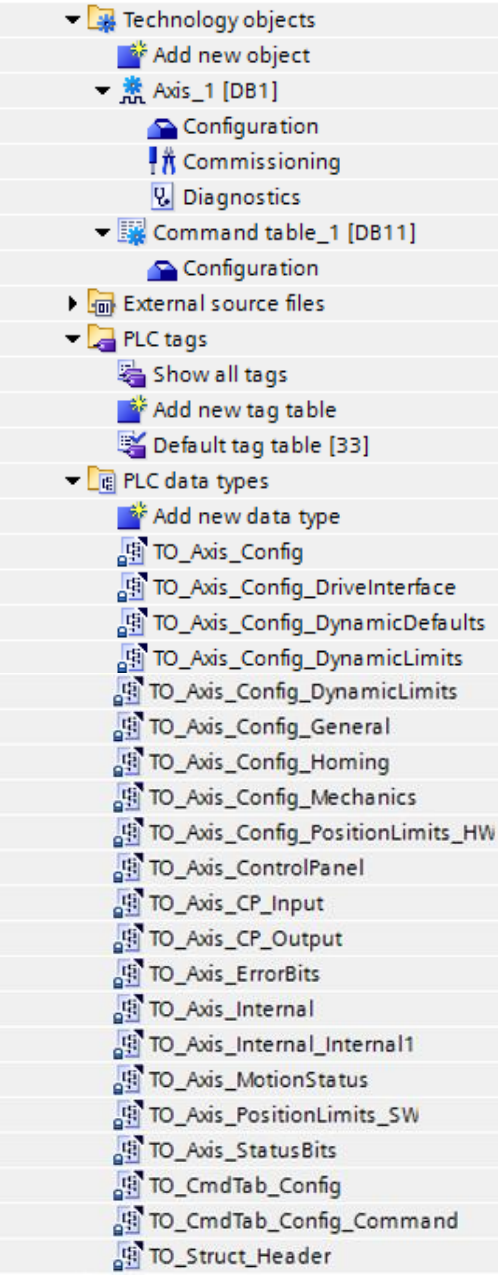
The Project Tree for a single axis PTO drive. There are a number of blocks programmed in the example which allow the user to experiment with the various blocks using the inputs from the selector switches. The selector switches are useful but must be supplemented by the HMI panel in order to provide a complete project.

The various function and data blocks as well as tags are created in the steps following.

Follow the steps below in the order given to use motion control with the CPU S7-1200. The subject will be broadly divided into the following steps. The text will only cover portions of these:

1. Add technological object Axis
2. Working with the configuration dialog
3. Download to CPU
4. Function test of the axis in the commissioning window
5. Programming
6. Diagnostics of the axis control

The axis program given already has commissioned the drive and has several programming blocks present to test the operation of the stepper. The following gives the list of programming blocks available plus the Command Table blocks. The Command Table block allows the user to enter a number of commands in a sequence for execution as a block.



The Project Tree continued

Homing modes:

Active homing

In active homing mode, the motion control instruction "MC_Home" performs the required reference point approach. When the homing switch is detected, the axis is homed according to the configuration. Active traversing motions are aborted.

Passive homing

During passive homing, the motion control instruction "MC_Home" does not carry out any homing motion. The traversing motion required for this step must be implemented by the user via other motion control instructions. When the homing switch is detected, the axis is homed according to the configuration. Active traversing motions are not aborted upon start of passive homing.

Direct homing absolute

The axis position is set regardless of the homing switch. Active traversing motions are not aborted. The value of input parameter "Position" of motion control instruction "MC_Home" is set immediately as the reference point of the axis.

Direct homing relative

The axis position is set regardless of the homing switch. Active traversing motions are not aborted. The following statement applies to the axis position after homing:
New axis position = current axis position + value of parameter "Position" of instruction "MC_Home".

Overview of the Motion Control Statements:

You control the axis with the user program using motion control instructions. The instructions start Motion Control jobs that execute the desired functions.

The status of the motion control jobs and any errors that occur during their execution can be obtained from the output parameters of the motion control instructions. The following Motion Control instructions are available:

MC_Power:	Enable, disable axis
MC_Reset:	Acknowledge error
MC_Home:	Home axes, set home position
MC_Halt:	Halt axis
MC_MoveAbsolute:	Absolute positioning of axes
MC_MoveRelative:	Relative positioning of axes
MC_MoveVelocity:	Move axes at preset rotational speed

MC_Moveog: Move axes in jogging mode
MC_CommandTable: Run axis jobs as movement sequence
MC_ChangeDynamic: Changing the dynamic settings for the axis

Creating a user program

Proceed as follows to create the user program:

1. In the project tree, double-click your code block (the code block must be called in the cyclic program). The code block is opened in the programming editor and all available instructions are displayed.

2. Open the “Technology” category and the “Motion Control” and “S7-1200 Motion Control” folders.

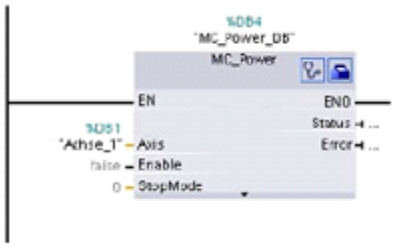
3. Use a drag-and-drop operation to move the “MC_Power” instruction to the desired network of the code block. The dialog box for defining the instance DB opens.

4. In the next dialog box, select from the following alternatives:
   Single instance
   Click “Single instance” and select whether you want to define the name and number of the instance DB automically or manually.
   Multi-instance
   Click “Multi-instance” and select whether you want to define the name of the multi-instance automatically or manually.

5. Click “ok”. The motion control instruction “MC_Power” is inserted into the network:

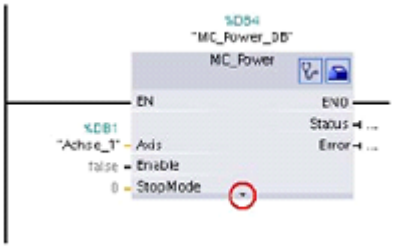
   Parameters marked with “<???>” must be initialized; all other parameters are assigned default values. Parameters displayed in black are required for use of the motion control instruction.

6. Select technology object in the project tree and drag-and-drop it on <???>.



Following selection of the technology object data block, the stethoscope button is available. Click the stethoscope if you want to open the diagnostics dialog for the technology object.

Click the toolbox icon if you want to open the configuration view of the technology object:



Click the toolbox icon if you want to open the configuration view of the technology object. Click the arrow down icon to view additional parameters of the motion control instruction.

7. Add your choice of motion control instructions from steps 3 to 6 above.

Programming notes:

When creating your user program, note the following:

- Cyclic call of utilized motion control instructions. The current status of command execution is available via the output parameters of the motion control instruction. The status is updated with every call of the motion control instruction. Therefore, make sure that the utilized motion control instructions are called cyclically.
- Transfer of parameter values of a motion control instruction. The parameter values pending for the input parameters are transferred with a positive edge at input parameter "execute" when the block is called. The motion control command is started with these parameter values. Parameter values that are subsequently changed for the motion control instruction are transferred until the next start of the motion control command. Exceptions to this are input parameters "StopMode" of motion control instruction "MC_Power" and "Velocity" of motion control instruction "MC_MoveJog". A change in the input parameter is also applied when "Enable" = true or "JogForward' and "JogBackward"…

- Programming under consideration of the status information. In a stepwise execution of motion control jobs, make sure to wait for the active command to finish before starting a new command. Use the status messages of the motion control instruction and the "StatusBits" tag of the technology object to check for completion of the active command.

  In the example below, observe the indicated sequence. Failure to observe the sequence will display an axis or command error.

  - Axis enable with motion control instruction "MC_Power"
    You must enable the axis before it can take on motion jobs. Use an AND operation of tag <Axis name>.StatusBits.Enable=TRUE with output parameter Status = TRUE of motion control instruction "MC_Power" to verify that the axis is enabled.
  - Acknowledge error with motion control instruction "MC_Reset". Prior to starting a motion control command, errors requiring acknowledgement must be acknowledged with "MC_Reset". Eliminate the cause of the error and acknowledge the error with motion control instruction "MC_Reset". Verify that the error has been successfully acknowledged before initiating a new command. For this purpose, use an AND operation of tag <Axis name>.StatusBits.Error = FALSE with output parameter Done = TRUE of motion control instruction "MC_Reset".
  - Home axis with motion control instruction "MC_Home"
    Before you can start an MC_MoveAbsolute command, the axis must be homed. Use an AND operation of tag <Axis name>.StatusBits.HomingDone= TRUE with output parameter Done = TRUE of motion control instruction "MC_Home" to verify that the axis has been homed.

- Override of motion control command processing
  Motion control jobs for moving an axis can also be executed as overriding jobs. If a new motion control command is started for an axis while another motion control command is active, the active command is overridden by the new command before the existing command is completely executed. The overridden command signals this using CommandAborted = TRUE in the motion control instruction. It is possible to override an active MC_MoveRelative command with a MC_MoveAbsolute command.
- Avoiding multiple use of the same instance
  All relevant information of a motion control command is stored in its instance. Do not start a new command using this instance, if you want to track the status of the current command. Use different instances if you want to track the commands separately. If the same instance is used for multiple motion control commands, the status and error information of the individual commands will overwrite each other.
- Call of motion control instructions in different priority classes (run levels). Motion control instructions with the same instance may not be called in different priority classes without interlocking. To learn how to call locked motion control instructions, refer to "Tracking commands from higher priority classes".

