LinMot®

Master Encoder User Manual

(Includes Release 1.3.16 and 1.4.11)

Note
The information in this documentation reflects the stage of development at the time of press and is therefore without obligation.
NTI reserves itself the right to make changes at any time and without notice to reflect further technical advance or product improvement. Please refer to the latest edition of our "General business terms"

Version 1.19 / May 8, 2008

1. System Overview

In automation applications it is often necessary to synchronize several systems not only in time lock but also in position lock mode. The master encoder add on module for *LinMot®* controllers opens solutions for position locked applications like cam, master encoder, winding etc.

A typical electronic line shaft solution with *LinMot®* would look like this:

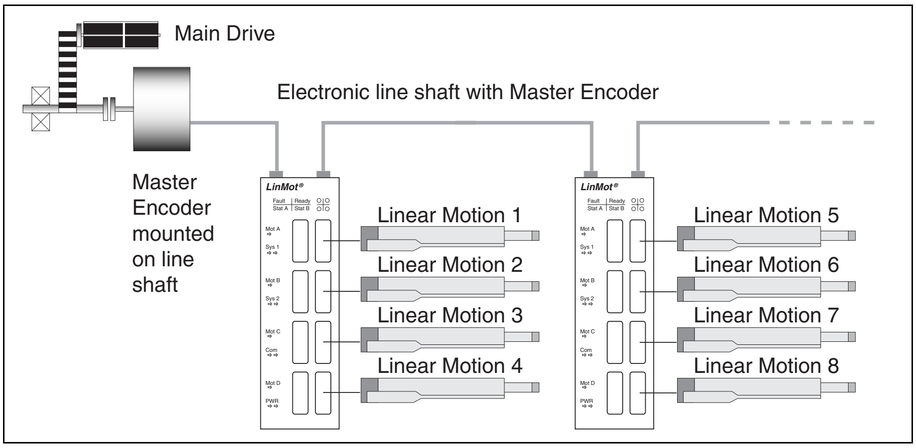

Figure 1-1: Electronic Line Shaft with Master Encoder

The master encoder module family consists of several types, which provide up to 2 complete encoder links and up to 8 digital inputs and outputs.

The following table gives an overview over the different encoder module types, which will be shortly describes below.

	ME01-01/08	ME01-02/08	DI01-08/08	EX001-ME, EX031-ME
Link A	ME / Pos. Sensor	ME / Pos. Sensor	-	ME / Pos. Sensor
Link B	loop through only	Pos. Sensor	-	loop through only
Digital Inputs	✘	✘	✘	✘
Digital Outputs	✘	✘	✘	✘

Version Ex001-ME, Ex031-ME

The version 3 controller series with the master encoder option (-ME) provide the master encoder functionality without an additional module. Most of the interfaces and features are the same on both systems. The porting from a ME01- module to a EX0x1-ME controller and a description of the differences can be found in chapter 3.10 “Changes in EX0X1-ME”.

Version ME01-01/08

The master encoder module ME01-01/08 provides one fully equipped encoder link (link A). The link B can be used for looping through the incremental signals of link A. In additionally there are eight digital inputs and outputs on this module.

Version ME01-02/08

The master encoder module ME01-02/08 provides two fully equipped encoder links (links A and B). Link B can also drive out the incremental signals from link A and link B can be used for external A/B position sensors. Additionally there are eight digital inputs and outputs on this module.

Version DI01-08/08

This module provides 8 digital inputs and 8 digital outputs without encoder links. The inputs and outputs are designed for a 24 Volt interface and the signal lines are galvanically isolated.

2. Hardware Installation

This chapter shows how to mount and wire the different combinations of main module, master encoder module and encoder.

2.1 Installation on Exx0

The following picture shows the power cabling of a master encoder module mounted on the controller type EX00-MT, DN or EX30-DP:

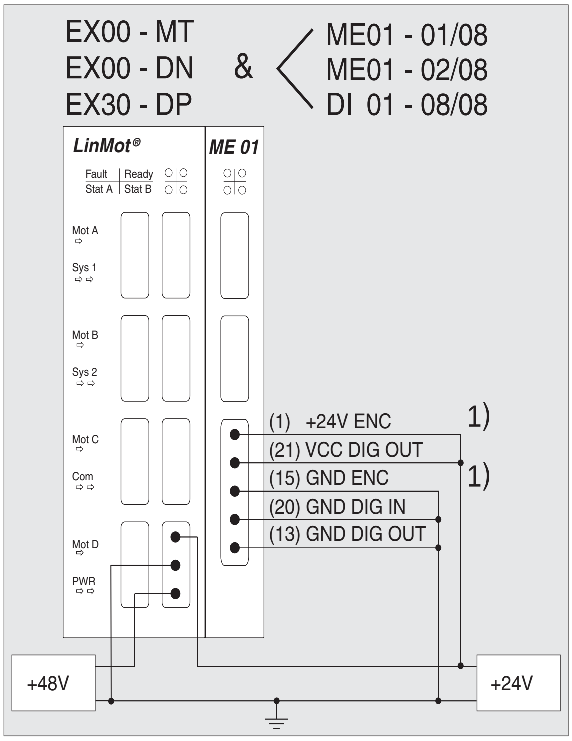

Figure 2-1: Power Supply

1) **24V ENC** and **GND ENC** are not used for DI01-08/08.

The pin numbers, which are set in brackets, belong to the **Control I/O** connector of the master encoder module.

2.2 Installation on Ex0x0

The following picture shows the power cabling of a master encoder module mounted on the controller EX000-MT, DN or EX030-DP:

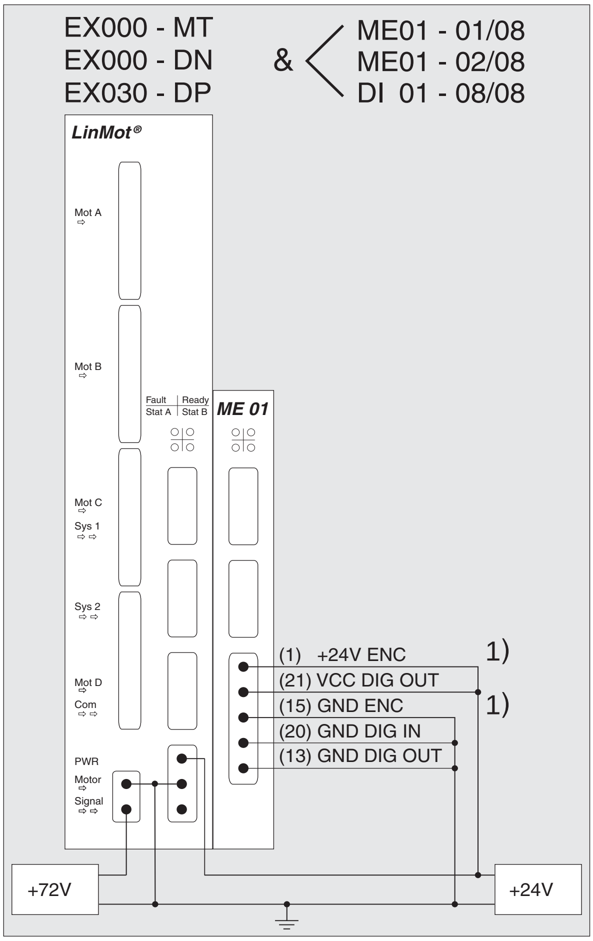

Figure 2-2: Power Supply

1) **24V ENC** and **GND ENC** are not used for DI01-08/08.

The pin numbers, which are set in brackets, belong to the **Control I/O** connector of the master encoder module.

2.3 Incremental Encoder

An incremental encoder with ABZ signals, where Z is optional, is connected to the master encoder module the following way:

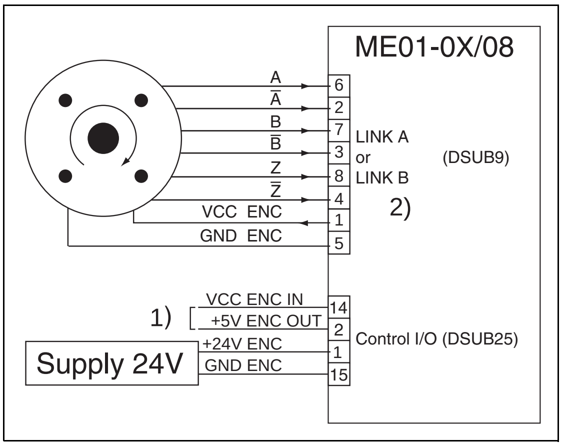

Figure 2-3: Wiring of an incremental ABZ encoder on ME01-0X/08 modules

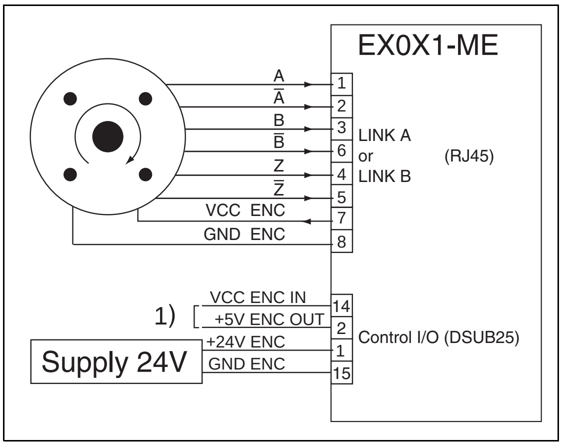

Figure 2-4: Wiring of an incremental ABZ encoder on EX0X1-ME controllers

1) If the encoder works with a 5V supply voltage, the internally generated 5V can be used by connecting the pins **VCC ENC IN** and **+5V ENC OUT**. If the encoder has a different supply voltage an additional power source is necessary.

2) ME01-01/08 modules do not provide this function on link B.

See for more information in chapter 3. “Hardware Description”.

2.4 Incremental Stepper Interface

A stepper motor interface can be connected to the master encoder module the following way:

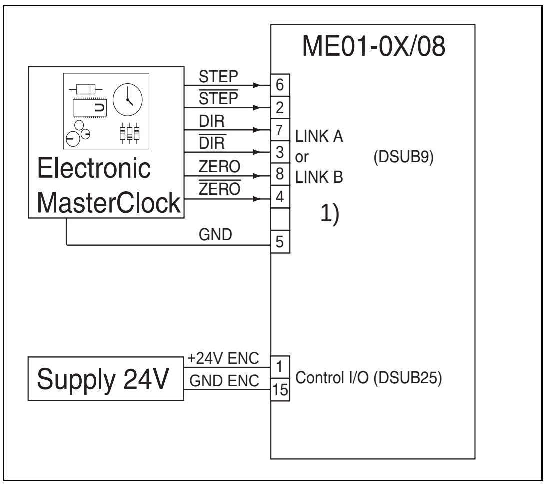

Figure 2-5: Wiring of a incremental stepper encoder on ME01-0X/08 modules

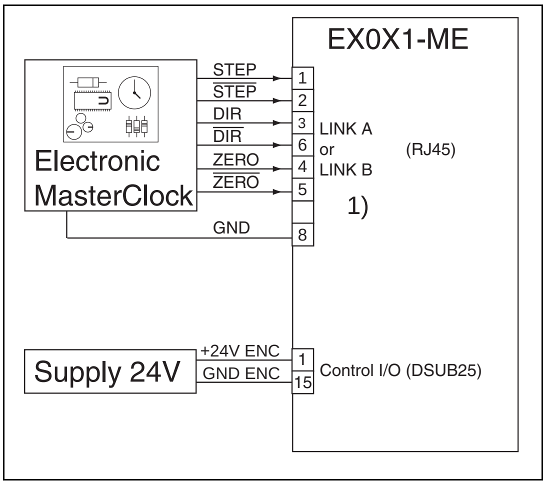

Figure 2-6: Wiring of a incremental stepper encoder on EX0X1-ME controllers

1) ME01-01/08 modules do not provide this function on link B.

The signals **STEP**, **DIR** and **ZERO** are RS422 compatible. **ZERO** can be wired optionally. See for more information in chapter 3. “Hardware Description”.

2.5 Absolute Encoder SSI

Absolute encoders are wired with the master encoder module the following way. **Note:** SSI modes are only supported by the ME01-0X/08 add-on modules, the controllers EX0X1-ME do not provide this function.

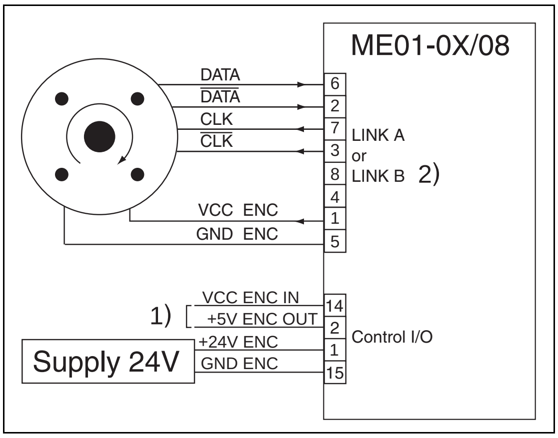

Figure 2-7: Wiring of an absolute encoder SSI on ME01-0X/08 modules

1) If the encoder works with a 5V supply voltage, the internally generated 5V can be used by connecting the pins **VCC ENC IN** and **+5V ENC OUT**. If the encoder has a different supply voltage an additional power source is necessary.

