Contents

How to Read these Operating Instructions

☐ How to Read these Operating Instructions

These Operating Instructions will help you get started, install, program, and troubleshoot your Motion Control Option MCO 305. Please read these operating instructions in full and, in order to be able to work with the system safely and professionally, particularly observe the hints and cautionary remarks.

Chapter **How to Read these Operating Instruction** introduces the manual and informs you about the approvals, symbols, and abbreviations used in this literature.

Page divider for 'How to Read these Operating Instructions'.

Chapter **Safety Instructions and General Warnings** entail instructions on how to handle the MCO 305 correctly.

Page divider for 'Instructions and General Warnings'.

Chapter **How to Install** informs you how to start up the FC 300 with the 'MCO 305 step-by-step'. This section offers you a quick general introduction which includes turning on the controller and familiarizing yourself with it, as well as starting operation of the control unit with the test programs provided and the most important basic settings.

Page divider for 'How to Install'.

__ How to Read these Operating Instructions __

Chapter **Optimizing the PID Controller** provides all about the PID control parameters and how to optimize the controller during one or more test runs, for example, in order to achieve better positioning results or shorter cycle times. Detailed instructions on how to proceed can be found in the section 'Ten steps for optimum control'.

Chapter **How to Install the Application** informs about Download, Backup, and Restore features.

Page divider for 'How to Optimize the PID Controller' and 'How to Install the Application'.

Chapter **General Specifications** entails technical data about MCO 305.

Page divider for 'General Specifications'.

Chapter **Troubleshooting** assists you in solving problems that may occur when installing MCO 305.

Page divider for 'Troubleshooting'.

Chapter **Appendix** provides information about parameters in clearly arranged lists. Please see the Parameter Reference in the MCO 305 DesignGuide for more details.

□ **Available Literature for FC 300, MCO 305 and MCT 10 Motion Control Tool**

–
– The MCO 305 Design Guide entails all technical information about the option board and customer design and applications.
– The VLT® AutomationDrive FC 300 Operating Instructions provide the necessary information for getting the drive up and running.
– The VLT® AutomationDrive FC 300 Design Guide entails all technical information about the drive and customer design and applications.
– The VLT® AutomationDrive FC 300 Profibus Operating Instructions provide the information required for controlling, monitoring and programming the drive via a Profibus fieldbus.
– The VLT® AutomationDrive FC 300 DeviceNet Operating Instructions provide the information required for controlling, monitoring and programming the drive via a DeviceNet fieldbus.
– The VLT® AutomationDrive FC 300 MCT 10 Operating Instructions provide information for installation and use of the software on a PC.

Danfoss Drives technical literature is also available online at www.danfoss.com/drives.

__ How to Read these Operating Instructions __

☐ Approvals
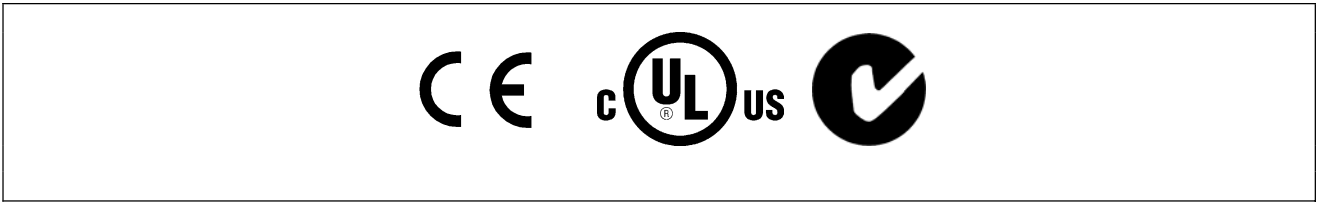
☐ Symbols and Conventions

Symbols used in this manual:

NB!:
Indicates something to be noted by the reader.

Indicates a general warning.

Indicates a high-voltage warning.

*   Indicates default setting.

Conventions

The information in this manual follows the system and uses the typographical features described below to the greatest extent possible:

Menus and Functions

Menus and functions are printed italics, for example: *Controller → Parameters*.

Commands and Parameters

Commands and parameter names are written in capitals, for example: AXEND and KPROP; Parameters are printed in italics, for example: *Proportional factor.*

Parameter Options

Values for use to select the parameter options are written in brackets, e.g. [3].

Keys

The names of keys and function keys are printed in brackets, for example the control key [Cntl] key, or just [Cntl], the [Esc] key or the [F1] key.

__ How to Read these Operating Instructions __

❑ Abbreviations

Automatic Motor Adaptation	AMA
Direct Current	DC
Frequency Converter	FC
Local Control Panel	LCP
Master Unit	MU
Millisecond	ms
Minute	min
Motion Control Option	MCO
Motion Control Tool	MCT
Parameter	par.
Protective Extra Low Voltage	PELV
Position Control Loop	PID
Pulses per Revolution	PPR
Quad-counts	qc
Revolutions per Minute	RPM
Second	s
Switch normally closed	nc
Switch normally open	no
User Unit	UU
Volts	V

Safety Instructions and General Warnings

MCO 305

Operating Instructions
Software version: 4.xx

CE   c(UL)us   C

These Operating Instructions can be used for the MCO 305 option board
with all FC 300 frequency converters with software version 4.0x.

The software version number can be seen from parameter 15-43.

□ **High Voltage Warning**

The voltage of the FC 300 is dangerous whenever the converter is connected to mains.

Incorrect fitting of the motor or FC 300 may cause damage to the equipment, serious injury or death. Consequently, it is essential to comply with the instructions in this manual as well as local and national rules and safety regulations.

At altitudes above 2 km, please contact Danfoss Drives regarding PELV.

□ **Safety Instructions**

–   Make sure the FC 300 is properly connected to earth.
–   Do not remove mains plugs or motor plugs while the FC 300 is connected to mains.
–   Protect users against supply voltage.
–   Protect the motor against overloading according to national and local regulations.

__ Safety Instructions and General Warnings __

–   Motor overload protection is not included in the default settings. To add this function, set par. 1-90 *Motor thermal protection* to value *ETR trip* or *ETR warning.* For the North American market:
–   ETR functions provide class 20 motor overload protection, in accordance with NEC.
–   The earth leakage current exceeds 3.5 mA.
–   The [OFF] key is not a safety switch. It does not disconnect the FC 300 from mains.

□ Before Commencing Repair Work

1. Disconnect FC 300 from mains.
2. Disconnect DC bus terminals 88 and 89.
3. Wait at least 4 minutes.
4. Remove motor plugs.

□ Avoid Unintended Start

While FC 300 is connected to mains, the motor can be started/stopped using digital commands, bus commands, references or via the LCP.

–   Disconnect the FC 300 from mains whenever personal safety considerations make it necessary to avoid unintended start.
–   To avoid unintended start, always activate the [OFF] key before changing parameters.
–   Unless terminal 37 is turned off, an electronic fault, temporary overload, a fault in the mains supply, or lost motor connection may cause a stopped motor to start.

□ Safe Stop of FC 302

The FC 302 can perform the Designated Safety Function *Uncontrolled Stopping* by removal of power (as defined by draft IEC 61800-5-2) or *Stop Category 0* (as defined in EN 60204-1). It is designed and approved suitable for the requirements of Safety Category 3 in EN 954-1. This functionality is called Safe Stop.

In order to install and use the Safe Stop function in accordance with the requirements of Safety Category 3 in EN 954-1, the related information and instructions of the FC 300 Design Guide MG.33.BX.YY must be followed! The information and instructions of the Operating Instructions are not sufficient for a correct and safe use of the Safe Stop functionality!

□ General Warning

 Warning:

Touching the electrical parts may be fatal – even after the equipment has been disconnected from mains.

Also make sure that all voltage inputs have been disconnected, such as load-sharing (linkage of DC intermediate circuit), as well as the motor connection for kinetic back-up.

Using VLT AutomationDrive FC 300 (at and below 7.5 kW):

wait at least xx minutes.

xx = The waiting time using the VLT AutomationDrive FC 300 is different for different sizes: Please see period of waiting time xx in the FC 300 Operating Instructions.

How to Install

▫ How to Get Started

Please consult the FC 300 Operating Instructions (MG.AX-XX) for correct installation of FC 300, this instruction covers installation of MCO 305.

The installation procedure depends on whether the MCO 305 was delivered as an option for upgrading an existing FC 300 or if it was preinstalled before delivery. Skip the next chapter (Mechanical installation) if the MCO 305 was preinstalled:

▫ Initialization to Default Settings

Initialize the frequency converter to default setting in two ways:

Recommended initialization (via par. 14-22):

1. Select par. 14-22
2. Press [OK]
3. Select "Initialization"
4. Press [OK]
5. Cut off the mains supply and wait until the displays turns off.
6. Reconnect the mains supply – the frequency converter is now reset.

NB!
MCO 305 programs and arrays are not affected.

Par. 14-22 initializes all except:	
14-50	*RFI 1*
8-30	*Protocol*
8-31	*Address*
8-32	*Baud Rate*
8-35	*Minimum Response Delay*
8-36	*Maximum Response Delay*
8-37	*Max Inter-char Delay*
15-00 to 15-05	Operating Data
15-20 to 15-22	Historic log
15-30 to 15-32	Fault log

__ How to Install __

Manual initialization

1. Disconnect from mains and wait until the display turns off.
2. Press [Status] – [Main Menu] – [OK] at the same time:
3. Reconnect mains supply while pressing the keys.
4. Release the keys after 5 s.
5. The frequency converter is now programmed according to default settings.

This method initializes all except:	
15-00	*Operating Hours*
15-03	*Power-up's*
15-04	*Over temp's*
15-05	*Over volt's*

NB!:

When you carry out manual initialization, you also reset serial communication and fault log settings.

And all MCO 305 programs and arrays are deleted!

□ System Overview

□ Mechanical Installation

(Only relevant if the MCO 305 is delivered as option for upgrading an existing FC300)
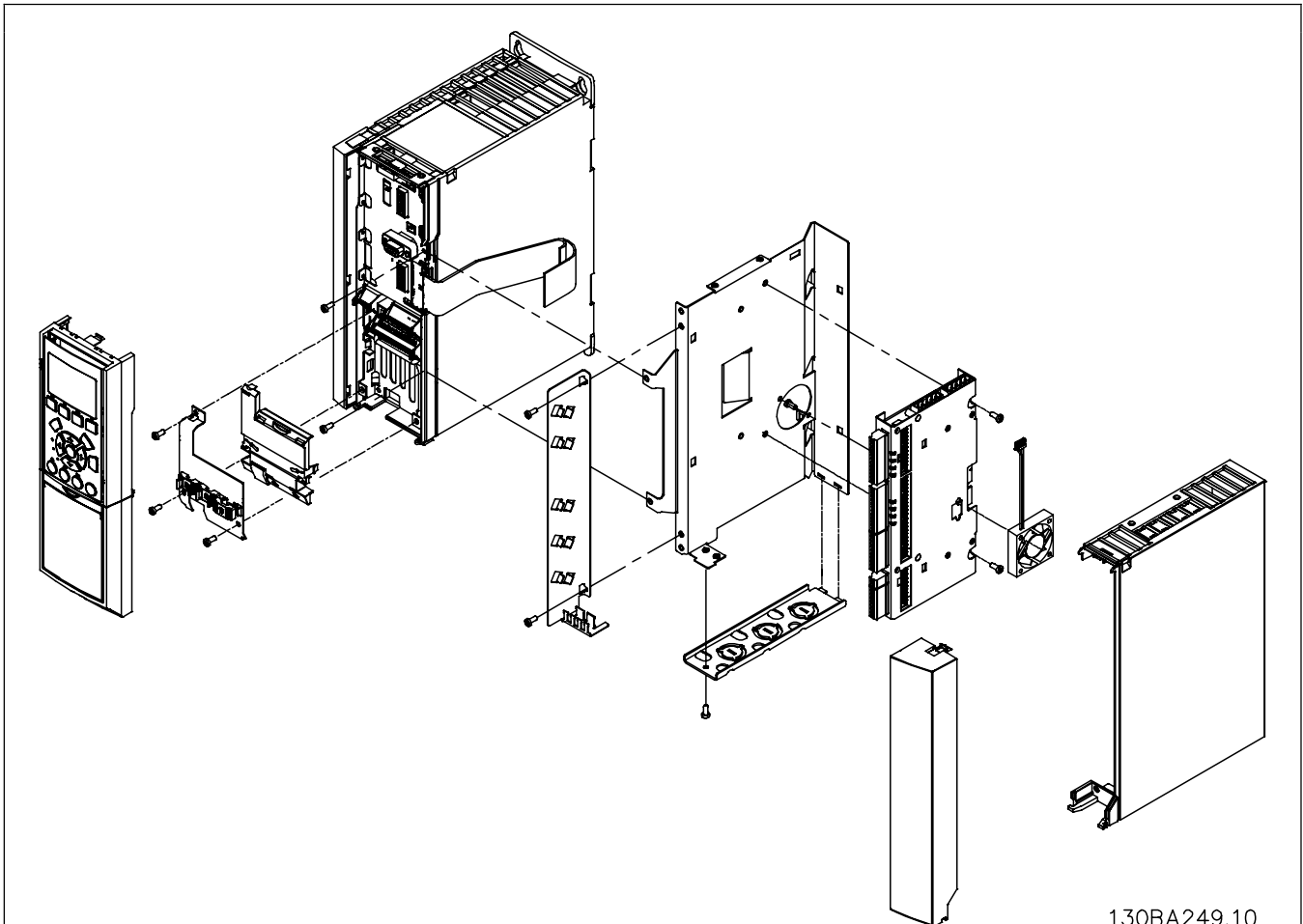
130BA249.10