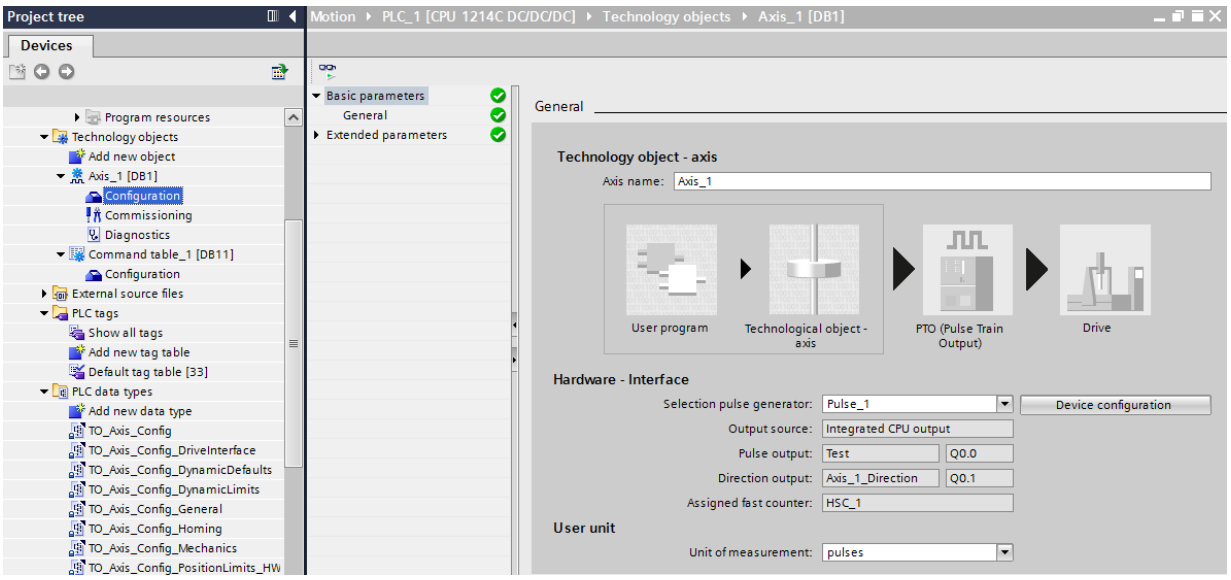
Monitoring active commands

There are three groups for tracking active motion control jobs, those with output parameter

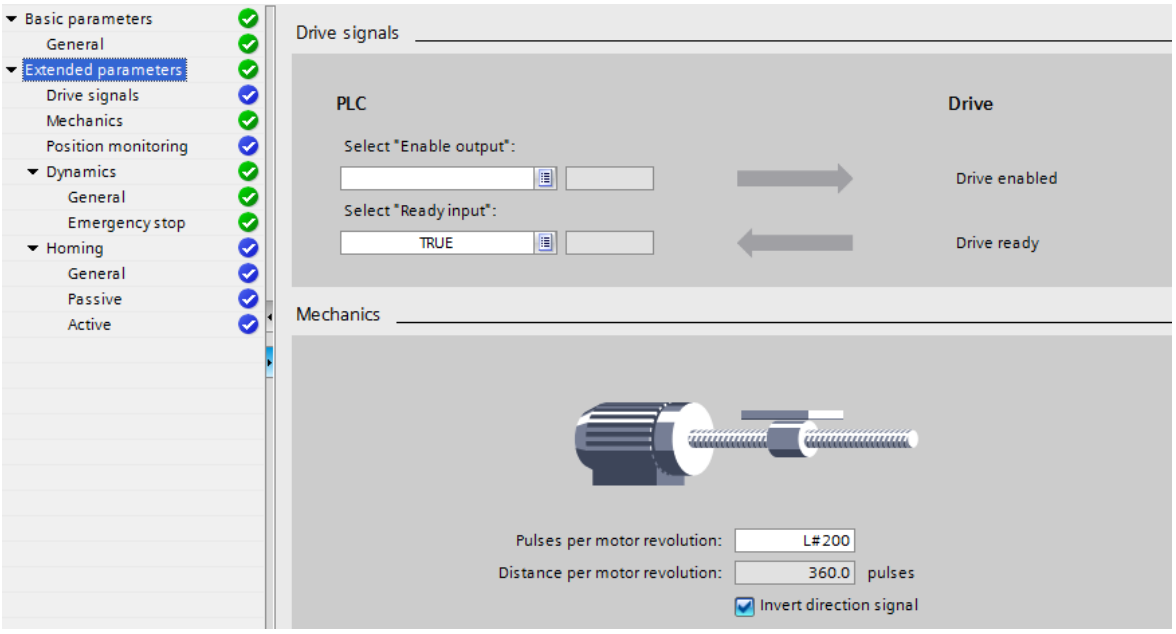
“Done”, instruction MC_MoveVelocity and instruction MC_MoveJog.

See the results of the instructions’ timing diagram with abort or error conditions. The WinCC book at about pg. 4000 shows these results.

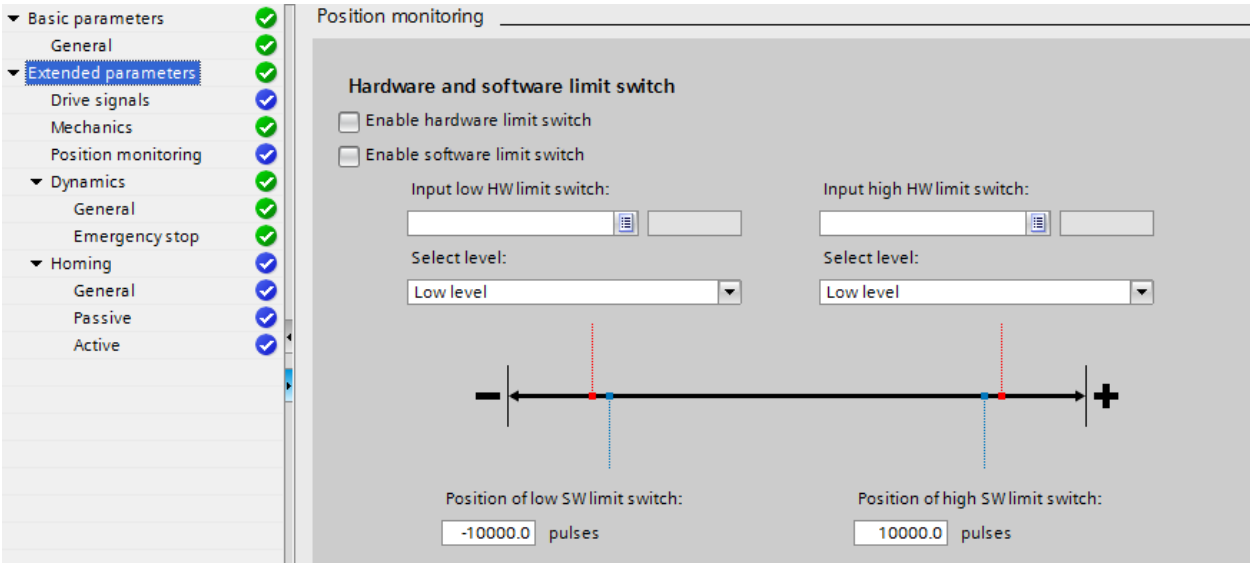
Create a Motion Technology Object:
Add new object> Select Axis> Axis_1 (define the axis)



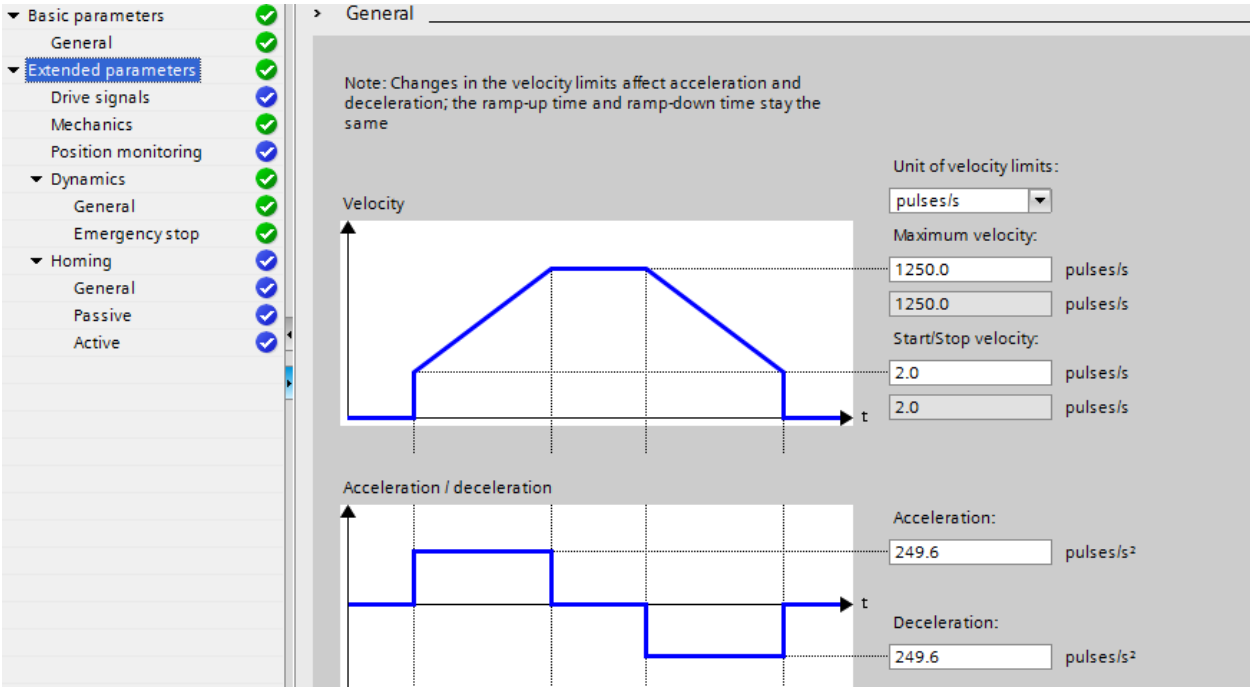
Define the mechanical parameters of the machine:
Steps per revolution
Distance per Motor Revolution
Invert Direction Signal