2) Only ME01-02/08 modules provide this function on link B.

See for more information in chapter 3. “Hardware Description”.

2.6 Loop Through (incremental)

If there is just one incremental encoder, on which several motors from different electronics are synchronized, there is the possibility to loop through the encoder signals.

The following figure shows the wiring of an incremental ABZ encoder in loop through mode.

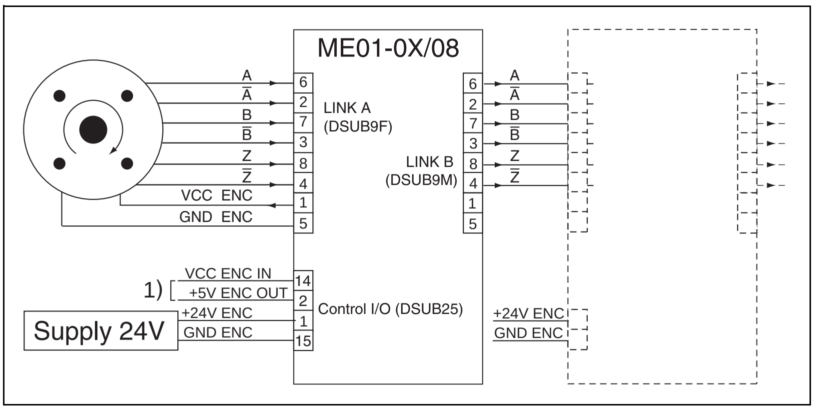

Figure 2-8: Loop through with a ABZ encoder on ME01-0X/08 modules

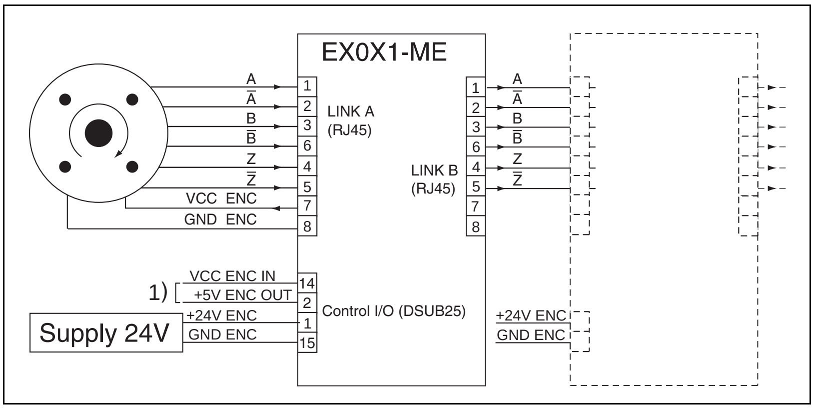

Figure 2-9: Loop through with a ABZ encoder on EX0X1-ME controllers

1) If the encoder works with a 5V supply voltage, the internally generated 5V can be used by connecting the pins **VCC ENC IN** and **+5V ENC OUT**. If the encoder has a different supply voltage an additional power source is necessary.

The next figure shows the wiring of a stepper interface used in loop through mode.

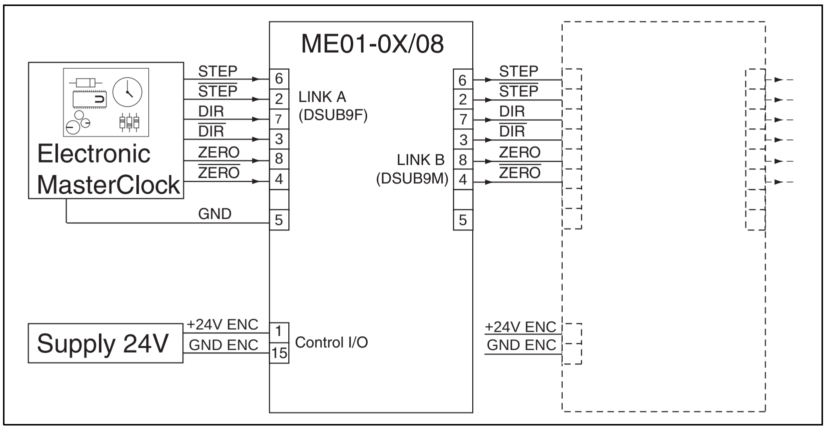

Figure 2-10: Loop through with a stepper interface encoder on ME01-0X/08 modules

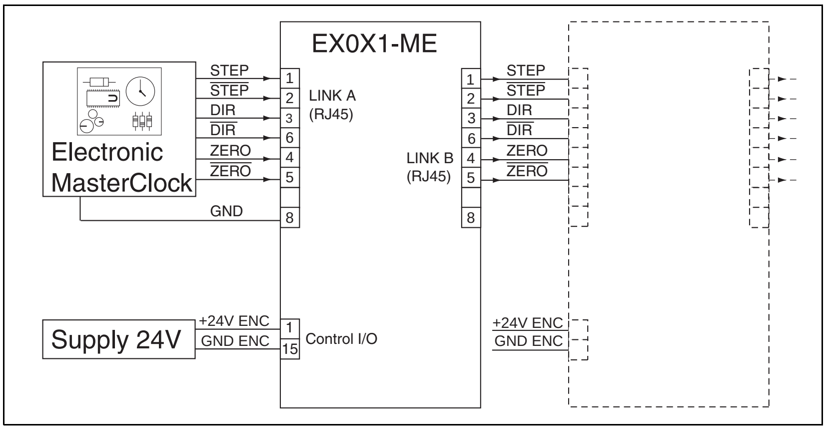

Figure 2-11: Loop through with a stepper interface encoder on EX0X1-ME controllers

2.7 Digital Inputs

The eight digital inputs are designed for a 24V interface. On the ME01-0X/08 and DI01-08/08 the inputs are galvanically isolated. The wiring can be done like this:

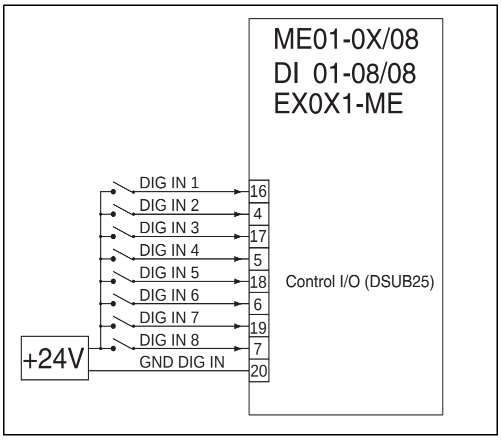

Figure 2-12: Wiring of the digital inputs

For more information see in chapter 3. “Hardware Description”.

2.8 Digital Outputs

All the eight digital outputs are designed for a 24V interface. On the ME01-0X/08 and DI01-08/08 the outputs are galvanically isolated.

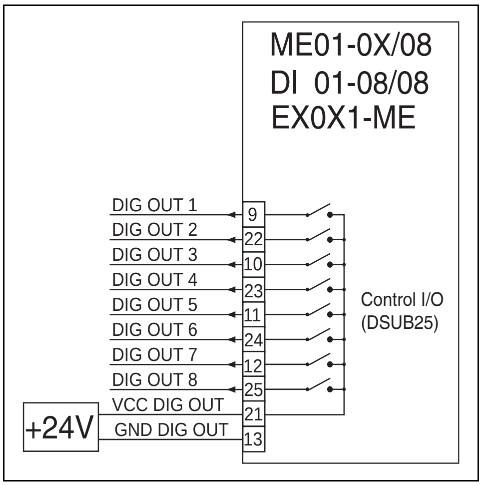

Figure 2-13: Wiring of the digital outputs

The outputs switch electronically the **+24V DIG OUT** to the output pin. The interface must be supplied (pins 21 and 13). For more information see in chapter 3. “Hardware Description”.

3. Hardware Description

This chapter describes the circuitry of the different links of the master encoder modules.

3.1 Mounting

The picture below shows how the master encoder add on module is mounted on a *LinMot*® EXX0 controller and the mechanical dimensions.

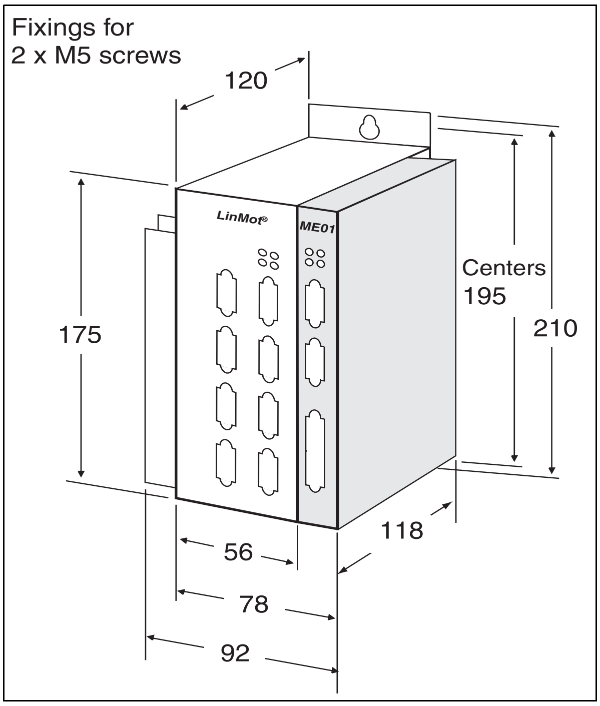

Figure 3-1: Mounting and dimensions

The picture below shows how the master encoder add on module is mounted on a *LinMot*® EX0X0 controller and the mechanical dimensions.

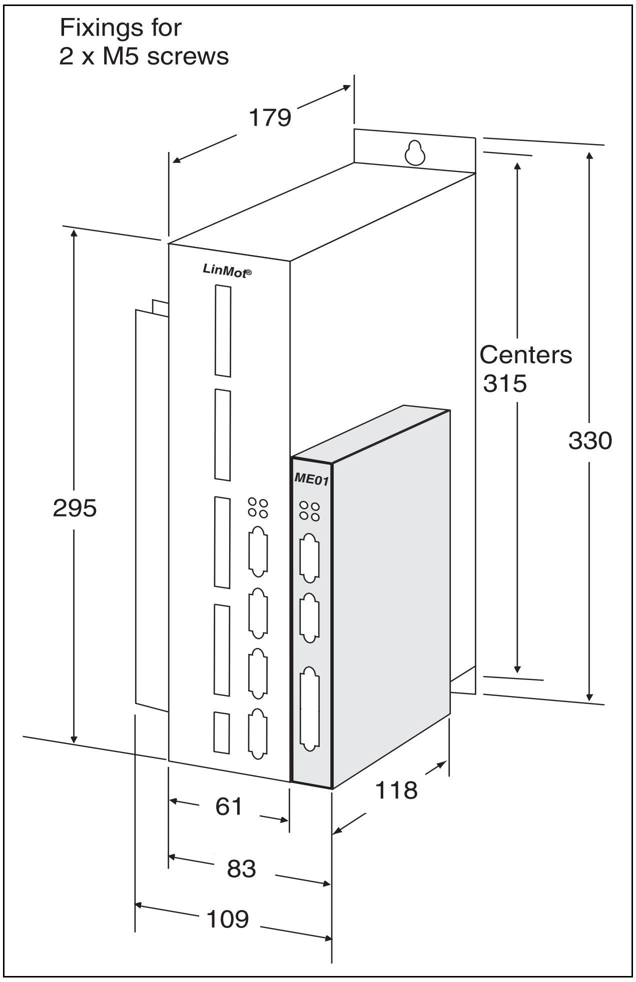

Figure 3-2: Mounting and dimensions.

The picture below shows the mounting dimension of a EX0X1-ME controller.

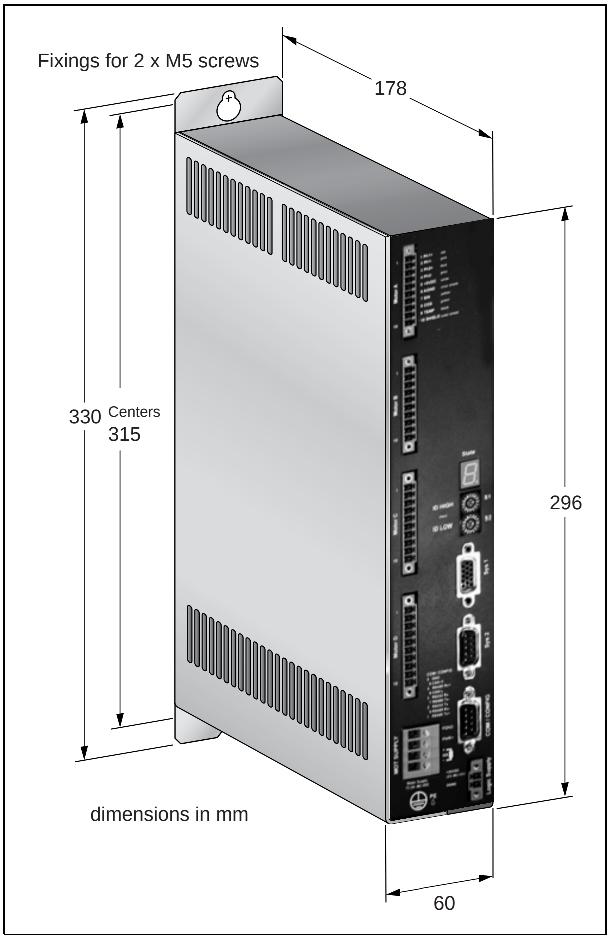

Figure 3-3: Mounting and dimensions.

