



APPLICATION*NOTE* 185

Operating a MC V3.0 EtherCAT driver as a CODESYS SoftMotion drive

Summary

This Product Application Note describes how to use a FAULHABER MC V3.0 EtherCAT driver connected to a CODESYS SoftMotion axis. A SoftMotion axis can be used to generate the complete motion trajectory directly within the master, different from a drive operated in Profile Position Mode where the built-in profile-generator does this job. Here the general setup of the CODESYS configuration is explained for a single axis system as well as the expected settings on the drive side.

Applies To

Any FAULHABER MC V3.0 EtherCAT drive MCS32xx BX4 ET, MCS3274 BP4 ET MC 5004 P ET, MC 5005 S ET, MC 5010 S ET

Description

Before a drive can be configured to be used as a SoftMotion drive, a CODESYS project as to be created and at least one CiA402 ET or CANopen¹ drive is to be connected to the system. In this example again a RaspberryPi is used as the CODESYS master whereas a FAULHABER MC 5005 S ET drive is used as the servo to be commanded.

General information about how to install the CODESYS environment on a RaspberryPi and to create a first project driving a FAULHABER MC V3.0 ET drive can be found in our Product Application Note 164 at our support page www.faulhaber.com/en/support/drive-electronics.

Prerequisites

Before starting to set-up your CODESYS environment please use the MotionManager to configure your drive:

- Select the motor and feedback-system using "Select motor" tool
- Configure the load inertia using the "Configure controller" tools
- Tune the control-loops
- Verify the dynamic parameters like acceleration and deceleration which can be safely used in your application by operating the drive using the MotionManager MotionCockpit
- Configure whatever additional I/Os might be used e.g. for a drive-based homing sequence
- Save your settings in the drive and store a copy of it at your PC

CODESYS project configuration

The general setup is the same as detailed in AppNote 164:

• Create a new project

¹ Use of a CiA402 CANopen servo-drive has not been tested. Please note SoftMotion will require high update-rates of the intermediate demand-values anyway, so ET might be the better choice.



- Select the correct target environment here it is a RaspberryPi
- Connect to the target device
- Add an EtherCAT Master to your project and configure the network interface to be used on your target device
- Add the FALUHABER MC as a subsystem to the EtherCAT master and check the communication settings
- Start thinking about the process-image, which is the collection of parameters to be exchanged cyclically when the system is running.

After these steps we end up with the PDO mappings page of the drive configuration. But before configuring the PDOs we add the SoftMotion axis.

Add a SoftMotion Axis to the drive

The SoftMotion CiA 402 Axis is added to the MC 5005 using the context menu of the driver in the project tree, Figure 1.



Whenever the drive is identified by CODESYS to be a CiA 402 servo-drive this option is visible in the context menu.

SoftMotion_ET_ConfigExample.proje	ct*	- CODESYS					
Eile Edit <u>Vi</u> ew <u>P</u> roject <u>B</u> uild <u>O</u> nl	ine	Debug Tools Window Help	2				
🎦 🚅 🖶 I 🎒 🗠 🗠 🔏 🛅 🛙	1	X M 😘 🖷 🛅 - 👘	🕮 I 📽 🧐 🕞 📲 🛠 I 🗔 🤊	1418 ¢ % =			
Devices		→ # ×	Device X				-
SoftMotion_ET_ConfigExample			Communication Collins	Scan Network Gateway	Device -		
Device (CODESYS Control for R	aspt	perry Pi SL)	Communication Settings	dateway •	Device		
Application			Applications				
Library Manager			Backup and Restore		•		:
PLC_PRG (PRG)							
i⊒ v MainTask			Files				•
EtherCAT_	Mas	ter.EtherCAT_Task	Log		Gatev	way	-
PLC_PRG	тм	actor)			Gateway-1	-	[0301.B0AB] (active) •
Etiercal_master (Etierca		aster)	PLC Settings		IP-Address: localhost		Device Name: RasPi3Bminus
OpModeDcSynchr	*	Cut	Shell		Port:		Device Address:
SoftMotion General Axis P Copy		Сору	is and Groups		1217		0301.B0AB
-àI²C (≧ Pa àSPI X D	Paste					Target ID:	
SPI SPI GPIOs_A_B (GPIOs A/B) Delete		Delete	Deployment				
a Onewire		Drowse Drowse	JS				4102
- 3 Onewire - 3 Camera device - ↓ <empty> (<empty>) -</empty></empty>		Refactoring					Target Vendor:
	•	Properties	mation				35 - Smart Software Solutions GmbH
		Add Object					Target Version: 3.5.10.20
		Add Folder					
		Disable Device					
		Update Device					
	î	Edit Object					
		Edit Object With					
		Edit IO mapping					
		Import mappings from CSV					
		Export mappings to CSV					
		Add SoftMotion CiA402 Axis					
<u>۲</u>		Add SoftMotionLight CiA402 Axis					
Sevices POUs		Simulation					
Messages - Total 12 error(s), 0 warni	ng(s), 0 message(s)					
					Last build: 🧿 0 😗 0	Precompile: 🗸	Current user: (nobody)

Figure 1 Add a CiA402 SoftMotion Axis to the MC 5005 S ET





FAULHABER drivers have not been tested at CODESYS so afterwards a warning will pop up – which can be ignored.

Adjust the settings of the SoftMotion drive to your axis

In the project tree the SoftMotion axis is organized as a sub-system of the MC 5005. There are two configuration tabs of the SoftMotion axis which are to be considered.

Scaling of the motion

As for the scaling there are two tabs where the necessary information must be entered. Most important is the correct scaling of the drive position or any transmission or gearhead being used. This information is entered on top of the Scaling/Mapping tab in Figure 3.

What is required	How to be set			
Increments per motor turn What is the encoder resolution in increments per motor turn?	This is the resolution that has been set in the drive. Default setting would be the native resolution of the motor feedback. Changing the settings of the Factor Group inside the MC is not recommended, the parameters of the transmission can be entered at the SoftMotion axis.			
	The value can be entered either hex (16#10000) or decimal. For a typical 12-bit feedback as e.g. our lin- ear hall sensors we would enter 4096 increments per turn.			
Gear reduction ratio	If the gearhead or transmission has an integer re- duction ratio it can be entered directly.			
the output shaft of the gearhead?	When there is a non-integer reduction ratio like $63/17 = 3.705$ for a 4:1 22F GH enter the numerator – the bigger number – at the motor side and the denominator – the smaller number – on the gear output side.			
Units to be used How do you want to count the position within your application? Default is 1.0 per one output turn.	If you got a linear system you might want to scale the position to be used by the pitch of a screw. So e.g. for a 1mm pitch a setting of 1 will have you po- sitions in mm because at one turn of the shaft the nut will move by 1mm.			
	If you got a rotating system a scaling in ° might also be a good option. So, use 360 units in your applica- tion per turn to have all your positions being in °.			

