Program example

M221 with ILX2T via Modbus TCP

Table of Contents

Introduction

1

Overview

This chapter gives the introduction.

Contents of this chapter

This chapter contains the following topics:

Before You Begin

General

The products specified in this document have been tested under actual service conditions. Of course, your specific application requirements may be different from those assumed for this and any related examples described herein. In that case, you will have to adapt the information provided in this and other related documents to your particular needs. To do so, you will need to consult the specific product documentation of the hardware and/or software components that you may add or substitute for any examples specified in this documentation. Pay particular attention and conform to any safety information, different electrical requirements and normative standards that would apply to your adaptation.

⚠ WARNING

REGULATORY INCOMPATIBILITY

Be sure that all equipment applied and systems designed comply with all applicable local, regional and national regulations and standards

Failure to follow these instructions can result in death, serious injury, or equipment damage.

Please Note

Electrical equipment should be installed, operated, serviced, and maintained only by qualified personnel. No responsibility is assumed by Schneider Electric for any consequences arising out of the use of this material. A qualified person is one who has skills and knowledge related to the construction and operation of electrical equipment and its installation, and has received safety training to recognize and avoid the hazards involved. Failure to observe this information can result in injury or equipment damage.

The use and application of the information contained herein require expertise in the design and programming of automated control systems. Only the user or integrator can be aware of all the conditions and factors present during installation and setup, operation, and maintenance of the machine or process, and can therefore determine the automation and associated equipment and the related safeties and interlocks which can be effectively and properly used. When selecting automation and control equipment, and any other related equipment or software, for a particular application, the user or integrator must also consider any applicable local, regional or national standards and/or regulations.

Some of the major software functions and/or hardware components used in the proposed architectures and examples described in this document cannot be substituted without significantly compromising the performance of your application. Further, any such substitutions or alterations may completely invalidate any proposed architectures, descriptions, examples, instructions, wiring diagrams and/or compatibilities between the various hardware components and software functions specified herein and in related documentation. You must be aware of the consequences of any modifications, additions or substitutions.

A residual risk, as defined by EN/ISO 12100-1, Article 5, will remain if
- it is necessary to modify the recommended logic and if the added or modified components are not properly integrated in the control circuit.
- you do not follow the required standards applicable to the operation of the machine, or if the adjustments to and the maintenance of the machine are not properly made (it is essential to strictly follow the prescribed machine maintenance schedule).
- the devices connected to any safety outputs do not have mechanically-linked contacts.

⚠ CAUTION
EQUIPMENT INCOMPATIBILITY Read and thoroughly understand all device and software documentation before attempting any component substitutions or other changes related to the application examples provided in the document **Failure to follow these instructions can result in injury, or equipment damage.**

Start-Up and Test

Before using electrical control and automation equipment after design and installation, the application and associated functional safety system must be subjected to a start-up test by qualified personnel to verify correct operation of the equipment. It is important that arrangements for such testing be made and that enough time is allowed to perform complete and satisfactory testing.

⚠ CAUTION
EQUIPMENT OPERATION HAZARD •Verify that all installation and set up procedures have been completed. •Before operational tests are performed, remove all blocks or other temporary holding means used for shipment from all component devices •Remove tools, meters, and debris from equipment. **Failure to follow these instructions can result in injury, or equipment damage.**

Verify that the completed system, including the functional safety system, is free from all short circuits and grounds, except those grounds installed according to local regulations. If high-potential voltage testing is necessary, follow the recommendations in equipment documentation to help prevent injury or equipment damage.

Operations and Adjustments

General

Regardless of the care exercised in the design and manufacture of equipment or in the selection and ratings of components, there are hazards that can be encountered if such equipment is improperly installed and operated.
In some applications, such as packaging machinery, additional operator protection such as point-of-operation guarding must be provided. This is necessary if the hands and other parts of the body are free to enter the pinch points or other hazardous areas where serious injury can occur. Software products alone cannot protect an operator from injury. For this reason, the software cannot be substituted for or take the place of point-of-operation protection.

⚠ WARNING
UNGUARDED MACHINERY CAN CAUSE SERIOUS INJURY •Do not use this software and related automation equipment on equipment which does not have point-of -operation protection. •Do not reach into machinery during operation. **Failure to follow these instructions can result in death, serious injury, or equipment damage.**

Ensure that appropriate safeties and mechanical/electrical interlocks related to point-of-operation protection have been installed and are operational before placing the equipment into service. All interlocks and safeties related to point-of-operation protection must be coordinated with the related automation equipment and software programming.

NOTE: Coordination of safeties and mechanical/electrical interlocks for point-ofoperation protection is outside the scope of the examples and implementations suggested herein. It is sometimes possible to adjust the equipment incorrectly and this produce unsatisfactory or unsafe operation. Always use the manufacturer instructions as a guide to functional adjustments. Personnel who have access to these adjustments must be familiar with the equipment manufacturer instructions and the machinery used with the electrical equipment. Only those operational adjustments actually required by the machine operator should be accessible to the operator. Access to other controls should be restricted to help prevent unauthorized changes in operating characteristics.

M221 with ILX2T via ModbusTCP

2

This document describes a small application using a M221-PLC and one ILx2TM drive with EthernetIP modul connected via ModbusTCP, using IO scanning with DriveProfileLexium.
It is not intended to replace any specific product documentation. On the contrary, it offers additional information to the product documentation, for installing, configuring and starting up the system. A detailed functional description or the specification for a specific user application is not part of this document. Nevertheless, the document outlines some typical applications where the system might be implemented.

The program example includes how to activate the operation modes:
-Homing
-Profile Velocity mode
-Profile Torque mode
-Point to point
-Manual mode
-How to write and read parameters

Designed 11.11.2015 by

Please contact your local Schneider Electric partner.
www.schneider-electric.com

System Configuration

M221ME32TK
IP address 192.168.100.1

ILX2TM
IP address 192.168.100.10

Configuration of ILx2T

There are several possibilities to set the IP address on the ILX2T, e.g. via rotary switch, DHCP, from the MAC address or from the EEPROM. Please see the manual for more detailed explanations.
This program example uses the default IP address (192.168.100.10), S2 is switched to 0, S1 is switched to STORED (C, D).

The two rotary switches for setting the IP address have the following positions and functions:

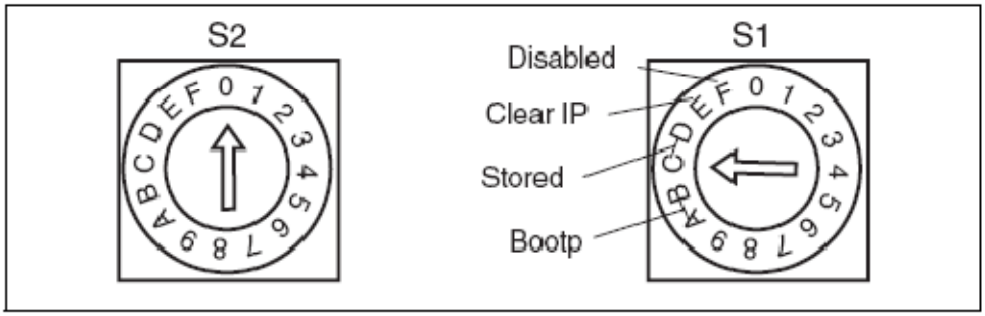


Figure 7.1 Settings of the rotary switches

ID	Name of switch function	Function	Valid positions (S2)	Valid positions (S1)
1	Device Name	DHCP server required; the IP address is assigned via the DeviceName.	0 - F	0 - 9
2	BootP	DHCP or Bootstrap Protocol server required; the IP address is assigned via the MAC address.	Any	A, B
3	Stored	Uses the IP parameters stored in the EEPROM	Any	C, D
4	Clear IP	Deletes the IP parameters stored in the EEPROM.	Any	E
5	Disabled	Disables the rotary switch settings.	Any	F

Following parameter settings are mandatory in ILX2T:

Name	Wert	Einheit	Beschreibung	Bereich	Modbus
EthIpAddr	192.168.100.10		Gespeicherte IP-Adresse	..	6658
EthIpAddrAct	192.168.100.10		Aktuelle verwendete IP-Adresse	..	6664
EthSubMask	255.255.255.0		Gespeichterte Subnet Mask	..	6660
EthSubMaskAct	255.255.255.0		Aktuell verwendete Subnet Mask	..	6666
EthGateway	192.168.100.1		Gespeicherter Ethernet Netzwerk-Gateway	..	6662
EthGatewayAct	192.168.100.1		Aktuell verwendeter Ethernet Netzwerk-Gateway	..	6668
_ethMacAdr1	0		Ethernet MAC-Adresse Teil 1	..	6672
_ethMacAdr2	0		Ethernet MAC-Adresse Teil 2	..	6674
MBTCPdword_order	HighLow		ModbusTCP Wortfolge für Doppelworte (32 Bit Werte)	0..1	6410
EthErrBehv	Warning		Fehlerverhalten für Ethernet-Echtzeitdaten	0..1	6412
EthIPConfInfo	Stored		IP-Konfigurationsinformation	0..65535	6694
EthFdrEnable	disabled		Fast Device Replacement (FDR) Enable	0..1	6696
EthFdrAutosave	disabled		Fast Device Replacement (FDR) Autosave	0..5	6698

For this example do not modify the double word order MBTCPdword_order.
The Gateway settings of Drive and PLC need to be identical.

Modbus TCP interface of ILX2T

There are two possibilities to control the drive and to activate the different modes of operation.

DCOM interface
The first one uses the DCOM interface which is similar to DSP402 on CANopen and the Modbus interface on Lexium05. The control of the drive (Enable, Fault reset, …) is possible via *DCOMcontrol* (Modbus 6914), the status is in *DCOMstatus* (Modbus 6916). To activate a mode of operation a sequence must be programmed in the plc. Here is an example that shows how to start a point to point movement:

-Write Set velocity *PPn_target* (Modbus 6942)
-Activate operation mode, set *DCOMopmode* (Modbus 6918) to 1
-Check actual operation mode *DCOMopmd_act* (Modbus 6920)
-Write Set position *PPp_targetusr* (Modbus 6940)
-Set start bit in *DCOMcontrol* (Modbus 6914)
-Check *DCOMstatus* (Modbus 6916) for the end of the movement (X_END bit)
-Reset start bit on *DCOMcontrol* (Modbus 6914)

As you can see this way of programming needs several plc cycles to activate any mode of operation.