Frame size A2 and A3

130BA250.10
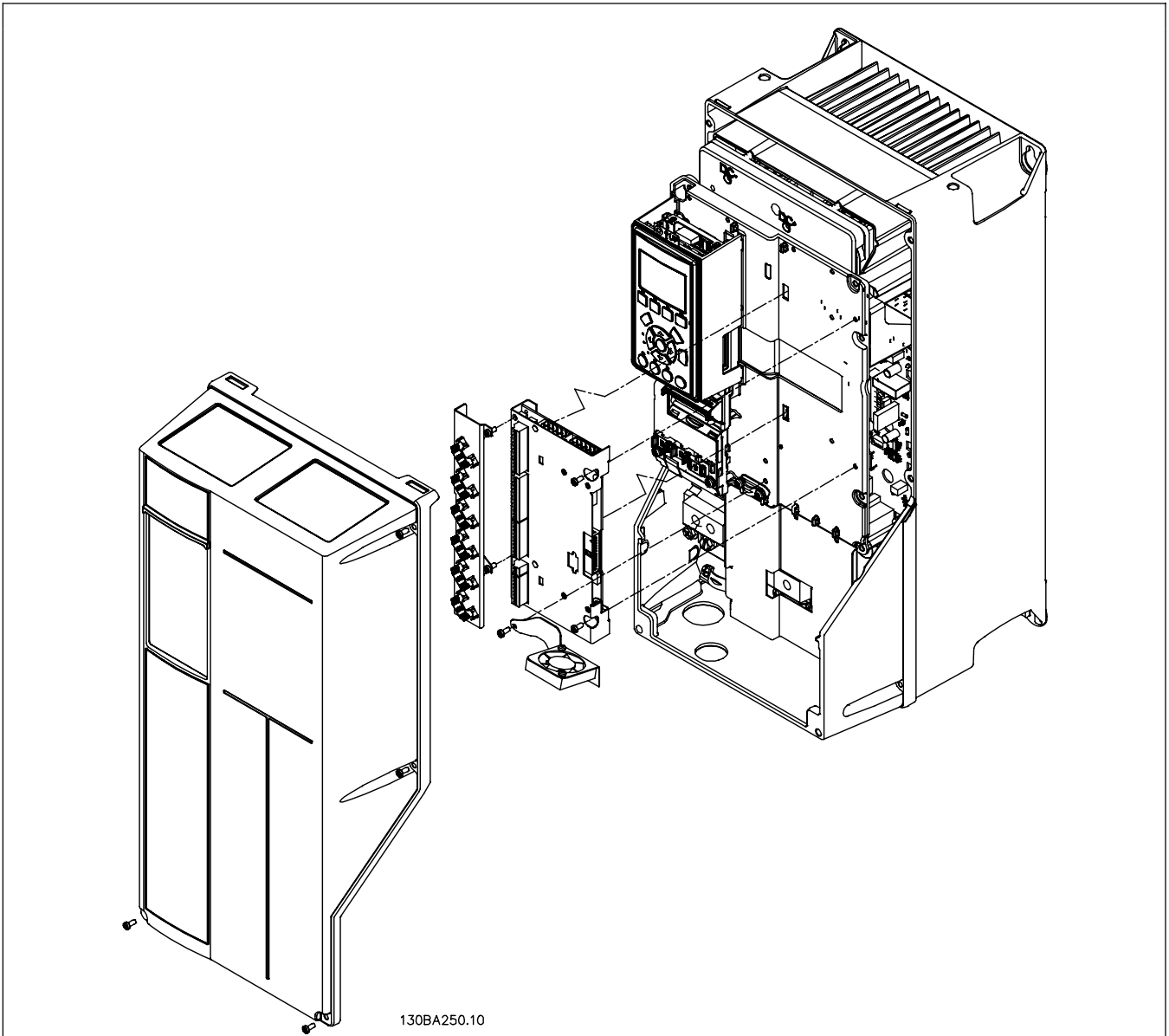
Frame size A5, B1 and B2

☐ **Accessory Bag**

Accessory bag A2 and A3 enclosures:
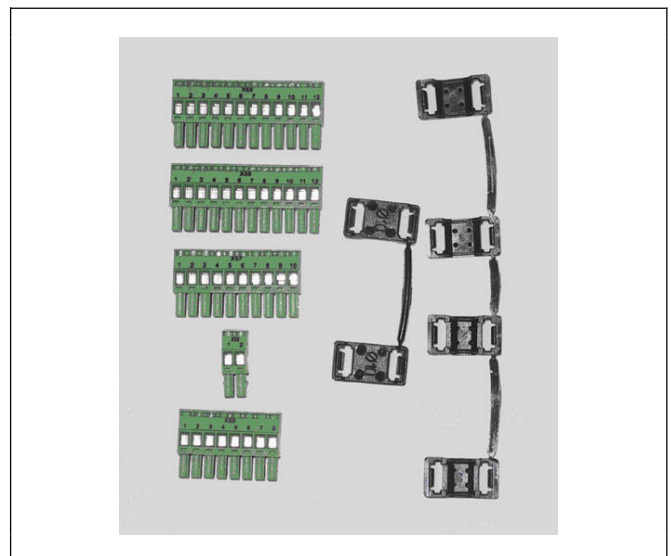
__ How to Install __

Installation kit A5 enclosure:
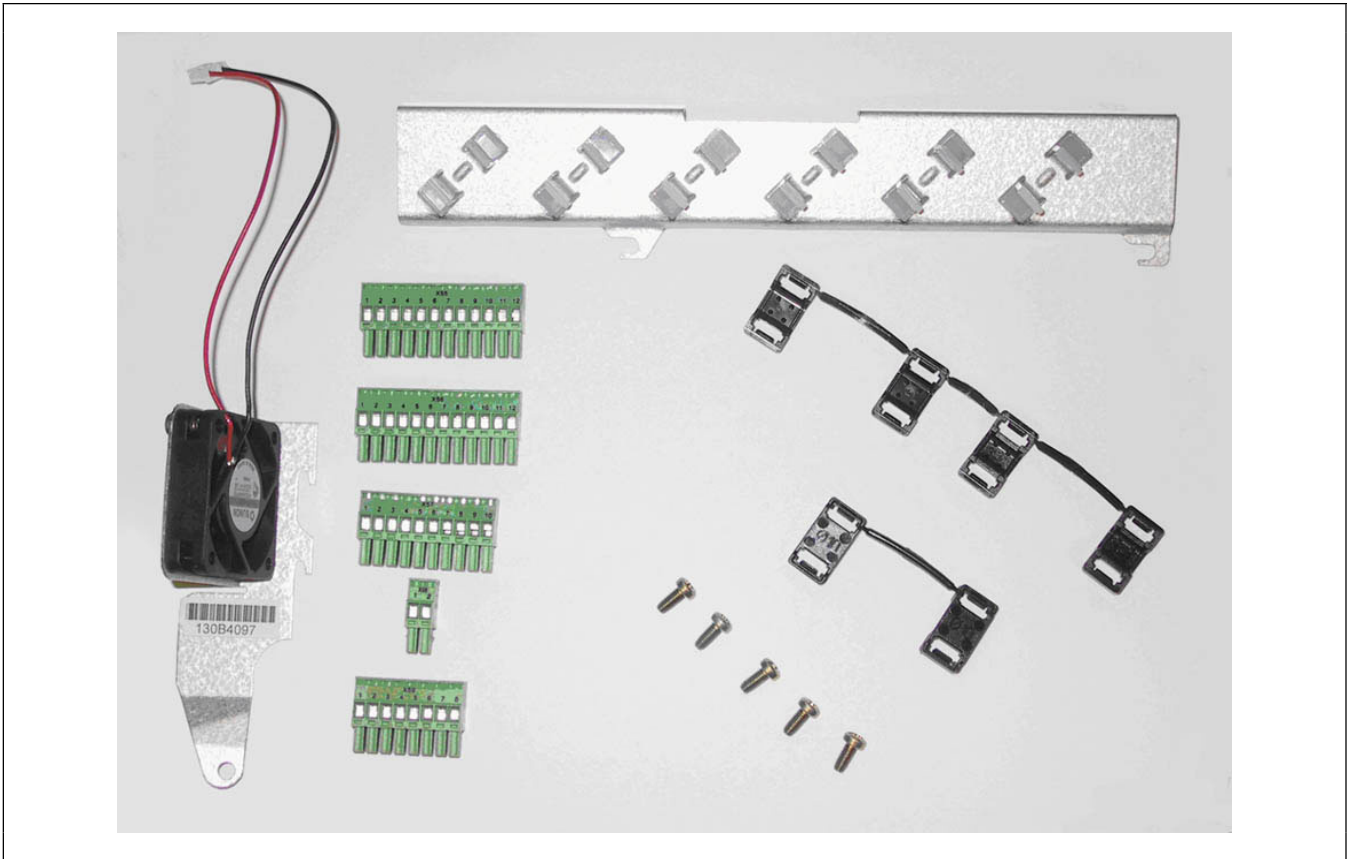
Installation kit B and C enclosure:
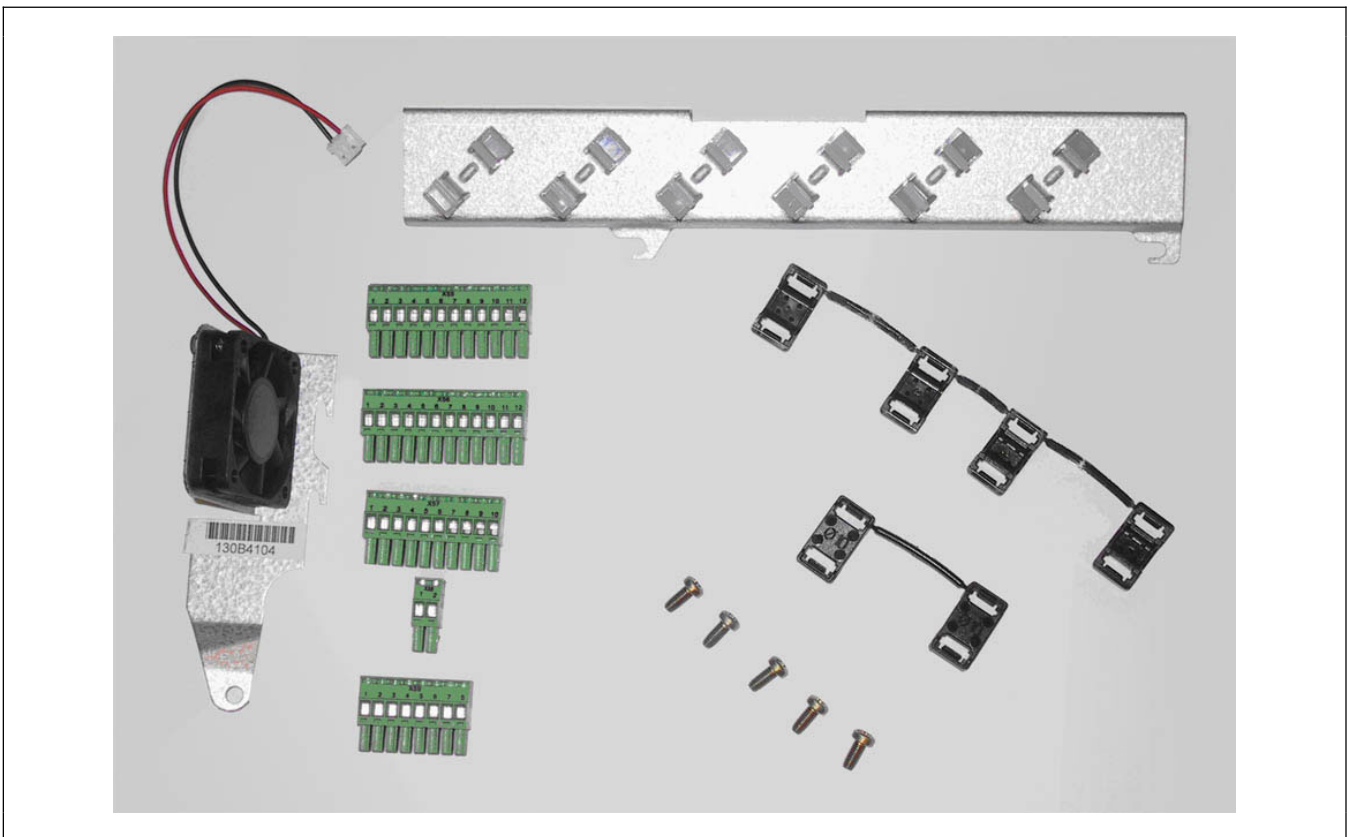
__ How to Install __

□ Electrical Installation

All control cables must be screened and the cable screen must be connected to earth at both ends. It is very important to use the correct cable for encoders to avoid noise problems, always follow the prescriptions of the encoder supplier. See also FC 300 Design Guide (MG.33.AX.XX) for more information regarding EMC correct cable installation.

□ **MCO 305 Control Terminals**

Frame size A2 and A3:

All MCO 305 terminals are located behind the C-option terminal cover, see illustration.

130BA248.10
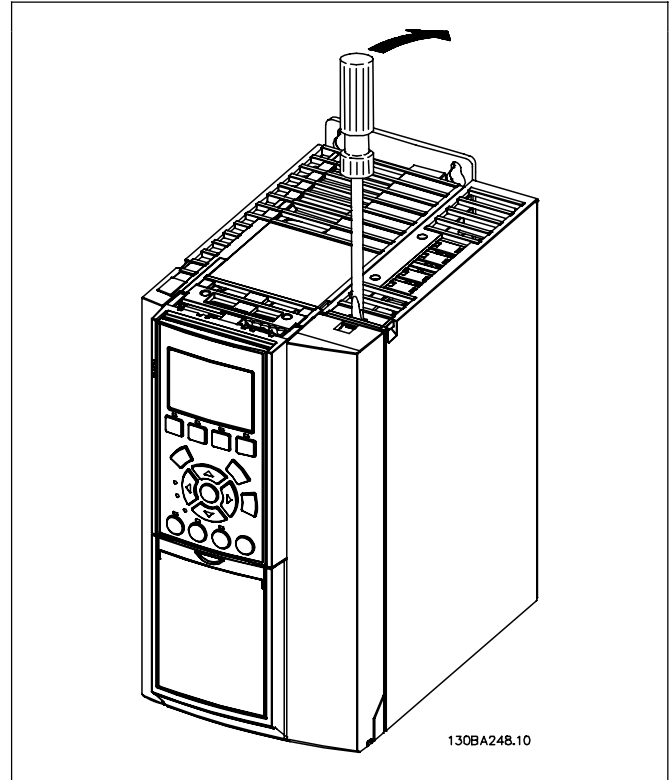
Frame size A5, B1 and B2:

All MCO 305 terminals are located next to the FC 300 control card. Remove the front cover to get access, see illustration.

130BT334
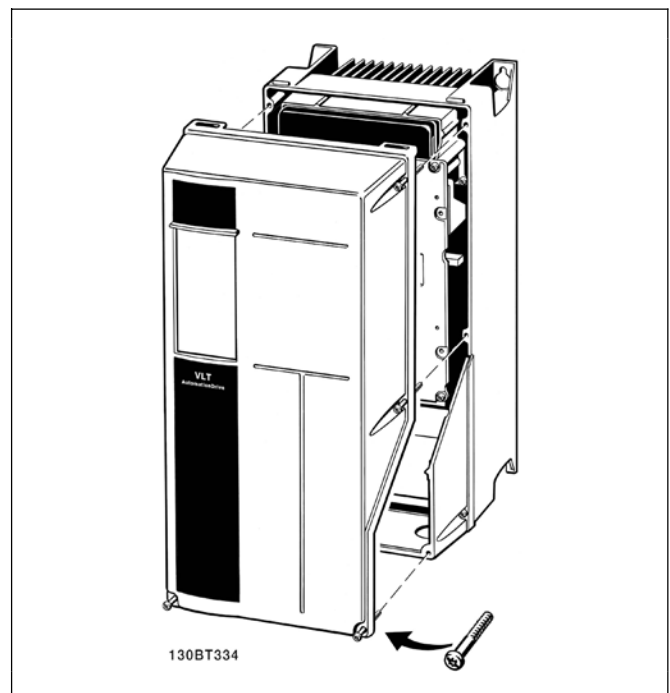
__ How to Install __

MCO 305 control terminals are plug connectors with screw terminals; the terminal blocks are duplicated to enable use of the same MCO 305 in all frame sizes. See illustration to locate the terminal blocks:

(1) is used with frame sizes A2 and A3,

(2) is used with frame sizes A5, B1 and B2:

   X55 = Encoder 2

   X56 = Encoder 1

   X57 = Digital inputs

   X58 = 24VDC supply

   X59 = Digital outputs
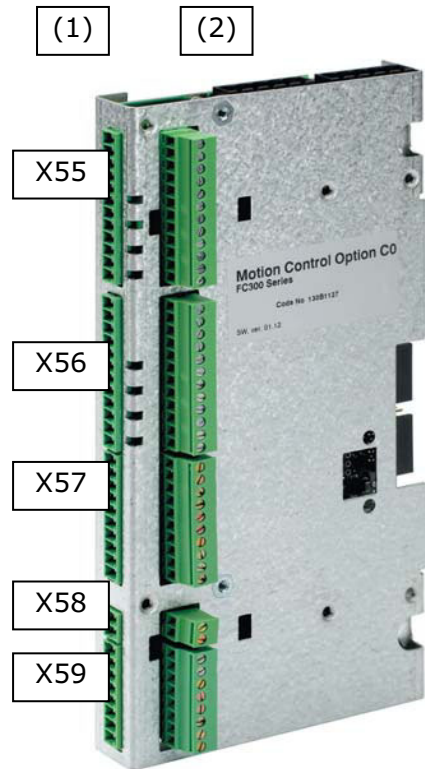
__ How to Install __