Next would be the dynamic limits of the drive (Figure 2), which is the maximum velocity [u/s], acceleration $[u/s^2]$, deceleration $[u/s^2]$ and jerk $[u/s^3]$.



These will be used to limit the values used in the application. The values can easily be taken from the drive configuration within the MotionManager, after these have been tested.

Control parameters	Velocity [u/s]
Position Velocity Current Profile parameters	Here this is the maximum speed in turns per s.
Values are sent immediately after a change!	Can be calculated out of the maximum profile ve-
Motion profile type: Linear ramp \sim	locity in the control parameters window of the Mo-
Max motor speed : 200000	tionManager.
Profile velocity:	Velocity $[u/s] = \frac{Profile \ velocity}{Velocity}$
Profile acceleration: 13100 1/s ²	60
Profile deceleration: 3930 1/s²	Acceleration [u/s ²]
Quick stop deceleration: 30000 T/S ²	Enter the maximum acceleration which still re- sulted in a satisfying behavior of your axis. Maxi- mum would be the Profile acceleration as in the control parameters window of the MotionManger.
	Deceleration [u/s ²]
	Enter the maximum deceleration which still re- sulted in a satisfying behavior of your axis. Maxi- mum would be the Profile acceleration as in the control parameters window of the MotionManger.
	Jerk [u/s³]
	The FAULHABER drivers don't limit the jerk so there is no value that can be transferred 1:1.
	Depends on the stiffness of the application and on the selected motion profile.
	Numbers are usually higher than the ones for accel- eration as the jerk is the derivative of the accelera- tion.



SoftMotion_ET_ConfigExample.project* - CODESYS	Y + [master [2]]	1 2 4 4 6	A 13 40	• B O		
Eile Edit View Project Build Online Debug Tools Window Help						
1111121121111111111111111111111111111	1 % 🧐 🕞 📲 🖎 l 🗇 🕾 🖆	*≣ 8 ≎ ∭	≓'			
Devices 👻 🕂 🗙	Device SM_Drive_Ge	nericDSP402 🗙				-
SoftMotion_ET_ConfigExample	General	Axis type and limits	Software limits			Velocity ramp type
General PLC Logic	Scaling/Mapping	Virtual mode Modulo	Activated	Negative [u]:	0.0	 Trapezoid Sin²
LiDrary Manager El PLC_PRG (PRG)	Commissioning	Finite		Positive [u]:	1000.0	Quadratic
Task Configuration Section MainTask	SM_Drive_ETC_GenericDSP402: I/O Mapping		Software error reac	tion Deceleration [u/s ²]:	0	Quadratic (smooth)
EtherCAT_Master.EtherCAT_Task D C C C C C	Status			Max. distance [u]:	0	ID: 0
- III EtherCAT_Master (EtherCAT Master)	Information	-Dynamic limits				Position lag supervision
MC5005 (MC5005 MotionController) MC5005 (MC5005 Vnchronous (Operation mode DC-Synchronous)		Velocity [u/s]:	Acceleration [u/s ²]	Deceleration [u/s ²]	Jerk [u/s³]:	deactivated
SM_Drive_GenericDSP402 (SM_Drive_GenericDSP402)		30	1000	1000	10000	Lag limit [u]: 1.0
SoftMotion General Axis Pool SoftMotion General Axis Pool SoftMotion General Axis Pool						
- à Onewire						
Camera device C						
Cevices POUs						
Messages - Total 12 error(s), 0 warning(s), 0 message(s)						
			Last build:	O 0 O Precom	pile: 🗸	Current user: (nobody)

Figure 2 General setting tab for the SoftMotion axis

PDO Mapping

After the Scaling, it is the Mapping of the process-image that must be considered. The Scaling/Mapping tab of the SoftMotion drive (Figure 3) gives an overview which parameters of a CiA 402 CANopen servo-drive would be used, if available. Plus, there is a checkbox to facilitate automated mapping. This mapping here applies for the mapping between the process image created by the PDOs of the drive and the connected SoftMotion axis only. It makes use of whatever is available in the process image created by the PDOs configured but it will not generate the PDOs configuration. That is still to be done and selected by hand depending on the application.