3.2 Link A

The link A supports full encoder functionality on the modules ME01-01/08 and ME01-02/08. For the pin out please seechapter 3.9 “Pinout”.

Characteristics

Item	Specification
Connector Type	DSUB-9 female / RJ45
Signal Levels	differential RS422
Encoder Modes	incremental (A, B, Z), up to 16 bits incremental (STEP, DIR, ZERO), up to 16 bits Only on ME01-0x/08: absolute SSI(DATA, CLK), up to 24 bits
Input Current	ME01-0x/08: +/- 0.5mA @ DC +/- 100mA max. peak when switching typ. +/- 25mA @ 500kHz Ex0x1-ME: typ. 50mA
Encoder Supply	internally generated (5V, 500mA max.) or externally (Control I/O, signal VCC ENC IN, pin 14)
Max. Signal Freq.	500 kHz
Protection	Only on ME01-0x/08: Galvanic Isolation
Termination	All signal lines are terminated. See chapter 3.4 “Encoder Signal Termination”.

3.3 Link B

The link B supports full encoder functionality on the module ME01-02/08. On ME01-01/08 only the loop through mode is supported. For the pin out see chapter 3.9 “Pinout”.

Characteristics

Item	Specification
Connector Type	DSUB-9 male / RJ45
Signal Levels	differential RS422
Encoder Modes on ME01-01/08	loop through (incremental only)
Encoder Modes	incremental (A, B, Z), up to 16 bits incremental (STEP, DIR, ZERO), up to 16 bits Only on ME01-0x/08: absolute SSI(DATA, CLK), up to 24 bits
Input Current	ME01-0x/08: +/- 0.5mA @ DC +/- 100mA max. peak when switching typ. +/- 25mA @ 500kHz Ex0x1-ME: typ. 50mA
Encoder Supply	internally generated (5V, 500mA max.) or externally (Control I/O, signal **VCC ENC IN**, pin 14)
Max. Signal Freq.	500 kHz
Protection	Only on ME01-0x/08: Galvanic Isolation
Termination	All signal lines are terminated. See chapter 3.4 “Encoder Signal Termination”.

3.4 Encoder Signal Termination

The figure below shows the termination of the RS422 signals on the encoder links A and B. These termination circuits are placed on all lines and cannot be disabled (by jumpers e.g.).

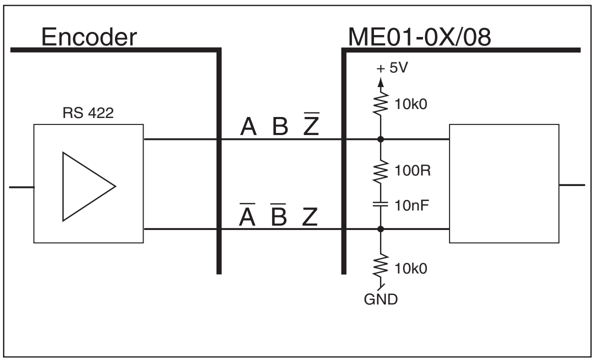

Figure 3-4: RS422 signal termination on ME01-0x/08

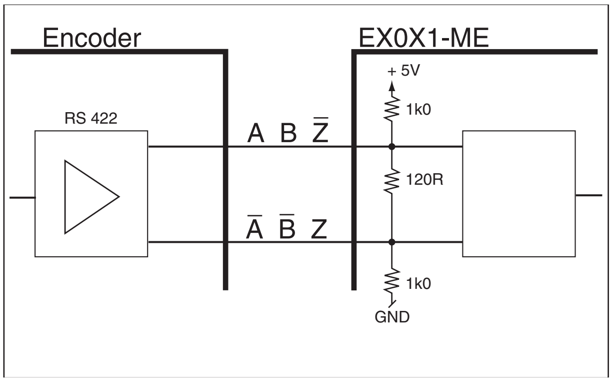

Figure 3-5: RS422 signal termination on EX0X1-ME

3.5 Single Line Wiring

The figure below shows the wiring of the RS422 inputs when only one signal line is used. In this case only the non-inverted input is used.

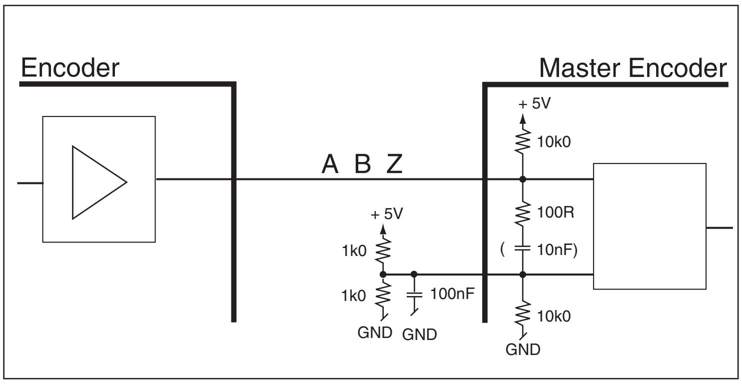

Figure 3-6: Single Line Wiring on ME01-0x/08

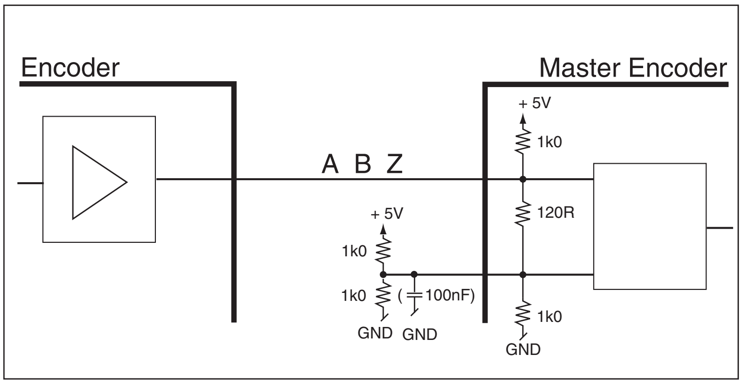

Figure 3-7: Single Line Wiring on EX0X1-ME

Note: For a better operating safety and higher frequencies, it is strongly recommended to use differential signal wiring. In practical applications single line wiring has not performed very stable (drift, noise...)!

3.6 Digital Inputs

On the master encoder module, there are eight galvanically isolated digital inputs. All inputs are protected against inverse polarity and have a common ground line. The pin out is shown in chapter 3.9 “Pinout”. The functionality of the digital inputs is mapped fix.

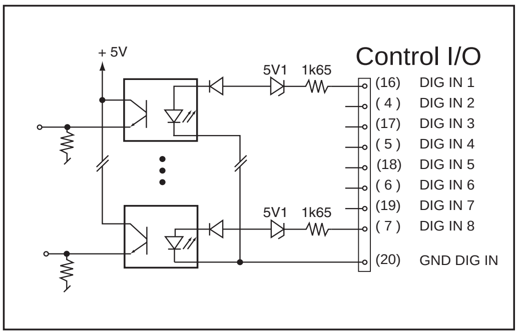

Figure 3-8: Digital Input Circuit on ME01-0X/08 and DI01-08/08 modules

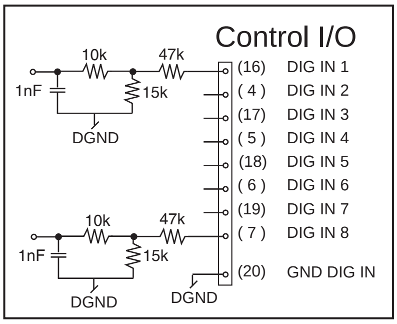

Figure 3-9: Digital Input Circuit on EX0X1-ME controllers

Characteristics

Item	Specification
Number of Inputs	8
Logic Levels	low < 5V, high > 15V
Nominal Voltage	24 V
Input Current	ME01-0x/08: typ. 11mA @ 24V (max. 15mA) Ex0x1-ME: typ. 0.5mA @ 24V (50kOhm input resistance)
Max. Frequency	2kHz
Protection	Inverse Polarity Protection Only on ME01-0x/08: Galvanic Isolation

Signals

In the present software version, the function of the digital inputs is mapped fix. The following table gives an overview, in chapter 4.6 “Digital Inputs” the software function is described.

Signal	Function
DIG IN 1	Trigger enable for motor A
DIG IN 2	Trigger enable for motor B
DIG IN 3	Trigger enable for motor C
DIG IN 4	Trigger enable for motor D
DIG IN 5	Recipe input bit 0
DIG IN 6	Recipe input bit 1
DIG IN 7	Recipe input bit 2
DIG IN 8	Recipe input bit 3

3.7 Digital Outputs

There are eight digital outputs on the master encoder module with the characteristics below. The pin out is shown in chapter 3.9 “Pinout”.

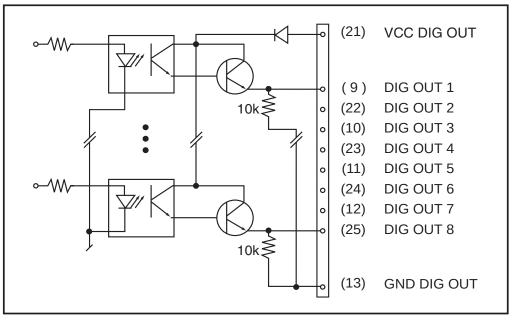

Figure 3-10: Digital Output Circuit on ME01-0X/08 and DI01-08/08 modules

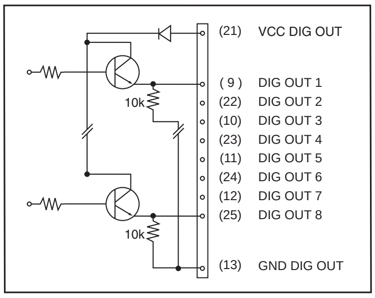

Figure 3-11: Digital Output Circuit on EX0X1-ME controllers

Characteristics

Item	Specification
Number of Outputs	8
Max. Current	0.5 A each output
Output Type	Electronic High Side Switch
Supply Voltage	6 .. 34 V (between VCC DIG OUT and GND DIG OUT)
Max. Frequency	1kHz
Protection	Overload Protection Inverse Polarity Protection Only on ME01-0x/08: Galvanic Isolation **Note**: There is no recovery diode placed for driving relais.

3.8 Power Supply

Each IO part of the master encoder module has its own galvanically isolated power supply. The master encoder interface needs 5V, which will be generated on the board from a 24V supply. These 5V are available for supplying the encoder.

Characteristics

Master Encoder Interface Supply	**Specification**
Voltage Range	20.4 .. 28.8 V
Power Consumption	max. 6 W
Pins Names (Numbers)	+24V ENC (1) and GND ENC (15) *)
+5V Output	+5V ENC OUT (2), max. 500mA
Encoder Supply	**Specification**
Pin Names (Numbers)	VCC ENC (14) and GND ENC (15) *)
Digital Inputs Supply	**Specification**
Pin Names (Numbers)	GND DIG IN (20) *), common ground line
Digital Outputs Supply	**Specification**
Voltage Range	6 .. 34 V
Max. Current	4 A
Pin Names (Numbers)	+24V DIG OUT (21) and GND DIG OUT (13)*)

*) On Ex0x1-ME types the pins 13, 15 and 20 are internally wired together and connected to DGND.

3.9 Pinout

The following figure shows the pin outs of the three connectors as a summary. The specifications of the signals can be found in the chapters above.