□ Terminal overview

Terminal block	Terminal number	Descriptive Name Encoder 2 (Feedback)
X55	1	+24V Supply
	2	+8V Supply
	3	+5V Supply
	4	GND
	5	A
	6	A not
	7	B
	8	B not
	9	Z / Clock
	10	Z not / Clock not
	11	DATA
	12	DATA not

Terminal block	Terminal number	Descriptive Name Encoder 1 (Master)
X56	1	+24V Supply
	2	N/A (not available)
	3	+5V Supply
	4	GND
	5	A
	6	A not
	7	B
	8	B not
	9	Z / Clock
	10	Z not / Clock not
	11	DATA
	12	DATA not

Terminal block	Terminal number	Descriptive Name Digital inputs
X57	1	Digital Input
	2	Digital Input
	3	Digital Input
	4	Digital Input
	5	Digital Input
	6	Digital Input
	7	Digital Input
	8	Digital Input
	9	Digital Input
	10	Digital Input

Terminal block	Terminal number	Descriptive Name Supply
X58	1	+24V Supply
	2	GND

Terminal block	Terminal number	Descriptive Name Digital outputs
X59	1	Digital Output/Input
	2	Digital Output/Input
	3	Digital Output
	4	Digital Output
	5	Digital Output
	6	Digital Output
	7	Digital Output
	8	Digital Output

__ How to Install __

☐ Connection Examples

5V incremental encoder (RS422) supplied by MCO 305:

5V incremental encoder
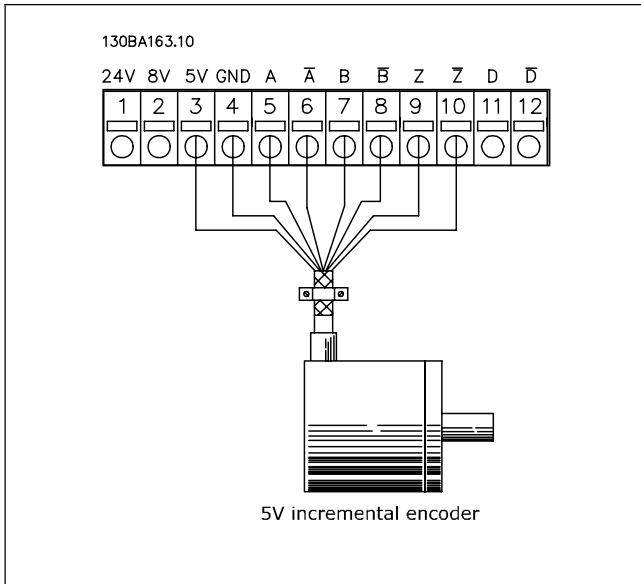
5 V incremental encoder (RS422) supplied by an external power source:

5V incremental encoder
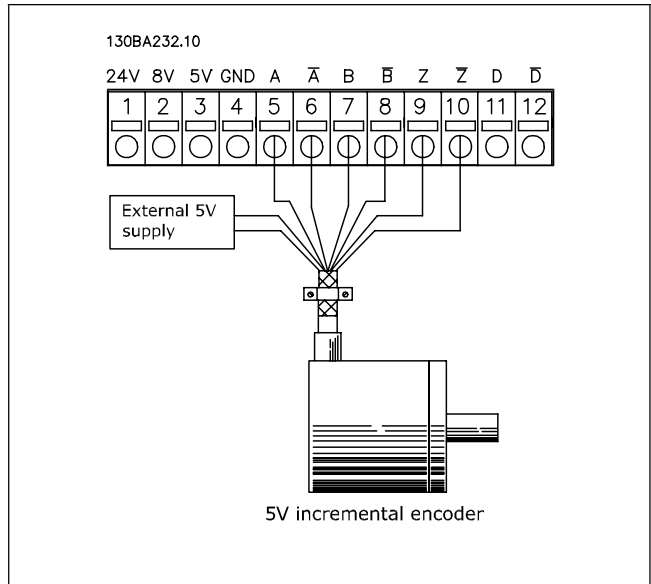
24 V absolute encoder (SSI) supplied by MCO 305:

24V absolute encoder
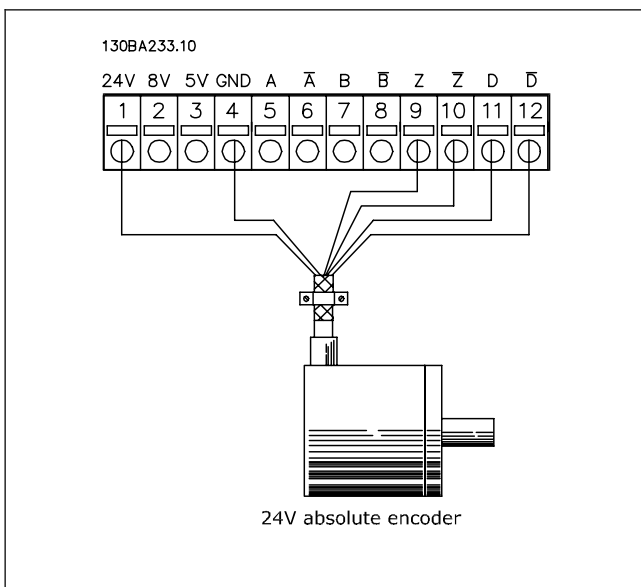
24 V absolute encoder (SSI) supplied by an external power source:

24V absolute encoder
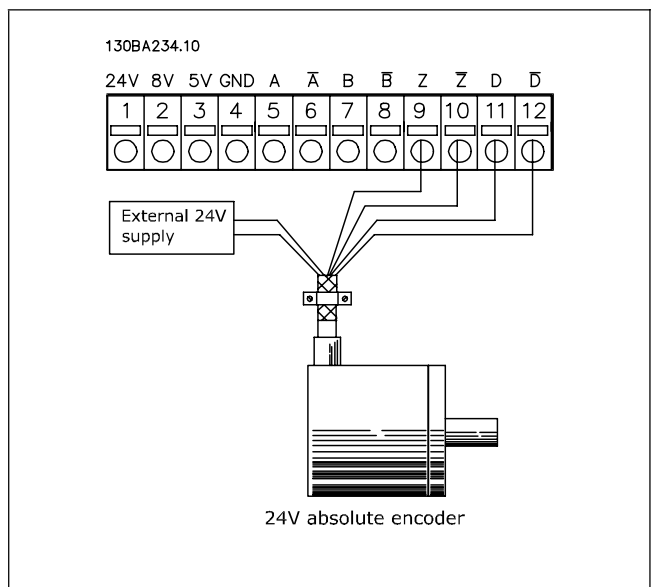
__ How to Install __

Virtual master connection:

When more than 2 MCO 305 master encoder interfaces are interconnected termination must be on (par. 32-40 = ON) at both ends of the bus. Termination must be off (par. 32-40 = OFF) in all other MCO's.
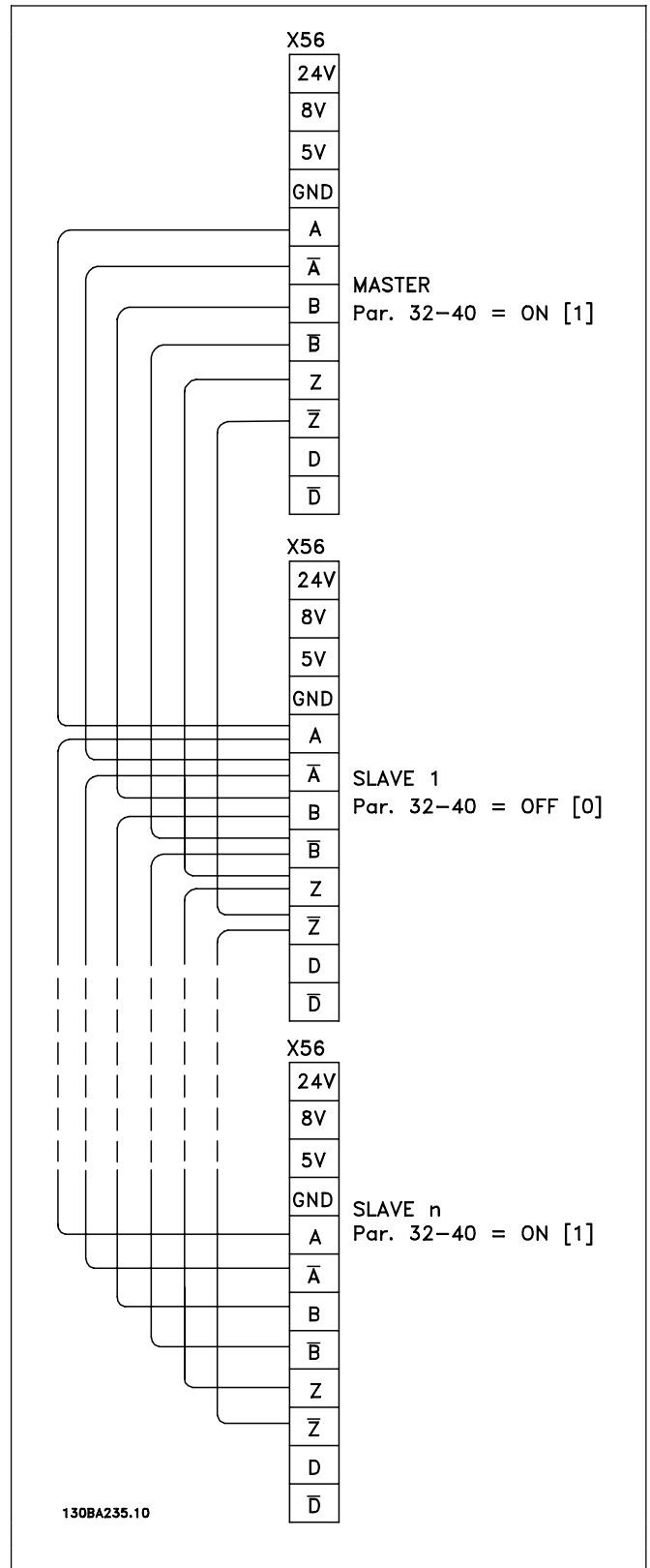
__ How to Install __

◻ Commissioning Guide

This guideline is only describing basic configuration and optimizing of MCO 305, before starting the following steps must be completed:

1. Basic commissioning of FC 300. NOTE: FC 300 must be optimized and have good control over the motor and application before commissioning MCO 305! Guidelines can be found in FC 300 Operating Instructions (MG.33.AX.XX).
2. Install the PC software tools (MCT 10 and APOSS)
3. Establish communication between PC and FC 300 using USB, RS485 or PROFIBUS DPV1.

Then follow these steps to set-up and optimize MCO 305:

1. Setting of basic MCO 305 parameters (groups 32-** and 33-**).
2. Connect and test encoder(s).
3. Optimize PID controller.

NB!:
Error 113 will show up right after power up if the FC 300 is not ready. The FC 300 is in the "Not ready" state when:

– it has an alarm (trip),
– it is in hand mode, [Hand on] key,
– local LCP stop is activated, [OFF] key,
– there is no signal on input 37 safe stop (only FC 302).

Error 113 can only be reset using the ERRCLR command or with BREAK [Esc] in the PC software and only when the FC 300 is in the "ready" state, which means none of the above may be true.

The FC 300 monitoring function can be switched off by selecting [0] in par. 33-82 *Drive Status Monitoring*.

Read the Safety Instructions before installing the drive.

◻ Basic MCO Parameter Settings

The following parameters must always be checked and if necessary adjusted. Depending on the require-ments of the application it might be necessary to adjust other parameters as well.

For the other parameters you can use the default settings at first and then optimize the controller as needed at a later point in time with a *Testrun*.

The MCT 10 Set-up Software Tool will support you to set the parameters in Local Control Panel LCP.

For a detailed description of all parameters and there options please refer to the chapter Parameter Reference.

Once the parameter settings are complete, it is recommended to save the settings.

__ How to Install __

□ Setting of Encoder Parameters

Define the used type of encoder for the slave in parameter group 32-0*.

These defaults are set:

32-0*	Encoder 2 - Slave	Default
32-00	Incremental Signal Type	[1] RS422
32-01	Incremental Resolution	1024
32-02	Absolute Protocol	[0] None
32-03	Absolute Resolution	8192
32-05	Absolute Encoder Data Length	25
32-06	Absolute Encoder Clock Frequency	262000
32-07	Absolute Encoder Clock Generation	[1] On
32-08	Absolute Encoder Cable Length	0
32-09	Encoder Monitoring	[0] Off

The following parameters are only relevant for synchronizing applications.

These defaults are set:

32-3*	Encoder 1 - Master	
32-30	Incremental Signal Type	[1] RS422
32-31	Incremental Resolution	1024
32-32	Absolute Protocol	[0] None
32-33	Absolute Encoder Resolution	8192
32-35	Absolute Encoder Data Length	25
32-36	Absolute Encoder Clock Frequency	262000
32-37	Absolute Encoder Clock Generation	[1] On
32-38	Absolute Encoder Cable Length	0
32-39	Encoder Monitoring	[0] Off
32-40	Encoder Termination	[1] On

□ Setting of Velocity Parameters

Define par. 32-80, the *Maximum Velocity* of the shaft where the encoder is mounted in RPM.

32-8*	Velocity & Acceleration	Default
32-80	Maximum Velocity (Encoder)	1500
32-81	Shortest Ramp	1.000

NB!:
For synchronizing the setting must be at least the same as the maximum velocity of the master in order to be able to synchronize. For position synchronizing it must even be higher so that the slave can catch up lost position related to the master. All velocity commands (VEL, CVEL) are related to this value.

Define par. 32-81 *Shortest Ramp.* That is the time from 0 to maximum velocity and from maximum velocity to 0. All acceleration and deceleration commands (ACC, DEC) are related to this value.

__ How to Install __

☐ Setting of Home Parameters

Homing is not necessary in standard synchronization applications and applications using an absolute encoder.

When using an incremental encoder the controller must be run to home after being switched on. During this process the reference switch defines the position at which 0 is located and how the FC 300 functions during a home run: input depends on the application.

Define par. 33-03 *Velocity of Home Motion:* Home speed is entered in % relative to the maximum speed of the drive.

33-0*	Home Motion	Default
33-00	*Force HOME*	[0] not forced
33-01	*Zero Point Offset from Home Position*	0
33-02	*Ramp for Home Motion*	10
33-03	*Velocity of Home Motion*	10
33-04	*Behavior during Home Motion*	[0] Reverse + Index

☐ Setting of Synchronization Parameters

The following parameters are only relevant for synchronizing applications:

Par. 33-10, 33-11: *Synchronizing Factor Master* and *Slave*

The *Synchronizing Factors Master* and *Slave* must be set according to the gear ratio between master and slave encoder.