SofMation_FC_confgleample Image: Configuration Image: Configuration Image: Configuration Image:	SoftMotion_ET_ConfigExample If Device (CODESYS Control for Raspberry Pi SL) Image: Device (CODESYS Control for Raspberry Pi SL)							_	
Scaling/Mapping 16#1000 increments <=> motor turns Image: Pice RG (PRG) Image: Pice RG (PRG) Image: Pice RG (PRG) Image: Pice RG (PRG) Image: Pice RG (PRG) Image: Pice RG (PRG) Image: Pice RG (PRG) Image: Pice RG (PRG) Image: Pice RG (PRG) Image: Pice RG (PRG) Image: Pice RG (PRG) Image: Pice RG (PRG) Image: Pice RG (PRG) Image: Pice RG (PRG) Image: Pice RG (PRG) Image: Pice RG (PRG) Image: Pice RG (PRG) Image: Pice RG (PRG) Image: Pice RG (PRG) Image: Pice RG (PRG) Image: Pice RG (PRG) Image: Pice RG (PRG) Image: Pice RG (PRG) Image: Pice RG (PRG) Image: Pice RG (PRG) Image: Pice RG (PRG) Image: Pice RG (PRG) Image: Pice RG (PRG) Image: Pice RG (PRG) Image: Pice RG (PRG) Image: Pice RG (PRG) Image: Pice RG (PRG) Image: Pice RG (PRG) Image: Pice RG (PRG) Image: Pice RG (PRG) Image: Pice RG (PRG) Image: Pice RG (PRG) Image: Pice RG (PRG) Image: Pice RG (PRG) Image: Pice RG (PRG) Image: Pice RG (PRG) Image: Pice RG (PRG) Image: Pice RG (PRG) Image: Pice RG (PRG) Image: Pice RG (PRG)		General	Scaling Invert direction						
Image: Commissioning 1 motor turns <=> gear output turns Image: Commissioning SM_Drive_ETC_GenericDSP402:10 1 gear output turns <=> units in application Image: Commissioning SM_Drive_ETC_GenericDSP402:10 Mapping Image: Commissioning Image: Commissioning Image: Commissioning SM_Drive_ETC_GenericDSP402:10 Mapping Image: Commissioning Image: Commissioning Image: Commissioning Status Image: Commissioning Image: Commissioning Image: Commissioning Image: Commissioning Status Image: Commissioning Image: Commissioning Image: Commissioning Image: Commissioning Status Image: Commissioning Status Image: Commissioning Image: Commissioning Status Image: Commissioning Image: Commissioning Image: Commissioning Image: Commissioning Status Image: Commissioning Image: Commissioning Image: Commissioning Image: Commissioning Status Image: Commissioning Image: Commissioning Image: Commissioning Image: Commissioning Status Image: Commissioning Image: Commissioning Image: Commissioning Image: Commissioning Status	C Application	Scaling/Mapping	16#10000	increments <=> i	motor turns		1		
Mapping Mapping Status Imputs: Cycle object Object number Address Type Status Information Status Imputs: Cycle object Object number Address Type Status Information Status Imputs: Cycle object Object number Address Type Status word (n.wStatusWord) 15#60411:15#00 %LD 'NWC 'NWC' UINT' SoftMotion General Axis Pool Status Information Status 'Nacture (MacVeolty) 15#6041:15#00 %LD 'NWC 'Nutr' SoftMotion General Axis Pool Status Information Status word (n.wStatusWord) 15#6041:15#00 'Nactus 'Nactus Vord' 'Nactus Vord'' 'Nactus Vord'' <td< td=""><th>- 🎁 Library Manager - 📄 PLC_PRG (PRG) 🖻 🌃 Task Configuration</th><td>Commissioning</td><td>1 m 1 gear</td><td>output turns <=> ge</td><td>ar output turr units in applic</td><td>ation</td><td>1</td><td></td><td>Scalin</td></td<>	- 🎁 Library Manager - 📄 PLC_PRG (PRG) 🖻 🌃 Task Configuration	Commissioning	1 m 1 gear	output turns <=> ge	ar output turr units in applic	ation	1		Scalin
Status Industry Dispersion Import Status Information Inputs: Information Object number Import Import	🖹 🍪 MainTask	Mapping	Mapping						
BetherCAT_Master (EtherCAT_Master) Information Information Information Cyclic object Object number Address Type Status word (n.wStatusWord) 16#6041116#00 96LW0' UINT' actual position (dActPosition) 16#6041116#00 96LW0' UINT' SoftMotion General Axis Pool if SoftMotion Galeral Axis Pool if if 000000000000000000000000000000000000	EtherCAT_Master.EtherCAT_Task	Status	Automatic mapping						
Image: Control of the second of the secon	EtherCAT Master (EtherCAT Master)	Information	Inputs:						
# OpModeDcSyndronous (Operation mode DC-Syndronous) Status word (n.wStatusWord) 16#6041:16#00 %GUV0 UINT a SoftWords General Axis Pool Status word (n.wStatusWord) 16#6041:16#00 %GUU0 WINT a SoftWords General Axis Pool Status word (n.wStatusWord) 16#6041:16#00 %GUU0 WINT a SoftWords General Axis Pool Status word (n.wStatusWord) 16#6041:16#00 %GUU0 WINT a Status word (n.wStatusWord) 16#6041:16#00 %GUU0 WINT actual solotity (dactWordsWord) 16#6041:16#00 %UU0 WINT a Status word (n.wStatusWord) 16#6041:16#00 %GUU0 Wint 0 %GUU0 %GUU0 %GUU0 %GUU0 %GUU0 %GUU0 0 %GUU0 %GUU1 %GUU1 %GUU0 %GUU	MC5005 (MC5005 MotionController)	Inomation	Cyclic object	Object number	Address	Туре		*	
By Sh_Dhive_GenericDSP402 (SM_Drive_GenericDSP402) actual position (dArtPosition) 15#00541.5#00 ** SoftMotion General Axis Pool actual position (dArtPosition) 15#00541.5#00 * * SoftMotion General Axis Pool actual position (dArtPosition) 15#0075.15#00 * * SoftMotion General Axis Pool actual position (dArtPosition) 15#0071.15#00 * * SoftMotion General Axis Pool actual position (dArtPosition) 15#0071.15#00 * * SoftMotion General Axis Pool actual position (dArtPosition) 15#0071.15#00 * * SoftMotion General Axis Pool actual position (dArtPosition) 15#0071.15#00 * * SoftMotion General Axis Pool actual position (dArtPosition) 15#0071.15#00 * * SoftMotion General Axis Pool actual position (dArtPosition) 15#0081.15#00 * * Concherols I falling edge 15#0081.15#00 * * * Touch Probe 1 falling edge 15#0081.15#00 * * * Touch Probe 2 faling edge 15#0081.15#00 *	OpModeDcSynchronous (Operation mode DC-Synchronous		status word (in.wStatusWord)	16#6041:16#00	'%IW0'	'UINT'			