Leave hardware limits not enabled

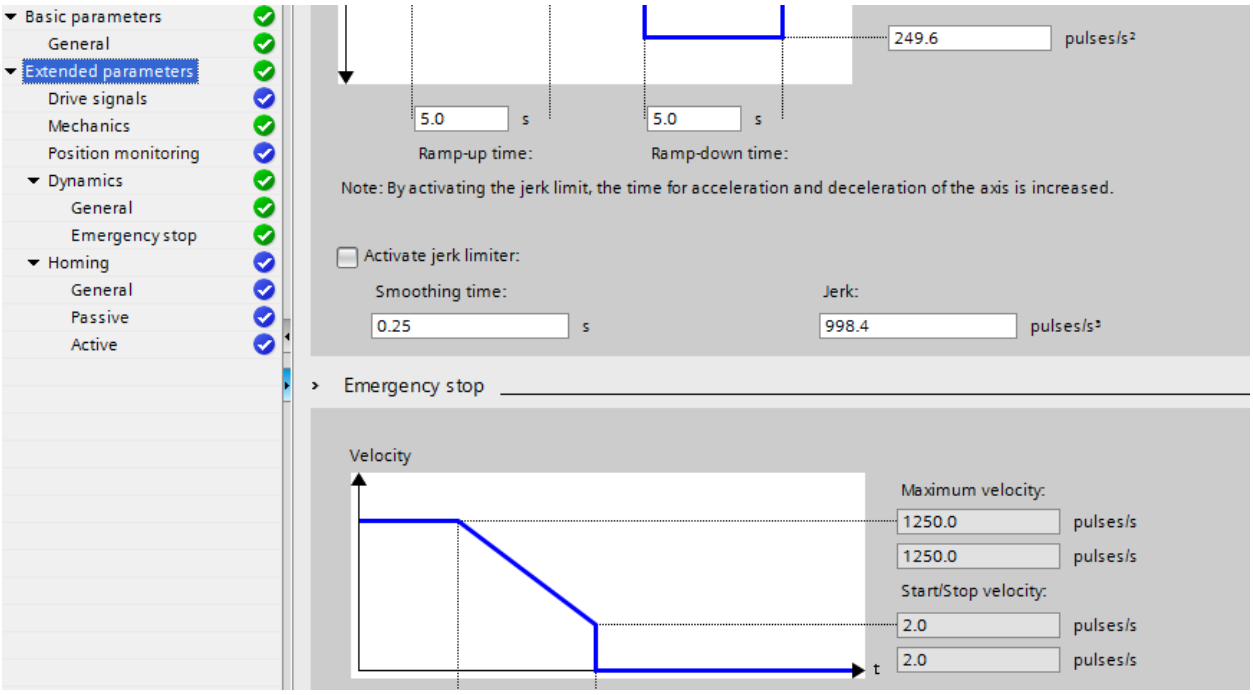


Determine the acceleration/deceleration for the axis:
Select units for pulses/s
Set maximum Velocity
Set Start/Stop Velocity

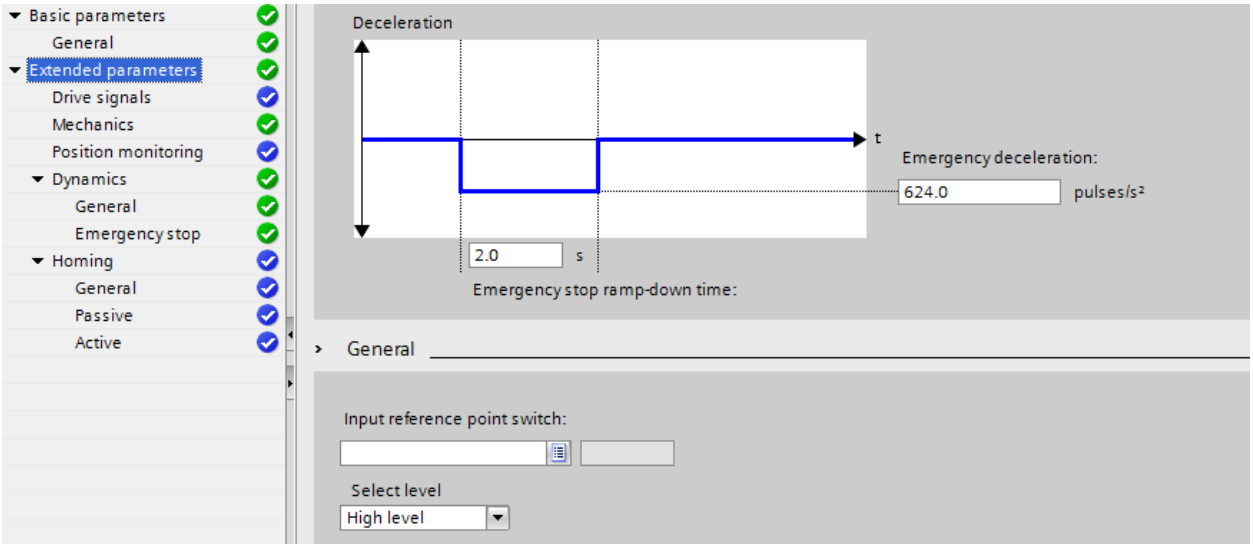


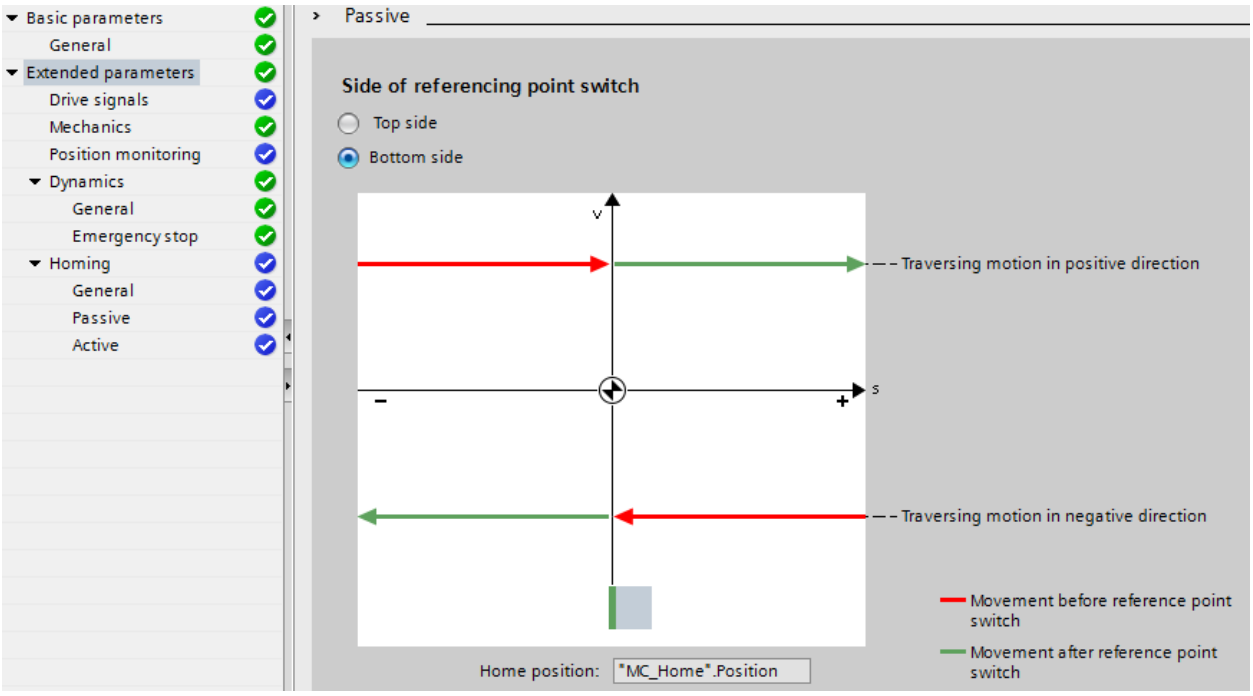
Set Ramp Up/Ramp Down Time

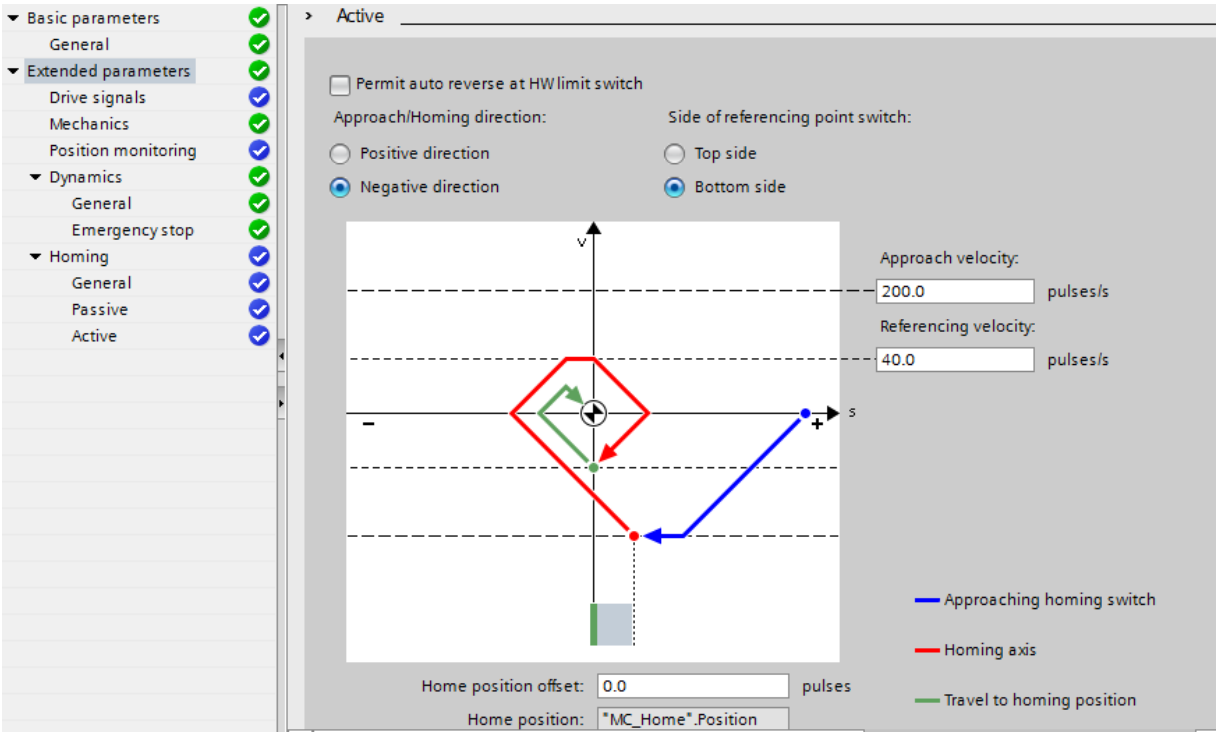
Set Emergency Ramp Time
Determine the Emergency Decel Time for immediate stops:



Emergency Stop Deceleration
Homing Configuration (leave open)





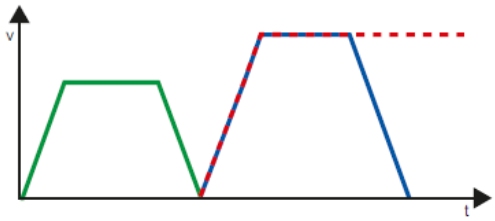
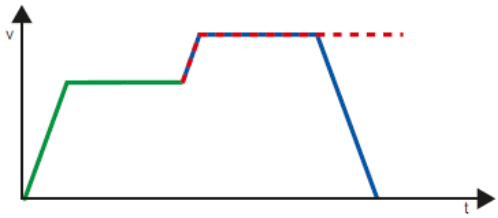
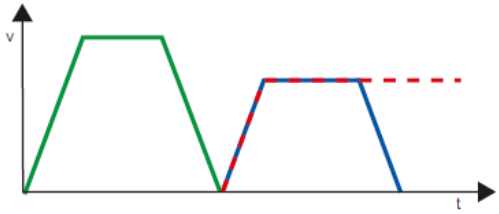
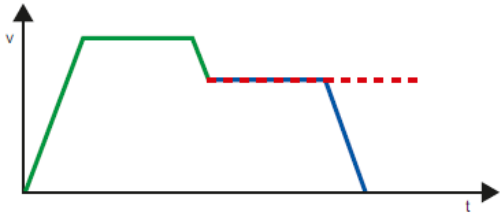


(The Technology Object for Motion configuration is complete.)

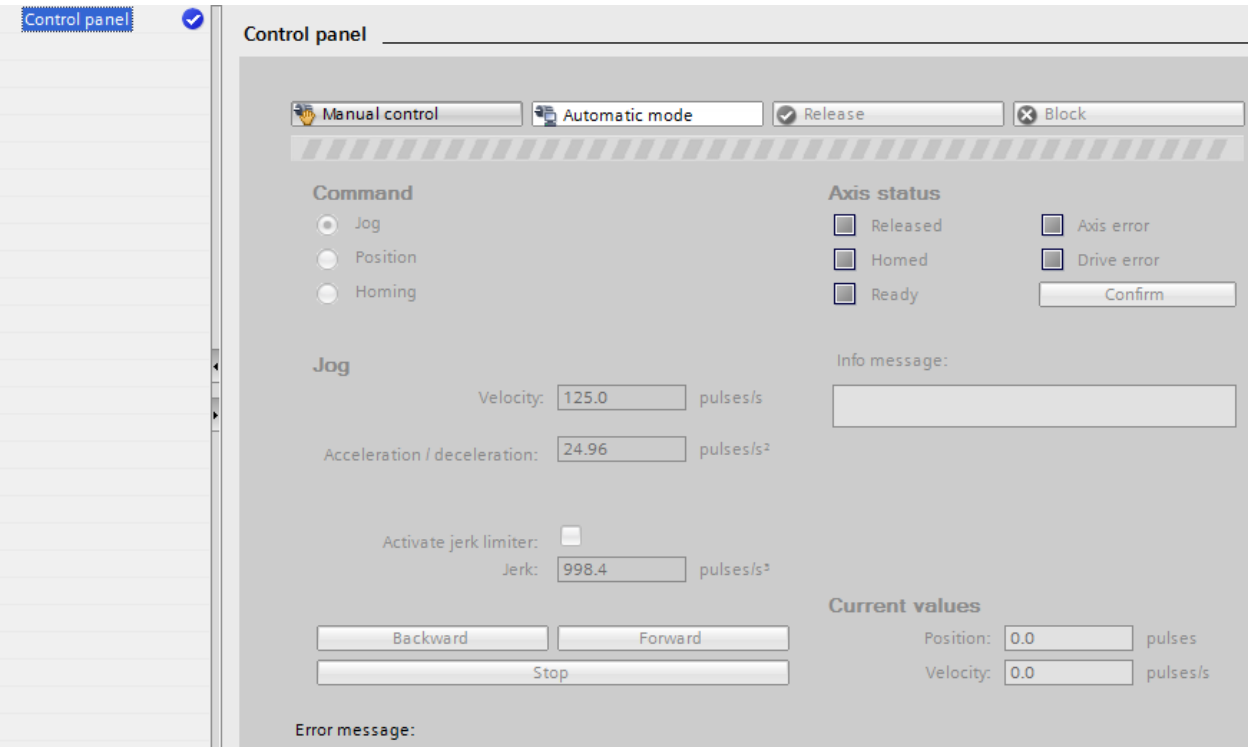
Motion transition with preceding velocity jobs

Transition from "Complete command" to "Blend motion"

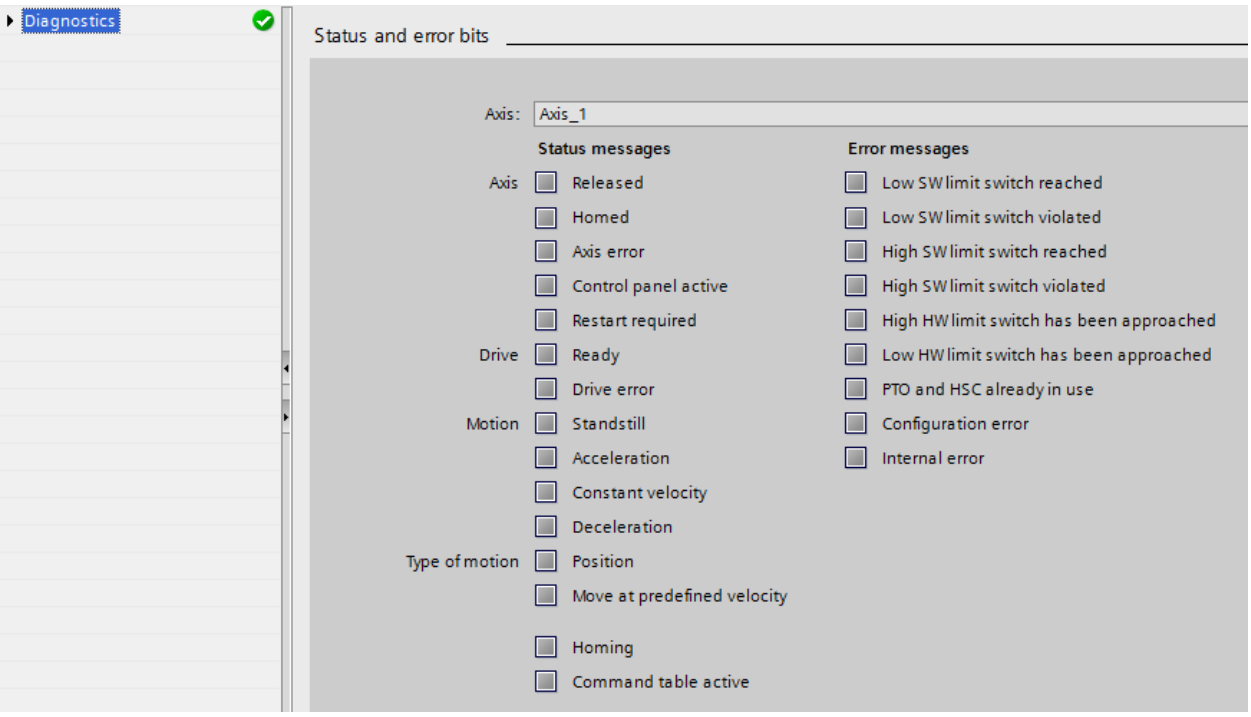
The charts below show the transition between movements in various different transition modes in the "Next step" column:

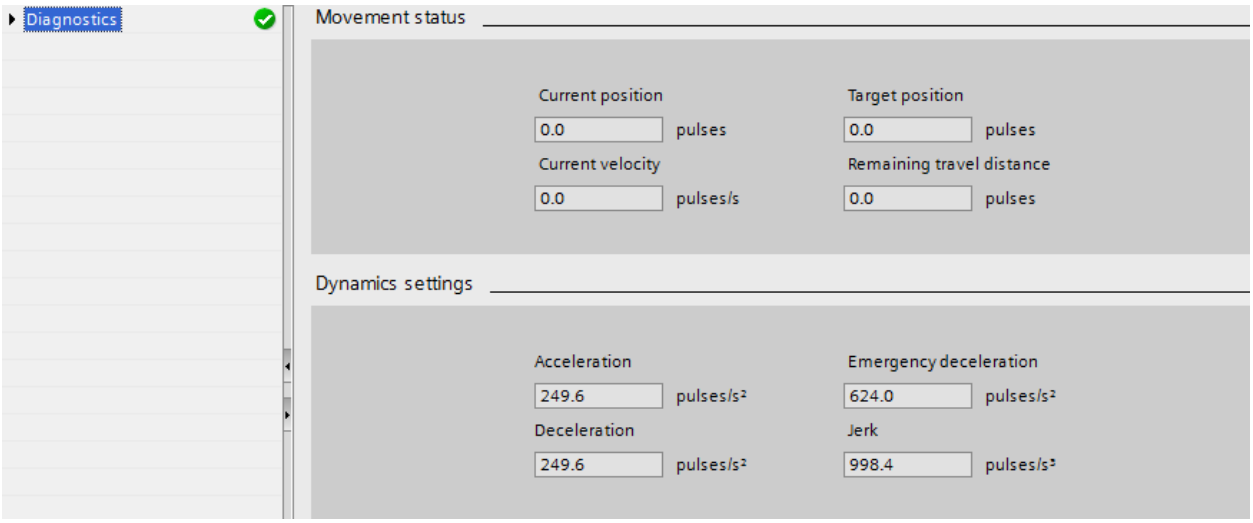
Complete job	Blend motion
Transition from lower to higher velocity  A job with high velocity is appended to a previous positioning job. The positioning job terminates at its target position at velocity "0". The second job starts from standstill.	Transition from lower to higher velocity  A job with high velocity is overlapped with a previous positioning job. The first positioning job terminates without standstill at its target position. The second job starts with the new velocity.
Transition from higher to lower velocity  A job with low velocity is appended to a previous positioning job. The positioning job terminates at its target position at velocity "0". The second job starts from standstill.	Transition from higher to lower velocity  A job with low velocity is overlapped with a previous positioning job. The first positioning job terminates without standstill at its target position. The first job starts with the new velocity.