Example:

Both encoders have 1024 PPR, master is running 305 RPM and slave must run 1220 RPM.

par. 33-10 *Syncfactor Master*	= 305 and	
par. 33-11 *Syncfactor Slave*	= 1220	

Alternative:

par. 33-10 *Syncfactor Master*	= 1	
par. 33-11 *Syncfactor Slave*	= 4	

The following parameters are only relevant when using synchronizing with marker correction (SYNCM):

Par. 33-15, 33-16: *Marker Number for Master* and *for Slave*

Marker Number for Master and *Slave* must be set according to the ratio between the number of marker signals from master and slave. A ratio of 1:1 means that each slave marker will be aligned with each master marker. A ratio of 2:1 means that each slave marker will be aligned with each second master marker.

33-1*	Synchronization	Default
33-10	*Synchronization Factor Master (M:S)*	1
33-11	*Synchronization Factor Slave (M:S)*	1
33-15	*Marker Number for Master*	1
33-16	*Marker Number for Slave*	1
33-17	*Master Marker Distance*	4096
33-18	*Slave Marker Distance*	4096
33-19	*Master Marker Type*	[0] Enc. Z pos.
33-20	*Slave Marker Type*	[0] Enc. Z pos.

Par. 33-17, 33-18: *Master* and *Slave Marker Distance*

When using the encoder zero pulse as marker signal the distance between 2 markers is the resolution (qc) of the encoder.

When external marker signals are used, the marker distance can be measured by means of the program sample "Marker count" (see chapter program samples) if it is unknown.

__ How to Install __

<u>Par. 33-19, 33-20:</u> *Master Marker* and *Slave Marker Type*

Master marker signal: Input 5
Slave marker signal: Input 6

Marker signal type must be selected for master and slave: Default setting is [0] Encoder Z positive flank for both.

☐ Connecting and Testing the Encoder(s)

NB!:
The controller and the motor must be able to be switched off at any time with an EMERGENCY STOP button.

The motor must be able to turn completely freely so that a sudden jolt can not cause damage.

If you have not yet done so, now is the time to connect and test the encoder. If you are using an absolute encoder, enter [0] in par. 32-00 and in par. 32-02 the used type of encoder. Then enter the resolution in par. 32-03.

NB!:
Remember to turn off the power before connecting the encoder.
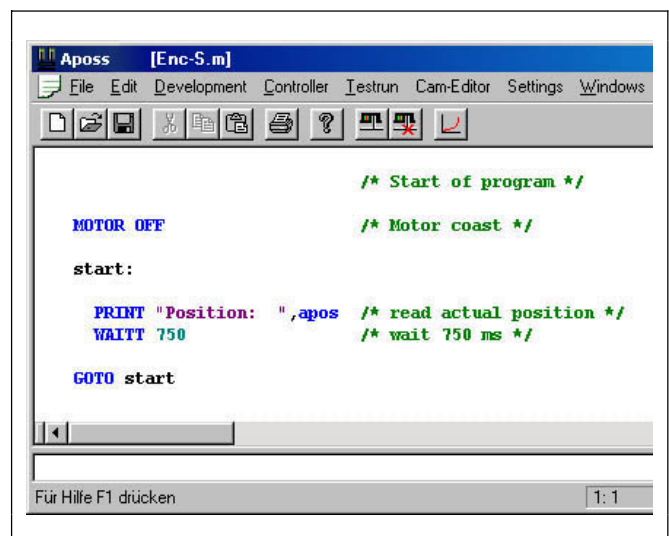
Each encoder channel (A, B, Z, and D) is monitored, open circuit as well as short circuit is detected. A green LED shows the status of each active channel, LED on means that the channel is ok. Monitoring if the channels are automatically enabled/disabled depending on the selected encoder type, e.g. A, B, and Z is monitored when an incremental encoder is selected. Additionally an alarm will be trigged if encoder monitoring is activated in parameters 32-09 and 32-39.

Check the encoder connections by means of the test program: Select the file "Enc-S.m" via the MCT 10. APOSS and with that the file is opened automatically .

☐ Execute the Encoder Test Program

In the *Development* menu click on → *Execute* to start the test program. Run the drive forward for example in–hand mode – [Hand on] key – then the position must count positive. If the position is counting negative you must swap A and B channels from the encoder or two motor phases.

If you turn the motor by hand (the motor should not be connected!), you can test whether the encoder functions: the position is continuously registered in the communications window. For a full rotation you should receive 4 times the value of the resolution of the encoder, that means 2000 if the *Encoder Counts per Revolution* is 500.

__ How to Install __

□ Checking Master Encoder for Synchronizing Applications

To check the master encoder change the test program: replace the command APOS by MAPOS in "Enc-S.m" and run the master forward then the master position must count positive as well. If the position is counting negative you must swap A and B channel from the master encoder.

□ Ending the Encoder Check

End the test of the encoder with the [Esc]-key and close the test program with *File → Close*. A successful test of the encoder is a requirement for further starting up of operations.

Repeat the test when using other encoders.

□ What to do, if the Encoder ... ?

What to do, if the encoder doesn't work?

This could be a result of incorrect cable installation. Measure the signals coming from the encoder and compare them to the values listed in the specifications. Check whether the connection was made according to the specification.

What to do, if the count is descending?

If the Count is Descending with an incremental encoder, exchange the feedback encoder track A by B and A/ by B/. If there is no counting in the display then check the wiring of the encoder.

□ PID Controller Settings

This is a quick guide to set-up the PID controller, further details can be found in the chapter "Optimizing the PID controller".

The following basic settings can be used in most applications with a 1024 PPR encoder connected directly to the motor shaft and maximum reference (parameter 3-03) is 1500 RPM:

Parameter 32-60 *Proportional Factor*	= 200
Parameter 32-61 *Derivative Value*	= 1000
Parameter 32-62 *Integral Factor*	= 5
Parameter 32-65 *Velocity Feed-forward*	= 10200

Velocity Feed-forward (par. 32-65) can also be calculated using the following formula:

$$Velocity\ Feed\ forward = \frac{62914560000}{envel \times eres \times tsample}$$

Where:

– "envel" is encoder velocity in RPM at maximum reference (parameter 3-03).

– "eres" is the encoder resolution
(Incremental encoder: eres = par. 32-01 * 4.
Absolute encoder: eres = par. 32-03.).

– "tsample" is PID sample time
(tsample = par. 32-69 except with SYNCV where tsample = par. 32-70).

The sample program "Feed-forward calculation" is making the above calculation and sets the parameters:

□ Sample Program: Feed-forward Calculation

This program calculates basic settings for Velocity Feed-forward and PID settings based on the following inputs: Encoder type (par. 32-00 and 32-02), Encoder resolution (par. 32-01 or 32-03), Encoder velocity (par. 19-00), and Motion type (par. 19-00). These parameters must thus be set before starting the calculation.

The program performs the following steps when setting input 1:

__ How to Install __

1. Check encoder type and resolution
2. Calculate PID sample time in order to get sufficient resolution
3. Check motion type and select sample time
4. Calculate Feed-Forward.

Parameters are updated when setting input 2.

The updated parameters can be saved by setting input 3.

New calculation is enabled by setting input 4.

```
/***************************************************************************/
// Inputs:     1       Start calculation
//             2       Set parameters
//             3       Save updated parameters
//             4       Enable new calculation
//             8       Clear Error
// Outputs:    1       Calculation done
//             2       Parameters updated
//             8       Error
/****************************** Interrupts **********************************/
ON ERROR GOSUB errhandle
    /* In case of error go to error handler routine, this must always be included */
/********************** Define application parameters  **********************/
LINKGPAR 1900 "Encoder velocity" 0 1073741823 0
    // Specify encoder velocity in RPM at maximum reference (par. 303)
LINKGPAR 1901 "Motion type" 0 1 0
    // Feed-Forward calculation is different for SYNCV compared to all other motions:
    // Select "1" to make the calculation for SYNCV applications, select "0" for all other applications.
/********************** Initializing flags and outputs  **********************/
calculation_done = 0
update_done = 0
save_done = 0
OUT 1 0
OUT 2 0
/*************************** main loop ************************************/
MAIN:
IF (IN 1 == 1) AND (calculation_done == 0) THEN        // Calculate once when input 1 is high
    GOSUB calculation                                  // Go to calculation routine
ELSEIF (IN 2 == 1)AND(update_done == 0)AND(calculation_done == 1) THEN
    // Update par. once when in 2 is high and calc. done
    GOSUB update_parameters                            // Go to parameter update routine
ELSEIF (IN 3 == 1) AND (update_done == 1) AND (save_done == 0) THEN
    // Save parameters once when in 3 high and parameters updated
    GOSUB save_par
ELSEIF (IN 4 == 1)AND(IN 1 == 0)AND(IN 2 ==0) THEN
    // Enable new calculation when input 4 is high and input 1/2 low.
    calculation_done = 0        // Reset flag
    update_done = 0             // Reset flag
    save_done = 0               // Reset flag
    OUT 1 0                     // Reset output
    OUT 2 0                     // Reset output
ENDIF
GOTO MAIN
/********************** Sub programs start ********************************/
SUBMAINPROG
```

__ How to Install __

```
/************************* Calculation *****************************************/
SUBPROG calculation
   IF (GET ENCODERTYPE != 0) THEN              // Check incremental encoder type
      eres = (4 * GET ENCODER)
      // Encoder resolution (par. 3201) multiplied by 4 (qc) when incremental encoder
   ELSEIF (GET ENCODERABSTYPE != 0) THEN       // Check absolute encoder type
      eres = (GET ENCODERABSRES)               // Absolute encoder resolution (par. 3203).
   ENDIF
   new_timer = 1 + (6000000 rnd ((GET 1900) * eres))       // Calculation of PID sample time.
   IF (GET 1901 == 1) THEN                      // Check motion type
      tsample = ((GET PROFTIME) rnd 1000)
      // tsample must be PROFTIME (par. 3270) when SYNCV is selected
   ELSE
      tsample = new_timer                       // tsample must be TIMER (par. 3269) for all other motion types
   ENDIF
   TempFF = (629145600 rnd (GET 1900) rnd eres * 100 rnd tsample)       // Calculate Feed-forward
   calculation_done = 1                         // Set done flag to ensure calculation is done only once.
   OUT 1 1                                      // Set "Calculation done" output
RETURN
/*********************** Update parameters *************************************/
SUBPROG update_parameters
   SET TIMER new_timer                // Set new PID sample time.
   SET FFVEL TempFF                   // Set FFVEL to result of calculations
   SET KPROP (TempFF rnd 50)          // P = FFVEL divided by 50
   SET KDER (TempFF rnd 10)           // D = FFVEL divided by 10
   SET KINT 5                         // I is set to 5
   update_done = 1                    // Set done flag to ensure update is done only once.
   OUT 2 1                            // Set "Parameters Updated" output
RETURN
/********************** Save parameters **************************************/
SUBPROG save_par
   SAVE AXPARS                        // Save new parameter settings.
   save_done = 1                      // Set done flag to ensure update is done only once.
RETURN
/********************** Error handler ****************************************/
SUBPROG errhandle
   err = 1         // Set error flag to remain in error handler until error is reset.
   OUT 8 1         // Set error output
   WHILE err DO                       // Remain in error handler until the reset is received
      IF IN 8 THEN                    // Reset error when Input 8 is high.
         ERRCLR                       // Clear error
         err=0                        // Reset error flag
      ENDIF
   ENDWHILE
   OUT 8 0                            // Reset error output
RETURN
/**************************************************************************/
ENDPROG
/*********************** End of program *************************************/
```

__ How to Install __

☐ Influence to the Control Loop when Changing Parameters

The gain of the control loop changes when one or more of the following parameters is changed: Maximum reference, encoder resolution, and gearing between motor and encoder.

When changing these parameters the following rules apply:

Increased maximum reference = P and D factors as well as Feed-forward must be reduced.

Higher encoder resolution = P and D factors as well as Feed-forward must be reduced.

Increase gear-ratio between motor and encoder (encoder turning slower) = P and D factors as well as Feed-forward must be increased.

Examples	32-60	32-61	32-62	32-65
1024 PPR encoder connected directly to the motor shaft and maximum reference 1500 RPM	200	1000	5	10200
4096 PPR encoder connected directly to the motor shaft and maximum reference 1500 RPM	50	250	5	2550
1024 PPR encoder connected behind a 10:1 gearbox and maximum reference 1500 RPM	2000	10000	5	102000
1024 PPR encoder connected directly to the motor shaft and maximum reference 3000 RPM	100	500	5	5100

Further optimizing can easily be done by means of the graphical Test-run tool which is included in the APOSS editor this however implies that the motor/application can move.

☐ Executing a Test Program

Now connect the motor to the FC 300, make sure that the motor can turn completely freely.
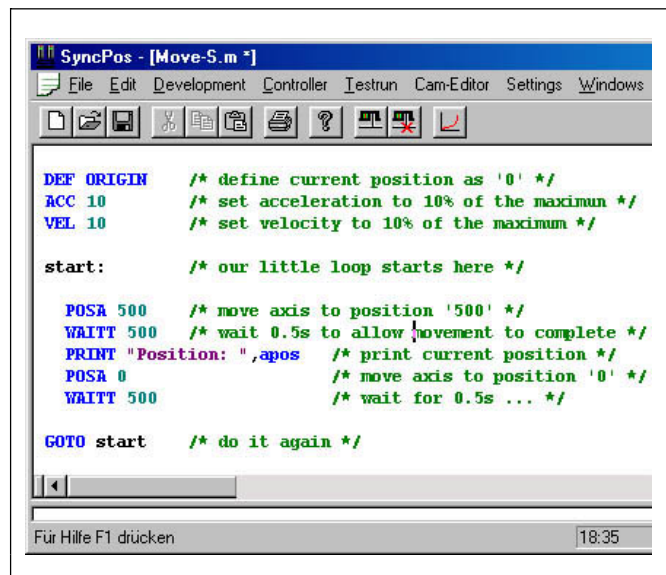
NB!:
The motor must be provided with an EMERGENCY STOP button.

Select the file "Move-S.m" via the MCT 10. APOSS and with that the file is opened automatically.

Click on *Development* and start the test program with → *Execute* or [F5].

The test is successful if the motor runs slowly back and forth and position 500 is registered.

End the test with [Esc] and → *Close* the *File*.


```
DEF ORIGIN      /* define current position as '0' */
ACC 10          /* set acceleration to 10% of the maximun */
VEL 10          /* set velocity to 10% of the maximum */

start:          /* our little loop starts here */

  POSA 500      /* move axis to position '500' */
  WAITT 500     /* wait 0.5s to allow movement to complete */
  PRINT "Position: ",apos    /* print current position */
  POSA 0                     /* move axis to position '0' */
  WAITT 500                  /* wait for 0.5s ... */

GOTO start      /* do it again */
```

__ How to Install __

☐ What to do, if … ?

What to do, if the motor sets off uncontrolled or vibrates heavily?

NB!:
Turn off the motor immediately with the EMERGENCY STOP button if it vibrates heavily or sets off uncontrolled.

If the motor sets off uncontrolled, but the encoder test previously was successful, then decrease the par. 32-60 *Proportional Factor*. (See Optimizing the PID controller.)

What to do, if the motor doesn't move?

If the motor doesn't move at all, then the *Proportional Factor* of the PID filter is probably too low or the FC 300 has not been enabled.

Check the FC 300 enable (terminal 37 = 24 V) and check that the FC 300 wasn't stopped via the LCP. Then increase the par. 32-60 *Proportional Factor*. (See Optimizing the PID controller.)