> SoftMotion General Axis Pool actual velocity (dActVelocity) 15#605:15#00 " > 1C sp1 if=607:15#00 " datual torque (wActTorque) 16#607:15#00 " > GriOs_A_B (cPIOS A/B) if=607:15#00 " " " > Onewire if=607:15#00 " " > Onewire if=607:15#00 " " > Camera device if=6081:15#00 " " Camera device if=6081:16#00 " " Touch Probe 1 falling edge if=6080:15#00 " " Touch Probe 2 falling edge if=6080:15#00 " " Outputs: Cyclic object Object number Address Type ControlWord (out.wControlWord) if=6071:15#00 " " set position (disetPosition) if=6071:15#00 " " et torque (wSetTorque) if=6071:15#00 "	SM Drive GenericDSP402 (SM Drive GenericDSP402)		actual position (diActPosition)	16#6064:16#00	'%ID1'	'DINT'			
¹ IC ¹ S IIC 	SoftMotion General Axis Pool		actual velocity (diActVelocity)	16#606C:16#00		"			
¹ / ₈ SPI ¹ / ₈ GPIOs A, B (GPIOs A, B) ¹ / ₈ GPIOs A, B (GPIOs A, B) ¹ / ₈ GPIOs A, B (GPIOs A, B) ¹ / ₈ GPIOs A, B (GPIOs A, B) ¹ / ₈ GPIOs A, B (GPIOs A, B) ¹ / ₈ GPIOs A, B (GPIOs A, B) ¹ / ₈ GPIOs A, B (GPIOs A, B) ¹ / ₈ GPIOs A, B (GPIOs A, B) ¹ / ₈ GPIOs A, B (GPIOs A, B) ¹ / ₈ GPIOs A, B (GPIOs A, B) ¹ / ₈ GPIOs A, B (GPIOs A, B) ¹ / ₈ GPIOs A, B (GPIOs A, B) ¹ / ₈ GPIOs A, B (GPIOs A, B) ¹ / ₈ GPIOs A, B (GPIOs A, B) ¹ / ₈ <t< td=""><th>≥ I²C</th><td></td><td>actual torque (wActTorque)</td><td>16#6077:16#00</td><td></td><td>"</td><td></td><td></td><td></td></t<>	≥ I ² C		actual torque (wActTorque)	16#6077:16#00		"			
GPIOs_A_B (GPIOs A,B) idigital inputs (m.d.vologitalinputs) id=60F0:15=00 " " Sonewire Camera device 16=60F0:15=00 " " Camera device Touch Probe 1 falling edge 16=60B1:15=00 " " Touch Probe 1 falling edge 16=60B2:15=00 " " " Touch Probe 1 falling edge 16=60B2:15=00 " " " Touch Probe 1 falling edge 16=60B2:15=00 " " " Touch Probe 1 falling edge 16=60B2:15=00 " " " Touch Probe 2 falling edge 16=60B2:15=00 " " " Outputs: ControlWord (out.wControlWord) 16=60B2:15=00 " " ControlWord (out.wControlWord) 16=60B2:16=00 " " " et position (disetPosition) 16=6040:16=00 " " " Outputs: ControlWord (out.wControlWord) 16=6071:16=00 " " et position (disetPosition) 16=6071:16=00 " " " et position (disetPosition) 16=6071:16=00 " " "	SPI SPI		Modes of operation display (OP)	16#6061:16#00		"		=	Mappi
¹ Onewire 156-6087:15400 * * ¹ Camera device 156-6087:15400 * * ¹ Camera device 156-6087:15400 * * * ¹ Cuch Probe 1 railing edge 156-6087:156400 * * ¹ Cuch Probe 2 railing edge 156-6087:156400 * * ¹ Cuch Probe 2 railing edge 156-6087:156400 * * ¹ Cuch Probe 2 railing edge 156-6087:156400 * * ¹ Cuch Probe 2 railing edge 156-6087:156400 * * ¹ Cuch Probe 2 railing edge 156-50407:15640 * * ¹ Cuch Probe 2 railing edge 156-5047:15640 * * ¹ Cuch Probe 2 railing edge 156-507:115640 * *	Bo GPIOS A B (GPIOS A/B)		digital inputs (in.dwDigitalInputs)	16#60FD:16#00		"			
Camera device Touch Probe 1 rising edge 156408.156400 " " Touch Probe 1 rising edge 156408.156400 " " " Touch Probe 1 rising edge 156408.156400 " " " Touch Probe 1 rising edge 156408.156400 " " " Touch Probe 1 rising edge 156408.156400 " " " Touch Probe 1 rising edge 156408.156400 " " " Outputs: Outputs: Outputs: " " ControlWord (out.wControlWord) 166409.156400 "SQUO" UINT" set position (dis@Position) 156407.156400 "SQUO" UINT" set torque (wSetTorque) 156407.116400 " "	a Onewire		Touch Probe Status	16#60B9:16#00	**	"			
C Empty> (<empty>) Touch Probe 1 ralling edge 16#60801 16#00 " Touch Probe 2 ralling edge 16#60801 16#00 " Touch Probe 2 ralling edge 16#60801 16#00 " Outputs: Cyclic object Object number Address Type ControlWord (out.wControlWord) 16#6040:16#00 " " " Set position (dSetPosition) 16#60740:16#00 " " " Set position (dSetPosition) 16#60711:16#00 " " "</empty>	🖹 🖕 Camera device		Touch Probe 1 rising edge	16#60BA:16#00					
Louch Probe 2 rising edge 16=608C: 16=00 " Touch Probe 2 failing edge 16=608C: 16=00 " " Outputs: Cyclic object Object number Address Type ControlWord (out.wControlWord) 16=6040: 16=00 "%CW0" 'UINT' set position (dis@Position) 16=60740: 16=00 "%CD1" 'DINT' set torque (/xGetVredotty) 16=6071: 16=00 " "	<pre><code code="" code<="" th=""><td></td><td>Touch Probe 1 falling edge</td><td>16#60BB:16#00</td><td></td><td></td><td></td><td></td><td></td></code></pre>		Touch Probe 1 falling edge	16#60BB:16#00					
Outputs: Object number Address Type ControlWord (out.wControlWord) 16#6049:16#00 "%cQW0" UINT" set position (disetPosition) 16#6074:16#00 "%cQU1" DINT" set velocity (disetVelocity) 16#6077:16#00 " " set torque (wSetTorque) 16#6071:16#00 " "			Touch Probe 2 rising edge	16#60BC:16#00					
Outputs: Object number Address Type ControlWord (out.wControlWord) 16=6049:16=00 %cQW0' UINT' set position (diseVealation) 16=6074:16=00 %cQW0' UINT' set velocity (diseVealation) 16=6071:16=00 " " set torque (wSetTorque) 16=6071:16=00 " "			Touch Probe 2 falling edge	16#60BD:16#00				-	
Cyclic object Object fundicer Address Type ControlWord (out.wControlWord) 16#6040:16#00 '96Q00' 'UINT' set position (discProsition) 16#6074:16#00 '96Q0' 'UINT' set velocity (dSetVelocity) 16#6076:16#00 " " set torque (wSetTorque) 16#6071:16#00 " "			Outputs:	Ohiostaushaa		T			
ControlWord (ult: WoohrdWord) 15=50/41(5):5=500 "M2Q/W0" JUNI" set position (disEtPosition) 16=6074:16=700 "M" "M" set veloitly (disEtVeloitly) 16=6071:16=700 "M" "M"			Cyclic object	Object number	Address	Type			
set position (abservation) 15+50/47.15+700 "V4_011 D1M1 set velocity (d544Velocity) 15+50/7F1:56-90 " " set torque (wSetTorque) 15+56071:16+900 " "			ControlWord (out.wControlWord)	16#6040:16#00	%QW0'	UINT'			
set veroury (userveroury) 15#50/PF15#00 set torque (wSetTorque) 16#6071:16#00 " "			set position (diset/osition)	10#00/A:10#00	%QD1	UINT			
sectorque (wsectorque) 10#00/1:10#00			set torque (wSetTorque)	16#6071:16#00					
Modes of operation (OP) 15#5050+15#00			Modes of operation (OP)	16#6060:16#00					
Truck probe Function 15#5000/15#00 " "			Touch Probe Function	16#6088:16#00					
Add velocit value 15#5081:15500 " "			Add velocity value	16#6081:16#00					
Add torque value 16#c002:16#00 " "			Add torque value	16#60B2:16#00					
Diotal outputs (A637) 16#A0FF-11#01 " "			Digital outputs (A637)	16#60FE:16#01					