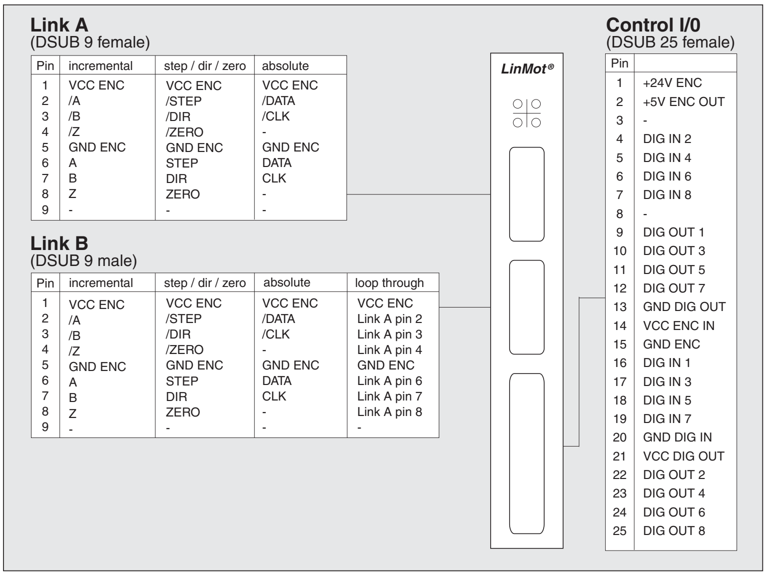

Figure 3-12: Pin Out Overview on ME01-0X/08 modules

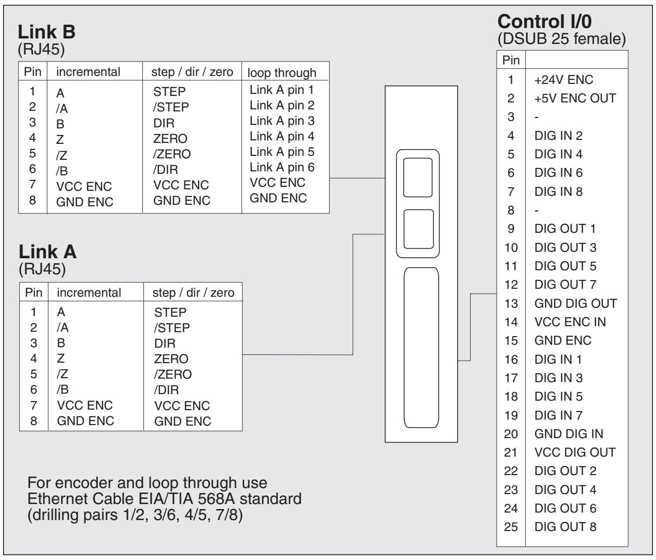

Figure 3-13: Pin Out Overview on EX0X1-ME controllers

The connector types are:

Name	Connector Type ME01-0X/08	Connector Type EX0X1-ME
Link A	DSUB-9, female	RJ45
Link B	DSUB-9, male	RJ45
Control I/O	DSUB-25, female	DSUB-25, female

3.10 Changes in EX0X1-ME

Differences

The following table shows the differences between the master encoder functionality on a EX0X1-ME controller and a ME01-0X/08 module.

Item	ME01-0X/08	EX0X1-ME
Location	Add On Module	Integrated on controller's bottom side
Dimension	see chapter 3.1 "Mounting"	see chapter 3.1 "Mounting"
Connector Type Enc. Link A	DSUB-9, female (see chapter 3.9 "Pinout")	RJ45 (see chapter 3.9 "Pinout")
Connector Type Enc. Link B	DSUB-9, male (see chapter 3.9 "Pinout")	RJ45 (see chapter 3.9 "Pinout")
Galvanic Isolation (Encoder Links & Digital I/Os)	Yes	No
Digital Input Current	typ. 11mA @ 24V (max. 15mA)	typ. 0.5mA @ 24V (50kOhm input resistance)
SSI Encoder Type	Supported	Not Supported

Application Conversion

The following steps are a guide to change an application realized with a ME01-0X/08 module to a system with a EX0X1-ME controller:

- Save the parameter configuration and curves from the old system. Therefore use the appropriate LinMot-Talk software version (typ. 1.3.14 or 1.3.12).
- Organize a EX0X1 controller with the -ME option.
- If encoder link A is used, with the adapter cable K025-RJ45-D/f-0.5 (article number 0150-1866) the DSUB9 from the encoder can be linked to the RJ45 jack.
- If encoder link B is used, with the adapter cable K025-RJ45-D/m-0.5 (article number 0150-1867) the DSUB9 from the encoder can be linked to the RJ45 jack.
- If several controllers are looped together, take a 1:1 Ethernet patch cable according to EIA/TIA 568A standard (drill pairs 1/2, 3/6, 4/5, 7/8), e.g. RJ45-08/0.3 (LinMot article number 0150-1852).
- Connect all necessary links to the new EX0X1-ME controller and install thenewest firmware release (1.3.16).
- Login to the controller and open the saved configuration offline. Set all parameters of the new system according to the settings in the offline configuration.
- Load the saved curves.

4. Software Description

4.1 Installation

The hardware and software commissioning is described in chapter 5. “Commissioning”.

4.2 Compatibility

The present software version (1.4.7ME) runs only on the newest *LinMot*®-MT controllers (version 2), this is because of the hardware compatibility. The system provides full AT and MT functionality. In this version, the function of the digital inputs is mapped fix.

For the EX0X1-ME controller types, the master encoder functionality is included in the release 1.3.16 itself. The main difference to the 1.4.7ME version is the lack of the SSI encoder type support, because of the different hardware,

LinMot® keeps the right to make changes in later version, which might not be downwards compatible.

4.3 Overview

Additionally to the AT and MT functionality the master encoder software provides the possibility of executing motion profiles based on an encoder position (cam locked mode).

This software is configurable with an extended parameter tree, in which the encoder type, the application mode and the behavior of the digital IOs can be set. Due to the additional digital inputs and outputs, the digital command interface has become more flexible. The ASCII protocol, which is fully supported, has become more powerful with the addition of some master encoder specific commands.

4.4 Application Modes

The master encoder software supports three basic application types: cam, parallel winding and cross winding.

Cam

The cam application mode supports the electronic cam functionality in general. The cams are defined like other motion profiles and are saved on the *LinMot®* controller. When executing cam profiles, the speed will be overtaken from the encoder motion. Cams can be started on encoder reference, on registration inputs or continuously. There are configuration parameters for cam length (over how many encoder counts a cam goes), cam delay (encoder counts delay after reference or registration input) and cam start point (the angle in the cam). Depending on the corresponding encoder type, the cam application mode can have three submodes, which are called and discussed as follows: Fixed Cam, Registered Cam and Continuous Cam.

Fixed Cam (Incr. (ABZ), STEP/DIR/ZERO or Absolute SSI)

The fixed cam mode is selected in combination with the encoder types “Incr. (ABZ)”, “STEP/DIR/ZERO” (or ”Absolute SSI” on ME01-0X/08 modules).

The relation between encoder position and motor position is visualized in the following figure:

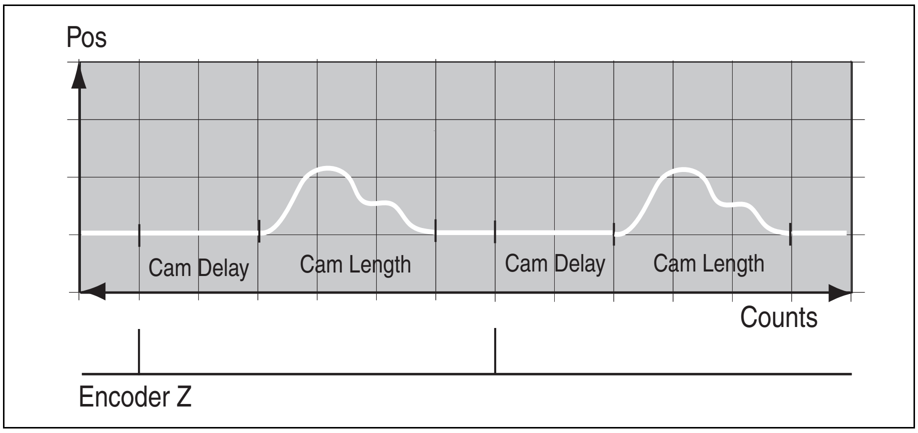

Figure 4-1: Motor Position according to Encoder Counts

The motion is strictly cyclic. If the encoder counts up, when Z occurs the motor stands still for “Cam Delay” encoder counts and then runs the cam over “Cam Length” counts. If the encoder counts down, after the Z occurs the motor stands still for (“Counts/Rev.” - “Cam Delay” - “Cam Length”), then runs the cam backwards over “Cam Length” counts and at the end stands still for “Cam Delay” counts. The sum of “Cam Delay” and “Cam Length” must be less or equal than “Counts/Rev”.

Registered Cam (Incr. (A B) with ext. Reg., STEP/DIR with ext. Reg. or Absolute SSI with ext. Reg.)

The registered cam mode is selected in combination with the encoder types “Incr. (A B) with ext. Reg.”, “STEP/DIR with ext. Reg.” or ”Absolute SSI with ext. Reg.”. This mode is typically used for applications with a conveyor belt and an light barrier, where the synchronization must be against the conveyor.

Independent from the encoder count direction, after the rising edge on the registration input occurs, the motor waits for “Cam Delay” counts and then starts the cam over “Cam Length” counts. If the encoder counts up, the following behavior is defined:

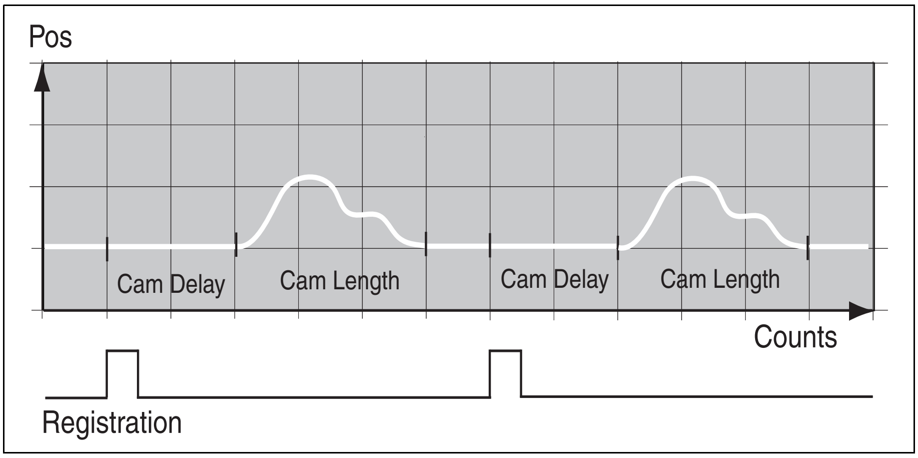

Figure 4-2: Registered Cam when encoder counts up

If the encoder counts down, the following behavior is defined:

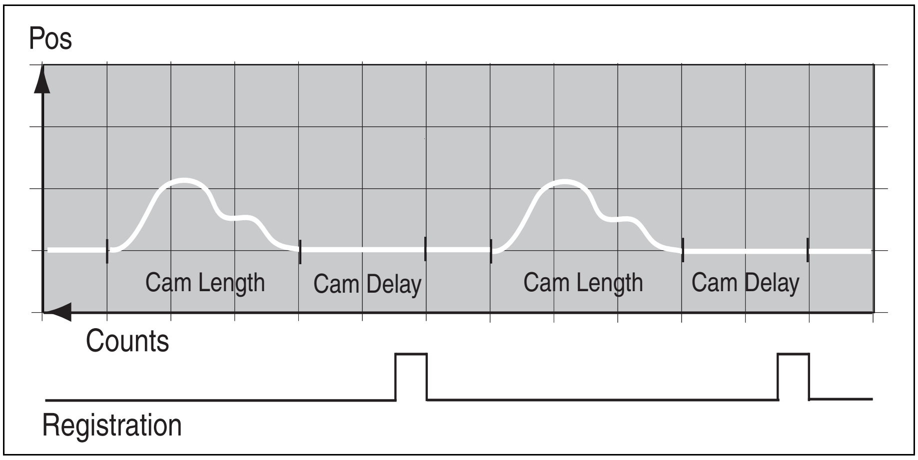

Figure 4-3: Registered Cam when encoder counts down

A new cam cycle is only initiated on a rising edge of the registration input and if the previous cycle has completely finished (minimal “Cam Delay” plus “Cam Length” counts).

Continuous Cam (Incr. (A B) Continuous or STEP/DIR Continuous)

The continuous cam mode is selected in combination with the encoder types "Incr. (A B) Continuous" or "STEP/DIR Continuous". This mode is typically used for cutting applications. The cutting itself is defined with the cam and the length of the cut object can be adjusted with the "Cam Delay" parameter.

A cycle is started when the cam enable input goes high or if the cam enable is not regarded when the software enters in RUN mode.

When the cam is finished and the cam enable is still active, the next cycle starts immediately with the cam delay phase.

One a cycle has started it will be completely finished.