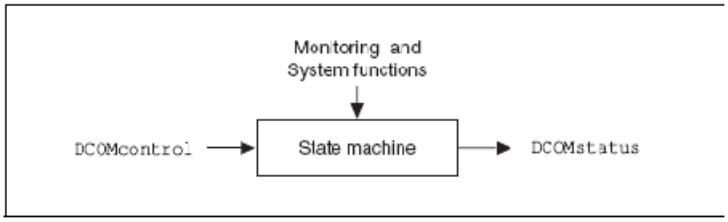


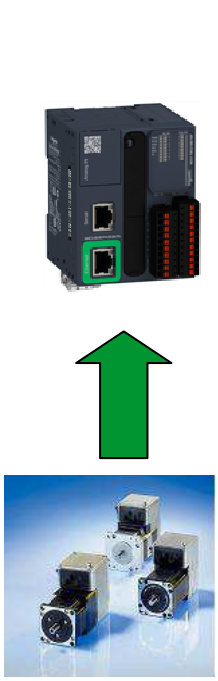
Figure 8.4 Changing and monitoring the operating state via parameters

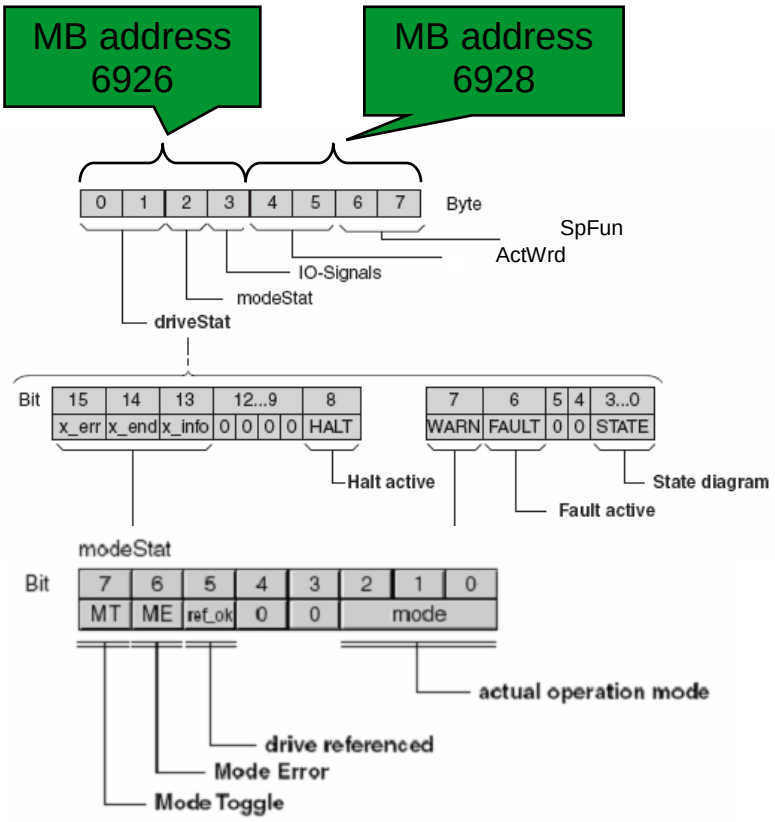
Parameter Name HMI menu	Description	Unit Minimum value Default value Maximum value	Data type R/W persistent Expert	Parameter address via fieldbus
DCOMcontrol	Drivecom control word Refer to chapter Operation, Operating States, for bit coding information. Bit 0: Switch on Bit 1: Enable Voltage Bit 2: Quick Stop Bit 3: Enable Operation Bit 4..6: Operating mode specific Bit 7: Fault Reset Bit 8: Halt Bit 9..15: Reserved (must be 0)	- - 0 -	UINT16 R/W - -	Modbus 6914

Parameter Name HMI menu	Description	Unit Minimum value Default value Maximum value	Data type R/W persistent Expert	Parameter address via fieldbus
DCOMstatus	Drivecom status word Refer to chapter Operation, State Machine for bit coding information. Bit 0-3,5,6: Status bits Bit 4: Voltage enabled Bit 7: Warning Bit 8: HALT request active Bit 9: Remote Bit 10: Target reached Bit 11: Reserved Bit 12: Operating mode specific Bit 13: x_err Bit 14: x_end Bit 15: ref_ok	- - 0 -	UINT16 R/- - -	Modbus 6916

IOscanning interface

Especially for the IOscanning there is another possibility to control the drive. This interface is easier and it is possible to start a mode of operation within one plc cycle. It uses the Modbus registers 6922 to 6928.



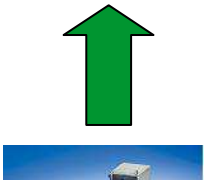


DriveStat gives information about the drive status, *ModeStat* gives information about actual operation mode. Byte 3 contains the status of the IOs. You will find the meaning of the action word (*ActWrd*) and the special functions (*SpFun*) below. For more information please see the manual.

Action Word

	15	14	13	12	11	10	9	8
Byte 4	0	CNST	ACC	DEC	TAR0	0	PWIN	MOTN
Byte 5	MOTP	MOTZ	0	0	0	0	0	0
	7	6	5	4	3	2	1	0

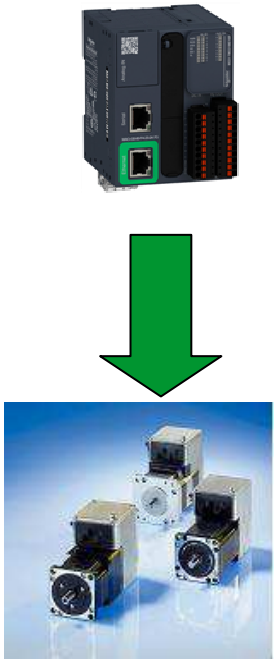
MOTZ:	Motion zero: actual speed is zero
MOTP:	Motion positive : motor turns in positive direction
MOTN:	Motion negative : motor turns in negative direction
PWIN:	inside position window
TAR0:	Profile generator: target speed is zero
DEC:	Profile generator: deceleration
ACC:	Profile generator: acceleration
CNST:	Profile generator: constant movement

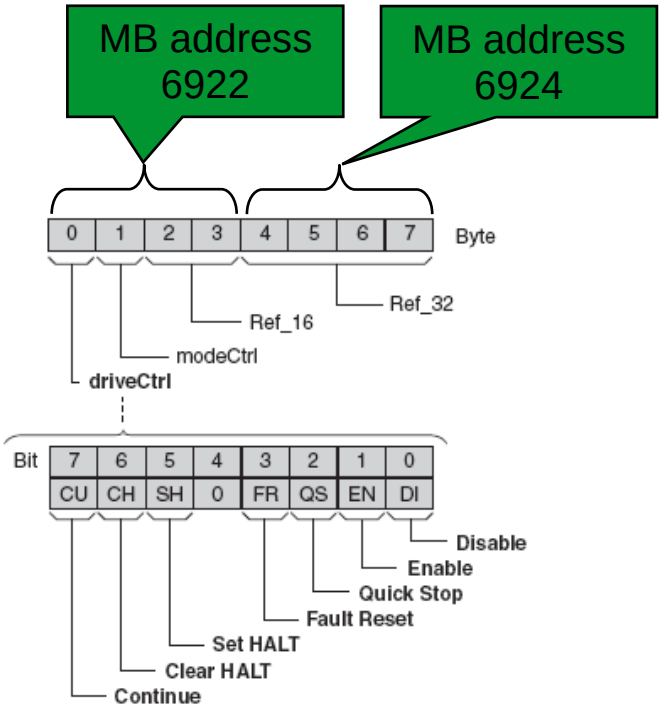


Special Function

	15	14	13	12	11	10	9	8
Byte 6	0	0	0	0	0	0	0	0
Byte 7	0	0	0	0	CAP2 1	CAP2 0	CAP1 1	CAP1 0
	7	6	5	4	3	2	1	0

- Indicates the status (bit 0 and bit1) of the Capture inputs CAP1 and CAP2
- Only at ILA and ILS