Figure 3 Scalings and mappings of the axis

So, what we must do is, try to make the necessary parameters available.

We do this, by creating an appropriate PDO-mapping for the MC 5005 used here and then un-check and recheck the automatic mapping checkbox.

The procedure for the PDO mapping itself is the very same as in Product Application Note 164.

First enable the expert settings for the MC on the General Settings Tab (Figure 4).



DefaultDemoET_ST.project* - CODESYS				
Eile Edit View Project Build Online Debug Tools Window Help				
◎ ○ □ ● ○ ○ ↓ ■ ◎ × ● ○ ○ ○ ○ ○	1 05 0 8 42 1 (11 91 41)	14 8 6 5 5 5		
Devices _ T_Y	(The Device of Market Star Star	the Vet and December of Vet	MCFOOF	
DefaultDemoET_ST	Device EtherCAT_Ma	ster 🖪 OpmodeDcSynchronous	MC5005 X	
Device (CODESYS Control for Raspberry Pi SL)	General	Address	- Additional	<hr/>
PLC Logic		AutoInc Address 0 🔶 🌔	Enable Expert Setting	as EtherCAT
E O Application	Expert Process Data	EtherCAT Address 1001	Continual	
Library Manager	Process Data			
PLC_PRG (PRG)		Distributed Clock		
I ask Configuration	Startup Parameters	> Startup checking	Dimeouts	
Hairrask Hercat Master EtherCat Task		DC cyclic unit control: assign to local uC		
PLC_PRG	EtherCAT I/O Mapping	be cyclic unic control. assign to local pe		
EtherCAT_Master (EtherCAT Master)	Status	V Watchuog		
🖹 👔 MC5005 (MC5005 MotionController)		Identification		
OpModeDcSynchronous (Operation mode DC-Synchronous)	Information	Oisabled		
SoftMotion General Axis Pool		 Configured Station Alias (ADO 0x0012) 	Value	1001
5 cor				
GPIOS A B (GPIOS A/B)		Explicit Device Identification (ADO 0x0134)		
a Onewire				
🗏 🚡 Camera device		 Data Word (2 Bytes) 	ADO (hex)	16#0 文
<pre>C <empty> (<empty>)</empty></empty></pre>				
Z Devices U POUs				
Messages - Total 0 error(s), 0 warning(s), 0 message(s)				
		Last build: 😲 0 🕐 0 🛛 F	Precompile: 🗸	Current user: (nobody)

Figure 4 General settings of the MC 5005 S ET

DefaultDemoET_ST.project* - CODESYS				
Elle Edit View Project Build Online Debug Tools Window Help				
物産目長しなよ時代×時代には物・学校	10808 v = 4816=9=4=	*= 8 o m =		
	Device EtherCAT_M	aster 🕅 OpModeDcSynchronous	/ MC5005 X	•
Deraurubemoer_ST Device (CODESYS Control for Raspberry Pi SL)	General	Sync Manager:	Add Carlete	
H- I PLC Logic			PD0 List:	
🖹 🚫 Application	Expert Process Data	0 0 Mailbox Out	Index Size Name	Elaar SM
Library Manager	Process Data	1 0 Mailbox In	16#1600 2.0 RxPD01	Tiags Sivi
PLC_PRG (PRG)	Flocess Data	2 6 Outputs	16#1601 6.0 RxPD02	2
Task Configuration	Startup Parameters	3 6 Inputs	16#1602 6.0 RxPD03	
= 💝 MainTask	· · · · · · · · · · · · · · · · · · ·		16#1603 4.0 RxPD 04	
EtherCAT_Master.EtherCAT_Task	EtherCAT I/O Mapping		16#1A00 2.0 TxPDO1	
			16#1A01 6.0 TxPDO2	3
EtherCAI_Master (EtherCAI Master)	Status		16#1A02 6.0 TxPDO3	
MCSOUS (MCSOUS MotionController)	Tefermetica		16#1A03 4.0 TxPDO4	
SoftMation General Avia Roal	mornation			(2)
				9
SPT				
GPIOS A B (GPIOS A/B)		PDO Assignment (16#1C12);	🕂 Insert 📝 Edit 🗙 Delete 🕆 Move Up 🐳 Move Down	
a Onewire		16#1600	PDO Content (16#1600):	
🖻 🚡 Camera device		✓ 16#1601	Index Size Offe Name	Tune
<pre>Comparison Comparison Compar</pre>		16#1602	16#6040000 2.0 0.0 control word	Туре
		16#1603	2.0 0.0 control word	0141
			2.0	
				~
		(1)		(২)
				•
		Download		
		PDO Assignment V PDO Configur	ration Load PDO Info from the Device	
Z vevices IL POUS				
Messages - Total 0 error(s), 0 warning(s), 0 message(s)				-
		Last	build: 😳 0 😗 0 Precompile: 🗸 Current use	r: (nobody) 🛛 🕅

Figure 5 Expert settings for the PDO mapping



Next modify and add the PDOs on the Expert Process Data Tab (Figure 5):

①: activate the number of PDOs required. A maximum of 4 parameters can be mapped into a single PDO. In an ET system there is no limitation in data length, different from a CANopen system.

②: select the PDO to be edited

(3): edit the contents of the selected PDO by removing unwanted objects and adding whatever is desired.

Finally make sure to have the checkboxes to download the PDO Assignment and the PDO configuration activated.