What to do, if the motor vibrates heavily?

If the motor vibrates then you have to optimize the PID controller and to adjust the other parameters of the controller: reduce either the par. 32-60 *Proportional Factor* or increase the par. 32-61 *Derivative Value for PID Control*.

What to do, if a "Tolerated Position Error is exceeded" is reported?

If the drive stops due to a "Tolerated Position Error is exceeded" message it is possible to determine whether the drive was rotating in the wrong direction by comparing the curves of the set and actual values.

Check the connections of the motor or encoder. If the connections are correct, then it is necessary to increase the par. 32-67 *Maximum Tolerated Position Error*. (See Optimizing the PID controller.)

Optimizing the PID Controller

□ How the Control Process Works

The *Testrun* explained in the chapter 'PC Software interface' of the APOSS Online help can be used as a tool to optimize the MCO 305 controller settings, thereby optimizing your system performance. To do that you only need to know a few things about the control scheme of MCO 305:

The APOSS position controller has two parts:

1. The *Setpoint Generator* interprets the various positioning commands in APOSS and generates a series of setpoint positions that eventually ends with the desired position.
    Normally, all positioning processes have a trapeze-shaped velocity curve. That means that after a phase of constant acceleration there is a phase with constant velocity and finally a phase with constant deceleration, which ends at the desired target position.
2. The *PID controller* receives the setpoint positions from the *Setpoint Generator* and in turn calculates the speed reference needed for the motor to follow the current setpoint position.
    By setting the PID control parameters you can directly influence to what degree and how quickly a deviation from a theoretical set path (as specified by the setpoint series) should be counteracted.

The following behavior indicate that the control parameters are not optimally adjusted:

–   Drive vibrates
–   Drive is loud
–   Frequent occurrence of position errors
–   Poor control accuracy

NB!:
The control parameters are load-dependent. Thus the drive must be optimized under the actual conditions of use.

In exceptional cases it may be necessary to determine various sets of control parameters while working with heavily changing load conditions and to re-program them in subsequent application programs depending on the motion process.

□ Significance and Influence of the Controller Parameters

The PID control unit of the APOSS motion controller transfers the necessary output frequency via an internal speed reference to the FC 300. This set value is periodically recalculated with an interval of one millisecond (interval is programmable by the TIMER parameter). The MCO 305 is by default set with soft 'fit for all' controller parameters.

__ Optimizing the PID Controller __

AV   is the actual velocity
AP   is the actual position (calculated from encoder feedback) in qc (Quad Counts).
CP   is the current position set point in qc.
CV   is the velocity set point in qc/ms
          (Position deviation) is calculated by CP–AP.
CA   is the acceleration set point
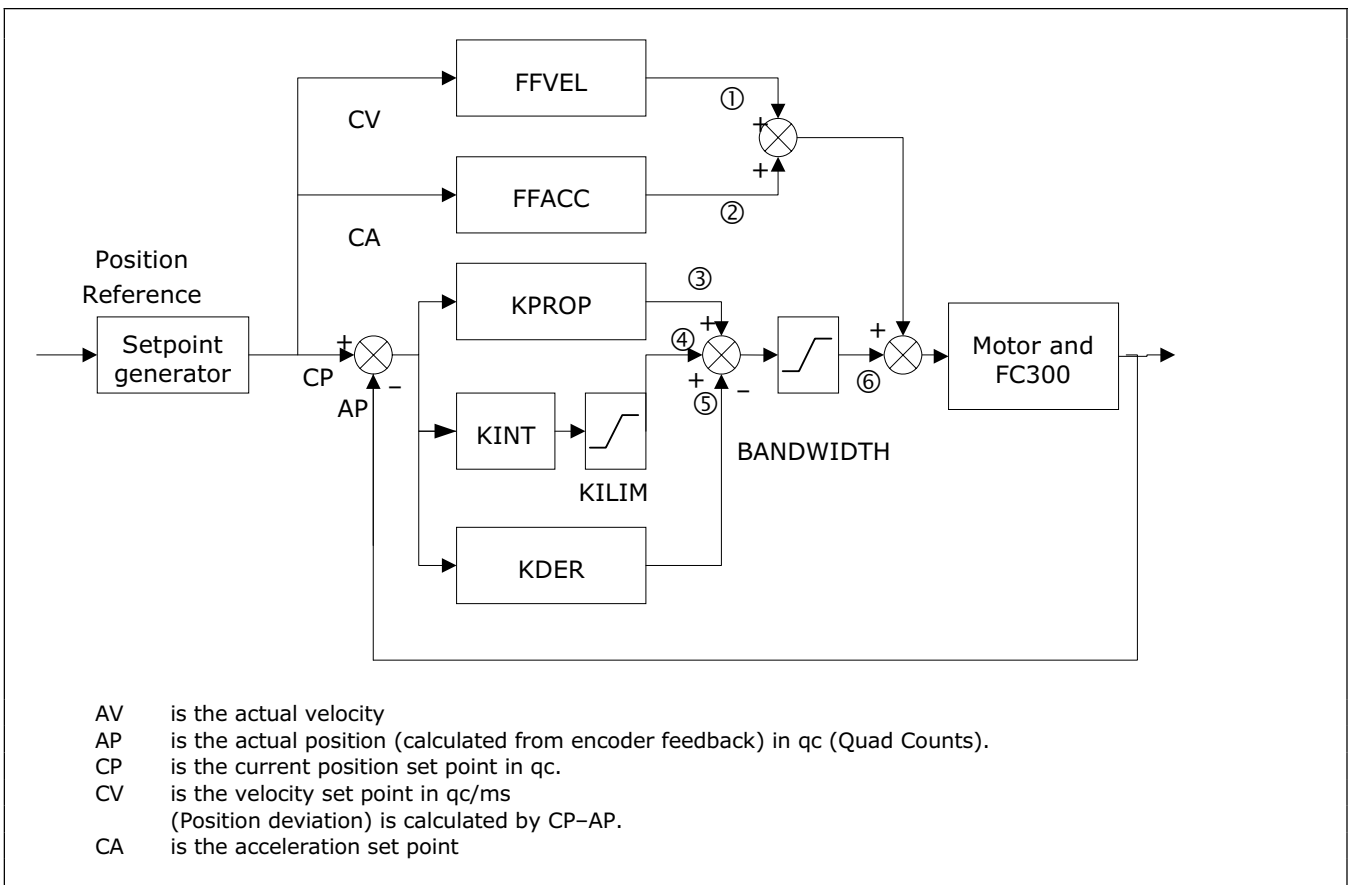
The PID filter works according to the following formula:

1 = FFVEL * (Setpoint Velocity)

2 = FFACC * (Setpoint Acceleration)

3 = KPROP * (Position Deviation)

4 = KINT * (Sum of all previous Position Deviations) (limited by KILIM)

5 = KDER * (Difference of Position Deviation)

6 = 3 + 4 + 5 (Limited by BANDWIDTH)

NB!:
In SYNCV mode the PID controller is working with speed deviation instead of position deviation. Speed deviation is calculated by CV–AV.

The controller in the MCO 305 utilizes two control strategies at the same time:

1. An open-loop feed-forward control. Since the asynchronous motor inherently has a good open loop performance the feed-forward control is a very important part of the controller in most applications. Benefits from using feed-forwards control is a very fast and accurate response to changes in the s*etpoint reference.*
2. A closed-loop PID control. The PID controller monitors the difference between the actual position and the setpoint position. Based on this information it calculates a control signal to minimize the position deviance. Thus the MCO is able to compensate for changes in load or friction. The PID controller is also necessary to compensate for any position deviance caused by inaccurate setting of the open-loop feed-forward controller.

__ Optimizing the PID Controller __

In short: The feed-forward control is used to handle changes in the setpoint reference (especially important in synchronization applications), while the PID control is used to handle changes in load conditions or inaccuracies of the feed-forward control.

□ PID Factors

□ Proportional Factor: KPROP

The par. 32-60 *Proportional Factor* is multiplied with the position deviance and the result is added to the control signal (the internal speed-reference to the FC 300). Since the calculated control signal is proportional to the position deviance (or error) this kind of control is called proportional control. The behavior of the proportional control is similar to that of a spring – the further the spring is extended the stronger the counter-force it produces.

Influence of the *Proportional Factor:*	
KPROP too small	large position deviation due to non-compensatable load and frictional moment
KPROP larger	quicker reaction, smaller steady-state deviation, larger overshoot, lesser damping
KPROP too great	heavy vibrations, instability

□ Derivative Value for PID Control: KDER

The par. 32-61 *Derivative Value* is multiplied with the derivative of the position deviance (the 'velocity' of the position deviance) and the result is added to the control signal. The behavior of the derivative control is similar to that of an absorber – the faster the absorber is extended the stronger the counter-force it produces. Thus using the *Derivative Value* increases damping in your system.

Influence of the *Derivative Value:*	
KDER small	no effect
KDER larger	better dampening, lesser overshoot; if KPROP is increased simultaneously: faster reaction to control deviation at the same level of vibration;
KDER too large	heavy vibrations, instability

□ Integral Factor: KINT

The sum of all error is calculated every time the control signal is updated. The par. 32-62 *Integral Factor* is then multiplied with the sum of all positioning errors and added to the overall control signal.

Thus in case that steady-state position errors occurs in your application, make sure you use the integral part of the controller. Steady-state errors will be leveled out as the summed error increases over time until the control signal eventually matches the load.

It is possible to set a limit for the control signal generated by the integral part of the controller (anti-windup).

Influence of the *Integral factor*:	
KINT very small	steady-state position deviance is very slowly regulated to zero
KINT larger	faster regulation towards zero steady-state position deviance, larger overshoot
KINT too large	heavy vibrations, instability

□ Limit Value for Integral Sum: KILIM

The par. 32-63 *Limit Value for Integral Sum* sets a limit for the control signal generated by the integral part of the controller. This helps to prevent the so called "wind-up" problem which typically occurs in applications where the overall control signal (the internal speed-reference) is maxed out for long periods of time.

This feature is also very helpful in applications where the power to the motor is turned off and on while the option card is controlling the FC 300. Cutting of power to the motor (by setting terminal 27 low) while little positioning deviance is present in the controller, could result in an enormous control signal being generated once the power is turned back on.

__ Optimizing the PID Controller __

☐ Velocity Feed-forward: FFVEL

The par. 32-65 *Velocity Feed-forward* is a scaling factor that is multiplied with the derivative of the setpoint position (the velocity of the setpoint). The result of this operation is added to the overall control signal. This feature is especially useful in applications where there is a good correlation between the control signal (the FC 300 speed reference), and the speed of the motor. This is indeed the case with most applications.

NB!:
The scaling of the FFVEL parameter is dependent on the correct setting of the *Maximum Reference* (par. 3-03) as well as the parameters 32-80 *Maximum Velocity* and 32-01 *Incremental Resolution*.

☐ Acceleration Feed-forward: FFACC

The par. 32-66 *Acceleration Feed-forward* is multiplied with the 2nd derivative of the setpoint position (the acceleration of the setpoint) and the result is added to the control signal. This feature should be used to compensate for the torque used to accelerate/decelerate the system inertia.

NB!:
Scaling of the *Acceleration Feed-forward* factor is depending on the setting of the *Shortest Ramp*. You should therefore increase the par. 32-66 *Acceleration Feed-forward* accordingly when decreasing par. 32-81 *Shortest Ramp* and vice versa.

☐ Sampling Time for PID Control: TIMER

For particularly slow systems you can slow down the entire control system by entering multiples of 1 ms as the sampling time. However, it is important to remember that such a change influences all the control parameters!

Therefore, normally you should not deviate from the value of 1 ms in par. 32-69 *Sampling Time for PID control.*

☐ PID Bandwidth

A *Bandwidth* of 1000 means that the set value is being executed 100%, thus *Derivative*, *Proportional* and *Integral Factors* are effective as defined.

But if you are operating a system which could be jeopardized by vibrations, for example, a crane with a heavy load then you can limit the bandwidth in which the PID controller should function. Par. 32-64 *PID Bandwidth* of 300 makes a limitation of 30% possible. The build-up of a vibration is thus prevented since the control is only moved to with 30% of the calculated set value.

However, then it is necessary also to use the feed-forward part of the controller in order to achieve the corresponding control.

__ Optimizing the PID Controller __

Optimizing your Controller Settings Step-by-Step

Before you adjust the controller parameters it is important to determine which controller behavior is to be achieved.

NB!:
The drive elements must never be operated outside of the technical specifications. Thus the maximum acceleration is determined by the "weakest" drive element.

–   "Stiff" axis: the fastest reaction possible is mainly influenced by the *Proportional Factor*. You can judge the results on the basis of the velocity graph.

–   Damping of vibrations is mainly influenced by the *Derivative Value*. The results can best be assessed in the velocity graph.

–   Temporary (static) deviations in position are mainly reduced by the *Integral Factor* and can best be judged on the basis of the positioning graph.

For best results use the functions in the *Testrun* menu for this purpose; these help you to evaluate and optimize the PID controller on the basis of graphs of the set and actual curves.

However, we recommend only changing one value at a time and subsequently determining the improvement with a *Testrun*.

Click on *Controller → Parameter → Axis* and select the drive, of which you are currently adjusting the settings.

Ten Steps for Optimum Control

The following procedure will optimize your controller settings in most applications:

1.   Make sure that you have specified correct values for FC 300 par. 3-03, as well as the parameters 32-80 *Maximum Velocity*, the encoder type and resolution in par. 32-00 and 32-01, and 32-81 *Shortest Ramp*. If you change these settings again at a later point you may need to optimize the controller again.

Par. #	Typical Settings	
3-03	*Maximum Reference*	1500,000
32-80	*Max. Velocity (Encoder)*	1500
32-00	*Incremental Signal Type*	[1] RS422
32-01	*Incremental Resolution*	1024
32-81	*Shortest Ramp*	1.000

2.   Set the par. 32-67 *Max. Tolerated Position Error* to a very high value e.g. 1000000 to avoid getting the error 108 during the following tests.

NB!:
To avoid damaging the system, make sure, that you set the par. 32-67 *Position Error* within the limits of the system, because the position error monitoring is not active with extremely high values.

3.   Optimize the *Velocity Feed-forward* control: Click on → *Execute Testrun* and → *Start* a testrun with following parameter settings:

32-60	*Proportional Factor*	KPROP	0
32-61	*Derivative Value for PID Control*	KDER	0
32-65	*Velocity Feed-forward*	FFVEL	100
32-66	*Acceleration FF*	FFACC	0

View the velocity profiles: If the Actual Velocity profile is <u>lower</u> than the Commanded Velocity profile, increase *Velocity Feed-forward* and → *Start* the test run again. Of course if the Actual Velocity profile is <u>higher</u> than the specified Commanded Velocity you should decrease *Velocity Feed-forward*.

__ Optimizing the PID Controller __

Run successive test runs until the two velocity profiles shown in the Testrun graph have the same maximum value.

Velocity Feed-forward is now optimized, save the current value.

4. In systems with large inertia and/or rapid changes in the reference velocity it is a good idea to use and optimize the *Acceleration Feed-forward* control (make sure the inertial load is connected when optimizing this parameter):

Execute a Testrun with KPROP=0, KDER=0, KINT=0, FFACC=0, and FFVEL at the optimized value found above. Use the highest possible acceleration setting. If par. 32-81 *Shortest Ramp* is adjusted properly an acceleration value of 100 and a deceleration value of 100 should be sufficient. Start out with a low setting of *Acceleration Feed-forward* approx. 10.