Next begin commissioning:
Manually move the axis to verify operation
1 - Select manual control
2 - Enable axis
3 - Select action (jog)
4 - Set Home Position Offset
5 - Set Accel Rate
6 - Make it go

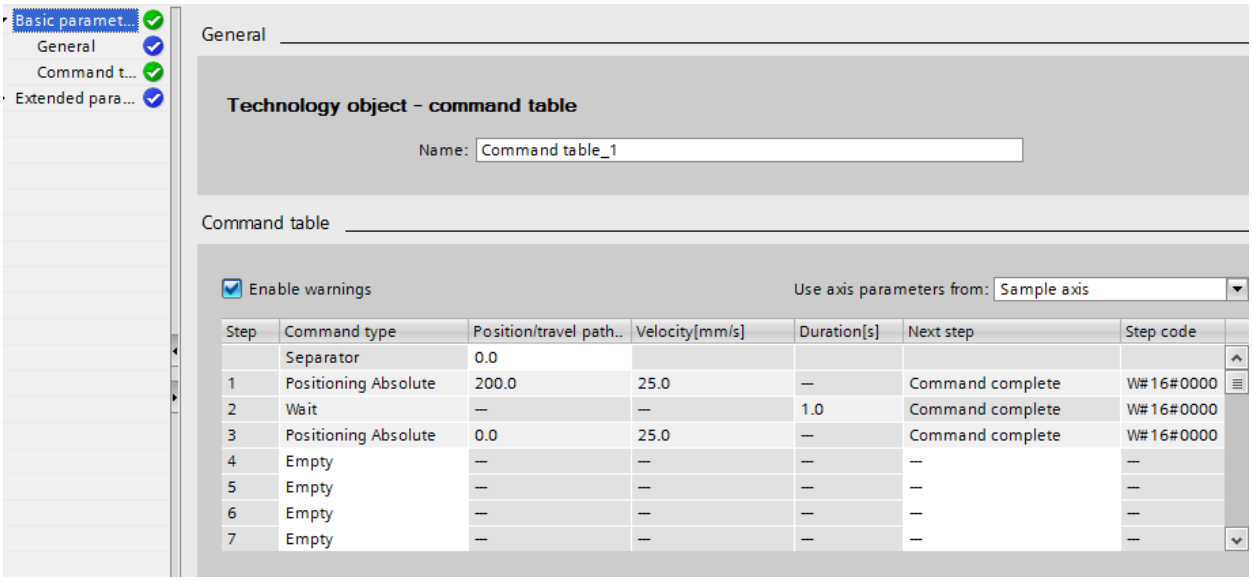

Control panel
Control panel
Manual control
Automatic mode
Release
Block
Command
Jog
Position
Homing
Axis status
Released
Axis error
Homed
Drive error
Ready
Confirm
Jog
Info message:
Velocity:
125.0
pulses/s
Acceleration / deceleration:
24.96
pulses/s²
Activate jerk limiter:
Jerk:
998.4
pulses/s³
Current values
Backward
Forward
Position:
0.0
pulses
Stop
Velocity:
0.0
pulses/s
Error message:

lk

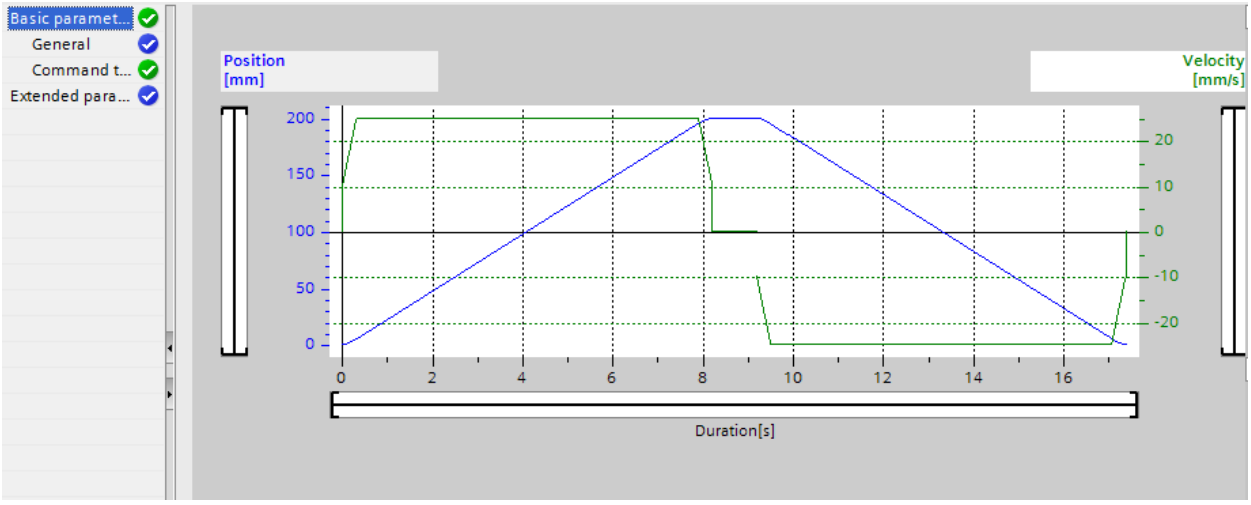

Diagnostics
Status and error bits
Axis:
Axis_1
Status messages
Error messages
Axis
Released
Homed
Axis error
Control panel active
Restart required
Drive
Ready
Drive error
Motion
Standstill
Acceleration
Constant velocity
Deceleration
Type of motion
Position
Move at predefined velocity
Homing
Command table active
Low SW limit switch reached
Low SW limit switch violated
High SW limit switch reached
High SW limit switch violated
High HW limit switch has been approached
Low HW limit switch has been approached
PTO and HSC already in use
Configuration error
Internal error



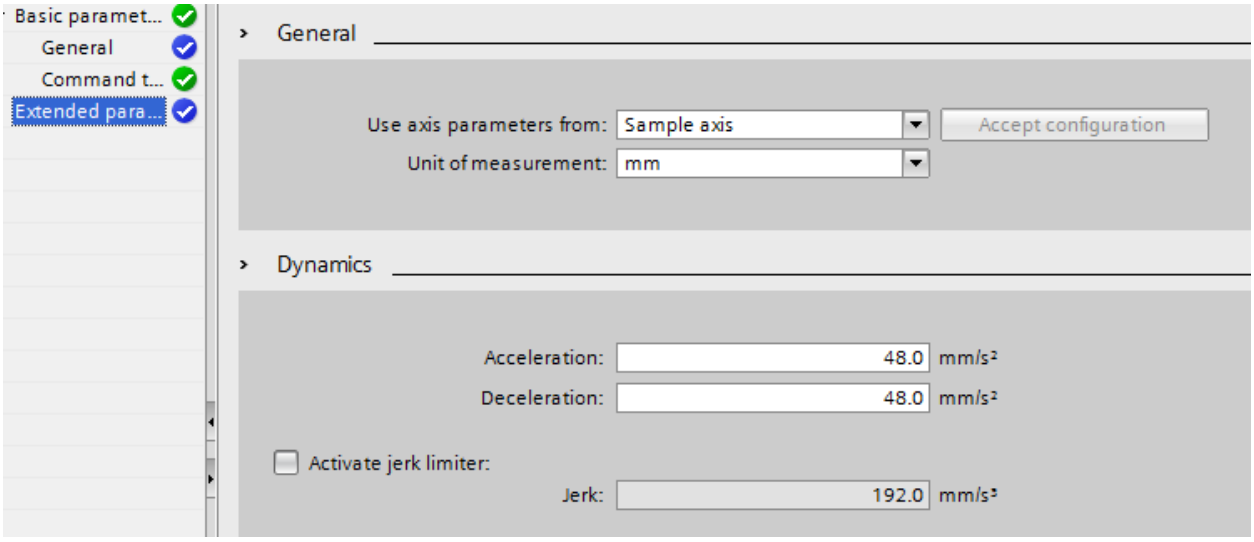
This figure shows Movement Status and Dynamic Settings