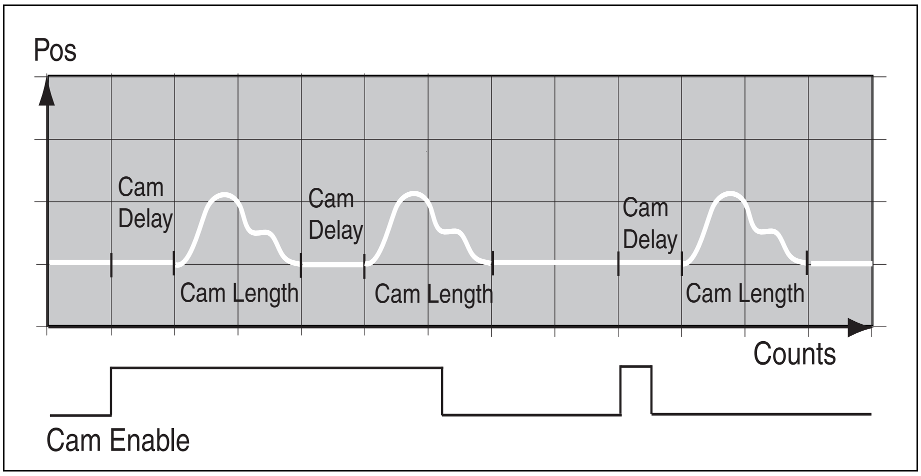

Figure 4-4: Continuous Cam when counting up

If the encoder counts down, the following behavior is defined:

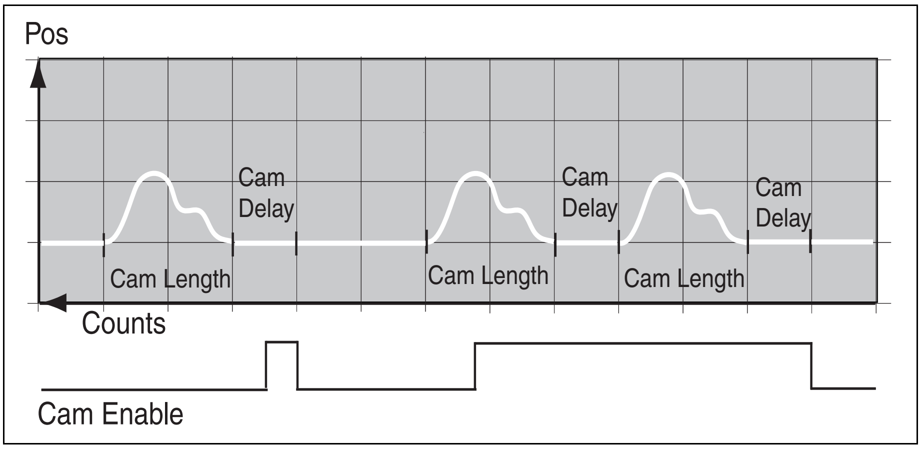

Figure 4-5: Continuous Cam when counting down

Parallel Winding

In this application type, a wire (e.g.) has to be guided according to the rotation speed of a coil on which it has to be wired up. The motion ratio between the *LinMot*® motor and the winder coil must be definable (diameter of the wire) and the turn around points have to be set on-line. This application is only for incremental encoder types.

The parallel winding mode generates a pattern like this:

Figure 4-6: Parallel winding mode pattern

Cross Winding

The cross winding application is quite similar to the parallel winding application. It will create another winding pattern and therefore it has some other parameters. See chapter 4.11 “Parameters”.

The cross winding mode generates a pattern like this:

Figure 4-7: Cross winding mode pattern

4.5 Encoder Types

The master encoder hardware module supports different encoder types in different modes, which can be selected in the parameter tree (under \Encoder\Encoder Setup\Type). Each encoder type (ABZ, STEP/DIR/ZERO and Absolute SSI) can be used with the encoder internal zero signal (for permanently clutched cams) or the cam motion can be started on an external registration input. See chapter 4.11 “Parameters”.

4.6 Digital Inputs

There are two selectable mappings for the digital inputs in the software version 1.4.1ME. The pinning is selected in the **\Encoder\Inputs\Input Location** parameter. See **chapter 4.11 “Parameters”.**

Normal Mapping

The following table shows the input mapping if the parameter is set to **Normal**.

Input Name	Function	Electronic Module	Connector and Pin
TRIG/ANALOG IN 1	Trigger Input Motor A	MT	Sys 2 Pin 1
TRIG/ANALOG IN 2	Trigger Input Motor B	MT	Sys 2 Pin 2
TRIG/ANALOG IN 3	Trigger Input Motor C	MT	Sys 2 Pin 6
TRIG/ANALOG IN 4	Trigger Input Motor D	MT	Sys 2 Pin 7
DIG IN 1	Cam Enable Motor A	ME	Control I/O Pin 16
DIG IN 2	Cam Enable Motor B	ME	Control I/O Pin 4
DIG IN 3	Cam Enable Motor C	ME	Control I/O Pin 17
DIG IN 4	Cam Enable Motor D	ME	Control I/O Pin 5
DIG IN 5	Recipe Bit 0 (LSB)	ME	Control I/O Pin 18
DIG IN 6	Recipe Bit 1	ME	Control I/O Pin 6
DIG IN 7	Recipe Bit 2	ME	Control I/O Pin 19
DIG IN 8	Recipe Bit 3 (MSB)	ME	Control I/O Pin 7

Exchanged Mapping

The following table shows the input mapping if the parameter is set to **Exchanged**.

Input Name	Function	Electronic Module	Connector and Pin
TRIG/ANALOG IN 1	**Cam Enable Motor A**	**MT**	**Sys 2 Pin 1**
TRIG/ANALOG IN 2	**Cam Enable Motor B**	**MT**	**Sys 2 Pin 2**
TRIG/ANALOG IN 3	**Cam Enable Motor C**	**MT**	**Sys 2 Pin 6**
TRIG/ANALOG IN 4	**Cam Enable Motor D**	**MT**	**Sys 2 Pin 7**
DIG IN 1	**Trigger Input Motor A**	**ME**	**Control I/O Pin 16**
DIG IN 2	**Trigger Input Motor B**	**ME**	**Control I/O Pin 4**
DIG IN 3	**Trigger Input Motor C**	**ME**	**Control I/O Pin 17**
DIG IN 4	**Trigger Input Motor D**	**ME**	**Control I/O Pin 5**
DIG IN 5	**Recipe Bit 0 (LSB)**	**ME**	**Control I/O Pin 18**
DIG IN 6	**Recipe Bit 1**	**ME**	**Control I/O Pin 6**
DIG IN 7	**Recipe Bit 2**	**ME**	**Control I/O Pin 19**
DIG IN 8	**Recipe Bit 3 (MSB)**	**ME**	**Control I/O Pin 7**

The inputs RUN +/-, INIT +/-, FREEZE +/- and EMERGENCY STOP +/- keep the same function and same pinning as in the AT-mode.

4.7 Digital Outputs

The function of the digital outputs on the *LinMot*® controller base module keeps the same with the master encoder software. The additional outputs on the master encoder module are configurable in the parameter tree (under Encoder\Ouputs\Output X). See chapter 4.11 “Parameters”.

4.8 Defining Cams

Cams can be defined in the **Curve Inspector** similar to curves for time lock mode.

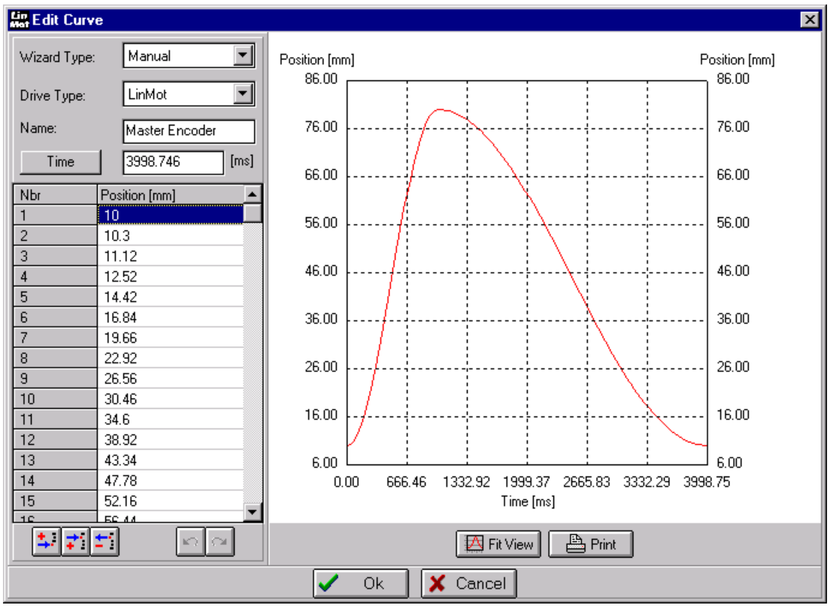

Figure 4-8: Defining a Cam in the Curve Inspector

In contrast to the time for time locked curves, the number of encoder increments over which the cam runs is not defined in the curve editor but can be set in the parameter tree under \Encoder\Encoder Application\Cam Config\Cam Length (X). See chapter 4.11 “Parameters”. For achieving the best motor performance, it is advisable to set the time to the expected value at 100% machine speed.

4.9 Defining Recipes

The master encoder software supports the use of recipes, which means the possibility of working with different cam sets while the system runs. A cam set consists of maximum four different cams, one for each motor. This functionality makes the system possible to proceed product changes on the fly.

The recipe function is activate if the system detects a curve named “Recipe”. Recipes are defined in the *LinMot*®-Talk Curve Editor.

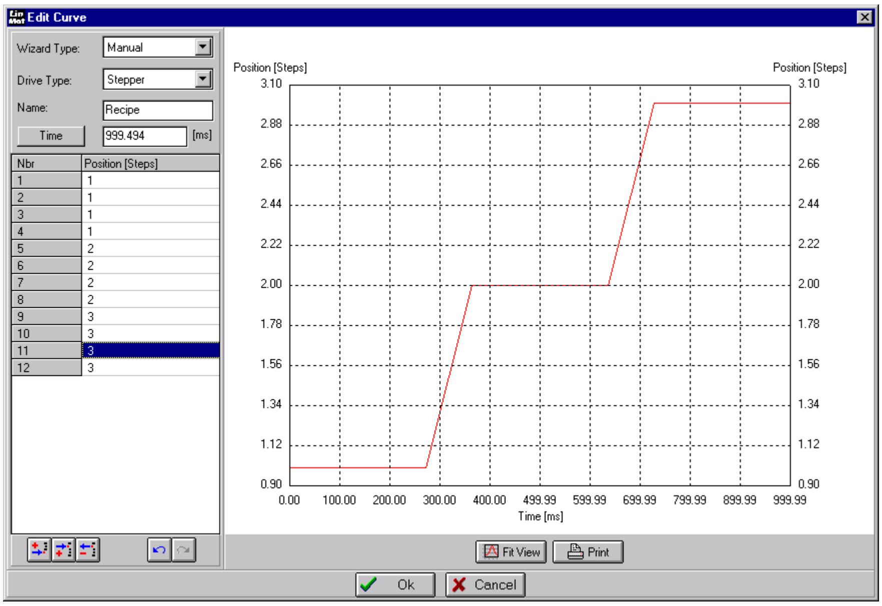

Figure 4-9: Creating a Recipe

A curve named “Recipe” must be defined in the “Manual” mode for a stepper motor. The value for “Time” or “Frequency” is irrelevant. “Recipe” must be written with a capital letter at the beginning and the rest in small letters.

One recipe is defined with four curve numbers (Recipe 1: Nbr. 1-4, Recipe 2: Nbr. 5-8, ...). The curve numbers are filled in the position table. (“1” means curve number 1, “2” curve number 2 ...). The first curve number of each recipe is for motor A, the second for motor B, and so on. The curve graphics, which is drawn automatically in the window on the right, does only show the curve numbers set on the left and has no relevance in this case.

In the curve editor the cam numbers for the different recipes are defined the following way:

Nbr	Position (Steps)	Recipe, Motor, Cam
1	a	0, A, a
2	b	0, B, b
3	c	0 C, c
4	d	0, D, d
5	e	1, A, e
6	f	1, B, f
7	g	1, C, g
8	h	1, D, h
9	i	2, A, i
...	...	..., ..., ...
64	xx	15, D, xx

If the recipe list has more than 64 items, all items with a higher number than 64 will be ignored. If the recipe list is shorter than 64 items and if a digital inputs combination is set which would access to not defined items, the combination will be ignored and the system will keep executing the former recipe.

A recipe (set of cams for each motor) are selected by the digital inputs DIG IN 5 to DIG IN 8 on the master encoder module. The following table shows the mapping of the digital input pattern an the corresponding recipe number.

DIG IN X				Recipe Number
8	7	6	5	
0	0	0	0	0
0	0	0	1	1
0	0	1	0	2
...	...	...	...	...
1	1	1	1	15

A new recipe will be loaded when the digital inputs have changed and the next registration event or zero signal of the encoder has occurred.

This user interface for defining recipes will become more comfortable in future versions.

Example

In this example, we assume the following “Recipe”.

Nbr	Position (Steps)
1	5
2	1
3	3
4	3
5	6
6	2
7	4
8	4

In this case there are two recipes defined, which can be selected the following way:

DIG IN 4	DIG IN 3	DIG IN 2	DIG IN 1	Motor A	Motor B	Motor C	Motor D
0	0	0	0	5	1	3	3
0	0	0	1	6	2	4	4

4.10 ASCII Commands

The ASCII command interface can also be used in combination with the master encoder. There are some additional commands for configuring and controlling. The mechanism of the ASCII command interface is described in the Addendum V1.3. chapter 8.

If the ASCII command interface has to be active, it has to be switched on in the base parameter tree under \System\Command Interface.

List of additional commands and afterwards the alphabetical list of the commands with detailed information.

ASCII Commands				L	S	M	E
	Value	**Set Command**	**Get Command**	**LinMot**	**Stepper**	**Magnet**	**System**
Get/Set	Cam Length	!MC	!NC	✘	✘		
	Cam Start Point	!MO	!NO	✘	✘		
	Minimal Position	!ML	!NL	✘	✘		
	Maximal Position	!MR	!NR	✘	✘		
	Motor Stroke	!MS	!NS				✘
	FF Reversion	!MF	!NF				✘
	Cam Delay	!MD	!ND	✘	✘		
	Cam Number (Rise Curve Number)	!PR	!GR	✘	✘		
	Counts per Stroke	!MT	!NT				✘
Get	Get wanted Position in Curve/Cam		!GH	✘	✘		
	Get actual Curve/Cam		!GG	✘	✘		
Act	Start Cam	!CU		✘	✘		
	Start Cam from actual Position	!CV		✘	✘		

CU

CU Start Cam	L
Direction	**ASCII sequence**
PC → ELO	'!CU'+ *cam*[uint16] + *drive*[drivecode] + 0x0D
ELO → PC	'#' + *ack*[uint16] + 0x0D

This command changes motor from time locked mode to cam mode. If the motor is already in cam mode and only the cam profile has to be changed, it is better to use the !PR command. The !CU command immediately switches the motor to the cam mode, so if according to the settings (cam start condition, Cam Delay parameter, Cam Length parameter, ...) the motor will jump to the corresponding position.