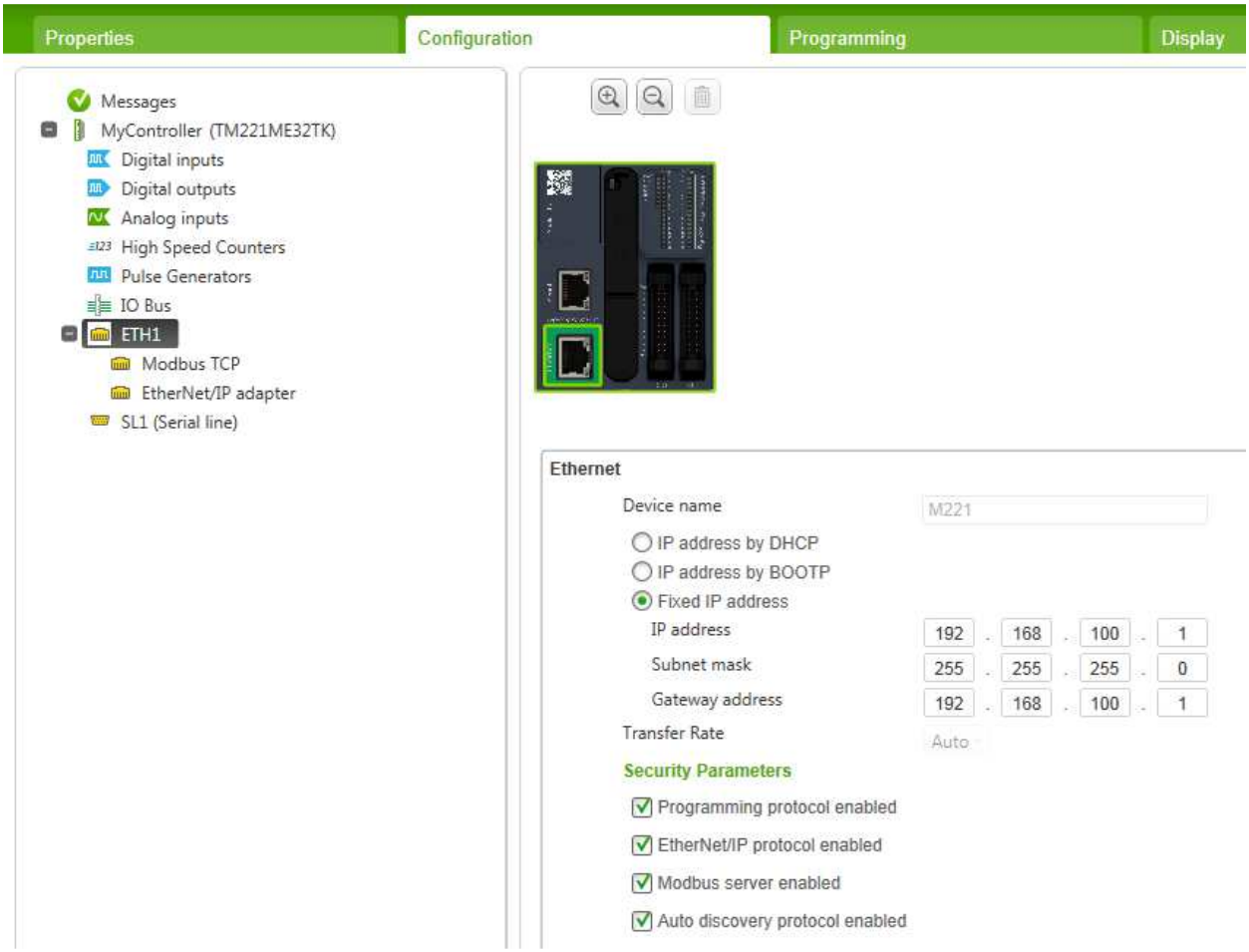


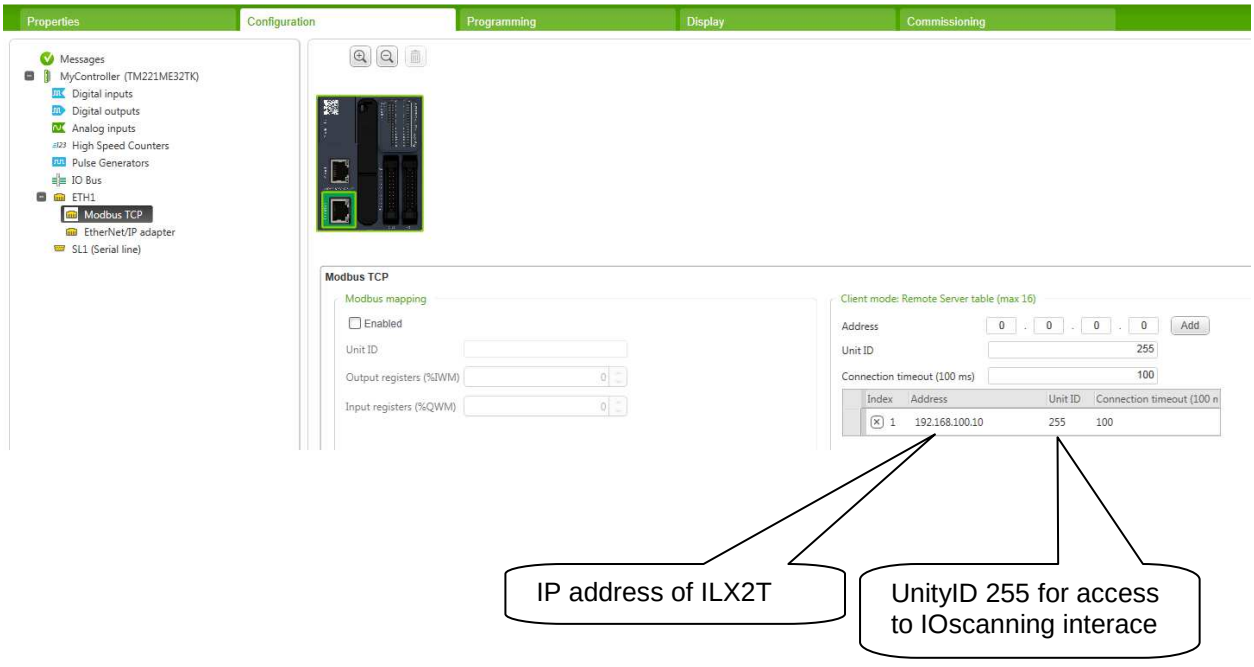


You can see the meaning of the drive control above. *modeCtrl* is necessary to start the different mode of operations. The meaning of Ref16 and Ref32 depends on the chosen mode of operation:

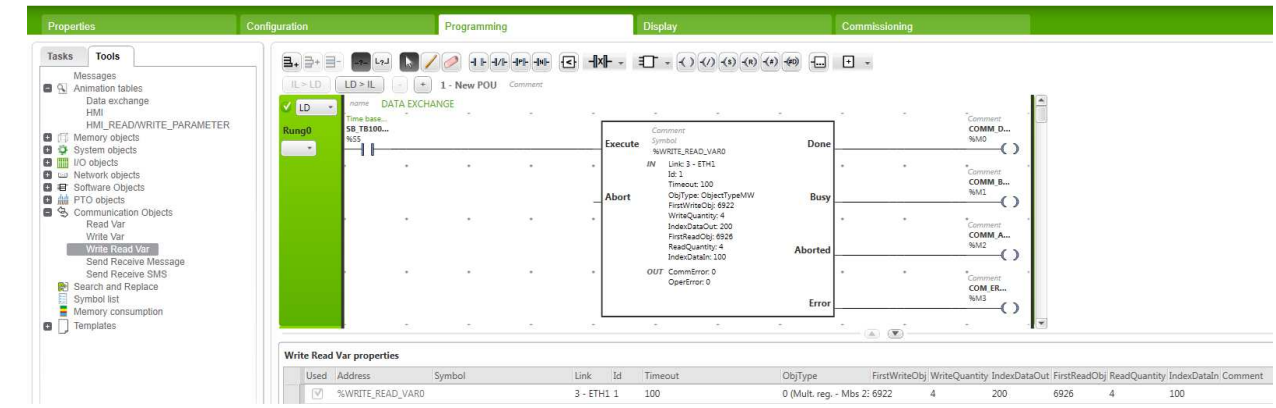
Mode	Action	modeCtrl* Bit 0..6	Description	Reference value Ref_16	Reference value Ref_32
1	0	01h	Jog	Corresponds to parameter JOGactivate	-
2	0	02h	Homing: Position setting	-	Position for position setting Corresponds to parameter HMp_homeusr
	1	12h	Homing: Reference movement	Homing method Corresponds to parameter HMmethod	-
3	0	03h	Profile Position: Absolute positioning	Target speed of rotation Corresponds to parameter PPn_target	Target position Corresponds to parameter PPp_absusr
	1	13h	Profile Position: Relative positioning with reference to the currently set target position	Target speed of rotation Corresponds to parameter PPn_target	Target position Corresponds to parameter PPp_relprefusr
	2	23h	Profile Position: Relative positioning with reference to the current motor position	Target speed of rotation Corresponds to parameter PPn_target	Target position Corresponds to parameter PPp_relpactusr
4	0	04h	Profile Velocity	Target speed of rotation Corresponds to parameter PPn_target	-
7	1	17h	Speed Control	Reference speed Corresponds to parameter SPEEDn_target	-