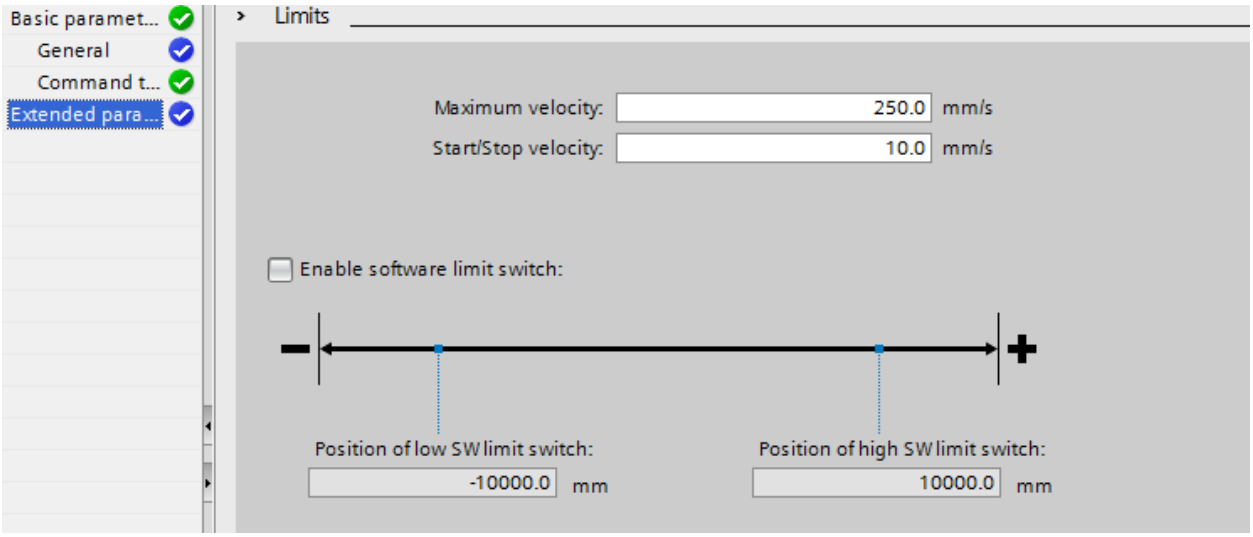


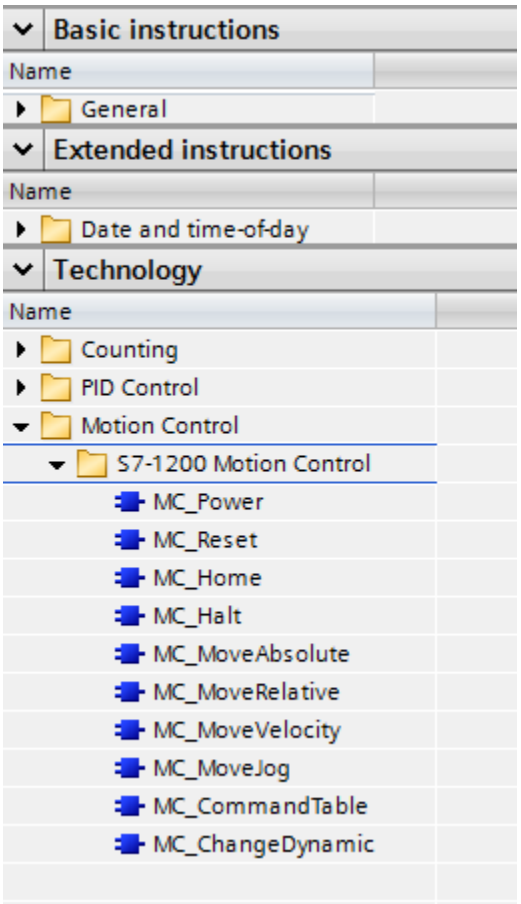
This figure shows a Command Table



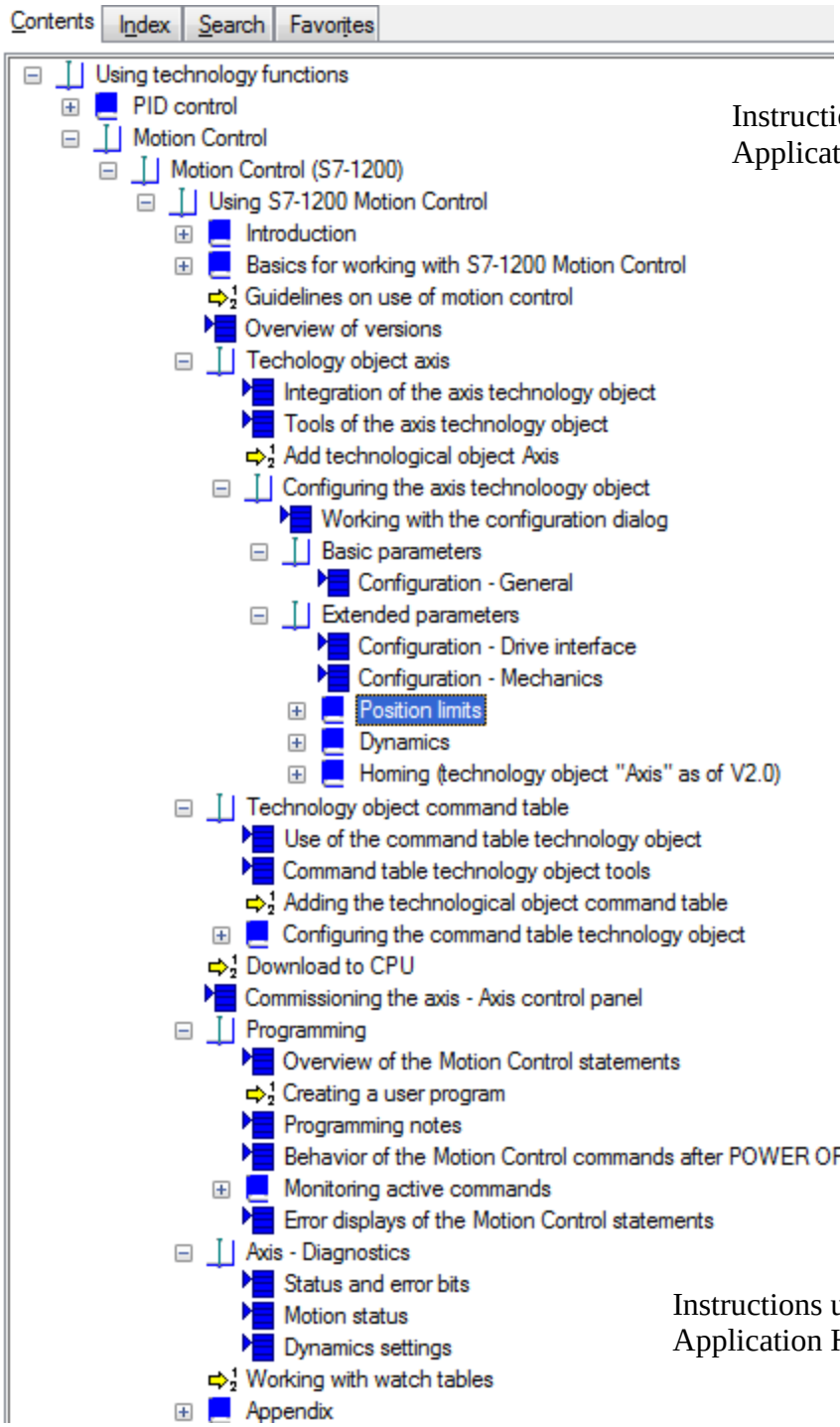
This figure shows the motion and position of the movements of the Command Table







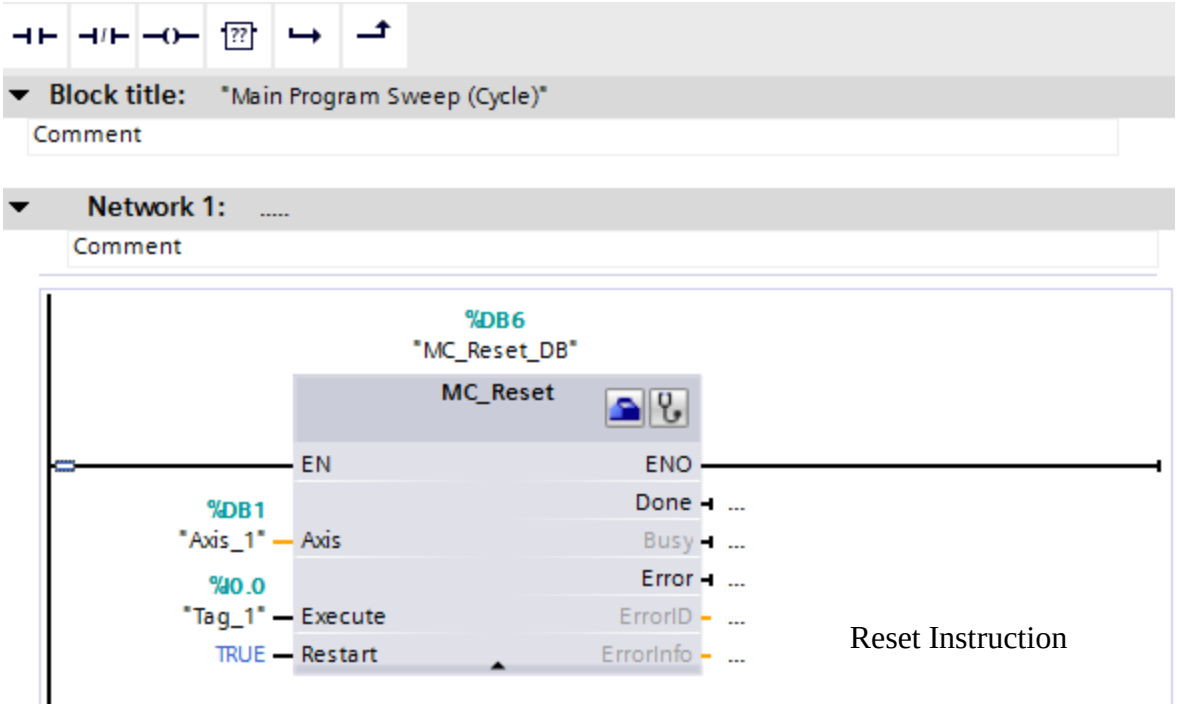
Instructions used in the Motion Application to control the axis