Steps how to configure the PDO mapping can again be found in the Product Application Note 164.



To have CODESYS re-check whether there are new useful parameters for a Soft-Motion axis, un-check and re-check the automatic mapping checkbox in Figure 3.

Most of the parameters used by the SoftMotion axis are available at the FAULHABER drive. Whether to map them or not depends on the application. If no torque mode or no speed mode is used, then mapping these target values is not necessary of course. The respective offsets however are used for the cyclic modes like CSP, nevertheless.

A mapping of the drive could therefor look like:

General	Select the Outputs		Select the Inputs	Select the Inputs			
	Name	Туре	Index	Name	Туре	Index	
Expert Process Data	✓ 16#1600 RxPDO1			✓ 16#1A00 TxPDO1			
Des server Deska	Target velocity	UDINT	16#60FF:00	Touch probe 1 positive edge	UDINT	16#60BA:00	
Process Data	Velocity offset	UDINT	16#60B1:00	Touch probe 1 negative edge	UDINT	16#60BB:00	
Startup Parameters	Target torque	UINT	16#6071:00	Touch probe 2 positive edge	UDINT	16#60BC:00	
	Torque offset	UINT	16#60B2:00	Touch probe 2 negative edge	UDINT	16#60BD:00	
EtherCAT I/O Mapping	✓ 16#1601 RxPDO2			✓ 16#1A01 TxPDO2			
	control word	UINT	16#6040:00	status word	UINT	16#6041:00	
Status	target position	DINT	16#607A:00	position actual value (user scaling)	DINT	16#6064:00	
	Touch probe function	UINT	16#60B8:00	Touch probe status	UINT	16#60B9:00	
Information	Modes of operation	USINT	16#6060:00	Modes of operation display	USINT	16#6061:00	
	16#1602 RxPDO3			✓ 16#1A02 TxPDO3			
	control word	UINT	16#6040:00	Velocity actual value	UDINT	16#606C:00	
	target velocity	DINT	16#60FF:00	Torque actual value	UINT	16#6077:00	
	16#1603 RxPDO4			Digital Inputs	UDINT	16#60FD:00	
	control word	UINT	16#6040:00	16#1A03 TxPDO4			
	target torque	INT	16#6071:00	status word	UINT	16#6041:00	
				torque actual value	INT	16#6077:00	

Figure 6 PDO-Mapping for a servo-drive linked to a SoftMotion axis

The SoftMotion axis will not only care for the target-values but also set the operating mode, use the feedforward offsets of the cyclic modes and seems to be able to use the touch-probe function built into these drives.

Here two RxPDOs and three TxPDOs are used. There are no requirements in how to assign the objects to the PDOs as the CSP mode doesn't require any special sequence.



From Master to	drive (RxPDO)	From drive to N	Master (TxPDO)
0x60FF.00	Target Velocity	0x60BA.00	Touch probe 1 positive edge
0x60B1.00	Velocity Offset	0x60BB.00	Touch probe 1 negative edge
0x6071.00	Target Torque (if used)	0x60BC.00	Touch probe 2 positive edge
0x60B2.00	Torque offset	0x60BD.00	Touch probe 2 negative edge
0x6040.00	Control word	0x6041.00	Status word
0x607A.00	Target position	0x6064.00	Position actual value
0x60B8.00	Touch probe function	0x60B9.00	Touch probe status
0x6060.00	Modes of operation	0x6061.00	Modes of operation display
		0x606C.00	Velocity actual value
		0x6077.00	Torque actual value
		0x60FD.00	Digital Inputs

Special settings in the MotionController

Scaling of the drive velocity

The scalings in Figure 2 and Figure 3 used the drive position in increments and rescaled them to whatever application specific scaling shall be used. When using the speed of the drive the scalings in Figure 2 and Figure 3 use turns/s but that's not how speed-values are expected to be scaled in the process image.

The profile generator built into the SoftMotion axis can generate not only whatever intermediate target position is to be used (position demand) but will also calculate the target speed required for this motion. This speed can be used as a feed-forward value for the control-loop in the FAULHABER MotionContoller – the velocity offset and the torque offset values are used here. The scaling of the velocity used between the drive and CODESYS has then to be in increments/s.

The scaling of the speed and the position can always be done using the factor group. When it is set to the default values, any actual- or target-position would be given in increments per turn of the motor, same scaling as received from the encoder. The speed is internally scaled to min⁻¹.



Default settings are

Drive functions				_		\times
₽ Search	₽	Ŷ				
Operating modes Operating modes Operating modes Operating modes			Device control - Factor group			
 General Motion profile Range limits Factor group Error handling Sequence programs Ambient parameters Signal management Communication 			Position encoder resolution (0x608F) Velocity encoder resolution (0x6090) Encoder increments: 4096 Motor revolutions: 1 Gear ratio (0x6091) Feed constant (0x6092) Motor shaft revolutions: 1 Driving shaft revolutions: 1 Velocity factor (0x6096) Numerator: Numerator: 1 Divisor: 4096			
			Function Active Reverse the speed			
			User units Position Velocity Current: incr. 1/min Reset to standard Standard: incr. 1/min			
			Send Save C	llose	Help)

Figure 7 Default Factor Group settings of the motor 2232 BX4 used here (linear hall sensors)

We don't need to rescale the position as the scaling here has already been entered into the SoftMotion axis in Figure 3. But we need to rescale the speed. The Velocity factor (Figure 7, Figure 8) is used to do this.

The default settings calculate the speed out of an internal one by dividing it by the encoder resolution. The result is a speed in min⁻¹. So, to have it in increments/s we remove the division by the encoder resolution, but we must divide the min⁻¹ value by 60 to have a 1/s value. So, the new setting for the velocity factor looks like in Figure 8.



Re-scaling of the drive speed is done on the drive side only. There is no change of configuration on the CODESYS side as it expects the feedback to be in this scaling anyway.