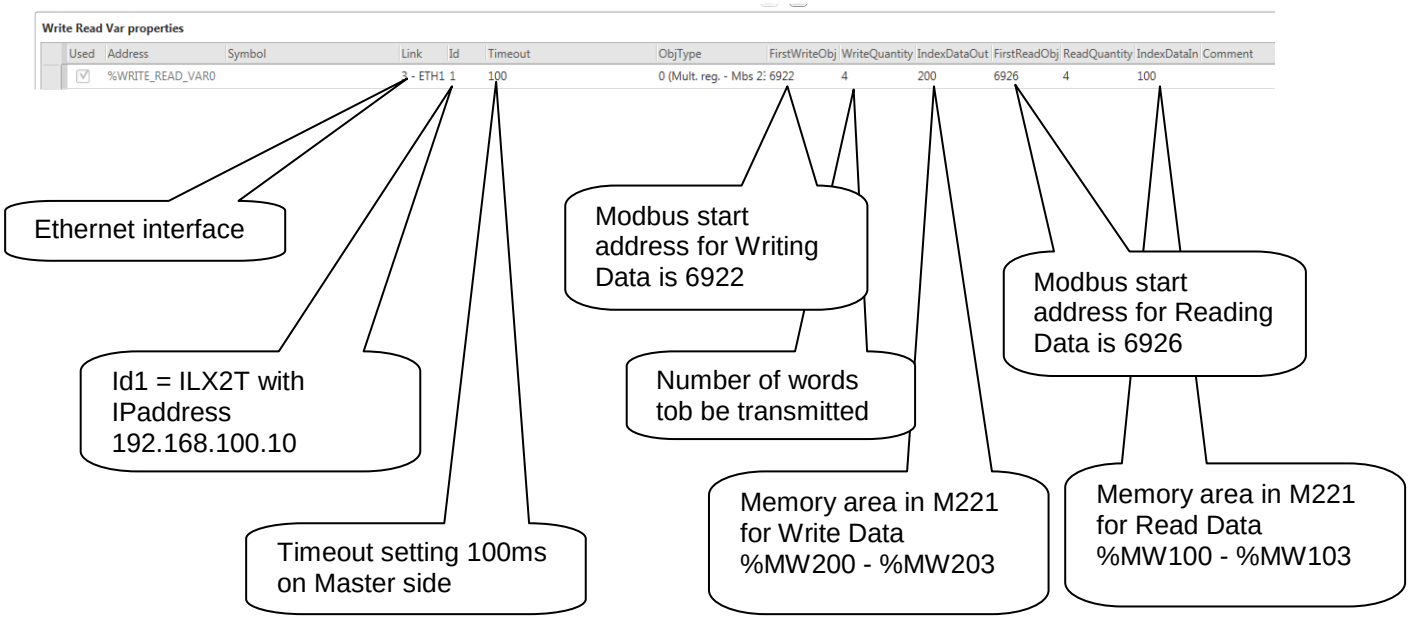
Ethernet configuration in M221

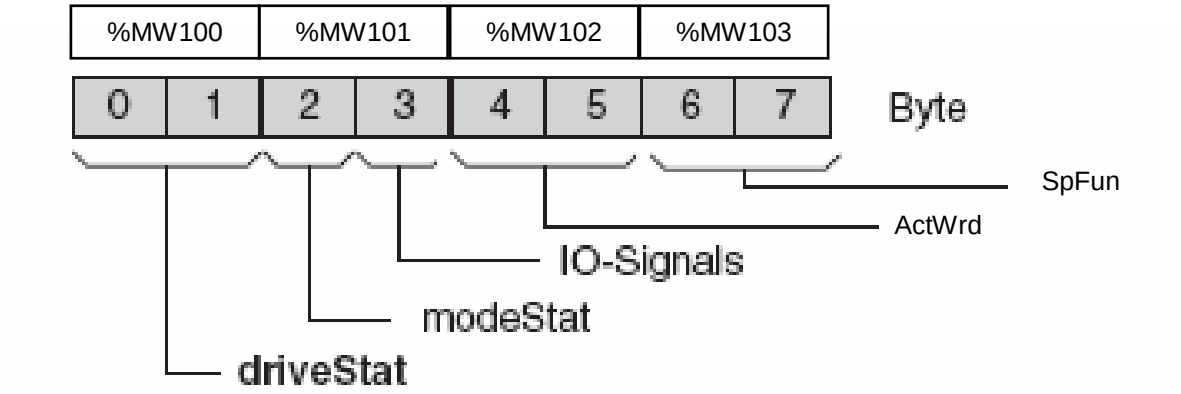


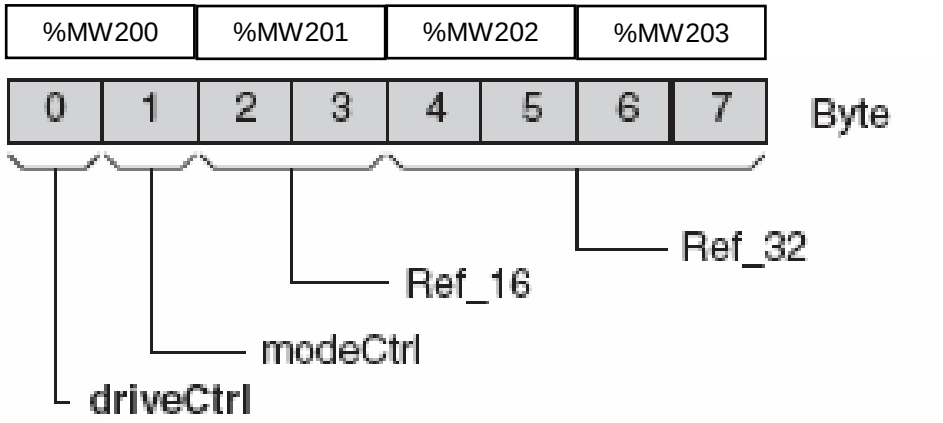


The data exchange is realized with the %WRITE_READ function block of SoMachineBasic:






%MW100
%MW101
%MW102
%MW103
0
1
2
3
4
5
6
7
Byte
SpFun
ActWrd
IO-Signals
modeStat
driveStat

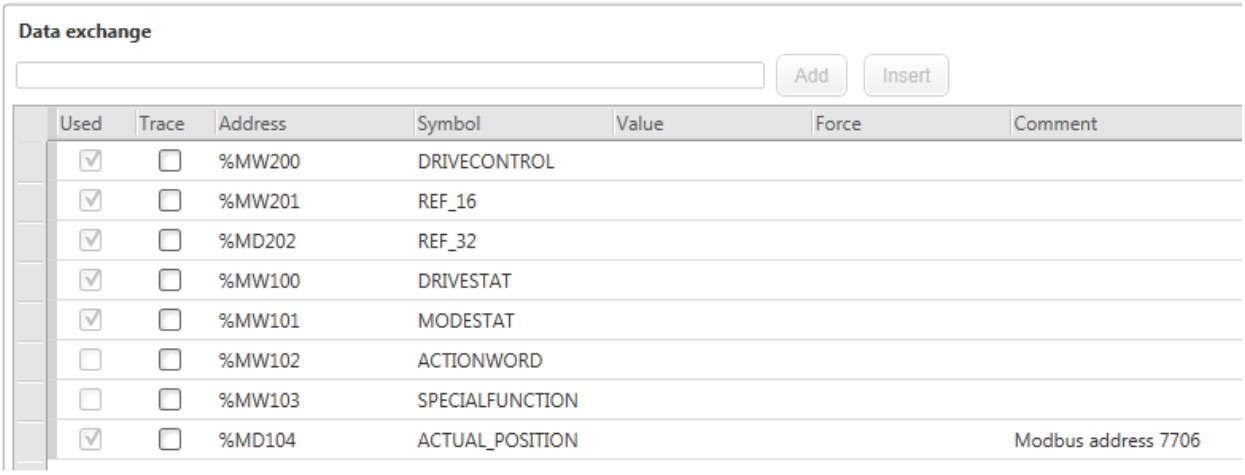

%MW200
%MW201
%MW202
%MW203
0
1
2
3
4
5
6
7
Byte
Ref_32
Ref_16
modeCtrl
driveCtrl

Explanation of Sourcecode

Animation table

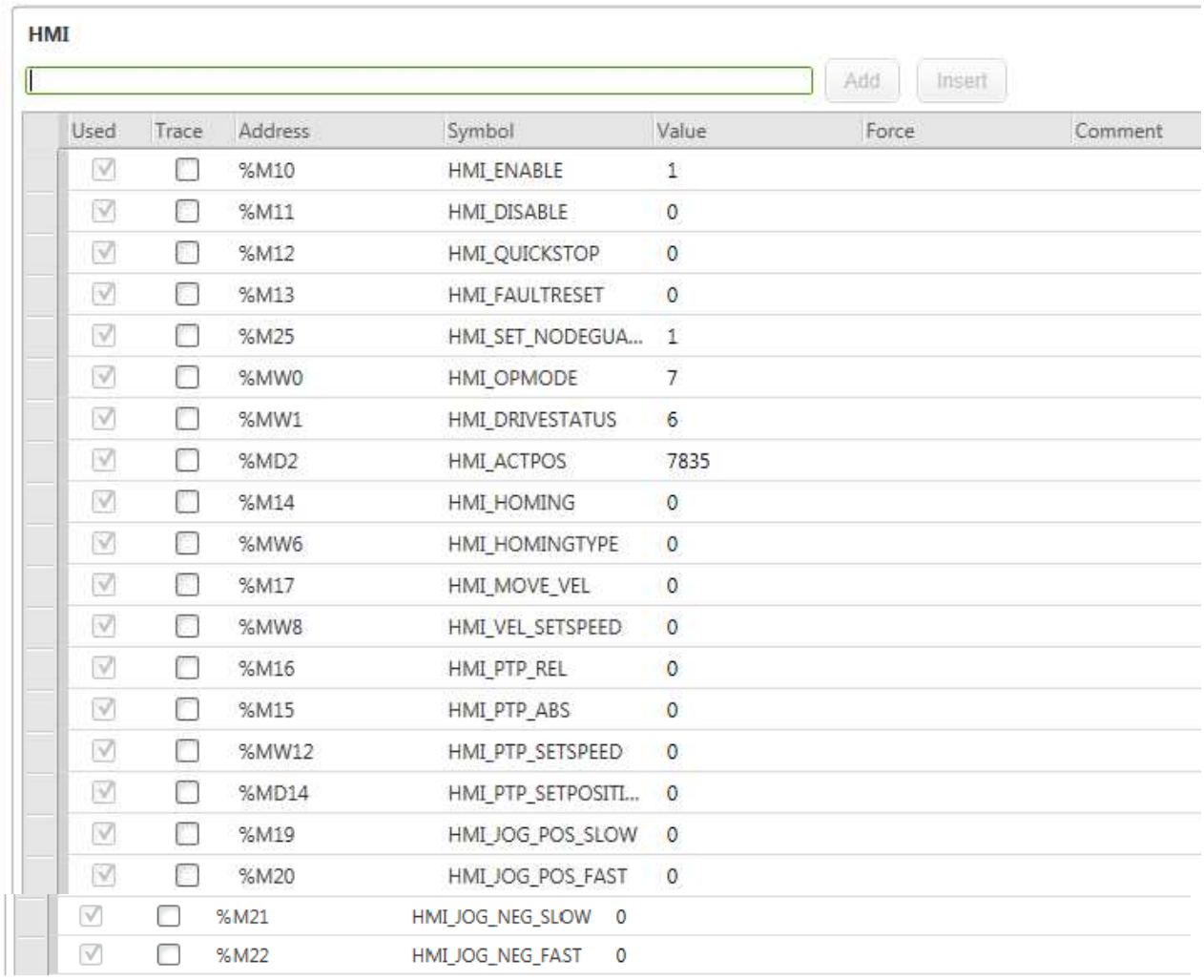
The program can be controlled and monitored via Animation tables.

“Data exchange” shows the data exchange between plc and drive



Used	Trace	Address	Symbol	Value	Force	Comment
☑	☐	%MW200	DRIVECONTROL			
☑	☐	%MW201	REF_16			
☑	☐	%MD202	REF_32			
☑	☐	%MW100	DRIVESTAT			
☑	☐	%MW101	MODESTAT			
☐	☐	%MW102	ACTIONWORD			
☐	☐	%MW103	SPECIALFUNCTION			
☑	☐	%MD104	ACTUAL_POSITION			Modbus address 7706

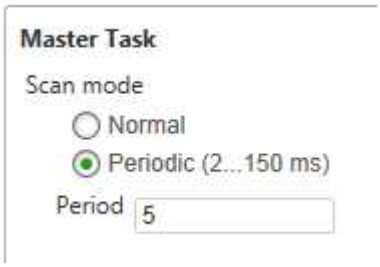
With “HMI” it is possible to control ILX2T.
E.g. by setting HMI_ENABLE the drive enables; HMI_FAULTRESET performs a fault reset.
HMI_OPMODE shows the current mode of operation; HMI_DRIVESTATUS shows the current drive status.
With HMI_HOMINGTYPE it is possible to choose the way homing is performed, by setting HMI_HOMING the movement will be started.
For the other modes of operation it is similar.



Used	Trace	Address	Symbol	Value	Force	Comment
☑	☐	%M10	HMI_ENABLE	1		
☑	☐	%M11	HMI_DISABLE	0		
☑	☐	%M12	HMI_QUICKSTOP	0		
☑	☐	%M13	HMI_FAULTRESET	0		
☑	☐	%M25	HMI_SET_NODEGUA...	1		
☑	☐	%MW0	HMI_OPMODE	7		
☑	☐	%MW1	HMI_DRIVESTATUS	6		
☑	☐	%MD2	HMI_ACTPOS	7835		
☑	☐	%M14	HMI_HOMING	0		
☑	☐	%MW6	HMI_HOMINGTYPE	0		
☑	☐	%M17	HMI_MOVE_VEL	0		
☑	☐	%MW8	HMI_VEL_SETSPEED	0		
☑	☐	%M16	HMI_PTP_REL	0		
☑	☐	%M15	HMI_PTP_ABS	0		
☑	☐	%MW12	HMI_PTP_SETSPEED	0		
☑	☐	%MD14	HMI_PTP_SETPOSITI...	0		
☑	☐	%M19	HMI_JOG_POS_SLOW	0		
☑	☐	%M20	HMI_JOG_POS_FAST	0		
☑	☐	%M21	HMI_JOG_NEG_SLOW	0		
☑	☐	%M22	HMI_JOG_NEG_FAST	0		