Instructions used in the Motion Application Help Screens

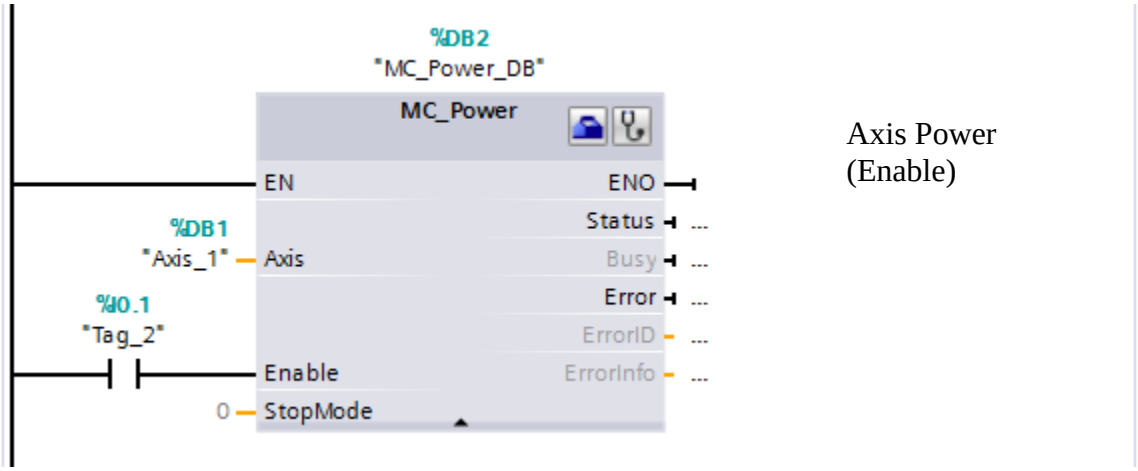
Instructions used in the Motion Application Help Screens

The following instructions are tied to inputs from the switched inputs directly wired to the PLC. Each input executes a specific action. For instance, I0.0, the first input, executes a drive reset instruction per the program below.



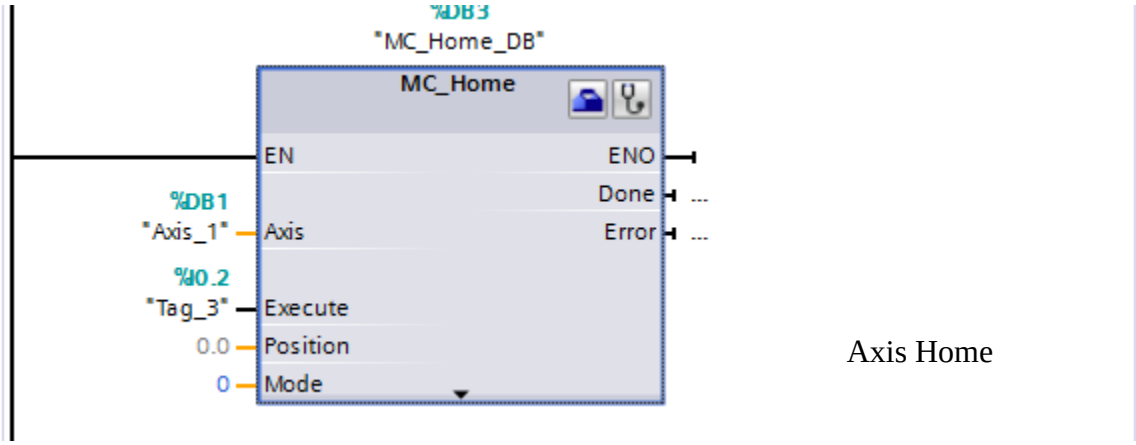
Reset Instruction

The Reset Instruction is used to reset the axis. It is referenced to I0.0 which is the first input on the switch panel on top of the PLC. Use this switch input to reset the axis.



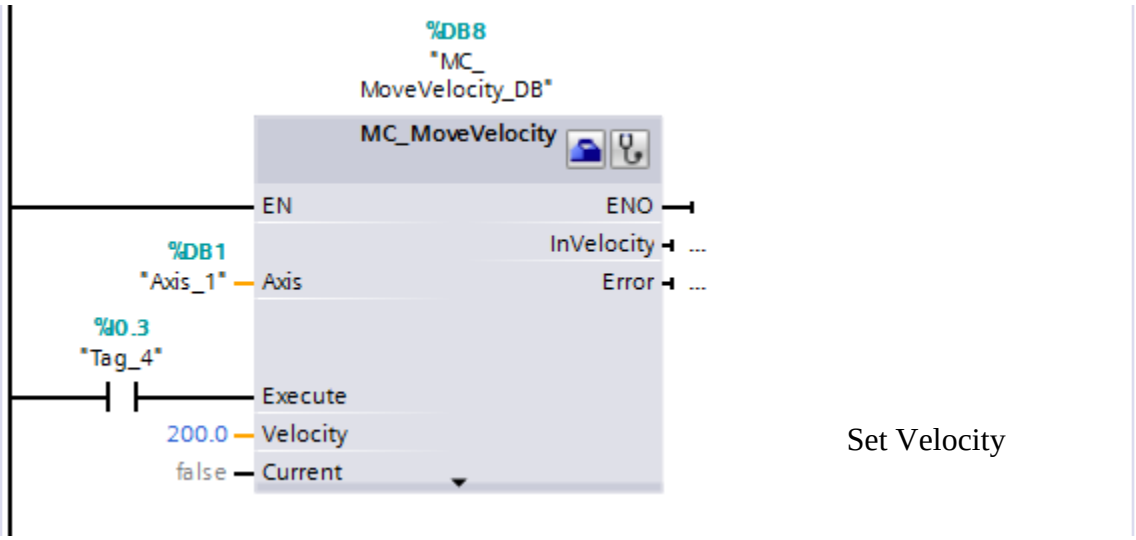
Axis Power (Enable)

The Power Instruction is used to enable the axis. It is referenced to I0.1 which is the second input on the switch panel on top of the PLC. Use this switch input to enable the axis.



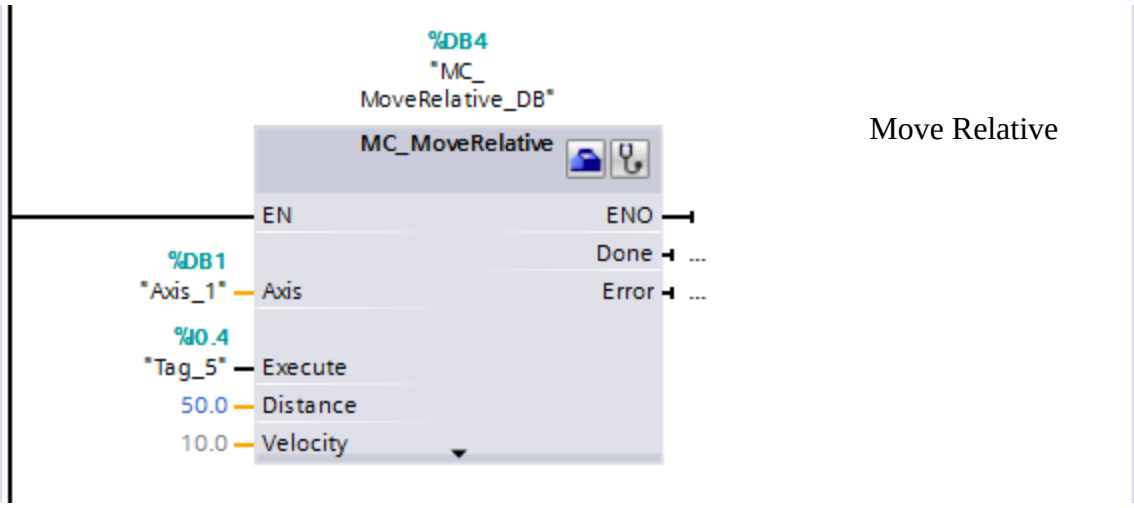
Axis Home

The Home Instruction is used to home the axis. It is referenced to I0.2 which is the third input on the switch panel on top of the PLC. If there are no home limit switches, the present position is used as the home position and absolute moves can be entered following the Home block being executed



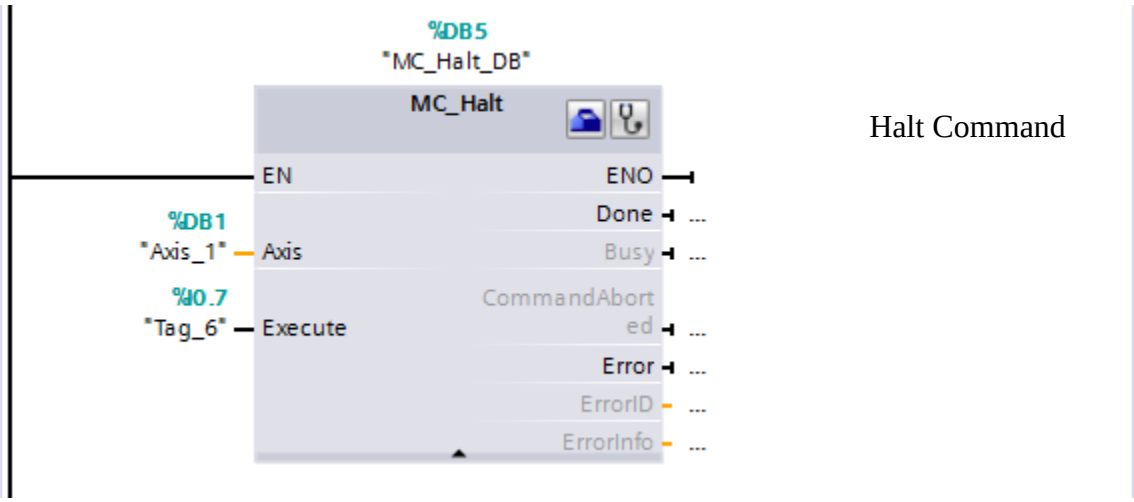
Set Velocity

The Move Velocity Instruction is used to set the velocity of the axis. It is referenced to I0.3.