Drive functions		· · · · ·	-	_		×			
👂 Search 🕹	1								
Operating modes Operating modes Operating modes Operating modes		Device control - Factor group							
Wotion profile Range limits Factor group Error handling Sequence programs Ambient parameters Signal management Communication			Position encoder resolution (0x608F) Velocity encoder resolution resolution (0x608F) Encoder increments: 4096 Motor revolutions: 1 Motor revolutions:	solution (0x6090) s: 4096 1					
								Gear ratio (0x6091) Motor shaft revolutions: 1 Feed:	4096
		Driving shaft revolutions: 1 Shaft revolutions:	1						
		Velocity factor (0x6096) Numerator: Divisor: Polarity (0x607E:00) Function Reverse the speed Reverse the position Value 0x00							
		User units Position Velocity Current: incr. v-unit Standard: incr. 1/min	to standard						
		Send	Save Close	•	Help				

Figure 8 Modified Factor Group settings to have the speed in increments/s

Using interpolation of the cyclic target values

Even when using EtherCAT from point of view of the control loop new target values are applied step wise. Reason is the different update rate. The control loop gets updated every 100µs whereas typical EtherCAT communication cycles could be in a range of 1ms ... 2ms. Here the default of the RaspberyPi CODESYS runtime of 4ms was used. Compared to the control-loop this is an updated target position only every 40th cycle which results in steps in the target position. The behavior can be improved, if the received target values are not applied immediately but are interpolated over one communication cycle.

Starting from firmware revision L3 of the FAULHABER MotionControllers and MotionControl Systems interpolation of the received target values is available for all cyclic modes – CSP, CSV and CST.

To use it, simply enter the communication cycle in multiples of 100μ s into object 0x2332.00 – interpolation rate (Figure 9).



E Object Browser							= x
🚯 🖄 👜 🚺		? 🔎 Sea	rch				
Communication Manuf	acturer	Device					
Object	Index	Subindex	Parameters	Current value	New value	Unit	^
		09	Reduction of thermal resistance	e 2 25		%	
Switch position for a	2330	00	Number of entries	3	-		
		01	Actual commutation angle sour	rce 1			
		02	Actual velocity source	1			
		03	Actual position source	1			
Discrete sources	2331	00	Number of entries	4	-		
		01	Target voltage source	0			
		02	Target current source	0			
		03	Target velocity source	0			
		04	Target position source	0			
Interpolation rate	2332	00	Interpolation rate	40			
Manufacturer specifi	233A	00	Number of entries	2	-		
		01	Bit mask for bit 14	0x00000000			
		02	Bit mask for bit 15	0x00000000			
Operation mode opti	233F	00	Operation mode options	0x0031			
General parameters	2340	00	Number of entries	8	-		
		01	Commutation type	3			
		02	Motor output voltage DC	0			
		03	Motor output voltage BL block	0			
		04	Motor output voltage Xd	0			
		05	Motor output voltage Xq	0			
		06	Sinus output voltage Lla	30632	-		*
Data type: U16 Min	: 1	1	Max: 65535 Default	:1			

Figure 9 set the interpolation rate for cyclic operating modes



Create your application

In Product Application Note 164 the complete application – the interaction with the drive – was implemented in structured text. When using a SoftMotion axis the standardized SoftMotion function blocks can be used to interact with the drive.

The SoftMotion FBs all use the drive data structure AXIS_REF_ETC_DS402_CS. Each SoftMotion axis creates

Device SM_Drive_GenericDSP402 X				
General	IEC Objects			
Scaling/Mapping	Variable	Mapping	Туре	
	SM_Drive_GenericDSP402	*	AXIS_REF_ETC_DS402_CS	
Commissioning				
SM_Drive_ETC_GenericDSP402: I/O Mapping				
Status				
Information				

one instance of it. This structure holds all the necessary data which are the ones exchanged with the drive plus maybe some internal ones too which we don't care about.

Here a short example using only a few of these FBs was implemented (Figure 10). There is no real automatic action implemented. The inputs are simply mapped to a visualization and are then switched manually. The used FBs were:

MC_Power

The MC_Power block is interacting with the control word and the status word to enable or disable the power-stage and control-loop. It requires to be connected to the correct instance of the axis structure – the very same which has been instanced by the SoftMotion axis. Additionally, there is a Boolean input which enables the FB to be evaluated and two Boolean inputs to start the drive and to enable the control -loop. As these FAULHABER low voltage drives don't need to be activated – not connected to the AC-grid by switching a contactor, we can use the same input variable for both.

There are some outputs for the MC_Power block which would have to be used, if this was an automated machine, but that's not required in this simple example.

MC_MoveAbsolute

The MC_MoveAbsolute block is used to move a single axis in a Point to Point manner (PTP). To do so, it takes an absolute target position – in whatever scaling we configured for the SoftMotion axis – here turns. And it takes some profile parameters as we would use them if we would use PP mode and the built-in profile generator. Here however CSP is used and automatically activated by the SoftMotion axis².

² At leat it has been activated when the Modes of Operation parameter was mapped to the SoftMotion axis. Whether it can force the OpMode using a SDO transfer has not been tested.



The block of course again needs to access the instance of the axis structure. Then there are the parameters for the motion itself and the binary "Execute" input which will start the motion at a rising edge of the input value. Here a RS-FF was additionally used to reset this execute signal once the target position has been reached.

MC_ReadActualPosition

The MC_ReadActualPosition block is used to read the actual drive position in an application specific scaling out of then instance of the axis structure. It again needs access to the axis structure and there is again an enable input to have it executed.



Figure 10 Example application moving between two discrete positions

Visualization

The Inputs in Figure 10 were connected to controls in a CODESYS visualization. Some of the drive outputs were displayed too. The first inputs are the two Boolean variables which input the MC_Power block. Its output is connected to a lamp.



The two absolute target positions were connected to a text-input each as well as the profile speed used by both moves. The StartA and StartB inputs finally were connected to two push-buttons on the top right of the visualization.

Please note, the positions have been scaled in turns and can be set here as floating values. The speed of the SoftMotion axis is calculated in turns/s. This example used fixed but different values for the actual acceleration and deceleration.





To interact with the drive, there is no need to be logged into it using the CODESYS engineering environment. Alternatively, you could use any web browser and log into the device using its name or its IP address:

<IP number of the target controller>:8080/webvisu.htm e.g.:

192.168.0.45:8080/webvisu.htm



Run the drive

Review the drive behavior using the MotionManager

Even when the drive is connected to a master via EtherCAT the MotionManager can be used in parallel to check the drive behavior. Any parameter which is part of the process image configured for the drive (Figure 6) will however be instantly overwritten by the PLC. Therefore, enabling or disabling the drive as well as changing the target position would be meaningless. We can however watch the drive move.