32-60	*Proportional*	KPROP	0
32-61	*Derivative*	KDER	0
32-62	*Integral*	KINT	0
32-65	*Velocity Feed-forward*	FFVEL	result of step 3
32-66	*Acceleration FF*	FFACC	10

View the velocity profiles: If during acceleration the actual velocity is constantly lower than the reference velocity profile, then set a higher value of *Acceleration Feed-forward* and → *Start* the Testrun again.

Run successive test runs until the two velocity profiles shown in the *Testrun* graph have similar ramp-up and ramp-down curves.

Acceleration Feed-forward is now optimized, save the current value.

5. Next step is finding the maximum stable value of the *Proportional Factor* in the PID controller. Execute a Testrun with KPROP=0, KDER=0, KINT=0. Set FFVEL and FFACC at the optimized values found above.

View the velocity profile. If the velocity profile is not oscillating then increase the *Proportional Factor*. Run successive test runs until the actual velocity profile is oscillating mildly.

32-60	*Proportional*	KPROP	0
32-61	*Derivative*	KDER	0
32-62	*Integral*	KINT	0
32-65	*Velocity Feed-forward*	FFVEL	result of step 3
32-66	*Acceleration Feed-forward*	FFACC	result of step 4

Decrease this "mildly" unstable *Proportional Factor* value to about 70 %. Save this new value.

6. In order to dampen the oscillations created by the *Proportional* part of the controller, the *Derivative Value* should now be optimized. → *Start* a Testrun with KINT=0 and KDER=200. Set FFVEL, FFACC and KPROP at the optimized values found above.

Run successive test runs with increasing values of the *Derivative Value*. At first the oscillations will gradually reduce. Stop increasing *Derivative Value* when the oscillations begin to increase.
Save the last value of *Derivative*.

32-60	*Proportional Factor*	KPROP	result of step 5
32-61	*Derivative*	KDER	200
32-62	*Integral*	KINT	0
32-65	*Velocity Feed-forward*	FFVEL	result of step 3
32-66	*Acceleration Feed-forward*	FFACC	result of step 4

7. In any system that requires a zero steady-state error, the *Integral* part of the controller must be used. Setting this parameter though is a trade-off between achieving zero steady-state error fast (which is good) and increasing overshoot and oscillations in the system (which is bad).

If you are using the *Integral* part of the PID controller, remember to reduce the par. 32-63 *Limit Value for Integral Sum* as much as possible (without losing the *Integral Factor* effect of course) in order to reduce oscillations and overshoot as much as possible.

9. Reduce the par. 32-64 *PID Bandwidth* as much as possible. With a properly optimized open-loop control Bandwidth could be reduced to as little as 6 or 12 % (60 – 120).

10. Set the par. 32-67 *Position Error* back to normal e.g. 20,000.

11. Once you have concluded the Testrun then → *Save* the new parameters as the user parameters. Thus, these parameters are saved in the controller and in the future will be used for all programs.

□ **What to do if . . . ?**

What to do, if there is a tendency towards instability?

In the event of a strong tendency towards instability reduce the parameters 32-60 and 32-61 *Proportional* and *Derivative Factors* again, or reset the par. 32-62 *Integral Factor.*

What to do, if stationary precision is required?

If stationary precision is required then you must increase the par. 32-62 *Integral Factor*.

What to do if the tolerated position error is exceeded?

If the test run is constantly interrupted with the message "position error" set the par. 32-67 *Maximum Tolerated Position Error*– within the tolerable limits– as large as possible.

If the position error occurs during the acceleration phase that suggests that the set acceleration cannot be achieved under the existing load conditions. Increase the par. 32-67 *Position Error* or determine a *Maximum Acceleration* suitable for the entire system.

If position errors do not occur until after the acceleration phase and they can be delayed but not eliminated by increasing the 32-67 *Max. Tolerated Position Error*, this suggests that the *Maximum Velocity* (RPM) chosen is too high. Determine a maximum velocity suitable for the entire system.

What to do if the maximum acceleration is not achieved?

In general, the technical data for a drive are only valid for a freely rotating axle end. If the drive is carrying a load the maximum acceleration is reduced.

The theoretical maximum acceleration will also not be achieved if, for example, the PID controller output is too small or the FC 300/motor is not sized correctly and therefore does not provide enough energy for peak consumption during acceleration.

How to Install the Application

□ Download Application Program and Configuration

Follow these steps to create and download MCO 305 program and configuration using MCT 10 and APOSS

1. Open MCT 10.

2. Open the folder "Project – Drive – MCO 305 – programs".

3. Right click in the right hand view and select:

   a. "Import program file" if using an existing program "outside" the project file or,

   b. "New MCO 305 program" when writing a new program. Right click the program file and double click or right click and select "Edit". Program editor will open.

4. Repeat 3. if more programs are required.

5. One program must be set as *Autostart* program: Right click the program file and select "Set as Autostart".

6. Open the folder "Project – Drive – MCO 305 – Arrays".

7. Right click on the file and select "Import" or "Edit".

8. Right click on the MCO 305 folder in the tree structure in the left-hand view of MCT 10 and select "Write to drive".

9. "Write MCO 305 Program Source Files to the Drive" must be ticked if it should be possible to read the program from the Drive.

□ Backup and Restore

□ Backup and Restore via MCT 10

All parameters and MCO 305 data can be read out and stored in an MCT 10 project and thus restored to the drive.

NB!:
MCO 305 programs can only be read-out when they have be downloaded including source code!

Use the following steps to back-up parameters and data by PC:

1. Go on-line with the drive via the Network folder.

2. Right click on the drive folder and select copy. "Include programs" must be ticked.

__ How to Install the Application __

3. When copy is completed, paste the drive in-to a project folder.
4. Save the project.

Now parameters and MCO 305 data is stored in the project file, restore by the following steps:

1. Open the project file with MCT 10.
2. Go on-line with the drive via the Network folder.
3. Open the project folder, Right click on the drive folder and select "Write to drive".

Now both parameters and MCO 305 data is restored in the drive.

❑ Backup and Restore via LCP

Both parameters and MCO 305 data can be stored in the LCP and thus restored from the LCP to another drive.

Use the following steps to back-up parameters data in the LCP:

1. Select "All to LCP" [1] in parameter 0-50 LCP copy.
2. Select "File from MCO to LCP" [4] in parameter 0-50 LCP copy.

Now parameters and MCO 305 data is stored in the LCP, restore by the following steps:

1. Select "All from LCP" [2] in parameter 0-50 LCP copy.
2. Select "File from LCP to MCO" [5] in parameter 0-50 LCP copy.

Now both parameters and MCO 305 data is restored in the drive.

❑ How to Connect and Run Multiple Drives

Use MCT 10 *Networking* or start APOSS directly by clicking on the application's icon. This stand-alone operation mode allows APOSS to switch drives or connect to multiple drives. In stand-alone mode, all file and parameter handling is done via APOSS. Thus the *File* menu also provides the functions for *File → New, → Open,* and → *Save As.* The *Settings* menu allows selecting the interface.

Run programs in several FC 300s

If you want to load the program into several controllers, link the program with the corresponding FC 300 and click on → *Execute.*

If you want to load a different program in each controller, open a different edit window for each FC 300, then open the desired program file and connect it to the FC 300 with → *Select Controller.* Then start each program, one after the other, with → *Execute*.

General Specifications

□ **Protection and Features**

–   All inputs, outputs and supply voltages are protected against short circuit.

–   All inputs, outputs and supply voltages are galvanic isolated from high voltages such as mains supply and motor voltage (PELV).

–   Encoder signals are monitored during operation and standstill.

–   Customer specific application programs can be copy protected.

–   All MCO 305 parameters including user defined application parameters are accessible via the FC 300 Local Control Panel.

–   MCO 305 can be combined with other FC 300 options e.g. PROFIBUS and DeviceNet interface.

–   All digital inputs and outputs are galvanic isolated from the internal electronics and can be sourced from an external 24V power supply.

□ **Technical Data**

Connection terminals

Type ....Plug connectors with screw terminals.
Maximum cross section, rigid wire .... 1,5 mm^2 / AWG 16
Maximum cross section, flexible wire .... 1,5 mm^2 / AWG 16
Maximum cross section, wire with enclosed core .... 1,5 mm^2 / AWG 16
Minimum cross section ....0,08 mm^2 / AWG 28

Digital inputs

Number of programmable digital inputs 10
Terminal block X57
Terminal number 1, 2, 3, 4, 5, 6, 7, 8, 9, 10
Logic ....PNP or NPN [1]
Voltage level .... 0 - 24 V DC
Voltage level, logic.0. PNP .... < 5 V DC
Voltage level, logic.1. PNP .... > 10 V DC
Voltage level, logic.0. NPN .... > 19 V DC
Voltage level, logic.1. NPN .... < 14 V DC
Maximum voltage on input .... 28 V DC

__ General Specifications __

1) Selected in parameter 5-00 Digital I/O mode.
The digital inputs are galvanic isolated from the internal electronics and can be sourced
by an external 24V power supply.

Digital outputs

Number of programmable digital outputs ....8 (6)[1]
Terminal block X59
Terminal number .... 1[1], 2[1], 3, 4, 5, 6, 7, 8
Driver type ....Push/Pull
Logic ....PNP or NPN [2]
Voltage level .... 0 - 24 V DC
Max. output current (sink or source) with internal power supply (total Σ) ....40 mA
Max. output current (sink or source) with external power supply (per output) ....100 mA

   1)   Terminals X59-1 and X59-2 can be programmed as input, parameter 33-60.
   2)   Selected in parameter 5-00 Digital I/O mode.

Combined digital inputs/outputs

Number of digital outputs which can be used as digital inputs ....2 [1]
Terminal block X59
Terminal number 1, 2
Logic ....PNP or NPN [2]
Voltage level .... 0 - 24 V DC
Voltage level, logic.0. PNP .... < 10 V DC
Voltage level, logic.1. PNP .... > 17 V DC
Voltage level, logic.0. NPN .... > 13 V DC
Voltage level, logic.1. NPN .... < 6 V DC
Maximum voltage on input .... 28 V DC

   1) Terminals X59-1 and X59-2 can be programmed as input, parameter 33-60.
   2) Selected in parameter 5-00 Digital I/O mode.

24 V DC supply output

Terminal block X58
Terminal number 1, 2
Maximum load ....65 mA

The internal 24V power supply can be disconnected via parameter 33-85, an external
24V power supply must then be connected to X58-1 and X58-2.

Encoder inputs

General specifications:
Number of encoder inputs 2
Terminal block ....X55 and X56
Terminal number 5, 6, 7, 8, 9, 10, 11, 12
Input impedance .... 120 Ω
Maximum voltage on inputs .... 5 V DC
Cable type ....Screened cable with a twisted pair of wires for each encoder channel[1]

Incremental encoder specifications
Incremental encoder type RS422/TTL
Maximum frequency .... 410 kHz
Phase displacement between A and B..90˚± 30˚
Maximum cable length ....300m [2]

__ General Specifications __

Absolute encoder specifications

Absolute encoder type .... SSI

Data coding .... Gray

Data length .... 12 – 37 bit

Clock frequency .... 78 kHz – 2 MHz [1]

Maximum cable length .... 150 m [2]

   1) Always observe specifications/limitations prescribed by the encoder supplier.

   2) 150 m cable is possible up to 500 kHz clock frequency, above 500 kHz cable length

   must be limited further.

Encoder output

Number of encoder outputs 1

Terminal block X56

Terminal number 5, 6, 7, 8, 9, 10, 11, 12

Signal type RS422

Maximum frequency .... 410 kHz

Maximum number of slaves .... 31 (more with repeater)

Maximum cable length .... 400 m

Encoder voltage supply

Number of supply voltages   3

Terminal block   X55 and X56

Terminal number   1, 2, 3, 4

24 V, max. load   250 mA [1]

8 V, max. load   250 mA [1] [2]

5 V, max. load   400 mA [1]

   1) This is maximum load when only one supply voltage is used; when 2 or 3 supply

   voltages are used simultaneously the load must be reduced accordingly. The following

   must be observed: $load^{24V} + load^{8V} + load^{5V} \leq 6W$ and $load^{8V} + load^{5V} \leq 2W$.

   2) 8 V is only available at terminal block X55.

Control characteristics

Sample time of position PID loop .... 1 ms

Positioning accuracy .... ± 1 increment [1]

Synchronizing accuracy .... ± 1 increment [1]

   1) This is the static accuracy; dynamic accuracy depends on many "external" factors

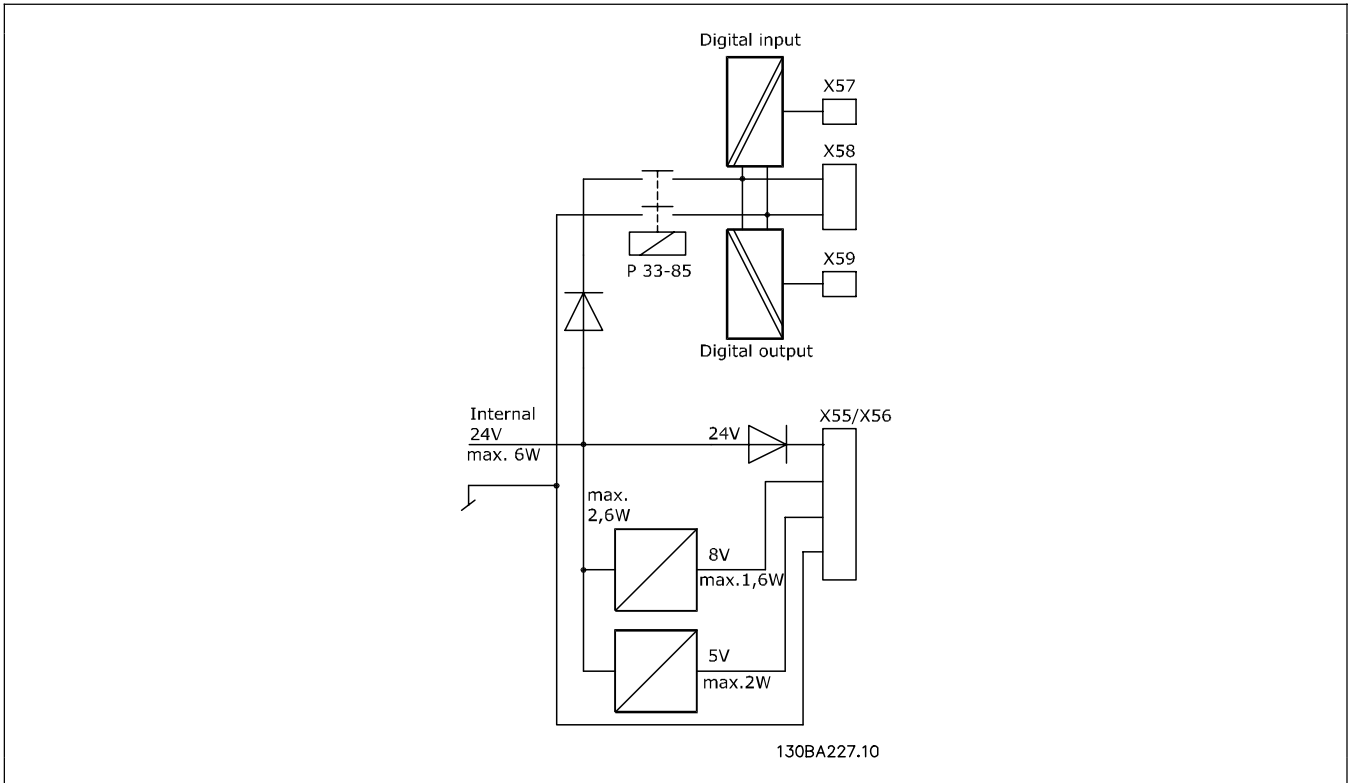   such as encoder resolution, moment of inertia, mechanical backlash and elasticity.