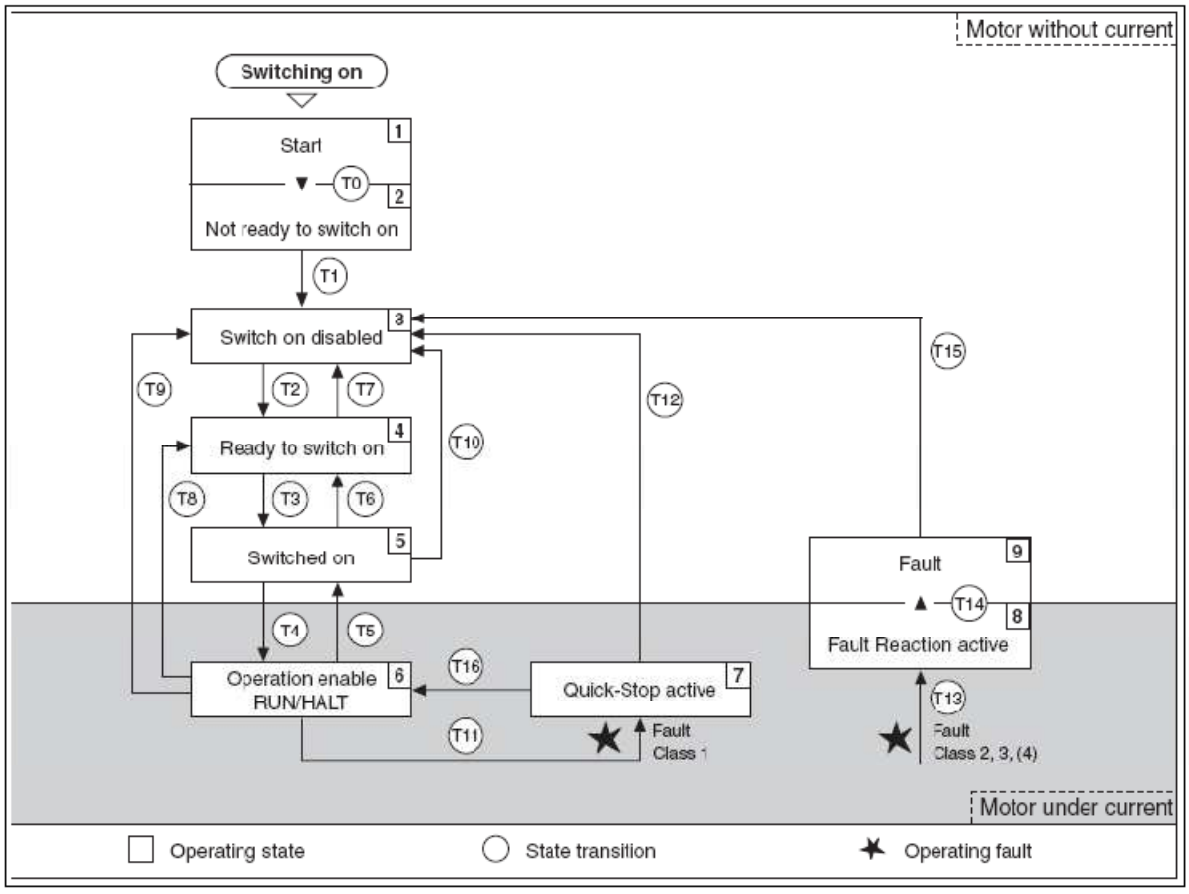
Task configuration

The rungs are called in a cyclic task:

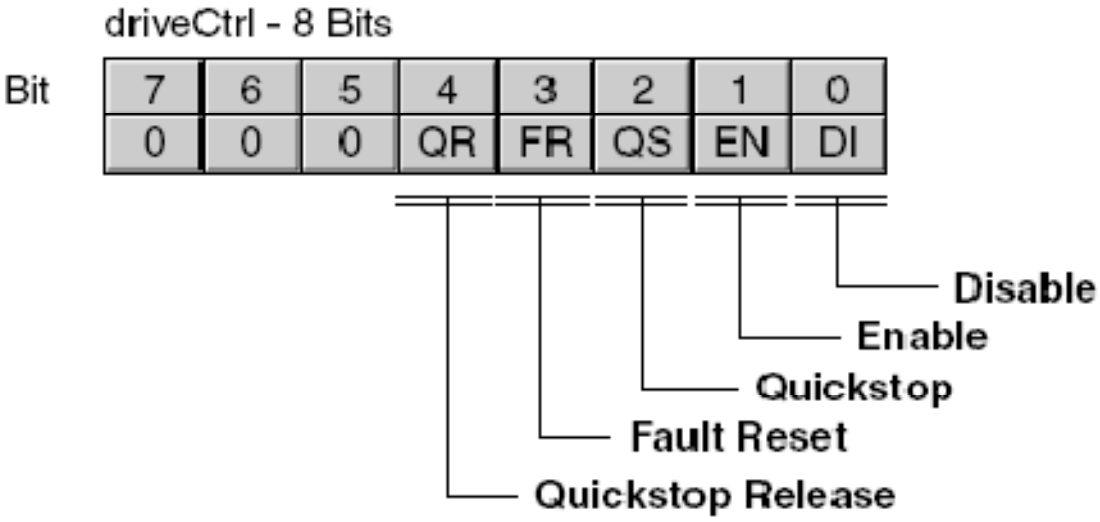


Status machine

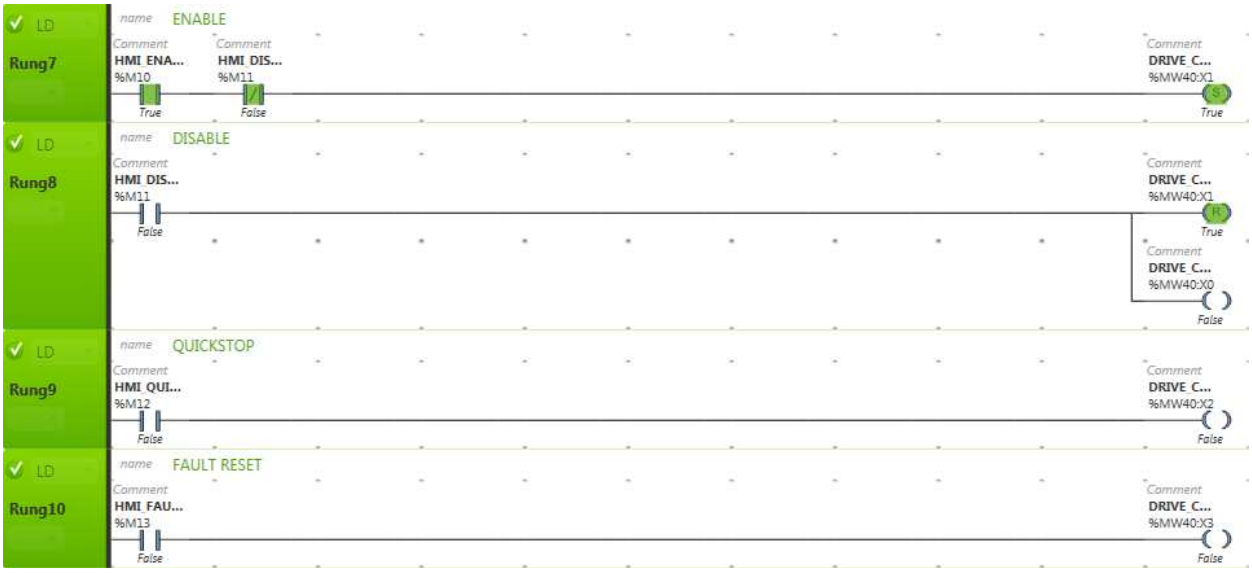
Before explaining the source code it is necessary to take a look to the state machine of the drive.
After switching on the drive a sequence of operating states is progressed through. If the drive is ok it goes automatically to state 4. When it is enabled and there is no error it changes to state 6.



To enable the drive the bit 1 of the drive control byte must be true. We have access to it directly via the low byte of %MW200.

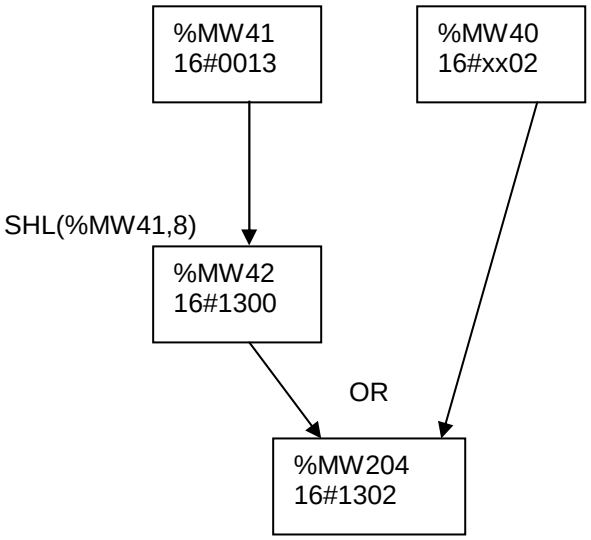


The source code corresponding to that is:



The bits %M10 (HMI_ENABLE), %M11 (HMI_DISABLE), %M12 (HMI_QUICKSTOP) and %M14 (HMI_FAULTRESET) manipulate the bits in %MW40 instead of %MW200. The reason is that drive control is only the high byte of %MW204, the low byte contains the mode control. Therefore the drive control is stored in %MW40 and the mode control in %MW41. In rung 17 both bytes are merged in %MW204:

An example:



The drive status %MW100 and the mode status %MW101 contain the information of the actual state of the drive.

driveStat

Bit	15	14	13	12 --- 8					7	6	5	4	3 --- 0
	x err	x_end	x_info	0	0	0	0	0	warn	Sig_SR	FltSig	0	cos

modeStat

Bit	7	6	5	4	3	2	1	0
	MT	ME	ref_ok	0	0	mode		

actual operation mode

drive referenced

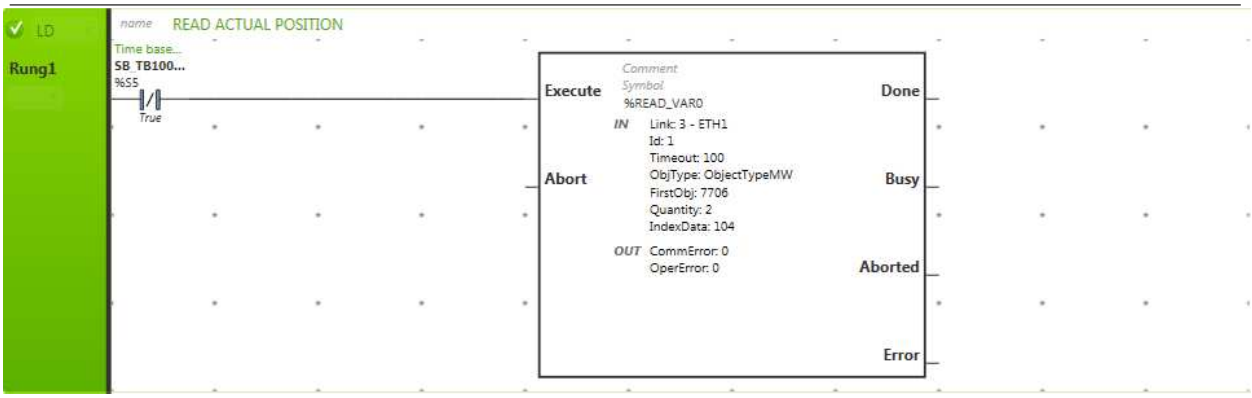
Mode Error

Mode Toggle

The field *cos* in the drive status contains the number of the actual status, the field *mode* in the mode control contains the number of the actual operation mode. For the HMI only the data in those fields are needed.