The Graphical analysis tool of the MotionManager is used to do this. The signals selected here were:

- Position actual value which at the drive is in increments here 4096 increments/turn
- Position demand value the interpolated target position same scaling as for the actual one
- Following error actual value the following error calculated out of the upper two same scaling
- Velocity actual value which we had to rescale to increments/s, results in big numbers

We can see what looks like an almost perfect following to the two steps, the red-line of the actual value can't be seen it's hidden by the orange one, the demand. The detailed analysis based on the following error shows some systematic lag which is proportional to the speed and some disturbances which might be caused by whatever disturbance the linear position feedback might have contained.



Figure 11 tracing the drive behavior using the MotionManager

Aside of the Graphical analysis tool the MotionManager can also be used in parallel to check the list of recorded errors and receive EMCY message sent by the drive.



Real-time view of the application

The second perspective on the system behavior would be the real-time view of the application within the CODESYS engineering environment. This view is available whenever you are logged into the target. Here you can see the states of the used FBs as well as the actual values of the numeric parameters. This view can be necessary when the drive behavior is somehow strange, and the interactions of whatever controlling flags shall be reviewed.





Additional Resources

FAULHABER Application Note	5
App-Note 164	How to control a FAULHABER MC V3.0 ET out of a CODESYS environment



FAULHABER manuals at https://www.faulhaber.com/en/



FAULHABER demo systems at YouTube



Rechtliche Hinweise

Urheberrechte. Alle Rechte vorbehalten. Ohne vorherige ausdrückliche schriftliche Zustimmung der Dr. Fritz Faulhaber & Co. KG darf diese Application Note oder Teile dieser unabhängig von dem Zweck insbesondere nicht vervielfältigt, reproduziert, gespeichert (z.B. in einem Informationssystem) oder be- oder verarbeitet werden.

Gewerbliche Schutzrechte. Mit der Veröffentlichung, Übergabe/Übersendung oder sonstigen Zur-Verfügung-Stellung dieser Application Note werden weder ausdrücklich noch konkludent Rechte an gewerblichen Schutzrechten, übertragen noch Nutzungsrechte oder sonstige Rechte an diesen eingeräumt. Dies gilt insbesondere für gewerbliche Schutzrechte, die mittelbar oder unmittelbar den beschriebenen Anwendungen und/oder Funktionen dieser Application Note zugrunde liegen oder mit diesen in Zusammenhang stehen.

Kein Vertragsbestandteil; Unverbindlichkeit der Application Note. Die Application Note ist nicht Vertragsbestandteil von Verträgen, die die Dr. Fritz Faulhaber GmbH & Co. KG abschließt, und der Inhalt der Application Note stellt auch keine Beschaffenheitsangabe für Vertragsprodukte dar, soweit in den jeweiligen Verträgen nicht ausdrücklich etwas anderes vereinbart ist. Die Application Note beschreibt unverbindlich ein mögliches Anwendungsbeispiel. Die Dr. Fritz Faulhaber GmbH & Co. KG übernimmt insbesondere keine Gewährleistung oder Garantie dafür und steht auch insbesondere nicht dafür ein, dass die in der Application Note illustrierten Abläufe und Funktionen stets wie beschrieben aus- und durchgeführt werden können und dass die in der Application Note beschriebenen Abläufe und Funktionen in anderen Zusammenhängen und Umgebungen ohne zusätzliche Tests oder Modifikationen mit demselben Ergebnis umgesetzt werden können. Der Kunde und ein sonstiger Anwender müssen sich jeweils im Einzelfall vor Vertragsabschluss informieren, ob die Abläufe und Funktionen in ihrem Bereich anwendbar und umsetzbar sind.

Keine Haftung. Die Dr. Fritz Faulhaber GmbH & Co. KG weist darauf hin, dass aufgrund der Unverbindlichkeit der Application Note keine Haftung für Schäden übernommen wird, die auf die Application Note und deren Anwendung durch den Kunden oder sonstigen Anwender zurückgehen. Insbesondere können aus dieser Application Note und deren Anwendung keine Ansprüche aufgrund von Verletzungen von Schutzrechten Dritter, aufgrund von Mängeln oder sonstigen Problemen gegenüber der Dr. Fritz Faulhaber GmbH & Co. KG hergeleitet werden.

Änderungen der Application Note. Änderungen der Application Note sind vorbehalten. Die jeweils aktuelle Version dieser Application Note erhalten Sie von Dr. Fritz Faulhaber GmbH & Co. KG unter der Telefonnummer +49 7031 638 345 oder per Mail von mcsupport@faulhaber.de.

Legal notices

Copyrights. All rights reserved. This Application Note and parts thereof may in particular not be copied, reproduced, saved (e.g. in an information system), altered or processed in any way irrespective of the purpose without the express prior written consent of Dr. Fritz Faulhaber & Co. KG.

Industrial property rights. In publishing, handing over/dispatching or otherwise making available this Application Note Dr. Fritz Faulhaber & Co. KG does not expressly or implicitly grant any rights in industrial property rights nor does it transfer rights of use or other rights in such industrial property rights. This applies in particular to industrial property rights on which the applications and/or functions of this Application Note are directly or indirectly based or with which they are connected.

No part of contract; non-binding character of the Application Note. The Application Note is not a constituent part of contracts concluded by Dr. Fritz Faulhaber & Co. KG and the content of the Application Note does not constitute any contractual quality statement for products, unless expressly set out otherwise in the respective contracts. The Application Note is a non-binding description of a possible application. In particular Dr. Fritz Faulhaber & Co. KG does not warrant or guarantee and also makes no representation that the processes and functions illustrated in the Application Note can always be executed and implemented as described and that they can be used in other contexts and environments with the same result without additional tests or modifications. The customer and any user must inform themselves in each case before concluding a contract concerning a product whether the processes and functions are applicable and can be implemented in their scope and environment.



No liability. Owing to the non-binding character of the Application Note Dr. Fritz Faulhaber & Co. KG will not accept any liability for losses arising from its application by customers and other users. In particular, this Application Note and its use cannot give rise to any claims based on infringements of industrial property rights of third parties, due to defects or other problems as against Dr. Fritz Faulhaber GmbH & Co. KG.

Amendments to the Application Note. Dr. Fritz Faulhaber & Co. KG reserves the right to amend Application Notes. The current version of this Application Note may be obtained from Dr. Fritz Faulhaber & Co. KG by calling +49 7031 638 345 or sending an e-mail to mcsupport@faulhaber.de.