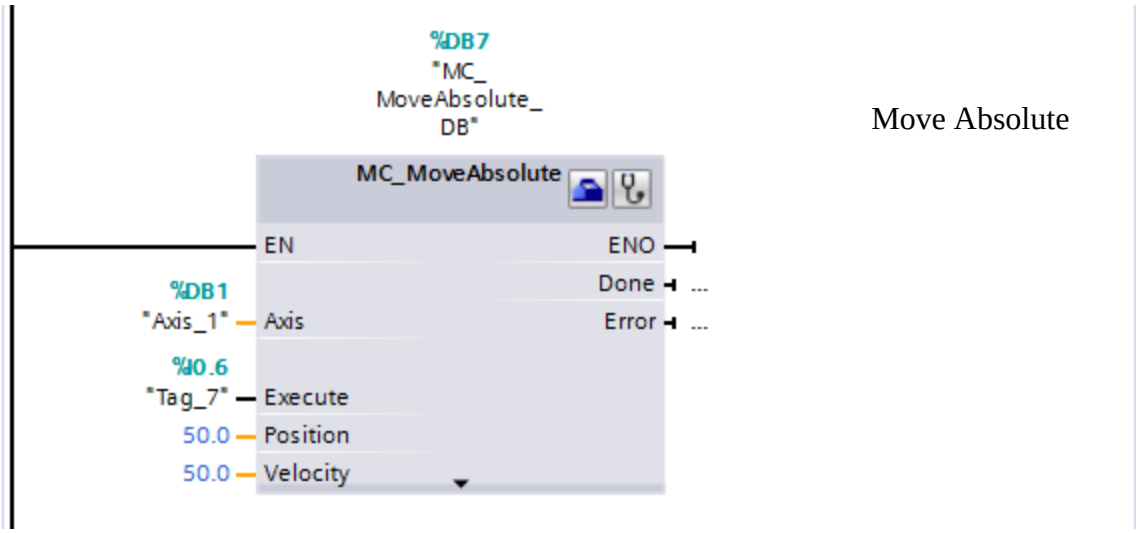
Move Relative

The Move Relative Instruction is used to trigger a relative move of the axis. It is referenced to I0.4.



Halt Command

The Halt Instruction is used to halt a move of the axis. It is referenced to I0.7.



Move Absolute

The Move Absolute Instruction is used to trigger an absolute move of the axis. It is referenced to I0.6. It will move the axis to a position relative to the home instruction.

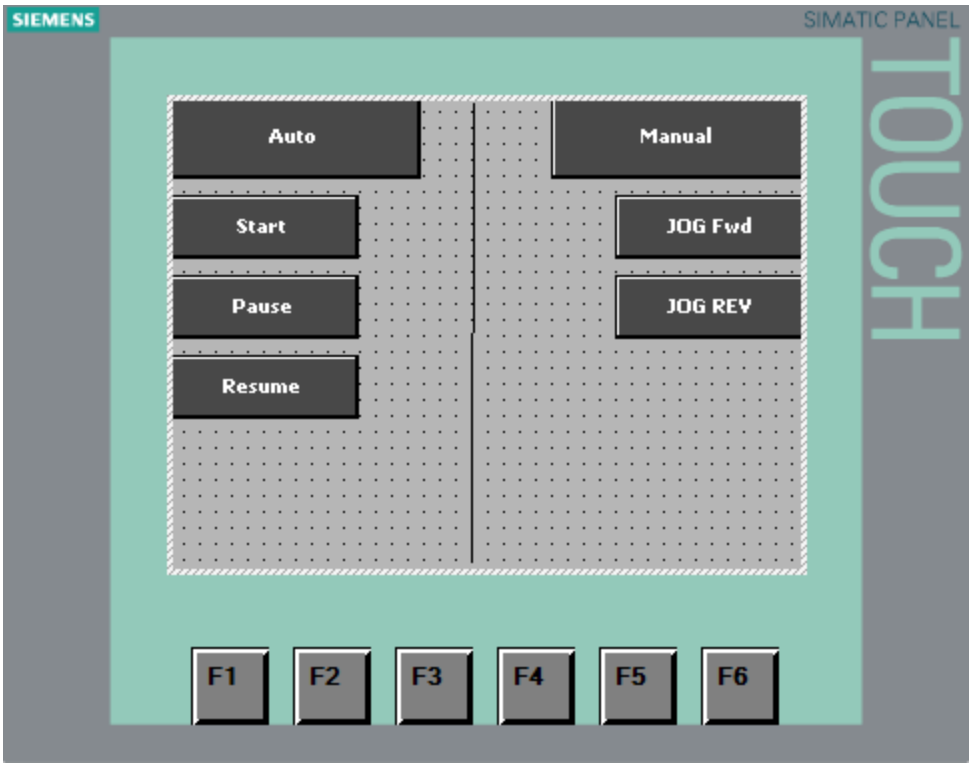
In addition to the discrete instruction listed above, a command table may be used to store the commands of a sequence of move commands for the Siemens processor. The same command table can be generated for Allen-Bradley although this feature was never completely made to work by the instructor or lab tech in the course to date. It would be a challenge for a student to execute the command sequence for both drives using that feature from the manufacturer's software.

Project

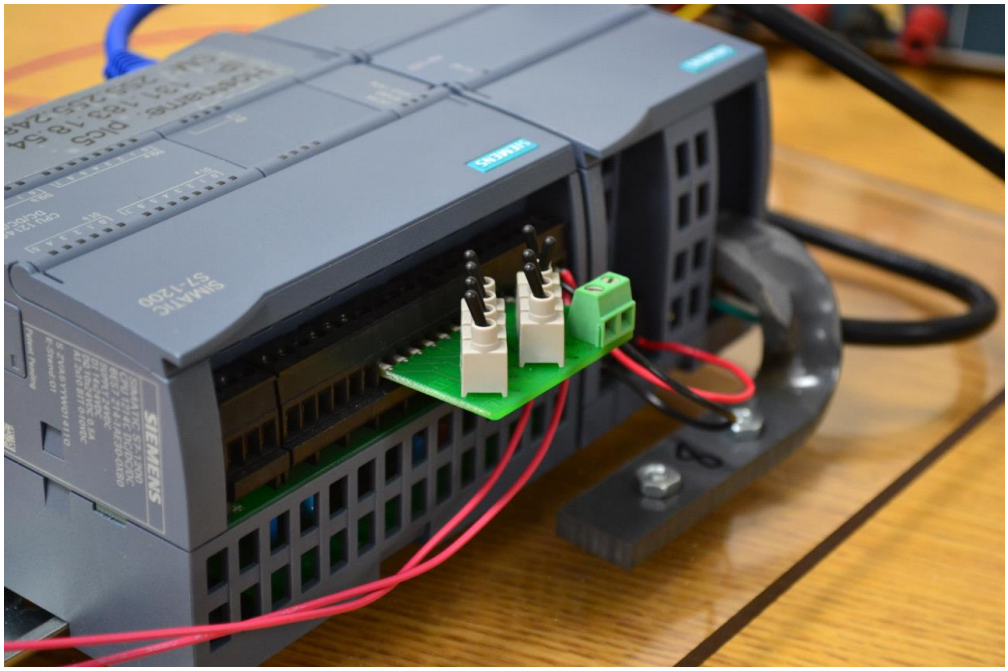
Program the Stepper Application using the Siemens PLC, Stepper Module and Siemens HMI to control the dial for a variable number of complete turns either forward (CW) or reverse (CCW) at a various number of speeds.

To design the HMI panel, use the description below. The description there describes a simple single-axis machine. As an automatic sequence, use at least two different rotations. One could be of 4 turns followed by a dwell followed by a second 4 turns followed by a dwell followed by a return to home.

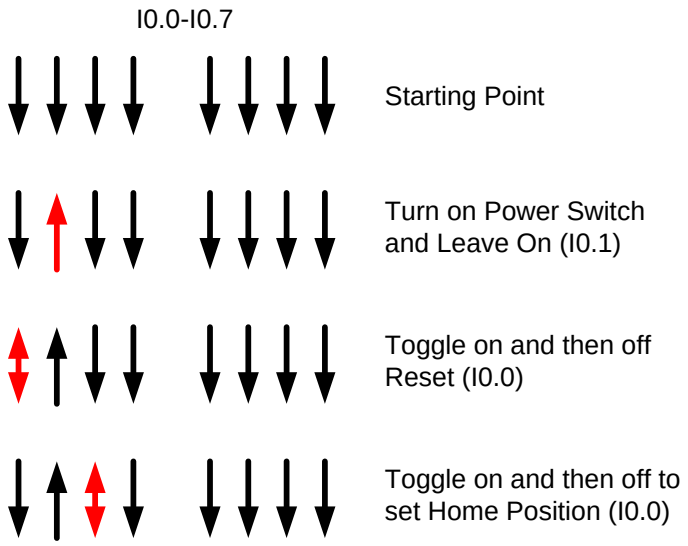
Notice the switches on the PLC. They are attached to inputs which set up the motion application. They may be used for all inputs except those specific to the operation of the machine such as the auto-manual, jog forward, jog reverse, halt and resume commands. Other commands such as 'home', 'reset' and 'power' can remain as switches instead of being incorporated into the HMI.



HMI Screen for Single Axis Stepper



The program given allows the user to toggle the various switches and rotate the motor. Some of the commands will be modified in the actual program. Notice that there is a command table that may be used. It is not usable if the pause and resume are to work properly. Try the toggle switches with the command table and then halt the motion. Then resume the motion. Notice that the motion is reset and starts again. The problem associated with the reset action requires the additional programming. Follow the commands below in this order to begin the Siemens stepper application.



Toggle Switch Settings for Siemens Stepper