Application program

Program memory size .... 100 Kbytes

Maximum number of application programs 90

Average command execution time .... 0.3 ms

Maximum reaction time on interrupt input .... 1 ms

__ General Specifications __

□ **Supply Voltage Overview**

130BA227.10

Troubleshooting

□ **Warnings and Error Messages**

All messages are shown in the LCP display of the FC 300 in short and in the APOSS software in plain text.

You can find brief information on the error messages in the table or detailed information in the following section.

The tables contain the messages in numerical order. Letters following a % sign represent variables which can be used in plain text at the corresponding locations.

Error no.	Error text	Description
103	Illegal axis num.	Axes not in system.
105	Error not reset	Error not cleared.
106	Home not done	Failed to move to HOME position.
107	Home vel. zero	Home was executed with Home Velocity set to zero.
108	Position error	Position error.
109	Index not found	Index pulse (encoder) not found.
110	Unknown cmm.	Unknown command.
111	SW end limit	Software end limit activated.
112	Unknown param.	Illegal parameter number.
113	FC not enabled	FC 300 is not ready but the PID controller is active.
114	Too many loops	Too many nested loops.
115	Par. save failed	Parameter save failed.
116	Param. memory	Parameters in memory are corrupted.
117	Progr. Memory	Programs in memory are corrupted.
118	Reset by CPU	Reset by CPU.
119	User abort	User abort.
125	HW end limit	HW end limit activated.
149	Too many inter.	Too many interrupt functions.
150	No ext. 24V	External supply is missing.

Error no.	Error text	Description
151	Too many gosub	Too many nested GOSUB commands
152	Too many return	Too many RETURN commands.
154	D. out overload	Digital Output overloaded.
155	LINK failed	LINKGPAR command failed.
162	Memory error	Error in verifying; EEPROM: address % defect.
170	Array size (DIM)	Error in DIM command.
171	Array too small	Attempt was made to cross array bounds.
179	Waitndx timeout	Timeout while waiting for index.
184	Too many ontime	Too many time interrupts.
187	Out of memory	No more room for variables.
190	Memory locked	The program memory is write-protected.
191	Illegal cam array	Curve array wrong.
192	Encoder error	Encoder error
199	Internal MCO fault	Internal MCO fault

Error 103

Illegal axis num.

An attempt has been made to find an axis which does not exist in the controller.

Check to see if the program axis command has an invalid number or a general axis command (...X(*)).

Error 105

Error not reset

An attempt has been made to execute a motion command, although a momentary error message has not been cleared.

Error 106

Home not done

Failed to move to HOME position. According to the axis par. 33-00 *Force HOME*, a forced move to the machine zero-point is demanded, before other motion commands can be executed. This move to the machine zero-point has not been executed.

Error 107

Home vel. Zero

An attempt was made to execute the HOME command but the motor is set to 0 in par. 33-03 Velocity of Home Motion.

Error 108

Position error

HOME was executed with HOME_VEL set to zero.

The distance between the set and the real position was greater than the *Maximum Tolerated Position Error* defined in par. 32-67 POSERR.

Causes:

− Mechanically blocked or overloaded drive,

− par. 32-67 *Maximum Tolerated Position Error* too small,

− commanded speed greater than FC 300 parameters 4-13 *Motor Speed High Limit* and 3-03 *Maximum Reference*,

− commanded acceleration too great,

− par. 32-60 *Proportional Factor* too small, or

− FC 300 not enabled.

__ Troubleshooting __

Error 109

Index not found

At reference or index search, the encoder index pulse could not be found within a motor rotation.

Causes

– An encoder without an index pulse has been used,

– index pulse not connected,

– index pulse incorrect (all three channels must have a simultaneous low), or

– the par. 32-01 *Incremental Resolution* (ENCODER) is set too low.

Error 110

Unknown comm.

Cause is a Communication or program error. The program must be re-compiled and re-loaded.

Error 111

SW end limit

A motion command will cause or has caused the software end limit to be activated.

Identification of attainment of software limit at a motion in the speed mode will only be made after the current position is identical to the software limit switch.

The control unit will be switched off and the drive must be manually moved back to within the admissible area, or the monitoring of the software limit switch must be temporarily de-activated via the *Negative and Positive Software End Limit* in par. 33-43 SWNEGLIMACT and par. 33-44 SWPOSLIMACT. Only then is it possible to clear the error.

In positioning mode, it will be known before motion start that the target position lies outside the path. In this case, the movement will not be executed and the error message can be cleared.

Error 112

Unknown param.

An attempt has been made to change a parameter (SET or SETVLT command), which does not exist.

Error 113

FC not enabled

FC 300 is not ready but the PID controller is active. The FC status word (Bit 09 and Bit 11) is monitored every 20 ms when the PID controller is active. The FC 300 is in the "Not ready" state when:

– it has an alarm,

– it is in local mode,

– local LCP stop is activated.

Error 114

Too many loops

Too many nested loops exist in the executed program.

Error 115

Par. Save failed

Saving of the option parameter failed.

Error 116

Param. memory

The parameters in EEPROM are no longer correct because of

– EEPROM defective or

– power outage while saving.

NB!:
You have to re-initialize the parameter with a 14-22 Reset and then overwrite these parameters again with your own application parameters.

Otherwise motion programs which require application parameters will no longer function correctly.

Error 117

Progr. Memory

The program data stored in EEPROM cannot be found or are no longer correct because of

– EEPROM defective or

– power outage while saving.

You have to do a 3-finger reset to reset all parameters to their defaults (ex factory) and to delete all user programs, arrays, and application parameters.

Afterwards re-load the programs and parameters.

This corresponds to a → *Reset complete* in the APOSS menu.

__ Troubleshooting __

Error 118

Reset by CPU

The processor has been stopped and a re-set has automatically been executed (watchdog).

Causes could be

– Short term voltage drop,

– voltage peak, or

– short circuit.

Error 119

User abort

The Autostart program has been aborted by the user.

Or the [CANCEL] key was pressed during switching on and a Master Reset triggered.

Error 125

HW end limit

A motion command has caused an axis limit switch to be activated.

Through activation of an end limit switch, the controller (depending on the par. 33-40 ENDSWMOD) is automatically switched off and the drive must be manually moved out of this position, before the error message can be cleared.

Error 149

Too many inter.

More interrupt functions than the maximum possible number were used. Permitted are:

32 ON INT
32 ON STATBIT
32 ON COMBIT
10 ON PARAM
20 ON APOS, ON MAPOS, ON MCPOS

Error 150

No ext. 24V

External supply is missing.

Error 151

Too many gosub

In the program exists too many calls from one subroutine to another subroutine.

The error usually occurs when there is a recurrent reference to one of the sub-programs in a sub-program.

Avoid too many (10 is maximum) opposing subroutine calls, and avoid subroutines which call themselves (re-cursive subroutine procedures).

Error 152

Too many return

There are either more RETURN than corresponding GOSUB commands in the program, or there is a direct jump from a subroutine with a GOTO command.

Only one RETURN is allowed per sub-program.

It is always better to jump to the beginning of a sub-program and then to jump with IF… to a previously defined label.

Error 154

D. out overload

Digital output overloaded.

Error 155

LINK failed

LINKGPAR command failed.

Error 162

Memory error

After saving something in the EEPROM (a program or parameters) an error was detected during verification.

Delete the EEPROM with a 3-finger reset and try to save the program or parameters again.

If this is not successful please call the technical service department.

Error 170

Array size (DIM)

The definition of an array in a DIM command does not correspond to an already existing array in the MCO 305.

Cause might be that the fields are from older SYNCPOS/APOSS programs. The current program has other definitions.

Either adapt the APOSS program to the correct array size or delete the old arrays.

NB!:
Remember to follow the recommendations concerning saving programs and parameters before deleting the EEPROM.

__ Troubleshooting __

Error 171

Array too small

An attempt was made to describe an array element that is located outside of the defined array limits.

Cause might be an error in the APOSS program. Array sizing does not agree with the space required (e.g. due to an incorrectly programmed loop).

Or the array is too small for the number of test drives triggered by TESTSTART.

Check loop variables.

Error 179

Waitndx timeout

The command WAITNDX was executed and the timeout listed was exceeded.

The timeout is probably too short or the index impulse could not found (see also Error 109).

Error 184

Too many ontime

Too many interrupts (ON TIME or ON PERIOD commands) were used within the program.

A maximum of 12 of these ON TIME and/or ON PERIOD commands are allowed within one program.

Error 187

Out of memory

No more space for variables.

When the APOSS program is started the space for the necessary variables is reserved dynamically. This space is now no longer available.

You may have selected a maximum number of variables which is too high. Reduce the maximum number in *Settings → Compiler* (Standard = 92).

Or the memory available is occupied with programs or arrays. Delete the programs or delete both the programs and arrays, i.e. by deleting the entire memory.

NB!:
Remember to follow the recommendations concerning saving programs and parameters before deleting the EEPROM.

Error 190

Memory locked

The program memory is write-protected and cannot be altered.

This means that auto recognition can neither be set nor deleted and programs can neither be saved nor deleted. Equally, → *RAM save* and → *EEPROM delete* will not be executed.

Error 191

Illegal cam array

An incorrect or old array is defined in the DIM instruction for SETCURVE.

An old array may exist if the CNF file with all parameters and arrays has not been loaded into the *CAM-Editor*.

An incorrect array could be caused by the following:

- It was not created by the curve editor.
- Previous version of a curve editor. Such an array must first be converted by the current CAM-Editor (→ *load* and *save*).
- Or the order of the arrays in the DIM instruction does not match the order in the cnf file. Refer to the number of the array in the title bar of the *CAM-Editor* in this respect.

Error 192

Encoder error

Error from encoder monitoring: open or short circuit in accordance with the displayed LED.

NB!:
An error will be indicated even if no encoder is connected.

Error 199

Internal MCO fault

If such an error should occur, please contact your dealer and report the error number displayed to the technical service department.

__ Troubleshooting __

□ APOSS Software Messages

The APOSS software messages are arranged in alphabetical order. Letters following a % sign represent variables which can be used in plain text at the corresponding locations.

Compilation error(s): program not saved!

A file is always compiled first and then saved. If you want to save the program, for example in the menu *Controller → Save program* and a syntax error is found during compilation this message will be displayed.

Start the *→ Syntax Check* in the menu *Development*, correct the syntax error and then save the program.

Connection to ... already exists ...

Connection to %d already exists [%s] - change to Window?

When opening a new window, or when trying to connect a window with a controller that is already linked to a window.

Yes: The controller is disconnected from the old window and linked to the new window.

No: The controller stays connected to the old window. The new window is not linked to a controller.

Connector pin is not valid

Connector %d pin %d is not valid in line %d column %d

An illegal combination or a pin number which cannot be set is used with the OUT command.

Controller is executing a program or command!

When the controller is executing a command or program it is not available for additional commands. You have to *Break* the new command and re-start it once the previous command has been completely.

Error in array part of file

When re-saving a configuration (e.g. *Controller → Parameters → Restore from file*) the computer recognizes that the data in the array area is formatted incorrectly.

In order to be able to save a file, the following conditions must be fulfilled:

– Identical software versions,
– same configuration (e.g. same number of axes),
– in the case that arrays have already been inputted, these must match the ones that are to be saved in terms of type and size.

Error in axis parameter part of file

When re-saving a configuration (e.g. *Controller → Parameters → Restore from file*) the computer recognizes that the data in the area of the axis parameters is formatted incorrectly. The parameter number and the sequence must be correct and numbering must be continuous.

In order to be able to save a file, the following conditions must be fulfilled:

– Identical software versions, that provides same number and order of the parameters,
– same configuration (e.g. same number of axes).

Error in global parameter part of file

When re-saving a configuration (e.g. *Controller → Parameters → Restore from file*) the computer recognizes that the data in the area of the global parameters is formatted incorrectly.

In order to be able to save a file, the following conditions must be fulfilled:

– Identical software versions, that provides same number and order of the parameters,
– same configuration (e.g. same number of axes).

Lost connection to ...

If the FC 300 is turned off or the plug is pulled, etc. the window is disconnected from the FC 300 and the lost connection is registered.

Timeout: no reply from FC

The FC 300 does not answer; check the connection.

Appendix

□ **Parameter Lists**

The parameters are determined by parameter numbers. We recommend using the overview as a guide; then you will be able to find detailed information very quickly using the number. Please see Parameter Reference in the MCO Design Guide for more details.

Changes during operation

"TRUE" means that the parameter can be changed, while the frequency converter is in operation.

"FALSE" means that the frequency converter must be stopped before a change can be made.

4-Set-up

"1-Set-up": Data value will be the same in all set-ups.

Conversion index

This number refers to a conversion figure used when writing or reading by means of a frequency converter.

Please see for all conversion indices the FC 300 Design Guide.

Conversion index	0
Conversion factor	1

Data type

Please see for all data types the FC 300 Design Guide.