Value	Min	Max
cam[uint16]	0	63

CV

CV	Start Cam from actual position
Direction	**ASCII sequence**
PC → ELO	'!CV'+ $cam_{[uint16]}$ + $drive_{[drivecode]}$ + 0x0D
ELO → PC	'#' + $ack_{[uint16]}$ + 0x0D

This command changes the motor from time locked mode to cam mode. If the motor is already in cam mode and only the cam profile has to be changed, it is better to use the !PR command. The !CU command immediately switches the motor to the cam mode, so if according to the settings (cam start condition, Cam Delay parameter, Cam Length parameter, ...). In contrast to the !CU command, the motor will start the cam from the actual position.

Note: The parameter Curve Position Offset will be changed.

Value	Min	Max
$cam_{[uint16]}$	0	63

GG

GG	Get actual Cam (Curve) Number
Direction	**ASCII sequence**
PC → ELO	'!GG'+ $drive_{[drivecode]}$ + 0x0D
ELO → PC	'#' + $cam_{[uint16]}$ + 0x0D

This command returns the active cam number. This command can be used to find out when the commanded recipe (!PR) becomes active, so the next cam number can be commanded.

Value	Min	Max
$cam_{[uint16]}$	0	63

GH

GH	Get wanted Cam Position
Direction	**ASCII sequence**
PC → ELO	'!GH'+ $drive_{[drivecode]}$ + 0x0D
ELO → PC	'#' + $pos_{[uint16]}$ + 0x0D

This command returns the actual wanted position in cam mode, according to the encoder position and the selected cam profile. This command can be used for changing from position to time locked mode.

Value	Min	Max
$pos_{[uint16]}$	-32256	32256

GR

GR	Get Cam (Curve) Number
Direction	**ASCII sequence**
PC → ELO	'!GR'+ *drive*$_{[drivecode]}$ + 0x0D
ELO → PC	'#' + *cam*$_{[uint16]}$ + 0x0D

This command returns the value of the **Curve Number** parameter.

Value	Min	Max
delay$_{[uint16]}$	0	63

MC

MC	Set Cam Length
Direction	**ASCII sequence**
PC → ELO	'!MC' + *length*$_{[uint16]}$ + *drive*$_{[drivecode]}$ + 0x0D
ELO → PC	'#' + *ack*$_{[ackcode]}$ + 0x0D

This command sets the **Cam Length (X)** parameter, which defines the motion ration between master encoder and motor movement for cam applications. This command must be used carefully! Because this command takes effect immediately, a running cam will make the motor jumping to its new position. This command is preferably used in the **DISABLE** state.

Value	Min	Max
length$_{[uint16]}$	16	32767

MT

MT	Set Counts per Stroke
Direction	**ASCII sequence**
PC → ELO	'!MT' + counts$_{[uint16]}$ + *drive*$_{[drivecode]}$ + 0x0D
ELO → PC	'#' + *ack*$_{[ackcode]}$ + 0x0D

This command sets the **Counts per Stroke** parameter, which is used in winding configurations. This command is preferably used in the **DISABLE** state.

Value	Min	Max
counts$_{[uint16]}$	64	16384

MD

MD Set Cam Delay	
Direction	**ASCII sequence**
PC → ELO	'!MD' + $delay_{[uint16]}$ + $drive_{[drivecode]}$ + 0x0D
ELO → PC	'#' + $ack_{[ackcode]}$ + 0x0D

This command sets the **Cam Delay (X)** parameter, which defines the number of increments the system will wait after an active edge on the registration input until the cam will be started. This command only takes effect if the encoder application mode is set to "Cam" and the registration input is used.

Value	Min	Max
$delay_{[uint16]}$	0	28672

MF

MF Set FF Reversion	
Direction	**ASCII sequence**
PC → ELO	'!ML' + $value_{[uint16]}$ + $drive_{[drivecode]}$ + 0x0D
ELO → PC	'#' + $ack_{[ackcode]}$ + 0x0D

This command sets the **FF Reversion** parameter, which is used for defining the position controller behavior at the reversion points in the level winding application. See chapter 4.11 "Parameters" for more details.

Value	Min	Max
$value_{[uint16]}$	0	32767

ML

ML Set Minimal Position	
Direction	**ASCII sequence**
PC → ELO	'!ML' + $pos_{[uint16]}$ + $drive_{[drivecode]}$ + 0x0D
ELO → PC	'#' + $ack_{[ackcode]}$ + 0x0D

This command sets the **Minimal Position** parameter, which is used for the lower turn around point in the level winding application.

Value	Min	Max
$pos_{[uint16]}$	-32256	32256

MO

MO Set Cam Start Point	L
Direction	**ASCII sequence**
PC → ELO	'!MO' + *startpoint*$_{[uint16]}$ + *drive*$_{[drivecode]}$ + 0x0D
ELO → PC	'#' + *ack*$_{[ackcode]}$ + 0x0D

This command sets the **Cam Start Point** parameter, which defines the start position of the cam on a registration or a zero indicator event. This command must be used carefully! Because this command takes effect immediately, the motor will jump to its new position. This command is preferably used in the **DISABLE** state. See chapter 4.11 "Parameters" for more details.

Value	Min	Max
startpoint$_{[uint16]}$	0	32767

MR

MR Set Maximal Position	L
Direction	**ASCII sequence**
PC → ELO	'!MR' + *pos*$_{[uint16]}$ + *drive*$_{[drivecode]}$ + 0x0D
ELO → PC	'#' + *ack*$_{[ackcode]}$ + 0x0D

This command sets the **Maximal Position** parameter, which is used for the upper turn around point for level winding applications.

Value	Min	Max
pos$_{[uint16]}$	-32256	32256

MS

MF Set Motor Stroke	L
Direction	**ASCII sequence**
PC → ELO	'!MF' + *stroke*$_{[uint16]}$ + *drive*$_{[drivecode]}$ + 0x0D
ELO → PC	'#' + *ack*$_{[ackcode]}$ + 0x0D

This command sets the **Motor Stroke** parameter, which is used for defining the ratio between **Encoder Counts** and **Motor Stroke** for level winding applications. See chapter 4.11 "Parameters" for more details.

Value	Min	Max
stroke$_{[uint16]}$	64	16384

NC

NC Get Cam Length	
Direction	**ASCII sequence**
PC → ELO	'!NC'+ $drive_{[drivecode]}$ + 0x0D
ELO → PC	'#' + $length_{[uint16]}$ + 0x0D

This command returns the value of the **Cam Length** parameter.

Value	Min	Max
$length_{[uint16]}$	16	32767

ND

ND Get Cam Delay	
Direction	**ASCII sequence**
PC → ELO	'!NC'+ $drive_{[drivecode]}$ + 0x0D
ELO → PC	'#' + $delay_{[uint16]}$ + 0x0D

This command returns the value of the **Cam Delay** parameter.

Value	Min	Max
$delay_{[uint16]}$	0	28672

NF

NF Get FF Reversion	
Direction	**ASCII sequence**
PC → ELO	'!NF'+ $drive_{[drivecode]}$ + 0x0D
ELO → PC	'#' + $value_{[uint16]}$ + 0x0D

This command returns the value of the **FF Reversion** parameter.

Value	Min	Max
$value_{[uint16]}$	0	32767

NL

NL Get Minimal Position	
Direction	**ASCII sequence**
PC → ELO	'!NL'+ $drive_{[drivecode]}$ + 0x0D
ELO → PC	'#' + $pos_{[uint16]}$ + 0x0D

This command returns the value of the **Minimal Position** parameter.

Value	Min	Max
$pos_{[uint16]}$	-32256	32256

NO

NO	Get Cam Start Point
Direction	**ASCII sequence**
PC → ELO	'!NO'+ *drive*[*drivecode*] + 0x0D
ELO → PC	'#' + *startpoint*[*uint16*] + 0x0D

This command returns the value of the **Cam Start Point** parameter.

Value	Min	Max
startpoint[*uint16*]	0	32767

NR

NR	Get Maximal Position
Direction	**ASCII sequence**
PC → ELO	'!NR'+ *drive*[*drivecode*] + 0x0D
ELO → PC	'#' + *pos*[*uint16*] + 0x0D

This command returns the value of the **Maximal Position** parameter.

Value	Min	Max
pos[*uint16*]	-32256	32256

NS

NS	Get Motor Stroke
Direction	**ASCII sequence**
PC → ELO	'!NS'+ *drive*[*drivecode*] + 0x0D
ELO → PC	'#' + *stroke*[*uint16*] + 0x0D

This command returns the value of the **Motor Stroke** parameter.

Value	Min	Max
stroke[*uint16*]	64	16384

NT

NT	Get Counts per Stroke
Direction	**ASCII sequence**
PC → ELO	'!MT' + *drive*[*drivecode*] + 0x0D
ELO → PC	'#' + counts[*uint16*] + 0x0D

This command returns the **Counts per Stroke** parameter.

Value	Min	Max
counts[*uint16*]	64	16384

PR

MC Set Cam (Curve) Number	
Direction	**ASCII sequence**
PC → ELO	‘!PR’ + $cam_{[uint16]}$ + $elo_{[elocode]}$ + 0x0D
ELO → PC	‘#’ + $ack_{[ackcode]}$ + 0x10D

This command sets the **Curve Number** parameter, which defines the next cam number which will be executed. This command can be used to command the recipe over the ASCII interface.

Value	Min	Max
$cam_{[uint16]}$	0	63

4.11 Parameters

The master encoder application needs some additional parameters, which are organized in an application parameter tree. The visibility feature (changing visibility of parameter tree structures depending on the different selections) can not be supported for these parameters. Some parameters of the base software parameter tree may change their function in combination with the master encoder applications. Please see chapter 4.12 “Base Parameters”.

The master encoder specific parameters are subdivided into three parts: encoder parameters, application parameters and I/O parameters.

Encoder Parameters

These parameters are used to specify the hardware of the connected encoder.

Type

This parameter defines the connected encoder type.

\Encoder\Encoder Setup\Type	
Incr. (A B Z)	The encoder type is incremental with a zero signal.
Incr. (A B) with ext. Reg.	The encoder type is incremental without a zero signal. The cam will be started and synchronized with a registration input.
Incr. (A B) Continuous	The encoder type is incremental without a zero track. The cam will be started as soon as the controller is in the RUN state and the cam enable is active (if used). When the cam is finished, it starts again immediately.See chapter “Continuous Cam (Incr. (A B) Continuous or STEP/DIR Continuous)”
STEP/DIR/ZERO	The encoder signals come from a stepper motor interface with a zero signal.
STEP/DIR with ext. Reg.	The encoder signals come from a stepper motor interface. The start of a motion is triggered by a registration input.
STEP/DIR Continuous	The encoder signals come from a stepper motor interface without a zero track. The cam will be started as soon as the controller is in the RUN state and the cam enable is active (if used). When the cam is finished, it starts again immediately. See chapter “Continuous Cam (Incr. (A B) Continuous or STEP/DIR Continuous)”
Absolute SSI	Only for ME01-0X/08 modules: The encoder type is absolute (SSI).
Absolute SSI with ext. Reg.	Only for ME01-0X/08 modules: The encoder type is absolute (SSI). The movement of the motor will be started with a registration input.

Configuration Incr.

In the **Configuration Incr.** directory are the parameters which define the incremental encoder more precisely. This directory is only relevant if the encoder type is set to **Incr. (A B Z)** or **Incr. (A B) with ext. Reg.**

\Encoder\Encoder Setup\Configuration Incr.	
Counts/Rev.	Defines the number of counts per revolution the encoder has. Range: 16 .. 32767
Decode	This parameter specifies the decode mode of the encoder. There are the following possibilities: 1x, 2x and 4x decode. See the figure below.
Zero Offset	This parameter is to compensate the difference between the machine and the encoder zero positions. This parameter is only relevant for encoders with a zero track.

This figure shows how the different decode modes generate the increments out of the signals A and B.