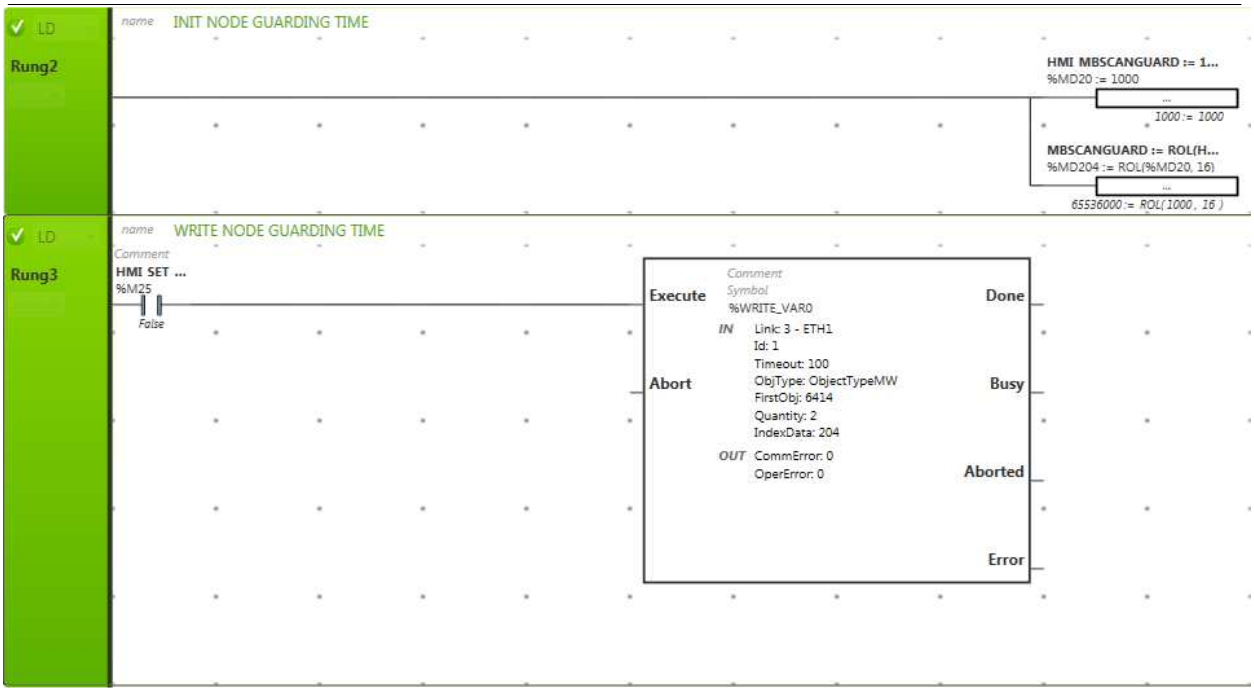
To get the current position a separate read request on Modbus Address 7706 needs to be done.

_p_actusr	Actual motor position in user units	usr - 0 -	INT32 R/- - -	Modbus 7706



The Nodeguarding monitoring on Drive side needs to be activated from the application. In our example it is set to 1000 msec.

MBscan_guard	ModbusTCP node guard for I/O scanning (91) 0: Inactive (default) >0: Monitoring time in ms The drive starts monitoring when it receives the first I/O scanning message. After a node guarding error the drive starts monitoring again when it receives the next I/O scanning message.	ms 0 0 10000	UINT16 R/W - -	Modbus 6414

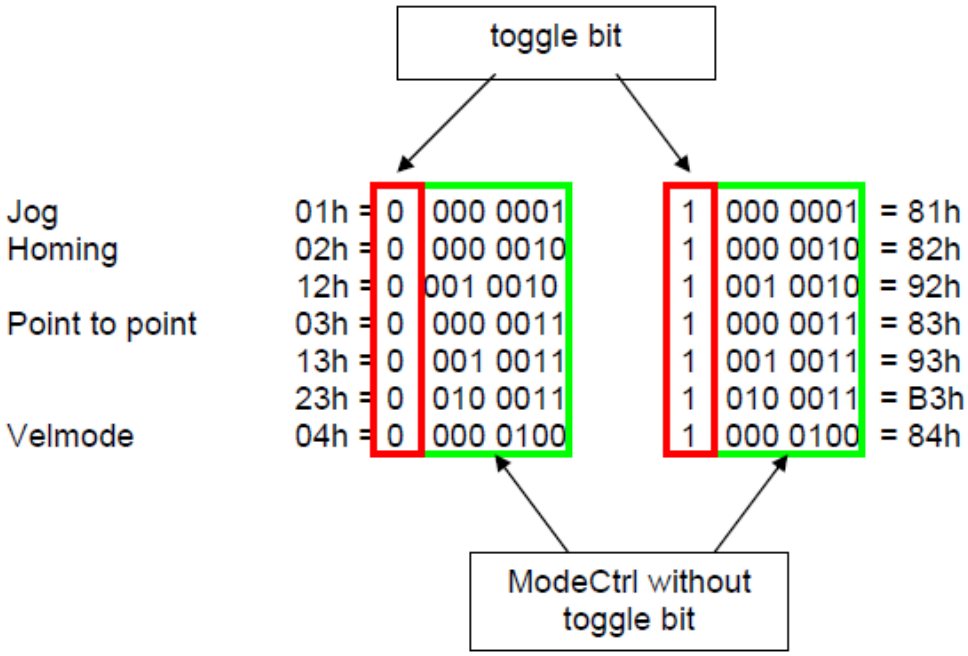


Operation modes

The ILX2T provides the following operation modes:

Mode	Action	modeCtrl* Bit 0..6	Description	Reference value Ref_16	Reference value Ref_32
1	0	01h	Jog	Corresponds to parameter JOGactivate	-
2	0	02h	Homing: Position setting	-	Position for position setting Corresponds to parameter HMp_homeusr
	1	12h	Homing: Reference movement	Homing method Corresponds to parameter HMmethod	-
3	0	03h	Profile Position: Absolute positioning	Target speed of rotation Corresponds to parameter PPn_target	Target position Corresponds to parameter PPp_absusr
	1	13h	Profile Position: Relative positioning with reference to the currently set target position	Target speed of rotation Corresponds to parameter PPn_target	Target position Corresponds to parameter PPp_relprefusr
	2	23h	Profile Position: Relative positioning with reference to the current motor position	Target speed of rotation Corresponds to parameter PPn_target	Target position Corresponds to parameter PPp_relpactusr
4	0	04h	Profile Velocity	Target speed of rotation Corresponds to parameter PPn_target	-
7	1	17h	Speed Control	Reference speed Corresponds to parameter SPEEDn_target	-

The left column shows the operation mode, the next column the entry in the mode control without the first bit, the toggle bit. This bit appears twice. Once in the *mode control* and once in the *mode status,* where it is mirrored. To start an operation mode its state has to change. When the toggle bit is false in the mode status it must be set to true in the mode control. After receiving the command it changes to true in the mode status. To activate the next operation mode it must be set to false. The hexadecimal values are:



Depending on the operation mode the meaning of the fields Ref16 and Ref32 is different. For example in velocity mode Ref16 contains the set speed, in homing it is the homing type.

Homing

The product manual contains detailed information about the operation mode. There are differences between an IFA, an IFE or an IFS. This program example uses an IFE.

8.5.4 Operating mode Homing

Overview of Homing

The operating mode Homing establishes an absolute position reference between the motor position and a defined axis position. Homing can be carried out by a means of a reference movement or by position setting.

- A reference movement is a movement to a defined point, the reference point, on the axis; the objective is to establish the absolute position reference between the motor position and the axis position. The reference point also defines the zero point that is used for all subsequent absolute positioning movements as a reference point. It is possible to parameterize a shift of the zero point.

  A reference movement must be completed for the new zero point to be valid. If the reference movement is interrupted, it must be started again. As opposed to the other operating modes, a reference movement must be completed before a new operating mode can be activated.

  The signals required for the reference movement must be wired. Monitoring signals that are not used must be deactivated.

- Position setting lets you set the current motor position to a desired position value to which the subsequent position values will relate.

Homing is not required for a motor with a multiturn encoder because it provides a valid absolute position immediately after being switched on.

Types of reference movements

There are 4 standard types of reference movements:

- Movement to negative limit switch `LIMN`
- Movement to positive limit switch `LIMP`
- Movement to reference switch `REF` with counterclockwise direction of rotation
- Movement to reference switch `REF` with clockwise direction of rotation

Reference movements are possible with or without index pulse.

- Reference movement without index pulse
  Movement from the switching edge to a distance distance from switching edge
- Reference movement with index pulse
  Movement from the switching edge to the closest index pulse of the motor. The current motor position can be read via the parameter `_p_absENCusr`. The index pulse is at position value 0.

Trigger homing

Homing is triggered via bit 4=1 in parameter `DCOMcontrol`.

Status messages The drive provides information concerning positioning via Bits 10 and 12 to 15 in the parameter `DCOMstatus`.

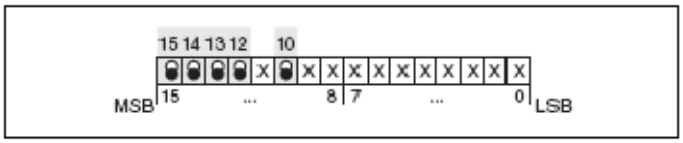


Figure 8.14 Status reports for operating mode

Parameter value	Description
Bit 10: Target reached	0: Homing not finished 1: Homing finished (even in the event of termination via "Halt")
Bit 12: Homing attained	1: Homing successfully completed
Bit 13: x_err	1: Error arisen
Bit 14: x_end	1: Homing completed, motor at a standstill
Bit 15: ref_ok	1: drive has valid reference point

8.5.4.1 Setting by parameters, general

Description There are various methods of homing which can be selected via the parameters `HMmethod`.