Data type	Description	Type
2	Integer 8	Int8
3	Integer 16	Int16
4	Integer 32	Int32
5	Unsigned 8	Uint8
6	Unsigned 16	Uint16
7	Unsigned 32	Uint32

__ Appendix __

□ Application Parameters, Parameter List

Par. No. #	Parameter name	Parameter description	Default setting	Changes during operation	4-set-up	Conver-sion index	Type
19-0*	**Application Parameters**						
19-00		Application parameters	0	TRUE			
…				TRUE			
19-89		Application parameters	0	TRUE			
19-9*	**Read-only Application Parameters**						
19-90		Application parameter 90	0	read only	1 set-up	0	Int32
19-91		Application parameter 91	0	read only	1 set-up	0	Int32
19-92		Application parameter 92	0	read only	1 set-up	0	Int32
19-93		Application parameter 93	0	read only	1 set-up	0	Int32
19-94		Application parameter 94	0	read only	1 set-up	0	Int32
19-95		Application parameter 95	0	read only	1 set-up	0	Int32
19-96		Application parameter 96	0	read only	1 set-up	0	Int32
19-97		Application parameter 97	0	read only	1 set-up	0	Int32
19-98		Application parameter 98	0	read only	1 set-up	0	Int32
19-99		Application parameter 99	0	read only	1 set-up	0	Int32

□ MCO Basics Settings, Parameter List

Par. No. #	Parameter name	Parameter description	Default setting	Changes during operation	4-set-up	Conver-sion index	Type
32-0*	**Encoder 2 - Slave**						
32-00	ENCODERTYPE	Incremental Signal Type	[1] RS422	TRUE	1 set-up		Uint8
32-01	ENCODER	Incremental Resolution	1024 PPR	TRUE	1-set-up		Uint32
32-02	ENCODER ABSTYPE	Absolute Protocol	[0] None	TRUE	1-set-up		Uint8
32-03	ENCODER ABSRES	Absolute Resolution	8192 PPR	TRUE	1-set-up		Uint32
32-05	ENCODER ABSTYPE	Abs. Encoder Data Length	25 Bit	TRUE	1-set-up		Uint8
32-06	ENCODERFREQ	Absolute Encoder Clock Frequency	262.000 kHz	TRUE	1-set-up		Uint32
32-07	ENCODER CLOCK	Absolute Encoder Clock Generation	[1] On	TRUE	1-set-up		Uint8
32-08	ENCODER DELAY	Absolute Encoder Cable Length	0	TRUE	1-set-up		Uint16
32-09	ENCODER MONITORING	Encoder Monitoring	[0] Off	TRUE	1-set-up		Uint8
32-10	POSDRCT	Rotational Direction	[1] No action	TRUE	1-set-up		Uint8
32-11	POSFACT_N	User Unit Denominator	1	TRUE	1-set-up		Uint32
32-12	POSFACT_Z	User Unit Numerator	1	TRUE	1-set-up		Uint32

∗ default setting   [ ] value for use in communication via serial communication port

__ Appendix __

Par. No. #	Parameter name	Parameter description	Default setting	Changes during operation	4-set-up	Conversion index	Type
32-3*	**Encoder 1 - Master**						
32-30	MENCODER TYPE	Incremental Signal Type	[1] RS422	TRUE	1-set-up		Uint8
32-31	MENCODER	Incremental Resolution	1024 PPR	TRUE	1-set-up		Uint32
32-32	MENCODER ABSTYPE	Absolute Protocol	[0] None	TRUE	1-set-up		Uint8
32-33	MENCODER ABSRES	Absolute Resolution	8192 PPR	TRUE	1-set-up		Uint32
32-35	MENCODER DATLEN	Abs. Encoder Data Length	25 Bit	TRUE	1-set-up		Uint8
32-36	MENCODER FREQ	Absolute Encoder Clock Frequency	262.000 kHz	TRUE	1-set-up		Uint32
32-37	MENCODER CLOCK	Absolute Encoder Clock Generation	[1] On	TRUE	1-set-up		Uint8
32-38	MENCODER DELAY	Abs. Encoder Cable Length	0	TRUE	1-set-up		Uint16
32-39	MENCODER MONITORING	Encoder Monitoring	[0] Off	TRUE	1-set-up		Uint8
32-40	MENCODER TERM	Encoder Termination	[1] On	TRUE	1-set-up		Uint8
32-5*	**Feedback Source**						
32-50	Source Slave		[2] Enc2	TRUE	1-set-up		
32-6*	**PID-Controller**						
32-60	KPROP	Proportional Factor	30	TRUE	1-set-up	0	Uint32
32-61	KDER	Derivative Value for PID Control	0	TRUE	1-set-up	0	Uint32
32-62	KINT	Integral Factor	0	TRUE	1-set-up	0	Uint32
32-63	KILIM	Limit Value for Integral Sum	1000	TRUE	1-set-up	0	Uint16
32-64	BANDWIDTH	PID Bandwidth	1000	TRUE	1-set-up	0	Uint16
32-65	FFVEL	Velocity Feed-forward	0	TRUE	1-set-up	0	Uint32
32-66	FFACC	Acceleration Feed-forward	0 %	TRUE	1-set-up	0	Uint32
32-67	POSERR	Maximum Tolerated Position Error	20000 qc	TRUE	1-set-up		Uint32
32-68	REVERS	Reverse Behavior for Slave	[0] Reversing	TRUE	1-set-up		Uint8
32-69	TIMER	Sampling Time for PID control	1 ms	TRUE	1-set-up		Uint16
32-70	PROFTIME	Scan Time for Profile Generator	[1] 1 ms	TRUE	1-set-up		Uint8
32-71	REGWINMAX	Size of the Control Window (Activation)	0 qc	TRUE	1-set-up		Uint32
32-72	REGWINMIN	Size of the Control Window (Deactivation)	0 qc	TRUE	1-set-up		Uint32
32-8*	**Velocity & Acceleration**						
32-80	VELMAX	Max. Velocity (Encoder)	1500 RPM	TRUE	1-set-up		Uint32
32-81	RAMPMIN	Shortest Ramp	1 s	TRUE	1-set-up		Uint32
32-82	RAMPTYPE	Ramp Type	0	TRUE	1-set-up		Uint8
32-83	VELRES	Velocity Resolution	100	TRUE	1-set-up		Uint16

∗ default setting   [ ] value for use in communication via serial communication port

__ Appendix __

Par. No. #	Parameter name	Parameter description	Default setting	Changes during operation	4-set-up	Conver-sion index	Type
32-84	DFLTVEL	Default Velocity	50	TRUE	1-set-up		Uint16
32-85	DFLTACC	Default Acceleration	50	TRUE	1-set-up		Uint16

□ MCO Advanced Settings, Parameter List

Par. No. #	Parameter name	Parameter description	Default setting	Changes during operation	4-set-up	Conver-sion Index	Type
33-0*	**Home Motion**						
33-00	HOME_FORCE	Force HOME	[0] not forced	TRUE	1-set-up		Uint8
33-01	HOME_OFFSET	Zero Point Offset from Home Position	0 qc	TRUE	1-set-up		Int32
33-02	HOME_RAMP	Ramp for Home Motion	10	TRUE	1-set-up		Uint16
33-03	HOME_VEL	Velocity of Home Motion	10	TRUE	1-set-up		Int16
33-04	HOME_TYPE	Behavior during Home Motion	[0] Reverse + Index	TRUE	1-set-up		Uint8
33-1*	**Synchronization**						
33-10	SYNCFACTM	Synchronization Factor Master (M:S)	1	TRUE	1-set-up		Int32
33-11	SYNCFACTS	Synchronization Factor Slave (M:S)	1	TRUE	1-set-up		Int32
33-12	SYNCPOSOFFS	Position Offset for Synchronization	0 qc	TRUE	1-set-up		Int32
33-13	SYNC ACCURACY	Accuracy Window for Position Synchronization	1000 qc	TRUE	1-set-up		Int32
33-14	SYNCVELREL	Relative Slave Velocity Limit	0 %	TRUE	1-set-up		Uint8
33-15	SYNCMARKM	Marker Number for Master	1	TRUE	1-set-up		Uint16
33-16	SYNCMARKS	Marker Number for Slave	1	TRUE	1-set-up		Uint16
33-17	SYNCMPULSM	Master Marker Distance	4096	TRUE	1-set-up		Uint32
33-18	SYNCMPULSS	Slave Marker Distance	4096	TRUE	1-set-up		Uint32
33-19	SYNCMTYPM	Master Marker Type	[0] Enc. Z pos.	TRUE	1-set-up		Uint8
33-20	SYNCMTYPS	Slave Marker Type	[0] Enc. Z pos.	TRUE	1-set-up		Uint8
33-21	SYNCMWINM	Master Marker Tolerance Window	0	TRUE	1-set-up		Uint32
33-22	SYNCMWINS	Slave Marker Tolerance Window	0	TRUE	1-set-up		Uint32
33-23	SYNCMSTART	Start Behavior for Synchronization	[0] Start Funct. 1	TRUE	1-set-up		Uint16
33-24	SYNCFAULT	Marker Number for Fault	10	TRUE	1-set-up		Uint16
33-25	SYNCREADY	Marker Number for Ready	1	TRUE	1-set-up		Uint16
33-26	SYNCVFTIME	Velocity Filter	0 µs	TRUE	1-set-up	0	Int32
33-27	SYNCOFFTIME	Offset Filter Time	0 ms	TRUE	1-set-up		Uint32

∗ default setting   [ ] value for use in communication via serial communication port

__ Appendix __

Par. No. #	Parameter name	Parameter description	Default setting	Changes during operation	4-set-up	Conver-sion Index	Type
33-28	SYNCMFPAR	Marker Filter Configuration	[0] Marker Filter 1	TRUE	1-set-up		Uint8
33-29	SYNCMFTIME	Filter Time for Marker Correction	0 ms	TRUE	1-set-up		Int32
33-30	SYNCM MAXCORR	Max. Marker Correction	[0] Off	TRUE	1-set-up		Uint32
33-31	SYNCTYPE	Synchronization Type	[0] Standard	TRUE	1-set-up		Uint8
33-4*	**Limit Handling**						
33-40	ENDSWMOD	Behavior at End Limit Switch	[0] Call error handler	TRUE	1-set-up		Uint8
33-41	NEGLIMIT	Negative Software End Limit	-500000 qc	TRUE	1-set-up		Int32
33-42	POSLIMIT	Positive Software End Limit	500000 qc	TRUE	1-set-up		Int32
33-43	SWNEGLIMACT	Negative Software End Limit Active	[0] Inactive	TRUE	1-set-up		Uint8
33-44	SWPOSLIMACT	Positive Software End Limit Active	[0] Inactive	TRUE	1-set-up		Uint8
33-45	TESTTIM	Time in Target Window	0 ms	TRUE	1-set-up		Uint8
33-46	TESTVAL	Target Window Limit Value	1 qc	TRUE	1-set-up		Uint16
33-47	TESTWIN	Size of Target Window	0 qc	TRUE	1-set-up		Uint16
33-5*	**I/O Configuration**						
33-50	I_FUNCTION_1	Terminal X57/1 Digital Input	[0] no function	TRUE	1-set-up		Uint8
33-51	I_FUNCTION_2	Terminal X57/2 Digital Input	[0] no function	TRUE	1-set-up		Uint8
33-52	I_FUNCTION_3	Terminal X57/3 Digital Input	[0] no function	TRUE	1-set-up		Uint8
33-53	I_FUNCTION_4	Terminal X57/4 Digital Input	[0] no function	TRUE	1-set-up		Uint8
33-54	I_FUNCTION_5	Terminal X57/5 Digital Input	[0] no function	TRUE	1-set-up		Uint8
33-55	I_FUNCTION_6	Terminal X57/6 Digital Input	[0] no function	TRUE	1-set-up		Uint8
33-56	I_FUNCTION_7	Terminal X57/7 Digital Input	[0] no function	TRUE	1-set-up		Uint8
33-57	I_FUNCTION_8	Terminal X57/8 Digital Input	[0] no function	TRUE	1-set-up		Uint8
33-58	I_FUNCTION_9	Terminal X57/9 Digital Input	[0] no function	TRUE	1-set-up		Uint8
33-59	I_FUNCTION_10	Terminal X57/10 Digital Input	[0] no function	TRUE	1-set-up		Uint8
33-60	IOMODE	Terminal X59/1 and X59/2 Mode	[0] Output	FALSE	1-set-up		Uint8
33-61	I_FUNCTION_11	Terminal X57/11 Digital Input	[0] no function	TRUE	1-set-up		Uint8

∗ default setting   [ ] value for use in communication via serial communication port

Par. No. #	Parameter name	Parameter description	Default setting	Changes during operation	4-set-up	Conver-sion Index	Type
33-62	I_FUNCTION_12	Terminal X57/12 Digital Input	[0] no function	TRUE	1-set-up		Uint8
33-63	O_FUNCTION_1	Terminal X59/1 Digital Output	[0] no function	TRUE	1-set-up		Uint8
33-64	O_FUNCTION_2	Terminal X59/2 Digital Output	[0] no function	TRUE	1-set-up		Uint8
33-65	O_FUNCTION_3	Terminal X59/3 Digital Output	[0] no function	TRUE	1-set-up		Uint8
33-66	O_FUNCTION_4	Terminal X59/4 Digital Output	[0] no function	TRUE	1-set-up		Uint8
33-67	O_FUNCTION_5	Terminal X59/5 Digital Output	[0] no function	TRUE	1-set-up		Uint8
33-68	O_FUNCTION_6	Terminal X59/6 Digital Output	[0] no function	TRUE	1-set-up		Uint8
33-69	O_FUNCTION_7	Terminal X59/7 Digital Output	[0] no function	TRUE	1-set-up		Uint8
33-70	O_FUNCTION_8	Terminal X59/8 Digital Output	[0] no function	TRUE	1-set-up		Uint8
33-8*	**Global Parameters**						
33-80	PRGPAR	Activated Program Number	-1	TRUE	1-set-up	0	Uint8
33-81	Power-up State	Power-up State	[1] Motor on	TRUE	1-set-up		Uint8
33-82	STATUS MONITORING	Drive Status Monitoring	[1] On	TRUE	1-set-up	0	Uint8
33-83	ERRCOND	Behavior after Error	[0] Coast	TRUE	1-set-up		Uint8
33-84	ESCCOND	Behavior after Escape	[0] Contr. Stop	TRUE	1-set-up		Uint8
33-85	EXTERNAL24V	MCO Supplied by External 24VDC	[0] No	TRUE	1-set-up		Uint8

□ **MCO Data Readouts, Parameter List**

Par. No. #	Parameter Name	Parameter description	Default setting	Changes during operation	4-Setup	Conver-sion Index	Type
34-0*	**PCD Write Parameters**						
34-01		PCD 1 Write to MCO			1-set-up		Uint16
34-02		PCD 2 Write to MCO			1-set-up		Uint16
34-03		PCD 3 Write to MCO			1-set-up		Uint16
34-04		PCD 4 Write to MCO			1-set-up		Uint16
34-05		PCD 5 Write to MCO			1-set-up		Uint16
34-06		PCD 6 Write to MCO			1-set-up		Uint16
34-07		PCD 7 Write to MCO			1-set-up		Uint16
34-08		PCD 8 Write to MCO			1-set-up		Uint16
34-09		PCD 9 Write to MCO			1-set-up		Uint16

∗ default setting   [ ] value for use in communication via serial communication port

Par. No. #	Parameter Name	Parameter description	Default setting	Changes during operation	4-Setup	Conver-sion Index	Type
34-10		PCD 10 Write to MCO			1-set-up		Uint16
34-2*	**PCD Read Parameters**						
34-21		PCD 1 Read from MCO			1-set-up		Uint16
34-22		PCD 2 Read from MCO			1-set-up		Uint16
34-23		PCD 3 Read from MCO			1-set-up		Uint16
34-24		PCD 4 Read from MCO			1-set-up		Uint16
34-26		PCD 6 Read from MCO			1-set-up		Uint16
34-27		PCD 7 Read from MCO			1-set-up		Uint16
34-28		PCD 8 Read from MCO			1-set-up		Uint16
34-29		PCD 9 Read from MCO			1-set-up		Uint16
34-30		PCD 10 Read from MCO			1-set-up		Uint16
34-4*	**Inputs & Outputs**						
34-40		Digital Inputs			1-set-up		Uint16
34-41		Digital Outputs			1-set-up		Uint16
34-5*	**Process Data**		**Unit**				
34-50		Actual Position	UU		1-set-up		Int32
34-51		Commanded Position	UU		1-set-up		Int32
34-52		Actual Master Position	qc		1-set-up		Int32
34-53		Slave Index Position	UU		1-set-up		Int32
34-54		Master Index Position	qc		1-set-up		Int32
34-55		Curve Position			1-set-up		Int32
34-56		Track Error	UU		1-set-up		Int32
34-57		Synchronizing Error	UU		1-set-up		Int32
34-58		Actual Velocity	UU/s		1-set-up		Int32
34-59		Actual Master Velocity	qc/s		1-set-up		Int32
34-60		Synchronizing Status			1-set-up		Int32
34-61		Axis Status			1-set-up		Int32
34-62		Program Status			1-set-up		Int32
34-7*	**Diagnosis Readouts**						
34-70		MCO Alarm Word 1		FALSE	1-set-up		Uint32
34-71		MCO Alarm Word 2		FALSE	1-set-up		Uint32

∗ default setting   [ ] value for use in communication via serial communication port

□ **Index**

__ Appendix __