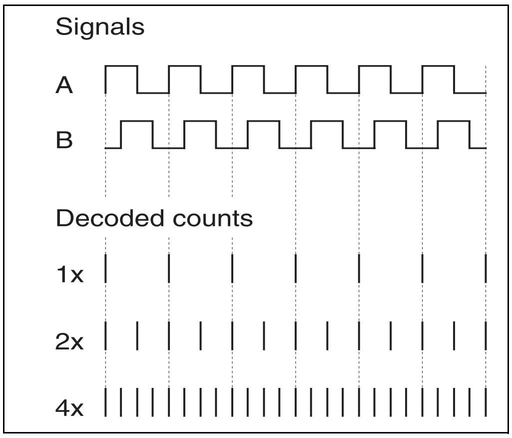

Figure 4-10: Decoding of incremental signals

Position Format

Only for ME01-0X/08 modules: In the **Position Format** directory are the parameters which define the number of bits transferred from the absolute encoder via SSI and how many of these bits will be processed. This directory is only relevant if the encoder type is set to **Absolute SSI** or **Absolute SSI with ext. Reg.**

\Encoder\Encoder Setup\Configuration Abs.\Position Format	
Encoder Readout Width	This parameter defines how many position bits have to be clocked in from the absolute encoder. The most significant bit of the position will be clocked first. Range: 2 .. 24
Processed Width	This parameter defines how many bits of the clocked in position will be processed. See the figure below. Range: 2 .. 15

The following figure shows how the position defining bits are processed.

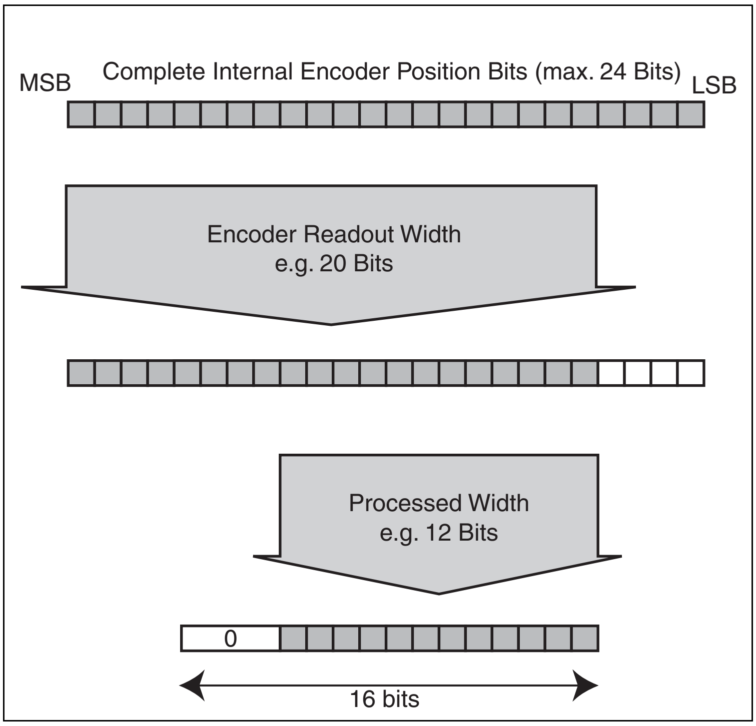

Figure 4-11: Position Format

Example

As a short practical example, an absolute SSI encoder is assumed, which has 24 position bits (12 bits revolution counter and 12 bits resolution in one revolution). A defined cam has to be run each two revolutions with 1024 encoder counts each revolution. The parameters have to be set to:

Encoder Readout Width = 21 Bits

Processed Width = 10 Bits

Position Coding

This parameter defines the coding of the serially transmitted position of the absolute encoder. This parameter is only to be set for absolute encoders.

\Encoder\Encoder Setup\Configuration Abs.\Position Coding	
Gray Code	The coding of the serially clocked in position is in Gray code.
Binary Code	The coding is binary.

Encoder Application Parameters

These parameters define the behavior of the master encoder application.

Mode

This parameter defines the application type.

\Encoder\Encoder Application\Mode	
Cam	The motor(s) controlled by the *LinMot*® unit run a predefined motion profile as a cam. Depending on the selected encoder type, the start and synchronization is controlled by the zero or registration inputs.
Parallel Winding	This application type will wind bobbins with the parallel winding pattern. This mode can only be used with incremental encoder types.
Cross Winding	This application type will wind bobbins with a cross winding pattern. This mode can only be used with incremental encoder types.

Cam Config

The parameters in this directory are to define the cam function. This directory is only used for cam applications (**Encoder Application\Mode** set to **Cam**). The “X” stands for the motor letters A to D.

\Encoder\ Encoder Application\Cam Config	
Cam Length (X) L	This parameter defines over how many encoder counts the cam for the motor is defined. Thus, the time information of the cam in the *curve editor* is not relevant. See the figure below.
Cam Start Point (X) L	This parameter defines the start point in the cam. E.g. if **Cam Length (X)** is set to 1000 and **Cam Start Point (X)** is set to 250, the cam will start at 90 degrees, run to 360, wrap around and finish up to 90 degrees.See the figure below.
Cam Delay (X) L	This parameter defines the delay in encoder counts, which the motor will wait on a registration event until the cam will be started.

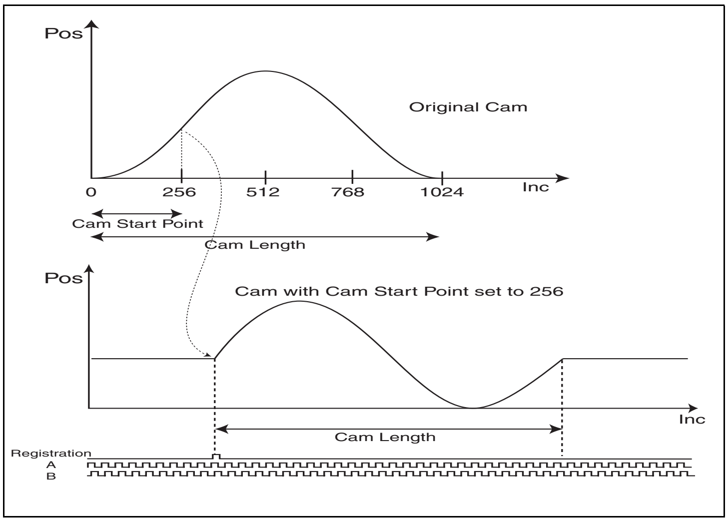

Figure 4-12: Parameters Cam Length (X) and Cam Start Point (X)

Winding Config

In this directory are the parameters for the level winding application. This directory is only relevant if the encoder application mode is set to **Parallel Winding** or **Cross Winding**.

\Encoder\ Encoder Application\Winding Config	
Motor Stroke [1]	This parameter works together with the **Counts per Stroke** parameter. It defines the motor stroke per encoder counts. See figure below. Range: 64 .. 16384
Counts per Stroke	In parallel winding applications this parameter works together with the **Motor Stroke** parameter. It defines the number of encoder counts per motor stroke. See figure below. In cross winding applications this parameter defines the number of encoder increments it will take the motor to move from one turn around point to the other. Range: 256 .. 16384
FF Reversion L	This parameter is used to optimize the motion of the *LinMot*® at the turn around points. See the figure below. Range: 0 .. 32767

1) Relevant only if the **Encoder Application Mode** is set to **Parallel Winding**.

The following figure shows the meaning of the **Motor Stroke** and **Counts per Stroke** parameters for parallel winding applications. The movement ratio between motor and encoder is defined as the quotient (Motor Stroke / Counts per Stroke). Please note: there is one degree of freedom for the selection of the two parameters. It is exactly the same if it is selected (20mm / 1000counts) or (100mm / 5000counts) e.g.

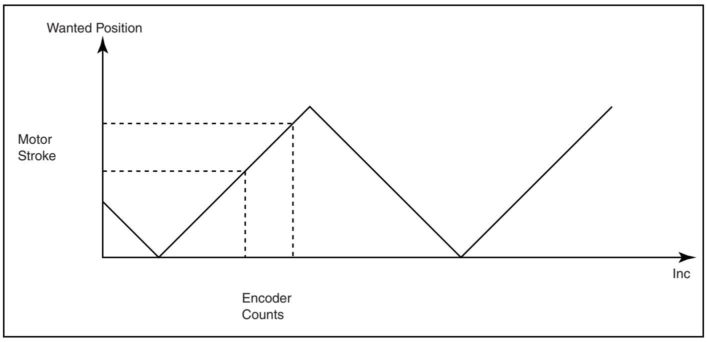

Figure 4-13: Meaning of Motor Stroke and Counts per Stroke

The two following figures show the effect of the **FF Reversion** parameter. If this parameter is set to zero the movement of the linear motor could look like this:

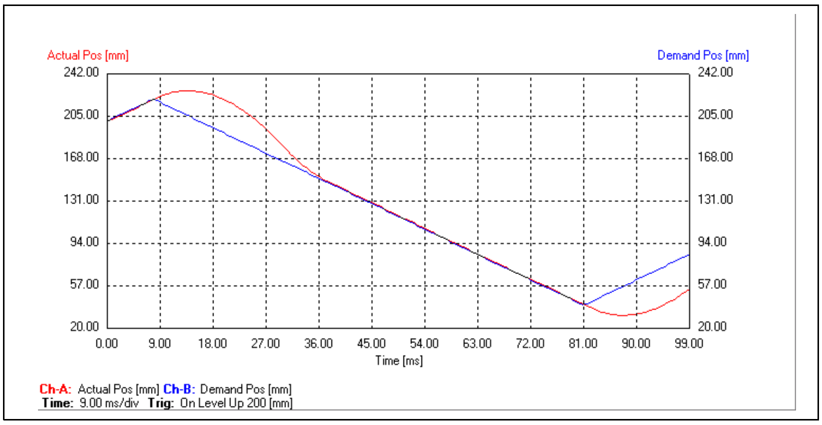

Figure 4-14: Motion with FF Reversion = 0

By setting the **FF Reversion** parameter the overshooting and the large position error can be minimized. It is necessary to perform some practical tests to find out the optimal value of **FF Reversion**.

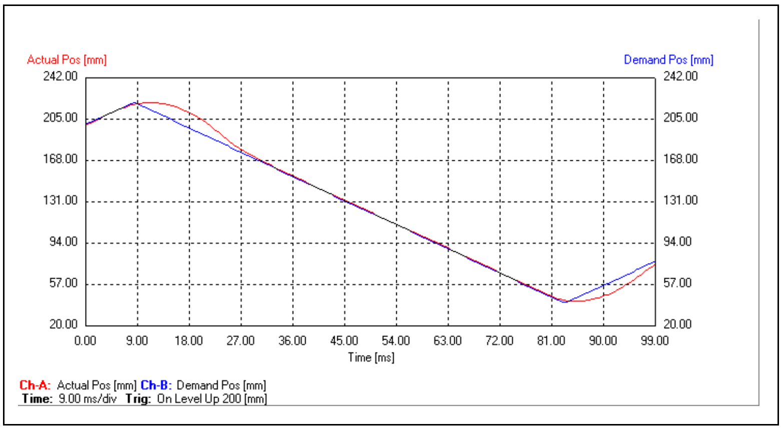

Figure 4-15: Motion with FF Reversion = 32

Master Encoder on Motors

Encoder on Motors

These on/off parameters define which motors will run in the position locked mode. All not selected motors will run the time locked mode.

\Encoder\Master Encoder on Motors\	
Motor A	Motor A runs in the cam mode, if selected.
Motor B	Motor B runs in the cam mode, if selected.
Motor C	Motor C runs in the cam mode, if selected.
Motor D	Motor D runs in the cam mode, if selected.

Each item is only visible if the corresponding motor is selected.

High Resolution Motors

High Resolution Motors

This on/off parameters defines if the motor will run in the high resolution registration mode. This mode is for high speed applications where it is necessary to start a movement exactly on the hardware start strobe even without any software synchronization. This mode is only for incremental encoder applications without zero track. In this release 1.4.1ME the high resolution mode is only supported for motor A.

See also chapter 4.6 “Digital Inputs”.

If any of the motors run in the high resolution mode, make sure the input location is set to **Exchanged.**

\Encoder\Encoder on Motors\	
Motor A	Motor A runs in the high resolution mode.
Motor B	Motor B runs in the high resolution mode.
Motor C	Motor C runs in the high resolution mode.
Motor D	Motor D runs in the high resolution mode.

Each item is only visible if the corresponding motor is selected.

Inputs

Location

This parameter defines the mapping of the digital inputs. Because of the high resolution input feature there are two mappings:

\Encoder\Inputs\Input Location	
Normal	The mapping is as describe in chapter 4.6 “Digital Inputs”.
Exchanged	The **Trigger Input Motor X** and **Cam Enable Motor X** signals are exchanged.

Cam Enables

These on/off parameters define which cam start enable inputs are turned on.

\Encoder\Inputs\Cam Enables	
Cam Enable Motor A	Cam enable input for motor A is read in.
Cam Enable Motor B	Cam enable input for motor B is read in.
Cam Enable Motor C	Cam enable input for motor C is read in.
Cam Enable Motor D	Cam enable input for motor D is read in.

Each item is only visible if the corresponding motor is selected.