Parameter Name	Description	Unit Minimum value Default value Maximum value	Data type R/W persistent Expert	Parameter address via fieldbus
HMmethod	Homing method 1: LIMN with index pulse 2: LIMP with index pulse 7: REF+ with index pulse, Inv., outside 8: REF+ with index pulse, Inv., inside 9: REF+ with index pulse, not Inv., inside 10: REF+ with index pulse, not Inv., outside 11: REF- with index pulse, Inv., outside 12: REF- with index pulse, Inv., inside 13: REF- with index pulse, not Inv., inside 14: REF- with index pulse, not Inv., outside 17: LIMN 18: LIMP 23: REF+, Inv., outside 24: REF+, Inv., inside 25: REF+, not Inv., inside 26: REF+, not Inv., outside 27: REF-, Inv., outside 28: REF-, Inv., inside 29: REF-, not Inv., inside 30: REF-, not Inv., outside 33: Index pulse neg. direction 34: Index pulse pos. direction 35: Position setting Abbreviations: REF+: Search movement in pos. direction REF-: Search movement in pos. direction Inv.: Invert direction in switch not Inv.: Direction not inverted in switch outside: Index pulse / distance outside switch inside: Index pulse / distance inside switch	- 1 18 35	INT16 R/W - -	Modbus 6936

The source code is very short. A rising edge on %M14 (*HMI_HOMING)* writes the homing type (HMI_HOMINGTYPE) into REF16 (%MW201). Depending on the toggle bit in *MODESTAT* (MW101:X15) the toggle bit in DRIVECONTROL is set or reset.

Profile velocity

The product manual contains detailed information.

8.5.3 Operating mode Profile velocity

In the operating mode Profile Velocity, the drive accelerates to an adjustable target speed of rotation. You can set a motion profile with values for acceleration and deceleration ramps.

Start operating mode

If the type of operation, the operating state and the parameter values are set, the operating mode can be started by transfer of a set velocity in the parameter `PVn_target`.

Status messages

The drive provides information concerning positioning via Bits 10 and 12 to 15 in the parameter `DCOMstatus`.

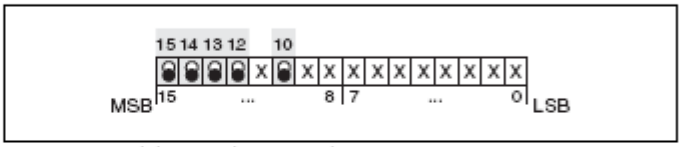


Figure 8.12 Status reports for operating mode

Parameter value	Description
Bit 10: Target reached	0: Reference speed not reached 1: Reference speed reached (even in the event of motor standstill via "Halt")
Bit 12: speed=0	0: motor moves 1: motor stopped
Bit 13: x_err	1: Error arisen
Bit 14: x_end	1: Operating mode finished
Bit 15: ref_ok	1: drive has valid reference point

Operating mode finished

The operating mode is completed and motor standstill achieved by "Halt", by an error or after a preset default = 0.

8.5.3.1 Parameterization

Overview The following overview shows the function principle of the parameters which can be set for the Profile Velocity operating mode.

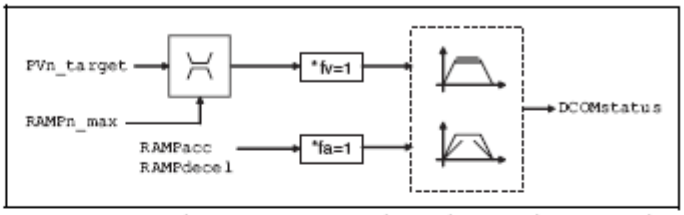


Figure 8.13 Operating mode Profile Velocity, effects of adjustable parameters

Set speed The set speed is transferred parameter `PVn_target` in rpm and can be changed during the movement. The operating mode is not limited by range limits of the positioning. New speed values are accepted immediately during a travel command.

Parameter Name HMI menu	Description	Unit Minimum value Default value Maximum value	Data type R/W persistent Expert	Parameter address via fieldbus
PVn_target	Reference speed in operating mode profile velocity The adjusted value is internally limited to the current parameter value in RAMPn_max.	min^{-1} - 0 -	INT32 R/W - -	Modbus 6938

Current speed The current speed is determined by using the 2 parameters `_n_act` and `_n_actRAMP`.

Parameter Name HMI menu	Description	Unit Minimum value Default value Maximum value	Data type R/W persistent Expert	Parameter address via fieldbus
_n_act	Actual motor speed	min^{-1} - 0 -	INT16 R/- - -	Modbus 7696
_n_actRAMP	Actual speed of motion profile generator	min^{-1} - 0 -	INT32 R/- - -	Modbus 7948

The velocity mode works in the same way. Now Ref16 contains the set speed.

Profile position (point to point)

The product manual contains detailed information.

8.5.2 Operating mode Profile position

In Profile Position operating mode, a movement with an adjustable motion profile is performed from a start position to a target position. The value of the target position can be specified as either a relative or an absolute position.

You can set a motion profile with values for acceleration ramp, deceleration ramp and target speed.

Relative and absolute positioning

In the case of absolute positioning, the positioning distance is specified absolutely with reference to the zero point of the axis. A zero point must be defined with the Homing operating mode before absolute positioning can be used for the first time.

In the case of a relative positioning, the positioning distance is specified relatively with reference to the current axis position or the target position.

Absolute positioning or relative positioning is set with bit 6 via the parameter `DCOMcontrol`.

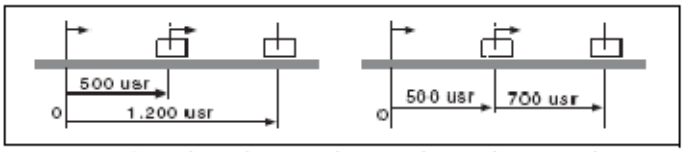


Figure 8.9 Absolute positioning (left) and relative positioning (right)

Triggering positioning

Parameter value	Meaning
Bit 4: New setpoint	0->1: start positioning or prepare following positioning
Bit 5: Change set immediately (Only applicable with New setpoint 0->1)	0: Enable new positioning values when target position is reached 1: Enable new position values immediately
Bit 6: Absolute / relative	0: Absolute positioning 1: Relative positioning

Start positioning with a rising edge of Bit 4 in the parameter `DCOMcontrol`.

The positioning can be triggered in 2 ways depending upon Bit 5.

- Bit 5=0:

  Position values (`PPp_targetusr`, `PPn_target`, `RAMPacc` and `RAMPdecel`), that are transferred during a positioning, are saved temporarily. The target position of the current positioning is approached. The new position values are executed only when the target position is reached.

  If new position values are transferred again, the temporarily saved position values are overwritten again.

- Bit 5=1:

  Position values (`PPp_targetusr`, `PPn_target`, `RAMPacc` and `RAMPdecel`), that are transferred during a positioning, are executed immediately. The target position of the new positioning is directly approached.

Status messages

The drive provides information concerning positioning via Bits 10 and 12 to 15 in the parameter `DCOMstatus`.

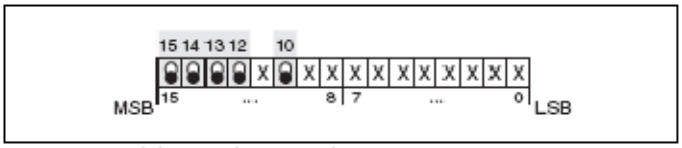


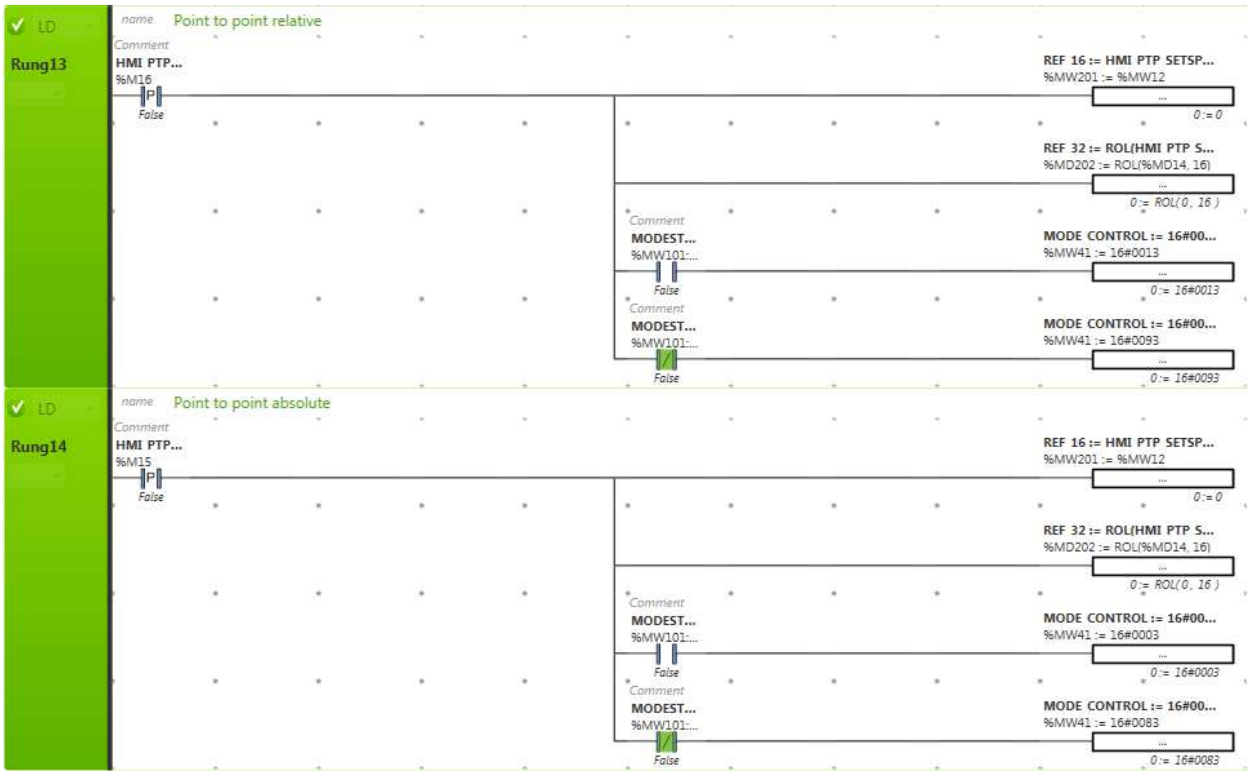
Figure 8.10 Status reports for operating mode

Parameter value	Description
Bit 10: Target reached	0: Target position not reached (even with "Halt" or error) 1: Target position reached
Bit 12: setpoint acknowledge	0: Transfer of new position possible 1: New target positioning accepted
Bit 13: x_err	1: Error arisen
Bit 14: x_end	1: Positioning completed, motor at a standstill
Bit 15: ref_ok	1: drive has valid reference point

Positioning finished

Bit 14 indicates whether positioning is complete. If this includes reaching the target position, then Bit 10 changes to 1. If the positioning has been interrupted by a "Halt" or a fault, Bit 10 remains at 0.