The following figure shows the logic how the cam is started:

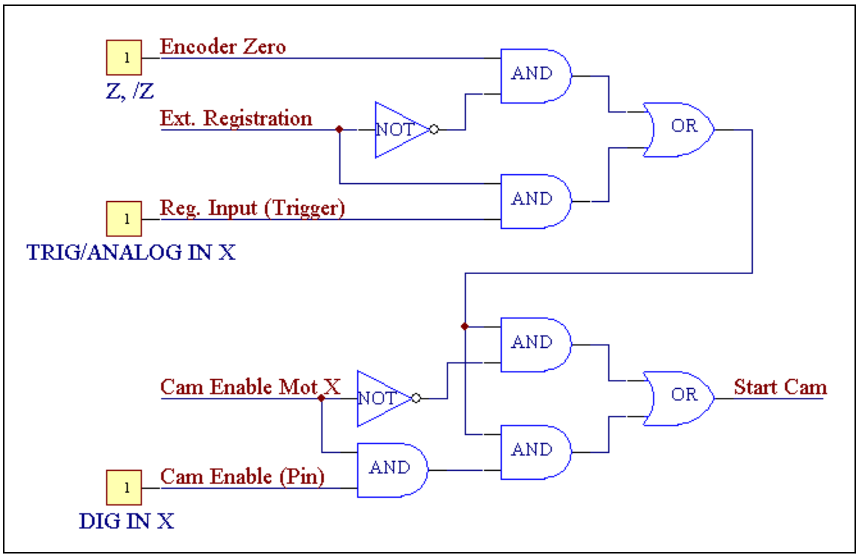

Figure 4-16: Cam Start Logic

Outputs

These parameters are used to define the behavior of the digital outputs.

Function

This parameter is to select the signal source for the output. The “Y” stands for the numbers 1 to 8 and the “X” is for the motor letters A to D. Please note: The **Encoder Position** and the **Cam Angle Mot. X** are not available on each output (**Encoder Position** is on Output 1, 3, 5 and 7, **Cam Angle Mot. X** on 2, 4, 6, 8).

\Encoder\Outputs\Output Y\Function	
None	The output is disabled.
Mot. X In Position	The output is high, if the corresponding motor is in position, which is defined by the **In Position-** and **In Position+** parameters.
Mot. X Pos. Range	The output is low, if the motor is in the position range defined by **Pos Range Min** and **Pos Range Max** in the base parameter tree. Please note: Outputs 1 and 2 can be mapped to motor A, outputs 3 and 4 to motor B etc.
Mot. X Following Error	The output is high, if a following error has occurred. The following error generation can be configured by the parameters **Following Error-** and **Following Error+** in the base parameter tree.
Encoder Position	The output is high, if the encoder position is within the count range defined by the parameters **Encoder Position Min** and **Encoder Position Max** under \Encoder\Outputs\Encoder Position Config. This choice is only selectable for the outputs 1, 3, 5 and 7
Mot X Cam Angle	The output is high, if the cam angle is within the count range defined by the parameters **Cam Angle Min** and **Cam Angle Max** under \Encoder\Outputs\Cam Angle Config. This choice is only selectable for the outputs 2, 4, 6 and 8.

Encoder Position Config

The parameters in this directory, which only exists for the outputs 1, 3, 5 and 7, are used to define the switching points of the digital output if the output function is set to **Encoder Position**. Between these two points the output is active (high).

\Encoder\Outputs\Output Y\Encoder Position Config	
Encoder Position Min L	Defines the encoder position at which the digital output Y switches on.
Encoder Position Max L	Defines the encoder position at which the digital output Y switches off.

Cam Angle Config

The parameters in this directory, which only exists for the outputs 2, 4, 6 and 8, are used to define the switching points of the digital output if the output function is set to **Mot. X Cam Angle**. Between these two points the output is active (high).

\Encoder\Outputs\Output Y\Cam Angle Config	
Cam Angle Min L	Defines the cam position at which the output Y switches on.
Cam Angle Max L	Defines the cam position at which the output Y switches off.

4.12 Base Parameters

In this section there will be the parameters discussed which have a changed function in the master encoder context.

Set Value configuration

In the two winding applications the **Minimal Position** and **Maximal Position** parameters define the reversal points. Please note: In contrast to other application modes these two parameters are not live.

\Drives\Drive X\Set Value Generation\Set Value Configuration	
Minimal Position	Determines the minimal position for the motor (lower limit of positioning range). In the level winding application it defines the lower turn around point.
Maximal Position	Determines the maximal position for the motor (upper limit of positioning range). In the level winding application it defines the upper turn around point.
Curve Number L	In the cam application the Curve Number parameter defines the cam number. This value will be overwritten if recipes are used.

5. Commissioning

This section deals with the steps necessary for successfully commissioning a *LinMot®* system for master encoder module applications.

5.1 Installing the *LinMot®* Master Encoder Software

The master encoder software is delivered as an *install package* on diskette. This install package contains everything needed to operate *LinMot®* in the master encoder mode.

Procedure for installing the master encoder software is as follows:

1 Install release R1.3.16 on your PC. (The software can be downloaded from the internet www.linmot.com)

2 Only for ME01-0X/08 modules: Mount the master encoder add on module on the *LinMot®* servo controller.Otherwise use a EX0X1-ME controller type.

3 Provide a serial link between PC and servo controller.

4 Supply servo controller and encoder module with power.

5 Start *LinMot®*-Talk 1.3.16.

6 Select “Install Package” from “Special” menu.

7 Log in with ID: “Install” and password: “NTI”.

8 In the appearing file dialogue, select the installation package Install.ipk under \LM1R3R16\Firmware\ for EX0X1-ME controllers or ME1R4R11.ipk under \LM1R3R16\Firmware\ME for ME01-0X/08 modules.

9 Wait until the installation is completed.

10 Setup the parameters (Encoder Type, Encoder Application, ...) according to chapter 4.11 “Parameters”.

5.2 Quick Start up Guide for Cross Winding Application

This section is a guide for a quick start up of a cross winding application for one axis. The encoder type in this example is assumed to be incremental, 5000 increments per revolution.

The movement ratio between linear motor and encoder increments is set in this case to 20mm per 5000 increments.

The following items show step by step which setting have to be done.

General System

Please set up the system like for a normal AT system. Select the correct drive type and set up its parameters. Possibly the command interface selection and the I/O configuration have to be changed.

Set Value Generation

The **Run Mode** has to be set to **Continuous Curve**.

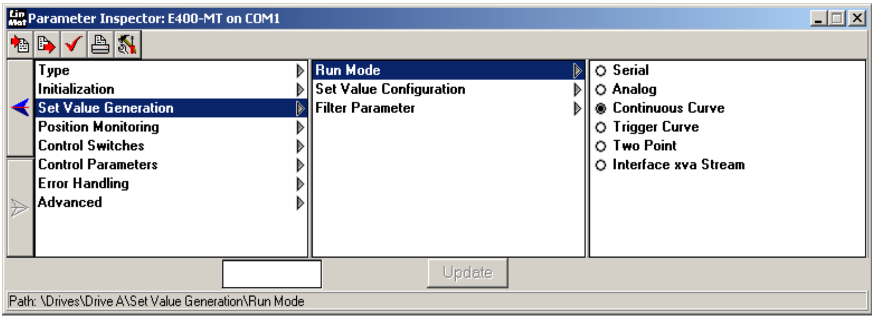

Figure 5-1: Set Value Generation

Set Value Configuration

The two parameters **Min Position** and **Max Position** are used in the cross winding application to define the two turn around points.

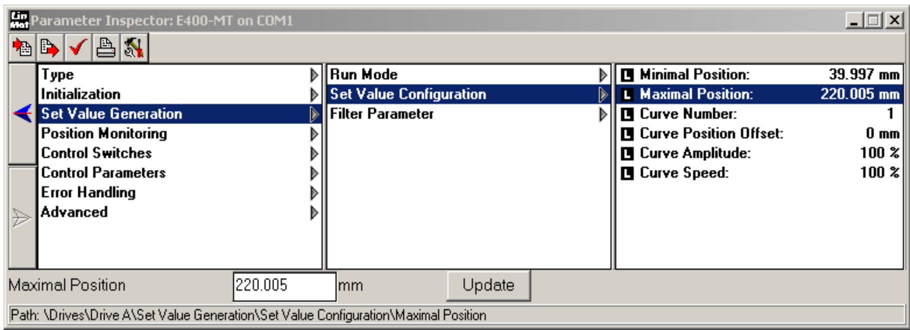

Figure 5-2: Set Value Configuration

Encoder Type

Under **Encoder\Encoder Setup\Type** the encoder has to be selected: **Incr. (A B) with ext. Reg.**

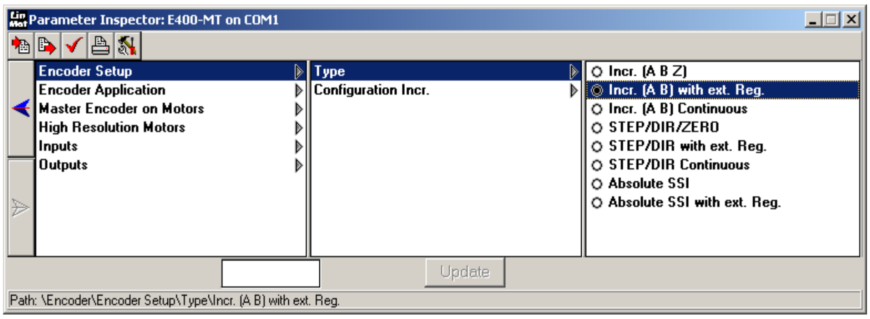

Figure 5-3: Encoder Type Selection

Encoder Cunfiguration

The decode mode (**Encoder\Configuration Incr.\Decode**) is to set to **1X Decode** for this application.

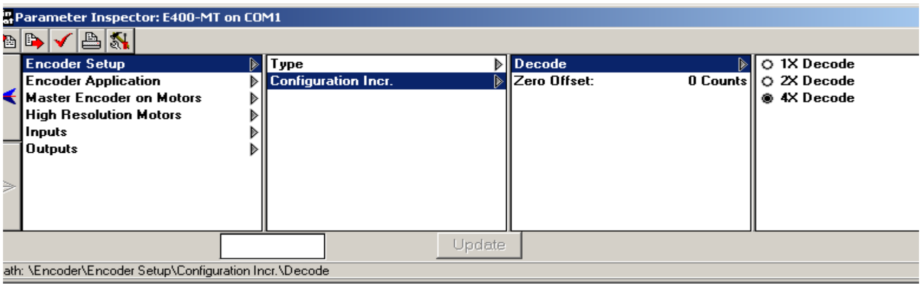

Figure 5-4: Encoder Configuration

Application Mode

In this example, the application is cross winding. Thus, **Cross Winding** is to be selected under **Encoder\Encoder Application\Mode.**

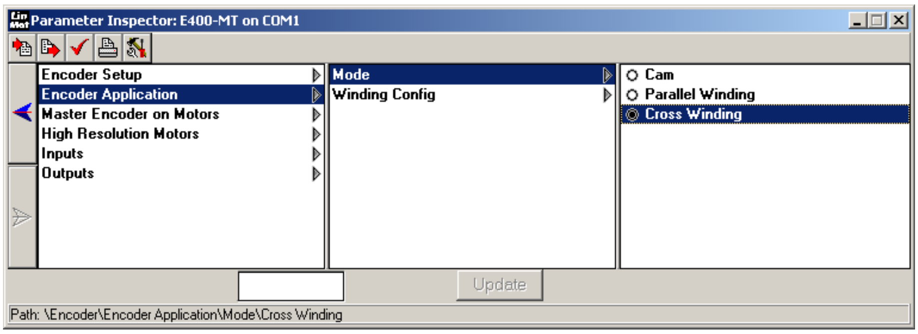

Figure 5-5: Application Mode

Winding Configuration

Under **Encoder\Encoder Application\Winding Config** the movement control of the motor is set up. With the parameters **Counts per Stroke** the ratio between motor speed and encoder speed is defined. In this example we set it to 5000 counts. The **FF Reversion** parameter influences the motor behavior at the turn around points, we set it to 32 mm/(m/s).

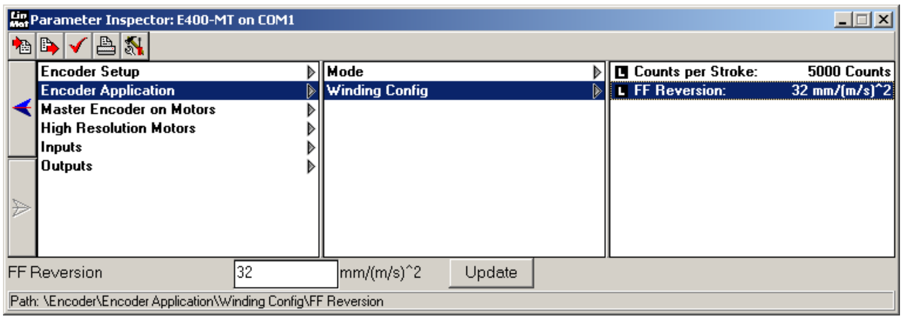

Figure 5-6: Winding Configuration

6. Contact Addresses

SWITZERLAND

NTI AG
Haerdlistr. 15
CH-8957 Spreitenbach

Sales and Administration:	office@linmot.com
Tech. Support:	+41-(0)56-544 71 00 support@linmot.com
Fax:	+41-(0)56-419 91 92
Web:	http://www.linmot.com

USA

LinMot, Inc.
5750 Townline Road
Elkhorn, WI53121

Sales and Aministration:	877-546-3270 262-743-2555
Tech. Support:	877-804-0718 262-743-1284
Fax:	800-463-8708 262-723-6688
E-Mail:	us-sales@linmot.com
Web:	http://www.linmot-usa.com

- Please visit http://www.linmot.com/ to find the distributor near you.

Index