The source code works also in the same way than the other operation modes. Now Ref16 contains the set speed and Ref32 the set position.



Jog mode (manual mode)

The product manual contains detailed information.

8.5.1 Operating mode Jog

Overview of jog

The motor moves by one jog unit or at constant speed of rotation in continuous operation. The length of the jog unit, the values for the speed of rotation and the waiting time prior to continuous operation can be set.

The current motor position is the start position for the Jog operating mode. The jog distance and the values for the speed of rotation are entered in user-defined units.

If a positive and a negative jog are requested at the same time, there is no motor movement.

Starting the operating mode

In the case of fieldbus control mode, the operating mode must be set using the parameter `DCOMopmode`.

With the start signal for the jog movement, the motor first moves by a defined jog distance `JOGstepusr`. If the start signal is still available after a specified waiting time `JOGtime`, the device switches to continuous operation until the start signal is canceled.

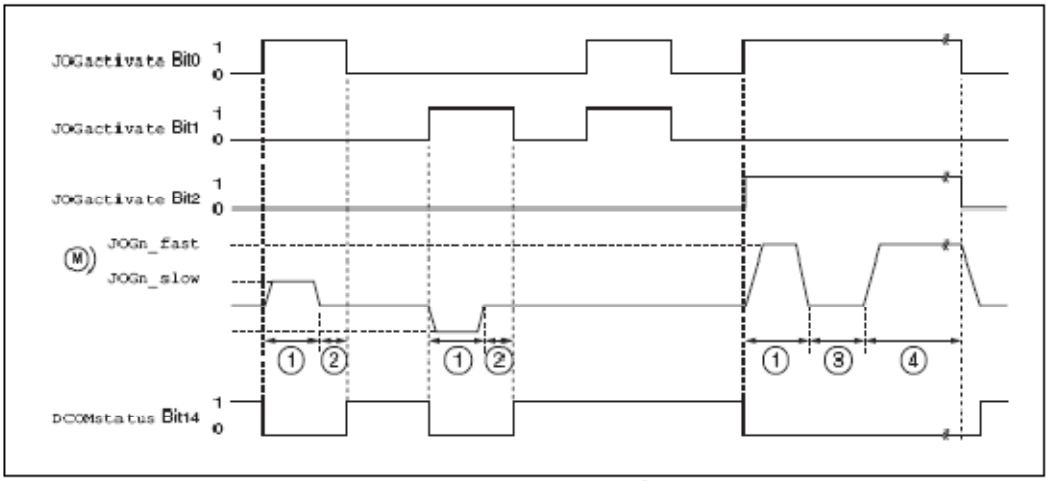


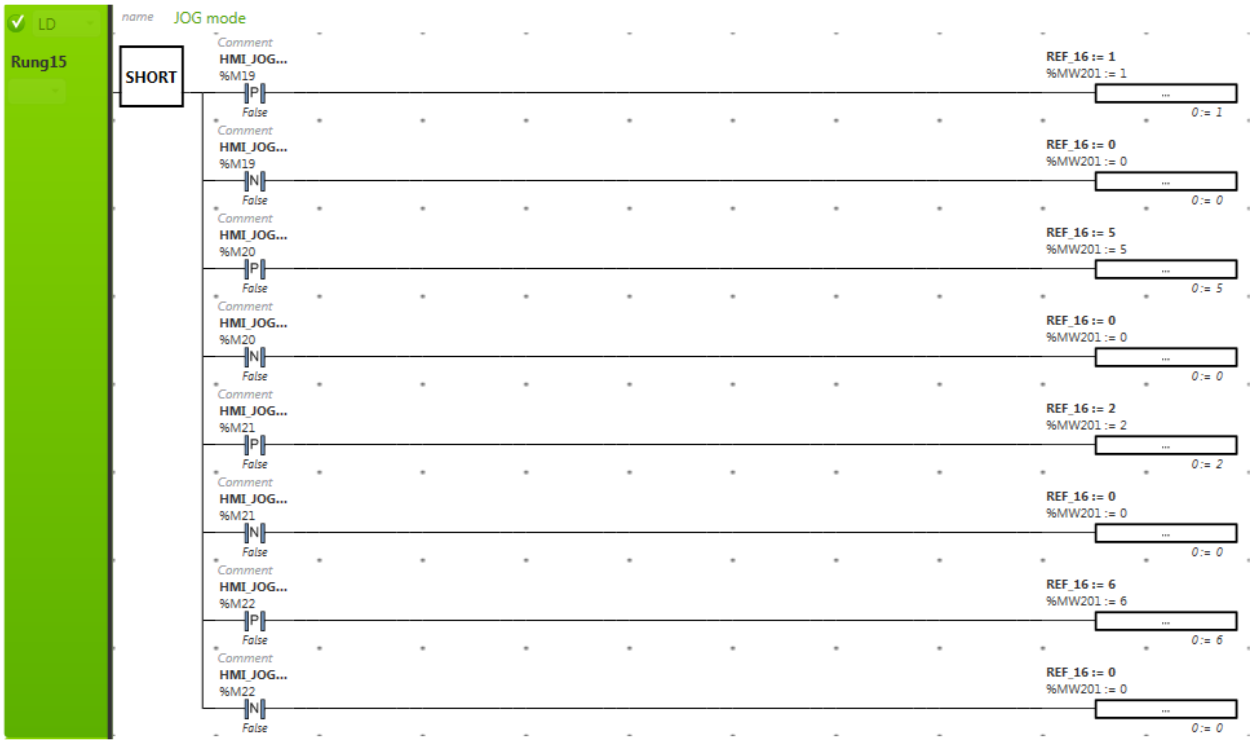
Figure 8.7 Jog, slow and fast

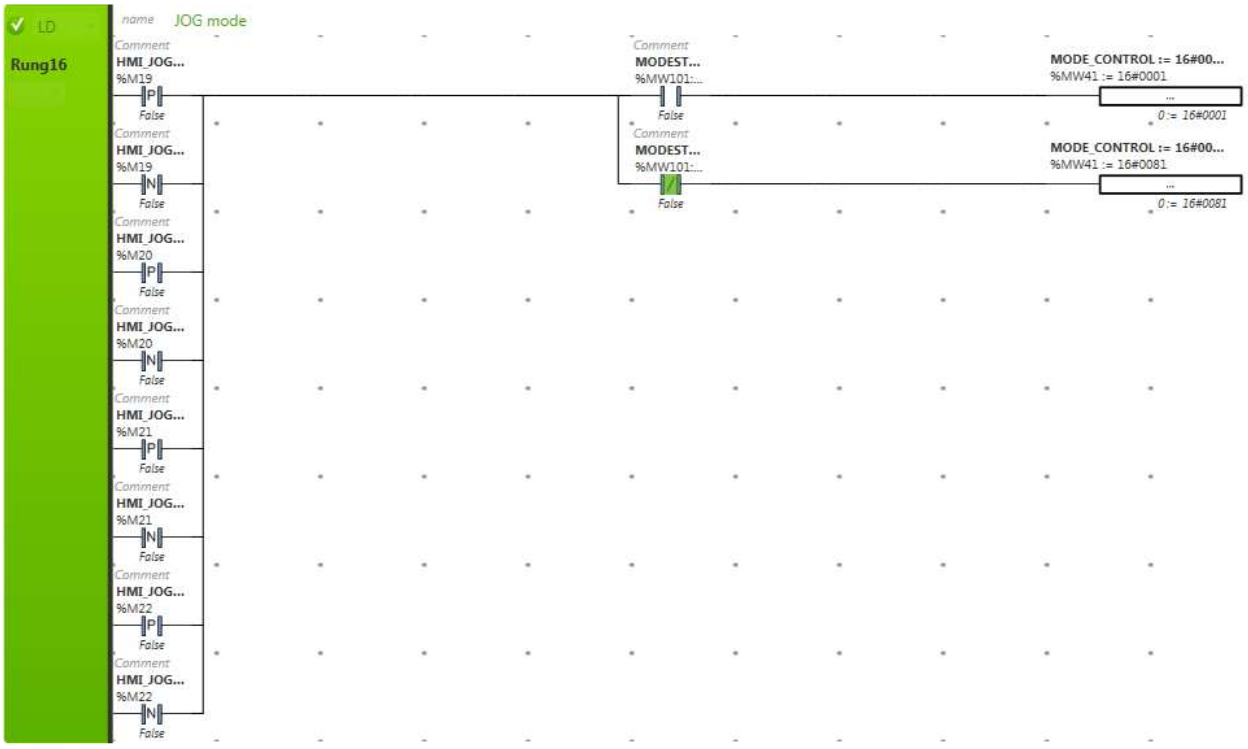
(1) Distance unit
(2) t < waiting time
(3) t > waiting time
(4) Continuous operation

The bits %M19 to %M22 are the signals to control JOG mode. For the manual mode Ref16 has the meaning of parameter *JOGactivate* (Modbus 6930). The resulting bit combinations to written in REF16 are:

no movement	000	=	0
move right slow	001	=	1
move left slow	010	=	2
move right fast	101	=	5
move left fast	110	=	6

A rising edge of a push button starts a movement. The falling edge sets REF16 to zero, the movement stops.




LD
Rung16
name JOG mode
Comment
HMI_JOG...
%M19
P
False
Comment
HMI_JOG...
%M19
N
False
Comment
HMI_JOG...
%M20
P
False
Comment
HMI_JOG...
%M20
N
False
Comment
HMI_JOG...
%M21
P
False
Comment
HMI_JOG...
%M21
N
False
Comment
HMI_JOG...
%M22
P
False
Comment
HMI_JOG...
%M22
N
False
Comment
MODEST...
%MW101:...
False
Comment
MODEST...
%MW101:...
/
False
MODE_CONTROL := 16#00...
%MW41 := 16#0001
...
0 := 16#0001
MODE_CONTROL := 16#00...
%MW41 := 16#0081
...
0 := 16#0081

Reading and writing parameter via SDO

For reading and writing of parameters an additional READ_VAR / WRITE_VAR function block is required. The general way to proceed such requests can be seen in rung 1 (read actual position) and rung 3 (write Nodeguarding time). Please take care that on Modbus all parameters are handeld as 32-bit-values, even if they are only 16-bit-values.

Summary

This program example shows an easy way how to start the operation modes homing, profile velocity, profile position and jog mode. It shows also how to read and write parameters out of the application.

For a sequence of different operation modes it is important that one movement has finished before the next starts. Therefore it is mandatory to check the bits *x_err* and *x_end* in the drive status. Here is an example: