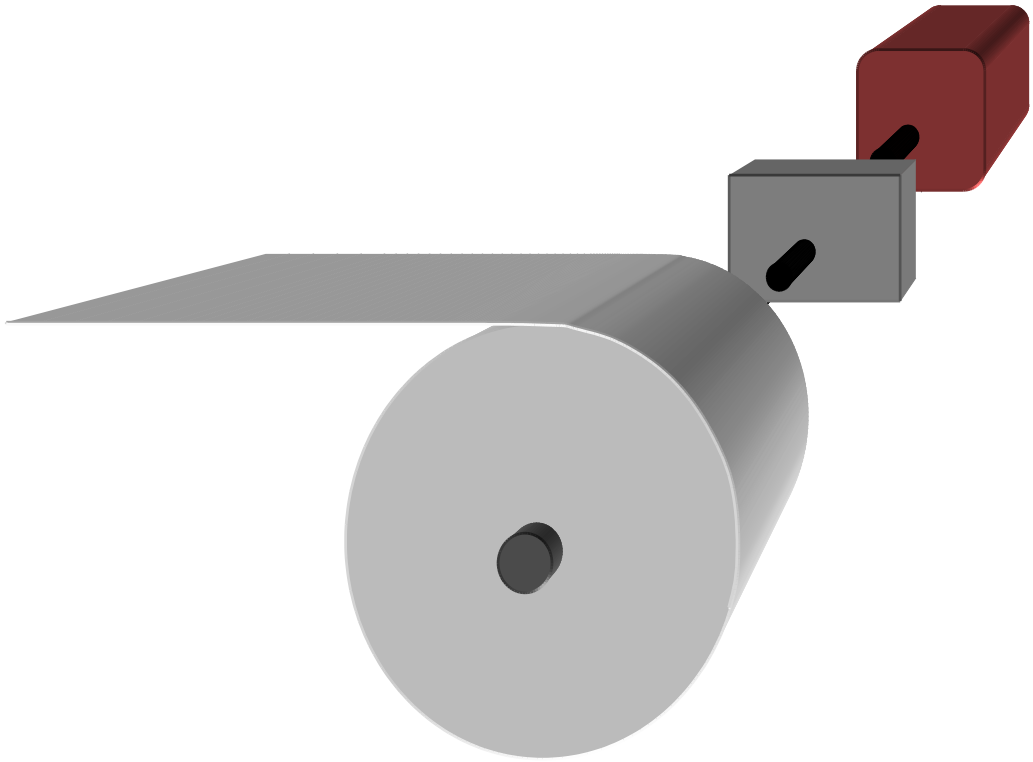
DCS800

DCS800 Winder Description

Indirect Tension Control Package



DCS800 Drive Manuals

All the documents available for the drive system DCS800 are listed below:

		Language					
	Public. number	E	D	I	ES	F	CN
DCS800 Quick Guide	3ADW000191	x	p	p	p	p	
DCS800 Tools & Documentation CD	3ADW000211	x					
DCS800 Converter module							
Flyer DCS800	3ADW000190	x	x	p	x	p	p
Technical Catalogue DCS800	3ADW000192	x	x	x	x	p	x
Hardware Manual DCS800	3ADW000194	x	x	p	p	p	p
Firmware Manual DCS800	3ADW000193	x	p	p	p	p	p
Installation according to EMC	3ADW000032	x					
Technical Guide	3ADW000163	x					
Service Manual DCS800	3ADW000195	x	p				
Planning and Start-up for12-Pulse converters	3ADW000196	p					
CMA-2 Board	3ADW000136	p					
Flyer Hard - Parallel	3ADW000153	p					
Drive Tools							
DriveWindow 2.x - User's Manual	3BFE64560981	x					
DriveOPC 2.x - User's Manual	3BFE00073846	x					
Optical DDCS Communication Link	3AFE63988235	x					
DDCS Branching Units - User´s Manual	3BFE64285513	x					
DCS800 Applications							
PLC Programming with CoDeSys	CoDeSys_V23	x	x			x	
61131 DCS800 target +tool description - Application Program	3ADW000199	x					
Winding with the DCS 800XXXXX	3ADW000058						
Winder application description	3ADW000253	x					
Flyer magnetic application							
Magnetic application description							
DCS800-E Panel Solution							
Flyer DCS800-E Panel solution	3ADW000210	x					
Hardware Manual DCS800-E	3ADW000224	x					
DCS800-A Enclosed Converters							
Flyer DCS800-A	3ADW000213	x					
Technical Catalogue DCS800-A	3ADW000198	x	p				
Installation of DCS800-A	3ADW000091	p	p				
DCS800-R Rebuild System							
Flyer DCS800-R	3ADW000007	x	x				
DCS800-R Manual	3ADW000197	x					
DCS500/DCS600 Size A5...A7, C2b, C3 and C4 Upgrade Kits	3ADW000256	x					
Extension Modules							
RAIO-01 Analogue IO Extension	3AFE64484567	x					
RDIO-01 Digital IO Extension	3AFE64485733	x					
AIMA R-slot extension	3AFE64661442	x					
Serial Communication							
Drive specific serial communication							
NETA Remote diagnostic interface	3AFE64605062	x					
Fieldbus Adapter with DC Drives RPBA- (PROFIBUS)	3AFE64504215	x					
Fieldbus Adapter with DC Drives RCAN-02 (CANopen)							
Fieldbus Adapter with DC Drives RCNA-01 (ControlNet)	3AFE64506005	x					
Fieldbus Adapter with DC Drives RDNA- (DeviceNet)	3AFE64504223	x					
Fieldbus Adapter with DC Drives RMBA (MODBUS)	3AFE64498851	x					
Fieldbus Adapter with DC Drives RETA (Ethernet)	3AFE64539736	x					
x -> existing p -> planned							
Status 09.2007							

Table of contents

DCS800 Drive Manuals ... 2
Table of contents ... 3
DCS800 Winder ... 5
Indirect tension control ... 5
Winder formulas and calculation ... 6
Diameter ... 6
Tension ... 7
Acceleration Torque ... 8
Losses ... 8
Winder motor ... 9
Winder structure ... 10
Interface DCS800 firmware and winder application ... 10
Signal and Parameter list ... 16
Group 7 ... 17
Group 8 ... 18
Group 60 ... 18
Group 61 ... 21
Group 63 ... 25
Group 64 ... 27
Group 65 ... 31
Group 66 ... 34
Appendix ... 35
Appendix ... 35
Safety instructions ... 36
What this chapter contains ... 36
To which products this chapter applies ... 36
Usage of warnings and notes ... 36
Installation and maintenance work ... 37
Grounding ... 38
Mechanical installation ... 40
Operation ... 41
Commissioning ... 43
Guidance ... 43
Control Bits (Example) ... 50
Appendix A - Application handling ... 51
Identification ... 51
Installation ... 51

Enable / disable application 52

DCS800 Winder

Winder drives can operate in

- velocity control
- indirect tension control
- direct tension control, equipped with a load cell
- dancer control, equipped with a dancer roll

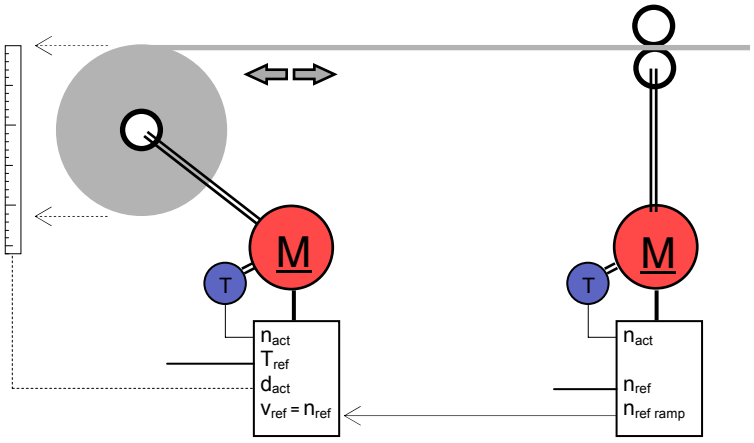
Winder control systems, except velocity control, needs the following conditions:

- The lead roll must be in velocity control
- The velocity signal for the winder drive comes from the lead roll or is only a reference signal
- Winder ratio should be lower than 1:12
- Quality of winding will be defined by the exactness of the calculation

This description contains the indirect tension control, which is also called as indirect torque control.

Indirect tension control

With indirect tension control the tension feedback must be calculated based on diameter and motor torque. Therefore it is important to adjust the torque- and compensation function blocks. The tension ratio should not be higher than 1:10, because quality winding without feedback signals isn't possible with a bigger ratio.



The following terms are necessary for this regulation scheme:

- Before starting, the actual diameter must be known
- Actual line speed (or line speed reference) is necessary during running
- Tension reference is necessary

Winder formulas and calculation

With DCS800 Winder Library it is possible to design winder applications using CoDeSys. For winders it is important that the following conditions are existing:

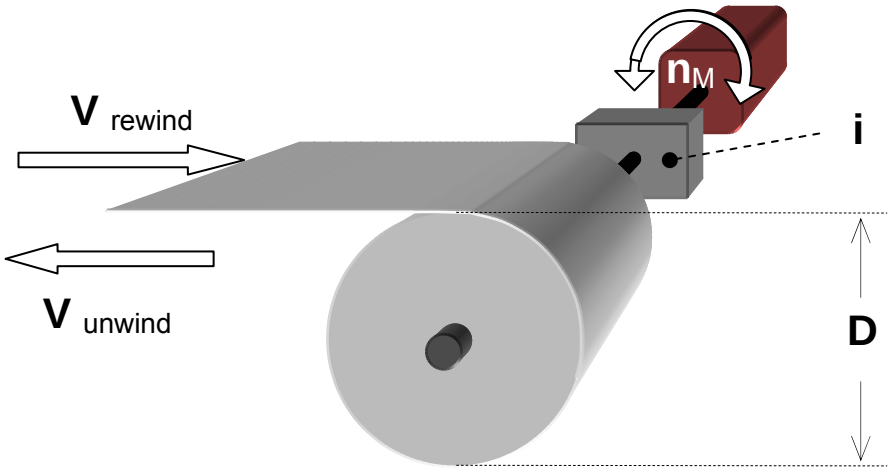
- The line speed (velocity) of the web is constant
- The material tension is constant (oscillating isn't allowed)
- Motor speed must be adapted to actual diameter
- Motor torque depends on the actual diameter

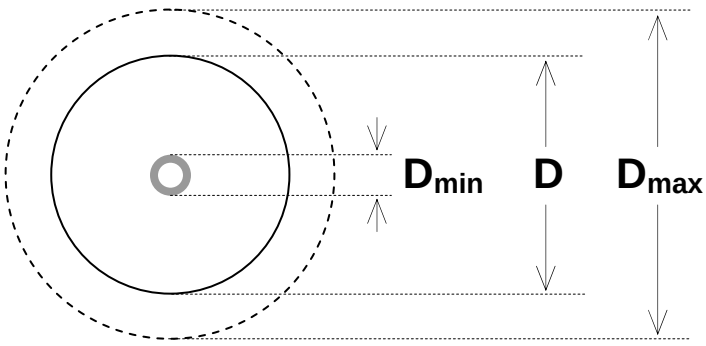
Diameter

In most cases the actual diameter must be calculated, because a sensor to enter the physical diameter from the material roll doesn't exist. But it is easy to calculate the diameter from the measured velocity and motor speed.

$$D = \frac{v}{n \cdot \pi}, \text{ unit: } [D] = m \tag{1.1}$$

In DCS800 we calculate with relative values. So it isn't essential to use the "PI".





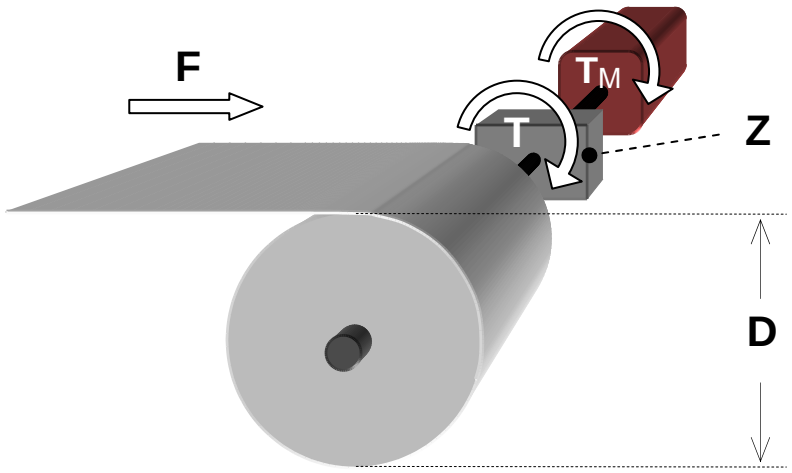
Diameter
[D] = m
Tension force
[F] = N
Motor speed
[n] = rpm
Velocity
[v] = m/s
Gear ratio
[i] Motor : Load

Tension

For winders it is important that the tension to be in line with the web material. If the tension is to low, the material at the roll doesn't wind correctly. When the tension is to powerful the web can be break. This is the hardest case, because the winder roll will accelerate, if there is no material break monitoring.
The tension is a force with units in Newtons. When the force is multiplied by the radius, the necessary torque for the selected tension can be calculated.

$$T_{tension} = F \cdot \frac{D}{2}, \text{ unit: } [T_{tension}] = Nm \quad (1.2)$$

The equation shows that the most torque is needed with the maximum diameter.



Actual diameter
[D] = m

Tension force
[F] = N

Total Torque
[T] = Nm

Tension Torque
[$T_{tension}$] = Nm

Motor speed
[n] = rpm

Gear ratio
Motor : Load

Note:
DCS800 Winder Software works with relative values (internal scaling). See scaling of the several parameters in function block description. For firmware structure the following rules applies:

- Velocity circuit is identical to speed circuit
  (100% velocity == 100% speed reference == 20000 internal scaling of 2.29)
- Tension circuit is identical to torque circuit
  (100% tension == 100% torque reference == 10000 internal scaling)
- Motor speed (100% == 20000 internal scaling)
- Diameter (100% == 10000 internal scaling, range: 100…10000 allowed)

Acceleration Torque

During the winding operation the motor must have only the torque from tension. But to accelerate it is necessary to have a torque buffer. The acceleration torque depends on the inertia of motor, mechanic, core and the material. If the diameter is small the inertia is also small. With increasing diameter the inertia will be bigger. That means more acceleration torque is needed. The problem in many applications is that the inertia isn't available. But with tests it is possible to measure the acceleration torque.

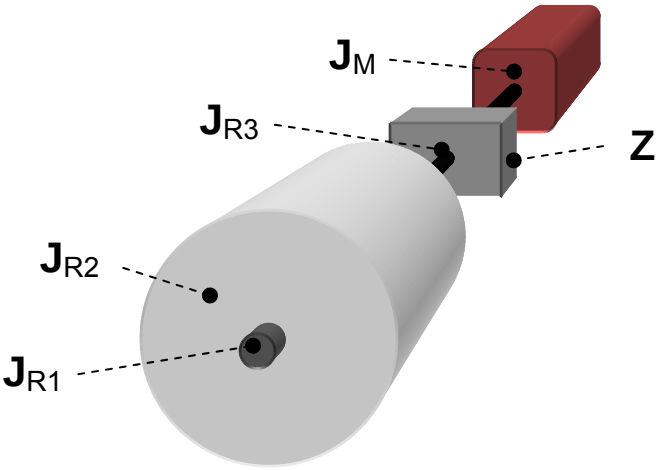
$$T_{acc} = J \cdot \frac{d\omega}{dt}, \text{ unit: } [T_{acc}] = Nm \tag{1.3}$$

After acceleration the acceleration torque is zero!

The acceleration torque depends extensive from the actual diameter. So the torque change with the fourth power.

$$T_{Motor} = \frac{F_{max} \cdot D}{2 \cdot i} + \left(J_M + \frac{J_{R1} + J_{R2} + J_{R3}}{i^2} + \frac{B \cdot \rho \cdot \pi}{32} \cdot \frac{D^4 - D_{min}^{\ 4}}{i^2} \right) \cdot \frac{2 \cdot i}{D} \cdot \frac{dv}{dt} + T_{losses}(n) \tag{1.4}$$

unit: $[T_{acc}] = Nm$



Torque inertia of core
$[J_{R1}]$ = kg m²

Torque inertia of material roll
$[J_{R2}]$ = kg m²

Torque inertia of gear box
$[J_{R3}]$ = kg m²

Torque inertia of the motor
$[J_M]$ = kg m²

Total torque inertia
$[J_c]$ = kg m²

Gear ratio
Motor : Load

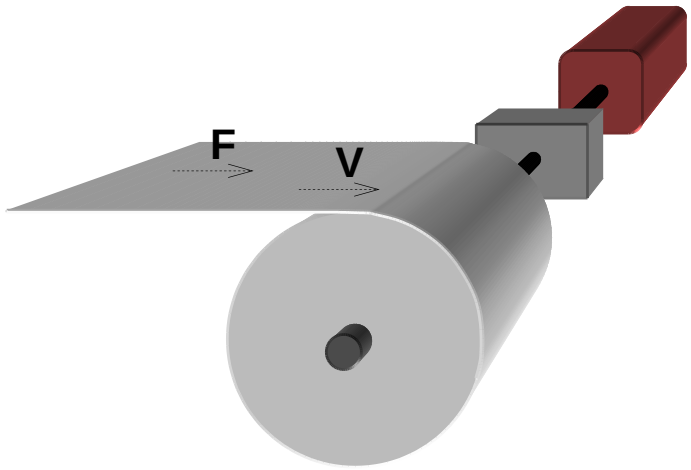
Losses

The mechanics of the winder generates losses in the form of friction and torsion. These losses depend on motor speed and can be measured in an idle test. They are non-linear and must be saved in a characteristic curve with supporting points.

$$T_{losses} = f(n), \text{ unit: } [T_{losses}] = Nm \tag{1.5}$$

Winder motor

To select the motor power the velocity and the tension force are important values that are needed to calculate it. Other values which are not included in this equation are the power to accelerate and the losses. They depend on the mechanics of the winder!



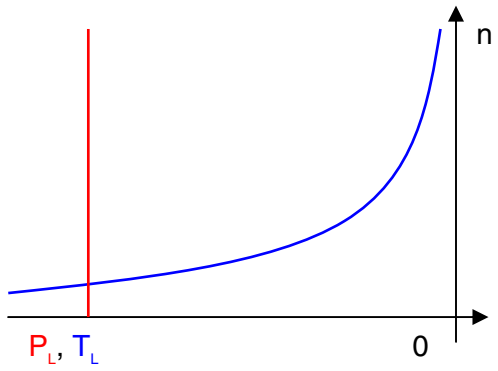
Tension Force
[F] = N

Velocity
[v] = m/min

Motor Power
[P] = kW

$$P = F \cdot v \qquad (1.6)$$

Winder characteristic curve:



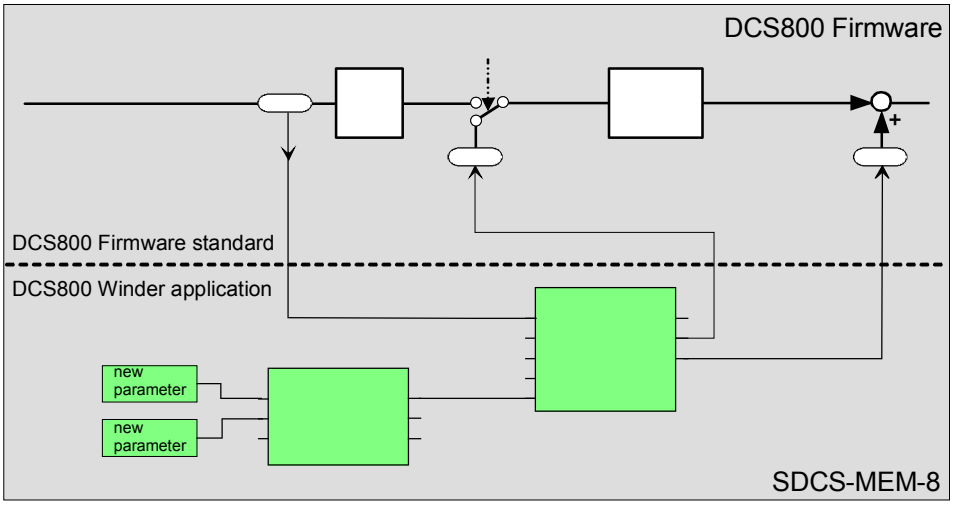
$$T_L = c \cdot \frac{1}{n}$$

$$P_L = \text{const.}$$

Note:
Winder motors are overloaded during the acceleration time! Dimensioning is normally necessary for torque from tension and losses. Overload must be allowed!

Winder structure

Interface DCS800 firmware and winder application

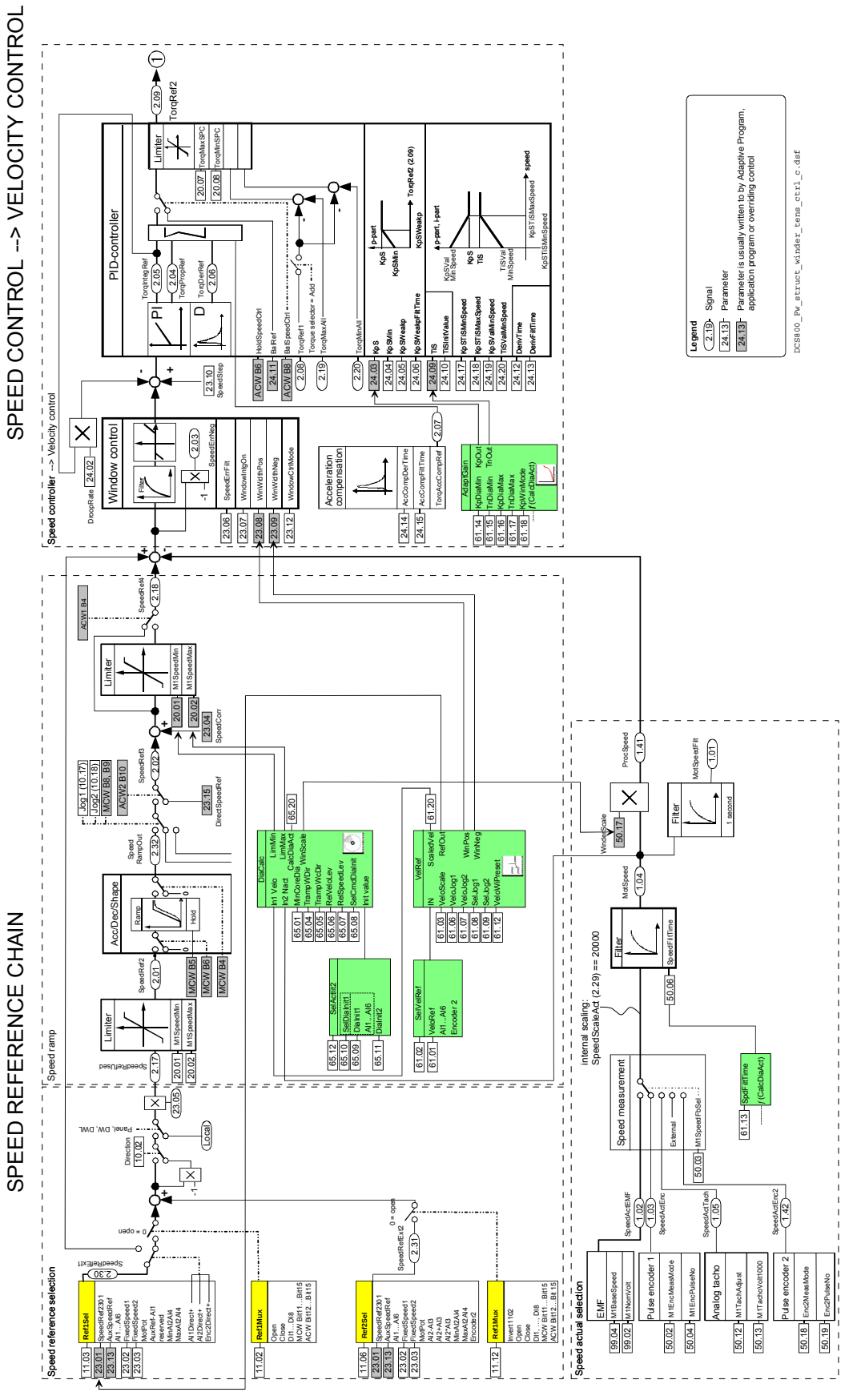


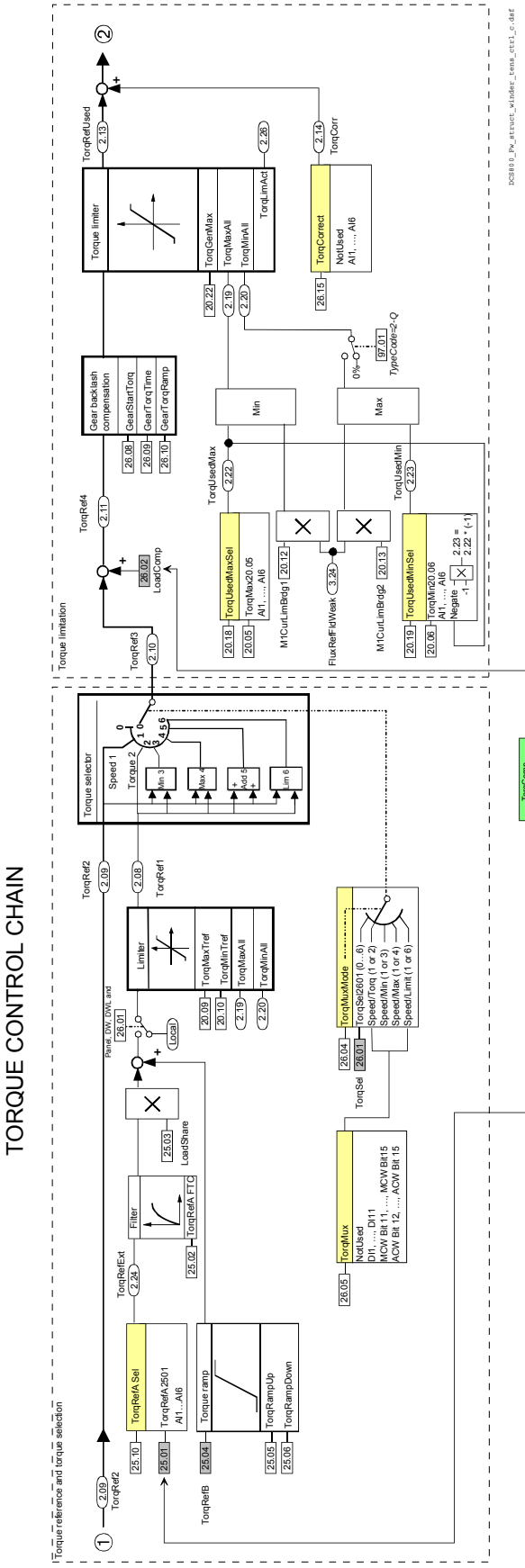
principle FW winder interf.dsf

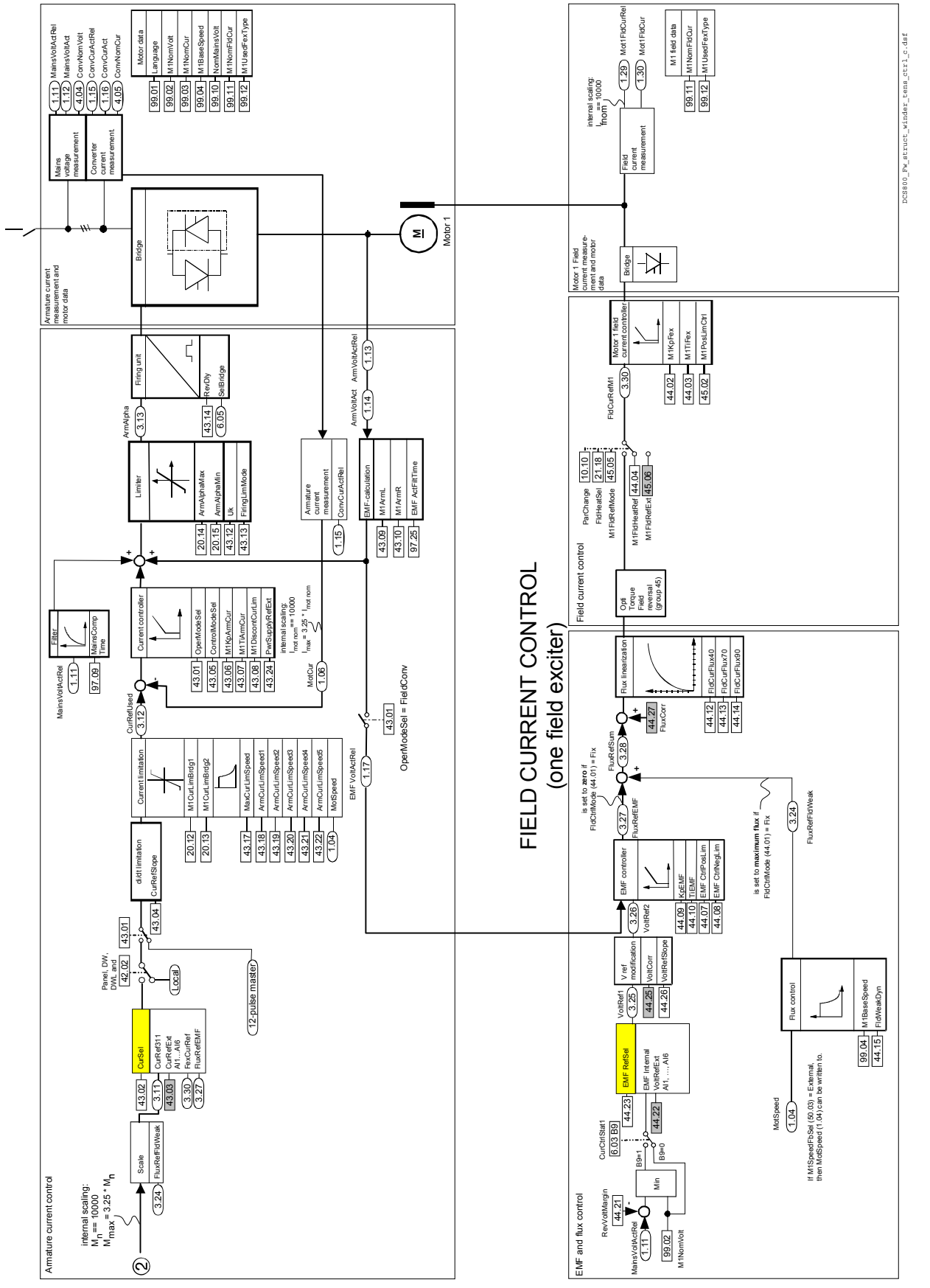
The DCS800 winder application is build as add-on to the DCS800 firmware. They interfaced by reading and writing parameters.

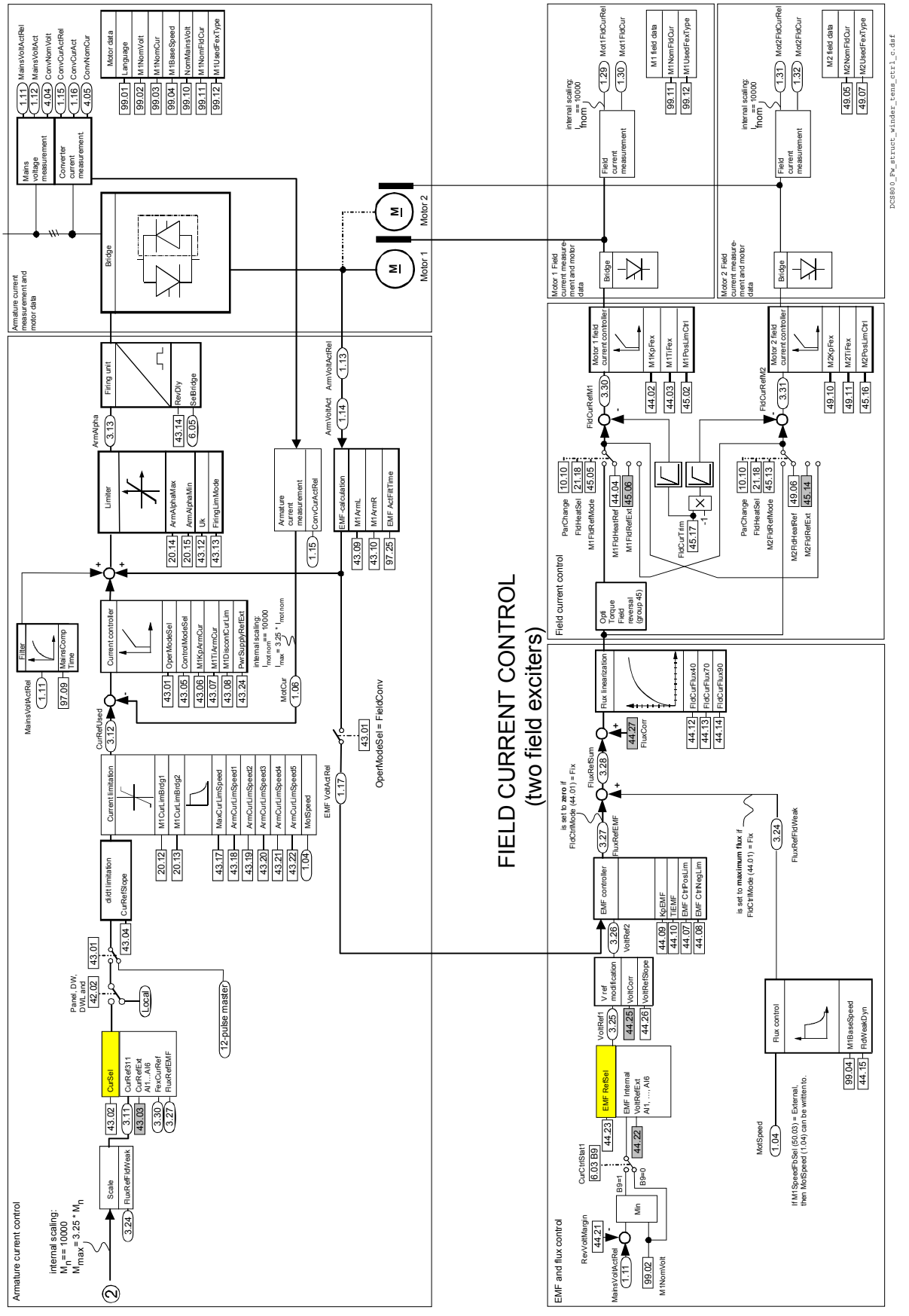
The DCS800 winder application is located in the SDCS-MEM-8 memory card and are clearly separated in hardware wise.

Detailed software structure diagrams are shown on the following pages.


SPEED CONTROL --> VELOCITY CONTROL
SPEED REFERENCE CHAIN
Speed controller --> Velocity control
PID-controller
Window control
Acceleration compensation
Speed reference selection
Speed ramp
Speed actual selection
Speed measurement
internal scaling:
SpeedScaleAct (2.29) == 20000
Legend
Signal
Parameter
Parameter is usually written to by Adaptive Program, application program or overriding control
DCS800_Pw_struct_winder_tens_ctrl_c.def


TORQUE CONTROL CHAIN
Torque reference and torque selection
Torque limitation
Torque selector
Torque limiter
Gear backlash compensation
Torque ramp
Limiter
Filter
TorqMux
TorqMuxMode
TorqComp
AccTorq
AccFilt
TorqScale
TorqRefA Sel
TorqRefA2501
TorqRefA FTC
TorqRampUp
TorqRampDown
TorqUsedMaxSel
TorqUsedMinSel
TorqCorrect
TorqRefUsed
TorqRef1
TorqRef2
TorqRef3
TorqRef4
TorqRefB
TorqRefExt
TorqSel
LoadShare
LoadComp
TorqGenMax
TorqMaxAll
TorqMinAll
TorqLimAct
TorqUsedMax
TorqUsedMin
TorqMaxTref
TorqMinTref
M1CurLimBrdg1
M1CurLimBrdg2
FluxReFldWeak
Min
Max
Local
Panel, DW, DWL and
TypeCode=2-Q
DCS810_Pw_struct_winder_tens_ctrl_c.daf


ARMATURE CURRENT CONTROL
FIELD CURRENT CONTROL
(one field exciter)
Armature current control
EMF and flux control
Field current control
Armature current measurement and motor data
Motor 1 Field current measure-ment and motor data
Bridge
Motor 1
Firing unit
Limiter
Current controller
Current limitation
di/dt limitation
Filter
Armature current measurement
EMF-calculation
Mains voltage measurement
Converter current measurement
Motor data
Field current measurement
Motor 1 field current controller
Flux linearization
EMF controller
V ref modification
EMF RefSel
Flux control
Opt Torque Field reversal (group 45)
internal scaling:
M_n== 10000
M_{max} = 3.25 * M_n
internal scaling:
$I_{mot\ nom}$== 10000
I_{max} = 3.25 * $I_{mot\ nom}$
internal scaling:
I_{nom} = 10000
Panel, DW, DWL and TDC
Local
12-pulse master
OpenModeSel = FieldConv
MainsVoltActRel
MainsVoltAct
ConvNomVolt
ConvCurActRel
ConvCurAct
ConvNomCur
ArmAlpha
ArmVoltActRel
ArmVoltAct
CurRefUsed
MotCur
EMF VoltActRel
MotSpeed
FluxRefSum
FluxCorr
FldCurFlux40
FldCurFlux70
FldCurFlux90
FldCurRefM1
M1FldHeatRef
M1FldRefExt
M1FldRefMode
FldHeatSel
ParChange
RevVoltMargin
MainsVoltActRel
M1NomVolt
CurCtrlStat1
VoltRef1
VoltRef2
VoltCorr
VoltRefSlope
FluxRefEMF
FluxRefFldWeak
M1BaseSpeed
FldWeakDyn
M1 field data
Mot1FldCurRel
Mot1FldCur
M1NomFldCur
M1UsedFexType
M1KpFex
M1TiFex
M1PosLimCtrl
M1KpEMF
M1TiEMF
EMF CtrlPosLim
EMF CtrlNegLim
is set to zero if FldCtrlMode (44.01) = Fix
is set to maximum flux if FldCtrlMode (44.01) = Fix
If M1SpeedFbSel (50.03) = External, then MotSpeed (1.04) can be written to.
DCS800_Pe struct_winder_tens ctrl_c.def


ARMATURE CURRENT CONTROL
FIELD CURRENT CONTROL
(two field exciters)
Armature current control
Armature current measurement and motor data
Bridge
Motor 1
Motor 2
Mains voltage measurement
Converter current measurement
Motor data
Firing unit
Limiter
Current controller
Current limitation
di/dt limitation
Filter
Armature current measurement
EMF-calculation
Field current control
Motor 1 Field current measurement and motor data
Motor 2 Field current measurement and motor data
Motor 1 field current controller
Motor 2 field current controller
Field current measurement
EMF and flux control
EMF controller
Flux linearization
V ref modification
EMF RefSel
Flux control
Panel, DW, DVIL and
Local
12-pulse master
internal scaling:
Mn== 10000
Mmax = 3.25 * Mn
internal scaling:
Inotnom== 10000
Imax = 3.25 * Inotnom
OperModeSel = FieldConv
Opt Torque Field reversal (group 45)
DCS800_Pe_struct_winder_tens_ctrl_c.def

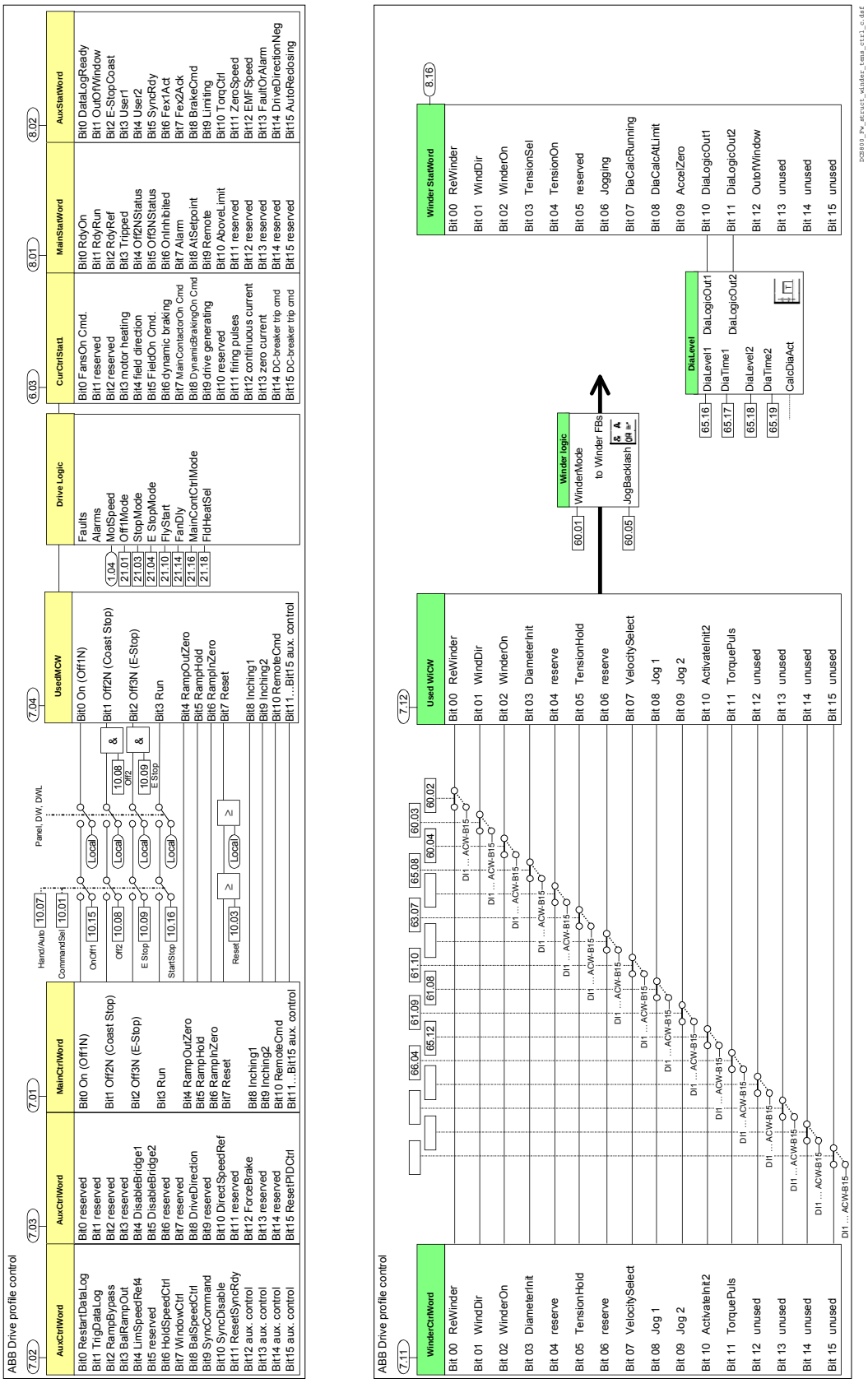


ABB Drive profile control
AuxCtrlWord
AuxCtrlWord
MainCtrlWord
Hand/Auto
CommandSel
Panel, DW, DWL
UsedMCW
Drive Logic
CurCtrlStat
MainStatWord
AuxStatWord
Bit0 On (Off1N)
Bit1 Off2N (Coast Stop)
Bit2 Off3N (E-Stop)
Bit3 Run
Bit4 RampOutZero
Bit5 RampHold
Bit6 RampInZero
Bit7 Reset
Bit8 Inching1
Bit9 Inching2
Bit10 RemoteCmd
Bit11...Bit15 aux. control
Faults
Alarms
MotSpeed
Off1Mode
StopMode
E StopMode
FlyStart
FanDly
MainContCtrlMode
FldHeatSel
Bit0 FansOn Cmd.
Bit1 reserved
Bit2 reserved
Bit3 motor heating
Bit4 field direction
Bit5 FieldOn Cmd.
Bit6 dynamic braking
Bit7 MainContactorOn Cmd
Bit8 DynamicBrakingOn Cmd
Bit9 drive generating
Bit10 reserved
Bit11 firing pulses
Bit12 continuous current
Bit13 zero current
Bit14 DC-breaker trip cmd
Bit15 DC-breaker trip cmd
Bit0 RdyOn
Bit1 RdyRun
Bit2 RdyRef
Bit3 Tripped
Bit4 OffZNStatus
Bit5 Off9NStatus
Bit6 OnInhibited
Bit7 Alarm
Bit8 AtSetpoint
Bit9 Remote
Bit10 AboveLimit
Bit11 reserved
Bit12 reserved
Bit13 reserved
Bit14 reserved
Bit15 reserved
Bit0 DataLogReady
Bit1 OutOfWindow
Bit2 E-StopCoast
Bit3 User1
Bit4 User2
Bit5 SyncRdy
Bit6 Fex1Act
Bit7 Fex2Ack
Bit8 BrakeCmd
Bit9 Limiting
Bit10 TorqCtrl
Bit11 ZeroSpeed
Bit12 EMFSpeed
Bit13 FaultOrAlarm
Bit14 DriveDirectionNeg
Bit15 AutoReclosing
OnOff1
Off2
E Stop
StartStop
Reset
Local
Off2
E Stop
ABB Drive profile control
WinderCtrlWord
Used WiCW
Winder logic
WinderMode
to Winder FBs
JogBacklash
DiaLevel
DiaLevel1
DiaLevel2
DiaTime1
DiaTime2
CalcDiaAct
DiaLogicOut1
DiaLogicOut2
Winder StatWord
Bit 00 ReWinder
Bit 01 WindDir
Bit 02 WinderOn
Bit 03 TensionSel
Bit 04 TensionOn
Bit 05 reserved
Bit 06 Jogging
Bit 07 DiaCalcRunning
Bit 08 DiaCalcATLimit
Bit 09 AccelZero
Bit 10 DiaLogicOut1
Bit 11 DiaLogicOut2
Bit 12 OutofWindow
Bit 13 unused
Bit 14 unused
Bit 15 unused
Bit 00 ReWinder
Bit 01 WindDir
Bit 02 WinderOn
Bit 03 DiameterInit
Bit 04 reserve
Bit 05 TensionHold
Bit 06 reserve
Bit 07 VelocitySelect
Bit 08 Jog 1
Bit 09 Jog 2
Bit 10 ActivateInit2
Bit 11 TorquePuls
Bit 12 unused
Bit 13 unused
Bit 14 unused
Bit 15 unused
DI1 ... ACW-B15

Signal and Parameter list

Index	Signal / Parameter name	min.	max.	def.	unit	E/C

General scaling of values:

	Percental	internal	physical
Velocity / line speed	100 %	20 000	x mps (also x mpm); → scaled with max. velocity, which is needed to get the max. motor speed while running with min. diameter
Speed (motor speed)	100 %	20 000	x rpm → scaled with parameter 2.29
Diameter	100 %	10 000	x mm = value of D_{max}
Tension	100 %	10 000	x N → depending on torque scaling via parameter 66.03
Torque	100 %	10 000	x Nm → depending on motor data
Current	100 %	10 000	x A = value of parameter 99.03 M1NomCur

Physical units of DCS800 standard parameters:

RPM The region of speed reference, ramp generator up to speed controller is defined as rpm. Unfortunately this physical unit of these parameters can not be modified.
By using the winder application these region is working as velocity.
The value in rpm can be calculated as following:

$$\text{value}\,[\text{m/s}] = \text{value of parameter}\,[\text{rpm}] \cdot \frac{\text{max. velocity}\,[\text{m/s}]}{\text{speed scaling}\,[\text{rpm}]}$$

speed scaling via parameter 2.29

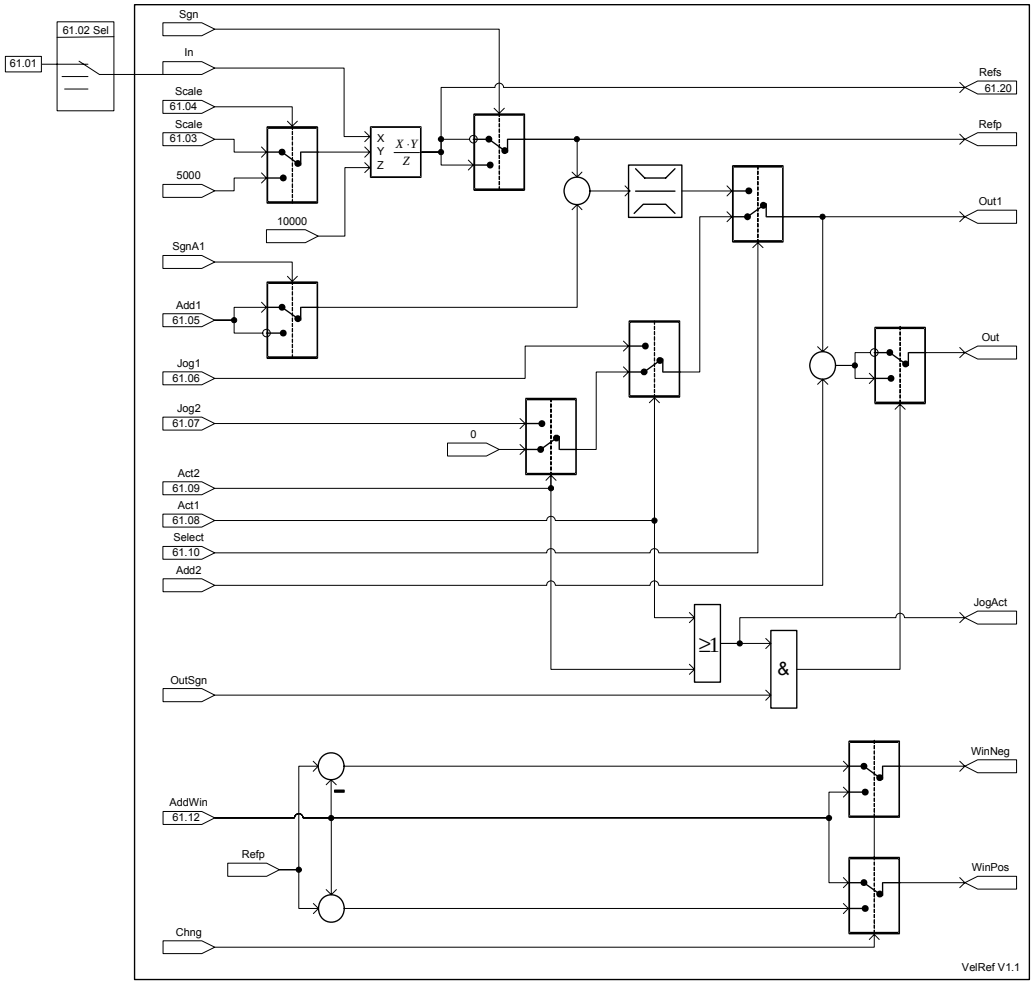
Index	Signal / Parameter name	min.	max.	def.	unit	E/C
Group 7	**Control Words additional to standard firmware**					
7.11	**Winder_CW (winder control word)** The control word contains all winder depending commands Please see the bit description of parameter 7.12. Please note that all used bits can also be selected via parameters.	'	'	'		
7.12	**Used_WiCW (used winder control word)** The used winder control word is read only and contains all winder depending commands, which sources are selectable (*1). **Bit / Name / Value / Comment** **0 / ReWin / 1 / Set as rewinder (*1)** **0 / Set as unwinder (*1)** **1 / WinDir / 1 / counter clockwise turning (*1)** **0 / clockwise turning (*1)** **2 / WinOn / 1 / Activates winding mode according winding mode selection** **0 / Activates speed (velocity) controlled mode** **3 / DiaSet / 1 / set initial diameter as actual diameter value** **0 / diameter calculation will run in winding mode** **4 / - - - / 1 / no action** **0 / no action** **5 / TeReHld / 1 / Set an internal value as tension setpoint** **0 / Set the external value as tension setpoint** **6 / VelScale / 1 / Switch over to 100% scaling** **0 / Scaling by using parameter (default 100%)** **7 / VelSelect / 1 / reserved for selection of external reference;** **0 / no action; jog commands takes priority** **8 / WinJog1 / 1 / Jog ref 1 selected** **0 / no action** **9 / WinJog2 / 1 / Jog ref 2 selected** **0 / no action** **10 / DiaActInit2 / 1 / value 2 is selected as initial diameter** **0 / value 1 is selected as initial diameter** **11 / TrqPuls / 1 / reserved for torque pulse will be activated** **0 / no action** **12 / - - - / 1 / no action** **0 / no action** **13 / - - - / 1 / no action** **0 / no action** **14 / - - - / 1 / no action** **0 / no action** **15 / - - - / 1 / no action** **0 / no action** *1 changed signal take effect, if RdyRef is false (8.01:2=0); means “not running”.	'	'	'		

Index	Signal / Parameter name	min.	max.	def.	unit	E/C
Group 8	**Status Words additional to standard firmware**					
8.16	**Winder_SW (winder status word)** Status word of winder function with following bits: **Bit / Name / Value / Comment** **0 ReWinder 1 Winder is selected as rewinder** **0 Winder is selected as unwinder** **1 WinDirection 1 Web is coming from above** **0 Web is coming from below** **2 WinOn 1 Winding mode is active** **0 Line speed mode is active** **3 TenSel 1 Tension mode is pre selected** **0 With WiOn the tension mode will not be active** **4 TenOn 1 Tension mode is active** **0 Tension mode is not active** **5 CtrlOn 1 Tension controller is active** **0 Tension controller is not active** **6 WinJogAct 1 Jog command is given.** **0 Jog command is not given.** **7 DiaCalcRun 1 Diameter calculator is released and running** **0 Diameter calculator is stopped** **8 DiaAtLimit 1 Output of diameter calculation is in working range.** **0 Output of diameter calculation is inside the limits.** **9 AccTqZero 1 The derivated signal is lower than the level. So the acceleration torque is zero.** **0 The derivated signal is higher than the level.** **10 WiLoOut1 1 Calculated diameter has just reached level 1 (impulse)** **0 no action** **11 WiLoOut2 1 Calculated diameter has just reached level 2 (impulse)** **0 no action** **12 OutofWindow 1 Delayed signal of out of window** **0 no action** **13..15 - - - 1 not used yet** **0 not used yet**	-	-	-		
Group 60	**Applic Controls Winder application**					
60.01	**WinderMode (selector of winding mode)** With this parameter the winding mode is to be selected: 0 = **None** 1 = Speed_Ctrl Without “Winder On” command the drive can always run in speed mode; e.g. for jogging. 2 = **Ind_Tension** with “Winder On” command the indirect tension mode is running 3 Dir_Tension Not available in this application 4 = Dancer_Ctrl Not available in this application	None	Dancer_Ctrl	Ind_Tension	-	

Index	Signal / Parameter name	min.	max.	def.	unit	E/C
60.02	**SelReWinder (selector of rewinder command)** The source for this rewinder command will be selected by this parameter: Signal = 0 means running as un winder Signal = 1 means running as re winder 00 = **Default** according winder control word (7.11); here bit 0 of Used_WiCW (7.12) 01 = **NotUsed** constant 0 (false) 02 = **On** constant 1 (true) 03 = **DI1** depending on digital input 1 04 = **DI2** depending on digital input 2 05 = **DI3** depending on digital input 3 06 = **DI4** depending on digital input 4 07 = **DI5** depending on digital input 5 08 = **DI6** depending on digital input 6 09 = **DI7** depending on digital input 7 10 = **DI8** depending on digital input 8 11 = **DI9** depending on digital input 9 12 = **DI10** depending on digital input 10 13 = **DI11** depending on digital input 11 14 = **DI12** depending on digital input 12 15 = **DI13** depending on digital input 13 16 = **DI14** depending on digital input 14 17 = **WiCW_Bit10** depending on winder control word (7.11), bit 10 18 = **WiCW_Bit11** depending on winder control word (7.11), bit 11 19 = **WiCW_Bit12** depending on winder control word (7.11), bit 12 20 = **WiCW_Bit13** depending on winder control word (7.11), bit 13 21 = **WiCW_Bit14** depending on winder control word (7.11), bit 14 22 = **WiCW_Bit15** depending on winder control word (7.11), bit 15 23 = **MCW_Bit00** depending on main control word (7.01), bit 00 24 = **MCW_Bit03** depending on main control word (7.01), bit 03 25 = **MCW_Bit08** depending on main control word (7.01), bit 08 26 = **MCW_Bit09** depending on main control word (7.01), bit 09 27 = **MCW_Bit11** depending on main control word (7.01), bit 11 28 = **MCW_Bit12** depending on main control word (7.01), bit 12 29 = **MCW_Bit13** depending on main control word (7.01), bit 13 30 = **MCW_Bit14** depending on main control word (7.01), bit 14 31 = **MCW_Bit15** depending on main control word (7.01), bit 15 **Int. Scaling: 1 == 1** **Type:** **C** **Volatile:**	Default	ACW_Bit15	Default	-	
60.03	**SelWinDir (selector of winder direction command)** The source for this winder direction command will be selected by this parameter: Signal = 0 means winding clockwise Signal = 1 means winding counter clockwise 00 = **Default** according winder control word (7.11); here bit 1 of Used_WiCW (7.12) 01 = **NotUsed** constant 0 (false) 02 = **On** constant 1 (true) … = **…** 31 = please see more possible setting in parameter 60.02 **Int. Scaling: 1 == 1** **Type:** **C** **Volatile:**	Default	ACW_Bit15	Default		

Index	Signal / Parameter name	min.	max.	def.	unit	E/C
60.04	**SelWinderOn (selector of winder on command)** The source for this winder on command will be selected by this parameter: Signal = 0 means running in line speed controlled mode Signal = 1 means running in indirect torque controlled mode 00 = **Default** according winder control word (7.11); here bit 2 of Used_WiCW (7.12) 01 = **NotUsed** constant 0 (false) 02 = **On** constant 1 (true) … = **…** 31 = please see more possible setting in parameter 60.02 **Int. Scaling: 1 == 1** **Type:** **C** **Volatile:**	Default	ACW_Bit15	Default		
60.05	**JogBacklash (Time of jog delay)** This time will delay the switch over from jogging reference to the external reference. **Int. Scaling: 1 == 1ms** **Type:** **I** **Volatile:**	0.001	30	2	s	

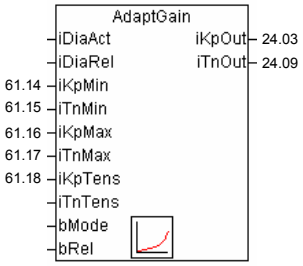
Index	Signal / Parameter name	min.	max.	def.	unit	E/C
Group 61	**Velocity Control Winder application**					

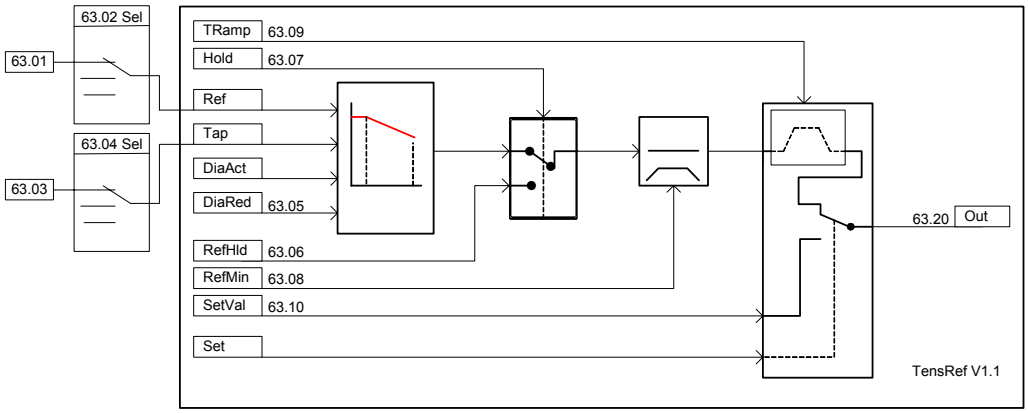


61.01	**VeloRef (Velocity reference; also called line speed)** This value is used as incoming velocity reference of the winder application. **Int. Scaling: 20000 == 100 %** **Type: SI** **Volatile:**	-32767	32767	0		

Index	Signal / Parameter name	min.	max.	def.	unit	E/C
61.02	**SelVelRef (selector of velocity reference)** Velocity reference selector for input as main reference (to function block VelRef): 0 = **Def_Para** VeloRef (Parameter 61.01) selected 1 = **AI1** analog input 1 2 = **AI2** analog input 2 3 = **AI3** analog input 3 4 = **AI4** analog input 4 5 = **AI5** analog input 5 6 = **AI6** analog input 6 7 = **Special** reserved 8 = **Extra** reserved 9 = **AITAC** tacho input 10 = **Encoder2** encoder input 2 **Int. Scaling: 1 == 1** **Type: C** **Volatile:**	Def_Para	Encoder2	Def_Para	-	
61.03	**VeloScale (scaling factor of velocity reference)** Function block : **VelRef** Input : **iScale** **Int. Scaling: 5000 == 100 %** **Type: SI** **Volatile:**	1	30000	5000	-	
61.06	**VeloJog1 (Jog reference 1 as velocity reference)** Function block : **VelRef** Input : **Jog1** **Int. Scaling: 20000 == 100 %** **Type: SI** **Volatile:**	-30000	30000	1000		
61.07	**VeloJog2 (Jog reference 2 as velocity reference)** Function block : **VelRef** Input : **Jog2** **Int. Scaling: 20000 == 100 %** **Type: SI** **Volatile:**	-30000	30000	-1000		

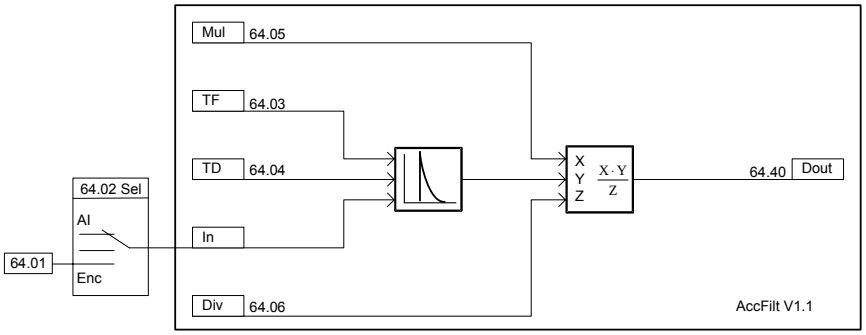
Index	Signal / Parameter name	min.	max.	def.	unit	E/C
61.08	**SelJog1 (selector of jog1 command)** Parameter selects the input of jog1 command; used in function block VelRef 00 = **Default** according winder control word ; here bit 8 of Used_WiCW (7.12). 01 = **NotUsed** constant 0 (false) 02 = **On** constant 1 (true) 03 = **DI1** depending on digital input 1 04 = **DI2** depending on digital input 2 05 = **DI3** depending on digital input 3 06 = **DI4** depending on digital input 4 07 = **DI5** depending on digital input 5 08 = **DI6** depending on digital input 6 09 = **DI7** depending on digital input 7 10 = **DI8** depending on digital input 8 11 = **DI9** depending on digital input 9 12 = **DI10** depending on digital input 10 13 = **DI11** depending on digital input 11 14 = **DI12** depending on digital input 12 15 = **DI13** depending on digital input 13 16 = **DI14** depending on digital input 14 17 = **WiCW_Bit10** depending on winder control word (7.11), bit 10 18 = **WiCW_Bit11** depending on winder control word (7.11), bit 11 19 = **WiCW_Bit12** depending on winder control word (7.11), bit 12 20 = **WiCW_Bit13** depending on winder control word (7.11), bit 13 21 = **WiCW_Bit14** depending on winder control word (7.11), bit 14 22 = **WiCW_Bit15** depending on winder control word (7.11), bit 15 23 = **MCW_Bit00** depending on main control word (7.01), bit 00 24 = **MCW_Bit03** depending on main control word (7.01), bit 03 25 = **MCW_Bit08** depending on main control word (7.01), bit 08 26 = **MCW_Bit09** depending on main control word (7.01), bit 09 27 = **MCW_Bit11** depending on main control word (7.01), bit 11 28 = **MCW_Bit12** depending on main control word (7.01), bit 12 29 = **MCW_Bit13** depending on main control word (7.01), bit 13 30 = **MCW_Bit14** depending on main control word (7.01), bit 14 31 = **MCW_Bit15** depending on main control word (7.01), bit 15 **Int. Scaling: 1 == 1** **Type: C** **Volatile:**	Default	Default	ACW_Bit15	-	
61.09	**SelJog2 (selector of jog2 command)** Parameter selects the input of jog2 command; used in function block VelRef 00 = **Default** according winder control word ; here bit 9 of Used_WiCW (7.12). 01 = **NotUsed** constant 0 02 = **On** 03 = … see more in parameter 61.08 **Int. Scaling: 1 == 1** **Type: C** **Volatile:**	Default	Default	ACW_Bit15	-	
61.12	**VeloWiPreset (lead value of velocity during winding mode)** Function block : **VelRef** Input : **iAddWin** **Int. Scaling: 20000 == 100 %** **Type: SI** **Volatile:**	-30000	30000	-1000		

Index	Signal / Parameter name	min.	max.	def.	unit	E/C
61.13	**SpdFiltTime (filter time of speed feedback depending on diameter** The value of filter time corresponds with the minimal diameter. The filter time will increase with growing diameter. With maximal diameter the following value will be active: max. filter time = value of 61.13 * max. diameter (100%) / value of 65.01 **Note:** The calculated value will be transmitted to parameter 50.06. **Int. Scaling: 1 == 1ms** **Type: SI** **Volatile:**	0	2000	1	ms	
						
61.14	**KpDiaMin (Kp value of speed controller with minimal diameter)** Function block : **AdaptGain0** Input : **iKpMin** **Note:** iKpOut will be transmitted to parameter 24.03. **Int. Scaling: 100 == 1 x** **Type: SI** **Volatile:**	0.00	325.00	5.00	-	
61.15	**TnDiaMin (Tn value of speed controller with minimal diameter)** Function block : **AdaptGain** Input : **iTnMin** **Note:** iTnOut will be transmitted to parameter 24.09. **Int. Scaling: 1 == 1 ms** **Type: SI** **Volatile:**	0	64000	2500	ms	
61.16	**KpDiaMax (Kp value of speed controller with maximal diameter)** Function block : **AdaptGain** Input : **iKpMax** **Note:** KpOut will be transmitted to parameter 24.03. **Int. Scaling: 100 == 1 x** **Type: SI** **Volatile:**	0.00	325.00	5.00	-	
61.17	**TnDiaMax (Tn value of speed controller with maximal diameter)** Function block : **AdaptGain** Input : **iTnMax** **Note:** KpOut will be transmitted to parameter 24.09. **Int. Scaling: 1 == 1 ms** **Type: SI** **Volatile:**	0	64000	2500	ms	

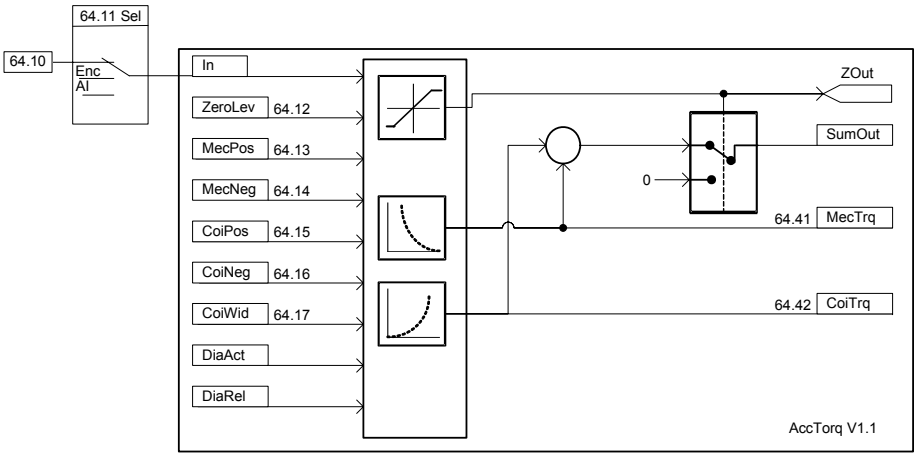
Index	Signal / Parameter name	min.	max.	def.	unit	E/C
61.18	**KpWinMode (Kp value of speed controller during window mode)** During tension mode the speed controller is running in window mode for limiting the speed range. Therefore a higher Kp value is desired. **Note:** In tension mode, this value will be transmitted to parameter 24.03. **Int. Scaling: 100 == 1 x Type: SI Volatile:**	0.00	325.00	10.00		
61.19	**Unused**	-	-	-	-	
61.20	**ScaledVel (Scaled velocity (line speed) from "VelRef")** This parameter is the output iRefs of the winder function block VelRef. **Int. Scaling: 20000 == 100% Type: SI Volatile:**	-30000	30000	0	-	
Group 63	**Tension Controls Winder application**					
						
63.01	**TensRef (input of tension reference)** Function block : **TensRef** Input : **iRef** **Int. Scaling: 100 == 1 % Type: SI**	0.00	100.00	0.00	%	
63.02	**SelTensRef (selector of tension reference)** Tension reference selector for input to function block TensRef: 0 = **Def_Para** TensRef (parameter 63.01) selected 1 = **AI1** analog input 1 2 = **AI2** analog input 2 3 = **AI3** analog input 3 4 = **AI4** analog input 4 5 = **AI5** analog input 5 6 = **AI6** analog input 6 **Int. Scaling: 1 == 1 Type: C Volatile:**	Def_Para	AI6	Def_Para	-	

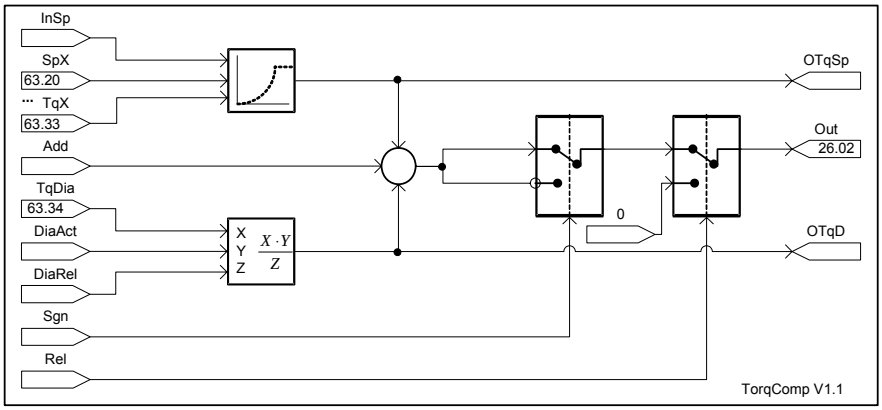
Index	Signal / Parameter name	min.	max.	def.	unit	E/C
63.03	**TapTens (reduction of tension dependend on diameter)** Function block : **TensRef** Input : **iTap** **Int. Scaling: 100 == 1 % Type: SI**	0.00	100.00	0.00	%	
63.05	**TapDia (value as from the tension reduction begins)** Function block : **TensRef** Input : **iDiaRed** **Int. Scaling: 100 == 1 % Type: SI**	1.00	100.00	1.00	%	
63.06	**TeRefHold (tension reference if hold command is active)** This tension reference will be active instead of external tension reference, if the incoming command, selected in 63.07, is active. **Int. Scaling: 100 == 1 % Type: SI**	0.00	100.00	15.00	%	
63.07	**SelTeRefHld (selector of hold tension command)** Parameter selects the input of tension hold reference command; used in function block TensRef 00 = **Default** according winder control word ; here bit 5 of Used_WiCW (7.12). 01 = **NotUsed** constant 0 (false) 02 = **On** constant 1 (true) 03 = … see more in parameter 61.08 **Int. Scaling: 1 == 1 Type: C Volatile:**	Default	Default	ACW_Bit15	-	
63.08	**TeRefMin (minimal tension reference)** Function block : **TensRef** Input : **iRefMinw** **Int. Scaling: 100 == 1 % Type: SI**	0.00	100.00	1.00	%	
63.20	**TeRefOut (Tension reference output of TensRef)** Function block : **TensRef** Input : **iOut** **Int. Scaling: 100 == 1 % Type: SI**	-320.00	320.00	0.00	%	

Index	Signal / Parameter name	min.	max.	def.	unit	E/C
Group 64	**Inertia+Friction Control Winder application**					



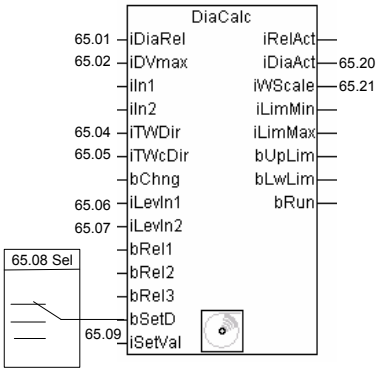
64.01	**DeriIn (input value for derivation)** **Int. Scaling: 1 == 1** **Type:** **I**	-30000	30000	0	-	
64.02	**SelDeriIn (selector for input, which is to be derivated)** Velocity reference selector for input as main reference (to function block VelRef): 0 = **Def_Para** DeriIn (parameter 64.01) selected 1 = **AI1** analog input 1 2 = **AI2** analog input 2 3 = **AI3** analog input 3 4 = **AI4** analog input 4 5 = **AI5** analog input 5 6 = **AI6** analog input 6 7 = **Special** ScaledVal (value out of parameter 61.20) 8 = **Extra** Speed RampOut (value out of parameter 2.32) 9 = **AITAC** tacho input 10 = **Encoder2** encoder input 2 **Int. Scaling: 1 == 1** **Type:** **C** **Volatile:**	Def_Para	Encoder2	Special	-	
64.03	**AccFilter (PT1 filter time for incoming signal)** **Int. Scaling: 1 == 1ms** **Type: SI** **Volatile:**	1	30000	100	ms	
64.04	**AccTD (derivation time)** **Int. Scaling: 1 == 1ms** **Type: SI** **Volatile:**	1	30000	100	ms	
64.05	**AccMul (multiplier to scale output)** **Int. Scaling: 1 == 1** **Type: SI** **Volatile:**	-32767	32767	1	-	

Index	Signal / Parameter name	min.	max.	def.	unit	E/C
64.06	**AccDiv (devisor to scale output)** **Int. Scaling: 1 == 1** **Type: SI** **Volatile:**	-32767	32767	1	-	
						
64.12	**AccTqZeroLev (Value below the output is set to zero)** Function block : **AccTorq** Input : **iZeroLev** **Int. Scaling: 1 == 1** **Type: I**	0	20000	100		
64.13	**AccMecPosSca (acceleration torque for mechanic part)** Function block : **AccTorq** Input : **iMecPos** **Int. Scaling: 10 == 1 %** **Type: I**	0.0	100.0	0.0	%	
64.14	**AccMecNegSca (deceleration torque for mechanic part)** Function block : **AccTorq** Input : **iMecNeg** **Int. Scaling: 10 == 1 %** **Type: I**	0.0	100.0	0.0	%	
64.15	**AccCoiPosSca (acceleration torque for the coil)** Function block : **AccTorq** Input : **iCoiPos** **Int. Scaling: 10 == 1 %** **Type: I**	0.0	100.0	0.0	%	
64.16	**AccCoiNegSca (deceleration torque for the coil)** Function block : **AccTorq** Input : **iCoiNeg** **Int. Scaling: 10 == 1 %** **Type: I**	0.0	100.0	0.0	%	
64.17	**AccCoiWidth (width of the coil)** Function block : **AccTorq** Input : **iCoiWid** **Int. Scaling: 100 == 1%** **Type: SI**	0.00	100.00	100.00	%	

Index	Signal / Parameter name	min.	max.	def.	unit	E/C
						
64.19	**TqCpSpIn (input of speed value)** **Read Only** This parameter shows the actual speed in percent for easy setting of all TqCpSpx parameters. Function block : **TorqComp** Input : **inSp** **Int. Scaling: 100 == 1% Type: I**	0	20000	0	%	
64.20	**TqCpSp0 (supporting point of speed value)** Function block : **TorqComp** Input : **iSpx** **Int. Scaling: 100 == 1% Type: I**	0	10000	0	%	
64.21	**TqCpTq0 (supporting point of torque value)** Function block : **TorqComp** Input : **iTqx** **Int. Scaling: 100 == 1 % Type: I**	0.00	100.00	2.00	%	
64.22	**TqCpSp1 (supporting point of speed value)** Function block : **TorqComp** Input : **iSpx** **Int. Scaling: 100 == 1% Type: I**	0.00	100.00	0.50	%	
64.23	**TqCpTq1 (supporting point of torque value)** Function block : **TorqComp** Input : **iTqx** **Int. Scaling: 100 == 1 % Type: I**	0.00	100.00	0.00	%	
64.24	**TqCpSp2 (supporting point of speed value)** Function block : **TorqComp** Input : **iSpx** **Int. Scaling: 100 == 1% Type: I**	0.00	100.00	20.00	%	

Index	Signal / Parameter name	min.	max.	def.	unit	E/C
64.25	**TqCpTq2 (supporting point of torque value)** Function block : **TorqComp** Input : **iTqx** **Int. Scaling: 100 == 1 % Type: I**	0.00	100.00	0.00	%	
64.26	**TqCpSp3 (supporting point of speed value)** Function block : **TorqComp** Input : **iSpx** **Int. Scaling: 100 == 1% Type: I**	0.00	100.00	40.00	%	
64.27	**TqCpTq3 (supporting point of torque value)** Function block : **TorqComp** Input : **iTqx** **Int. Scaling: 100 == 1 % Type: I**	0.00	100.00	0.00	%	
64.28	**TqCpSp4 (supporting point of speed value)** Function block : **TorqComp** Input : **iSpx** **Int. Scaling: 100 == 1% Type: I**	0.00	100.00	60.00	%	
64.29	**TqCpTq4 (supporting point of torque value)** Function block : **TorqComp** Input : **iTqx** **Int. Scaling: 100 == 1 % Type: I**	0.00	100.00	0.00	%	
64.30	**TqCpSp5 (supporting point of speed value)** Function block : **TorqComp** Input : **iSpx** **Int. Scaling: 100 == 1% Type: I**	0.00	100.00	80.00	%	
64.31	**TqCpTq5 (supporting point of torque value)** Function block : **TorqComp** Input : **iTqx** **Int. Scaling: 100 == 1 % Type: I**	0.00	100.00	0.00	%	
64.32	**TqCpSp6 (supporting point of speed value)** Function block : **TorqComp** Input : **iSpx** **Int. Scaling: 100 == 1% Type: I**	0.00	100.00	100.00	%	
64.33	**TqCpTq6 (supporting point of torque value)** Function block : **TorqComp** Input : **iTqx** **Int. Scaling: 100 == 1 % Type: I**	0.00	100.00	0.00	%	
64.34	**TqCpDia (torque value dependend on diameter)** Function block : **TorqComp** Input : **iTqDia** **Int. Scaling: 100 == 1 % Type: SI**	-100.00	100.00	0.00	%	

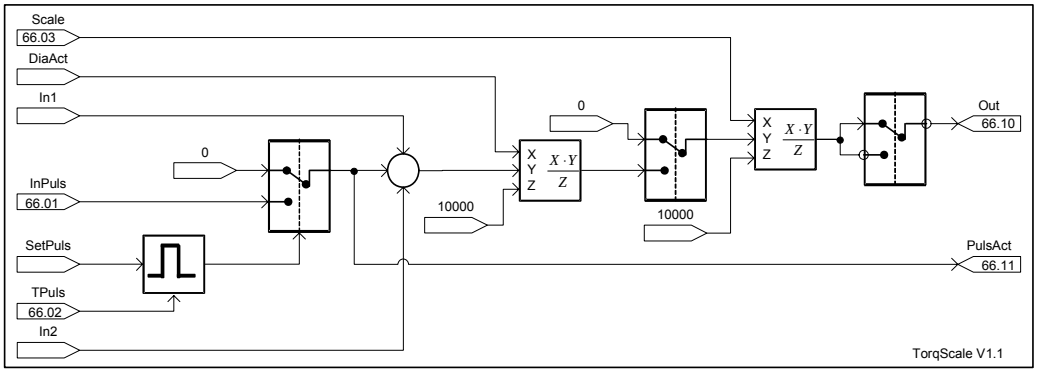
Index	Signal / Parameter name	min.	max.	def.	unit	E/C
64.39	**TqSpeedCtrl (torque compensation also during speed control mode)** 0 = **No** Torque compensation is active during tension controlled mode 1 = **Yes** Torque comp. is also active during normal speed controlled mode **Int. Scaling: 1 == 1 Type: SI**	No	Yes	No	-	
64.40	**AccFiltOut (output of derivated signal)** Output signal of the derivated velocity signal, which can be used for accelerating torque. **Int. Scaling: 1 == 1 Type: SI**	-30000	30000	0	/	
64.43	**AccTqOut (output of acceleration torque)** **Int. Scaling: 100 == 1% Type: I**	-320.00	320.00	0.00	%	
64.46	**TrqCpOut (totalized output of torque compensation)** **Int. Scaling: 1 == 1 Type: I**	-32767	32767	0.00	-	
Group 65	**Diameter calculator** **Winder application**					



65.01	**MinCoreDia (diameter relation = min. core diameter)** Function block : **DiaCalc** Input : **iDiaRel** **Int. Scaling: 100 == 1 % Type: I**	1.00	100.00	10.00	%	

Index	Signal / Parameter name	min.	max.	def.	unit	E/C
65.02	**DiaVmax (diameter with fastest line speed and max. speed)** Function block : **DiaCalc** Input : **iDVmax** **Note:** This parameter has no influence, if the value is lower than value of 65.01. Please let this value lower, because this feature is not tested inside the whole winder application. **Int. Scaling: 100 == 1 %** **Type: I**	1.00	100.00	10.00	%	
65.04	**TrampDir (ramp time of calculation in winding direction)** Function block : **DiaCalc** Input : **iTWDir** **Int. Scaling: 1 == 10 ms** **Type: SI**	-300.00	300.00	10.00	s	
65.05	**TrampWcDir (Ramp time of calcul. in counter winding direct.)** Function block : **DiaCalc** Input : **iTWcDir** **Int. Scaling: 1 == 10 ms** **Type: SI**	-300.00	300.00	10.00	s	
65.06	**RelVeloLev (Release level of velocity (line speed))** Function block : **DiaCalc** Input : **iLevIn1** **Int. Scaling: 1 == 1** **Type: I**	0	32000	20		
65.07	**RelSpeedLev (Release level of actual motor speed)** Function block : **DiaCalc** Input : **iLevIn2** **Int. Scaling: 1 == 1** **Type: I**	0	32000	20		
65.08	**SelDiaInit (selector of initialization command)** Parameter selects the input of initialization command for setting the initial value. 00 = **Default** according winder control word ; here bit 3 of Used_WiCW (7.12). 01 = **NotUsed** constant 0 (false) 02 = **On** constant 1 (true) 03 = … see more in parameter 61.08 **Int. Scaling: 1 == 1** **Type: C** **Volatile:**	Default	Default	ACW Bit15	-	
65.09	**DiaInit 1 (parameter for initial diameter value 1)** Instead of this value other analogue inputs can be selected, please see parameter 65.10 **Int. Scaling: 100 == 1 %** **Type: I**	1.00	100.00	10.00	%	

Index	Signal / Parameter name	min.	max.	def.	unit	E/C
65.10	**SelDiaInit1 (selector of initial value 1)** Velocity reference selector for input as main reference (to function block VelRef): 0 = **Def_Para** DiaInit1 (parameter 65.09) selected 1 = **AI1** analog input 1 2 = **AI2** analog input 2 3 = **AI3** analog input 3 4 = **AI4** analog input 4 5 = **AI5** analog input 5 6 = **AI6** analog input 6 **Int. Scaling: 1 == 1** **Type:** **C** **Volatile:**	Def_Para	AI6	Def_Para	-	
65.11	**DiaInit 2 (parameter for initial diameter value 2)** This value can be selected via parameter 65.12. **Int. Scaling: 100 == 1 %** **Type: I**	1.00	100.00	10.00	%	
65.12	**SelActInit2 (selector of activation initial value 2)** Parameter selects the switch over between initial value 1 and 2. 00 = **Default** according winder control word ; here bit 10 of Used_WiCW (7.12). 01 = **NotUsed** constant 0 (false) 02 = **On** constant 1 (true) 03 = … see more in parameter 61.08 **Int. Scaling: 1 == 1** **Type:** **C** **Volatile:**	Default	Default	ACW_Bit15	-	
65.16	**DiaLevel1 (diameter level 1 for output signal)** Function block : **WinLogO** Input : **iIn** **Int. Scaling: 100 == 1 %** **Type: I**	1.00	100.00	50.00	%	
65.17	**DiaTime1 (time for diameter level 1 output)** Function block : **WinLogO** Input : **iTime1** **Int. Scaling: 1000 == 1 s** **Type: I**	0.000	30.000	2.000	s	
65.18	**DiaLevel2 (diameter level 2 for output signal)** Function block : **WinLogO** Input : **iLogIn1** **Int. Scaling: 100 == 1 %** **Type: I**	1.00	100.00	50.00	%	
65.19	**DiaTime2 (time for diameter level 2 output)** Function block : **WinLogO** Input : **iTime2** **Int. Scaling: 1000 == 1 s** **Type: I**	0.000	30.000	2.000	s	
65.20	**CalcDiaAct (signal: calculated diameter)** Function block : **DiaCalc** Input : **iSetVal** **Int. Scaling: 100 == 1 %** **Type: I**	1.00	100.00	1.00	%	

Index	Signal / Parameter name	min.	max.	def.	unit	E/C
65.21	**WinScale (output of diameter calculator)** Function block : **DiaCalc** Input : **iWScale** **Int. Scaling: 100 == 1 %** **Type: I**	1.00	100.00	100.00	%	
Group 66	**Torque Control** **Winder application**					
						
66.01	**TorquePuls (Value of start impulse)** Function block : **TorqScale** Input : **iInPuls** **Int. Scaling: 100 == 1 %** **Type: I**	0.00	320.00	0.00	%	
66.03	**TorqueScale (Scaling factor of torque sum)** Function block : **TorqScale** Input : **iScale** **Int. Scaling: 100 == 1 %** **Type: I**	-320.00	320.00	0	%	
66.10	**TrqScaOut (Actual output signal of torque scaling)** Function block : **TorqScale** Input : **iPulsAct** **Int. Scaling: 100 == 1 %** **Type: SI**	-320.00	320.00	0.00	%	
66.11	**TePulsAct (Actual signal of tension puls)** Function block : **TorqScale** Output : **Int. Scaling: 100 == 1 %** **Type: SI**	-320.00	320.00	0.00	%	

Index	Signal / Parameter name	min.	max.	def.	unit	E/C
Appendix	**Writing values of standard parameters**					
	This winder application is overwriting the following standard parameters: 07.02 AuxCtrlWord 07.03 AuxCtrlWord2 22.08 BalRampRef 23.01 SpeedRef 23.08 WinWidthPos 23.09 WinWidthNeg 23.15 DirectSpeedRef 24.03 KpS 24.09 TiS 25.01 TorqRefA 26.02 LoadComp 50.06 SpeedFiltTime 50.17 WinderScale **Please don't write on these parameters, too!**					
Appendix	**Presetting values of standard parameters**					
	This application needs the following basic presetting, compared to the factory setting: 23.12 WinCtrlMode = SpeedActWin 26.01 TorqSel = Add 26.03 TorqSelMod = Fix 90.02 DsetXVal2 = 6101 90.03 DsetXVal3 = 6401					

Safety instructions

What this chapter contains

This chapter contains the safety instructions you must follow when installing, operating and servicing the drive. If ignored, physical injury or death may follow, or damage may occur to the drive, the motor or driven equipment. Read the safety instructions before you work on the unit.

To which products this chapter applies

The information is valid for the whole range of the product DCS800, the converter modules DCS800-S0x size D1 to D7, field exciter units DCF80x, etc. like the Rebuild Kit DCS800-R00-9xxx.

Usage of warnings and notes

There are two types of safety instructions throughout this manual: warnings and notes. Warnings caution you about conditions which can result in serious injury or death and/or damage to the equipment, and advise on how to avoid the danger. Notes draw attention to a particular condition or fact, or give information on a subject. The warning symbols are used as follows:

Dangerous voltage warning warns of high voltage which can cause physical injury or death and/or damage to the equipment.

General danger warning warns about conditions, other than those caused by electricity, which can result in physical injury or death and/or damage to the equipment.

Electrostatic sensitive devices warning warns of electrostatic discharge which can damage the equipment.

Installation and maintenance work

These warnings are intended for all who work on the drive, motor cable or motor. Ignoring the instructions can cause physical injury or death and/or damage to the equipment.

WARNING!

- **Only qualified electricians are allowed to install and maintain the drive!**
- Never work on the drive, motor cable or motor when main power is applied.
  Always ensure by measuring with a multimeter (impedance at least 1 Mohm) that:
  1. Voltage between drive input phases U1, V1 and W1 and the frame is close to 0 V.
  2. Voltage between terminals C+ and D- and the frame is close to 0 V.
- Do not work on the control cables when power is applied to the drive or to the external control circuits. Externally supplied control circuits may cause dangerous voltages inside the drive even when the main power on the drive is switched off.
- Do not make any insulation resistance or voltage withstand tests on the drive or drive modules.
- Isolate the motor cables from the drive when testing the insulation resistance or voltage withstand of the cables or the motor.
- When reconnecting the motor cable, always check that the C+ and D- cables are connected with the proper terminal.

Note:

- The motor cable terminals on the drive are at a dangerously high voltage when the main power is on, regardless of whether the motor is running or not.
- Depending on the external wiring, dangerous voltages (115 V, 220 V or 230 V) may be present on the relay outputs of the drive system (e.g. SDCS-IOB-2 and RDIO).
- DCS800 with enclosure extension: Before working on the drive, isolate the whole drive system from the supply.

Grounding

These instructions are intended for all who are responsible for the grounding of the drive. Incorrect grounding can cause physical injury, death and/or equipment malfunction and increase electromagnetic interference.

WARNING!

- Ground the drive, motor and adjoining equipment to ensure personnel safety in all circumstances, and to reduce electromagnetic emission and pick-up.
- Make sure that grounding conductors are adequately sized and marked as required by safety regulations.
- In a multiple-drive installation, connect each drive separately to protective earth (PE ⏚).
- Minimize EMC emission and make a 360° high frequency grounding (e.g. conductive sleeves) of screened cable entries at the cabinet lead-through plate.
- Do not install a drive equipped with an EMC filter to an ungrounded power system or a high resistance-grounded (over 30 ohms) power system.

Note:

- Power cable shields are suitable as equipment grounding conductors only when adequately sized to meet safety regulations.
- As the normal leakage current of the drive is higher than 3.5 mA AC or 10 mA DC (stated by EN 50178, 5.2.11.1), a fixed protective earth connection is required.

Printed circuit boards and fiber optic cables

These instructions are intended for all who handle the circuit boards and fiber optic cables. Ignoring the following instructions can cause damage to the equipment.

WARNING! The printed circuit boards contain components sensitive to electrostatic discharge. Wear a grounding wrist band when handling the boards. Do not touch the boards unnecessarily.

Use grounding strip:

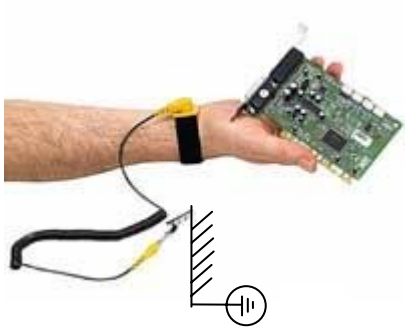


ABB order no.: 3ADV050035P0001

WARNING! Handle the fiber optic cables with care. When unplugging optic cables, always grab the connector, not the cable itself. Do not touch the ends of the fibers with bare hands as the fiber is extremely sensitive to dirt. The minimum allowed bend radius is 35 mm (1.38 in.).

Mechanical installation

These notes are intended for all who install the drive. Handle the unit carefully to avoid damage and injury.

WARNING!

- DCS800 sizes D4 ... D7: The drive is heavy. Do not lift it alone. Do not lift the unit by the front cover. Place units D4 and D5 only on its back.
  DCS800 sizes D5 ... D7: The drive is heavy. Lift the drive by the lifting lugs only. Do not tilt the unit. The unit will overturn from a tilt of about 6 degrees.
- Make sure that dust from drilling does not enter the drive when installing. Electrically conductive dust inside the unit may cause damage or lead to malfunction.
- Ensure sufficient cooling.
- Do not fasten the drive by riveting or welding.

Operation

These warnings are intended for all who plan the operation of the drive or operate the drive. Ignoring the instructions can cause physical injury or death and/or damage to the equipment.

WARNING!

- Before adjusting the drive and putting it into service, make sure that the motor and all driven equipment are suitable for operation throughout the speed range provided by the drive. The drive can be adjusted to operate the motor at speeds above and below the base speed.
- Do not control the motor with the disconnecting device (disconnecting mains); instead, use the control panel keys and , or commands via the I/O board of the drive.
- Mains connection
  You can use a disconnect switch (with fuses) to disconnect the electrical components of the drive from the mains for installation and maintenance work. The type of disconnect switch used must be as per EN 60947-3, Class B, so as to comply with EU regulations, or a circuit-breaker type which switches off the load circuit by means of an auxiliary contact causing the breaker's main contacts to open. The mains disconnect must be locked in its "OPEN" position during any installation and maintenance work.
- EMERGENCY STOP buttons must be installed at each control desk and at all other control panels requiring an emergency stop function. Pressing the STOP button on the control panel of the drive will neither cause an emergency stop of the motor, nor will the drive be disconnected from any dangerous potential.
  To avoid unintentional operating states, or to shut the unit down in case of any imminent danger according to the standards in the safety instructions it is not sufficient to merely shut down the drive via signals "RUN", "drive OFF" or "Emergency Stop" respectively "control panel" or "PC tool".
- Intended use
  The operating instructions cannot take into consideration every possible case of configuration, operation or maintenance. Thus, they mainly give such advice only, which is required by qualified personnel for normal operation of the machines and devices in industrial installations.
  If in special cases the electrical machines and devices are intended for use in non-industrial installations - which may require stricter safety regulations (e.g. protection against contact by children or similar) - these additional safety measures for the installation must be provided by the customer during assembly.

Note:

- When the control location is not set to Local (L not shown in the status row of the display), the stop key on the control panel will not stop the drive. To stop the drive using the control panel, press the LOC/REM key and then the stop key  .

Commissioning

After hardware installation, the drive must be commissioned. The following tools are recommended for commissioning:

- DriveWindow light (start-up assistant) See also *DCS800 Quick Guide 3ADW000191*
- DriveWindow 2.2 (actual value recording and analysis) **COM8-board needed!**

Guidance

Please check that the steps can be done in terms of electrical and mechanical construction of your application.

The preconditions of the following steps are the default parameter setting of DCS800 firmware and the software application. Partition: **S**tandard, **A**pplication, **W**inder

- S01 Connect DriveWindow light and open the start-up assistant “Wizard”.
  Follow the instructions! For more Wizard support please see *DCS800 Quick Guide 3ADW000191*

- S02 Please take care that the winder is equipped with the empty core,
  because the maximum motor speed will be reached with the following steps.

- S03 1. Name plate data
  Special hints for winder application:
  20.02 The **maximum** (20.01 as invers minimum) **speed** is the value, which is needed for the maximum velocity (line speed) measured on the core (minimum diameter)

$$n_{max} = \frac{v_{max}}{\pi \cdot D_{min}} \cdot i$$

n_{max}	maximal needed motor speed in rpm
v_{max}	maximum velocity (line speed) in m/min
D_{min}	minimum diameter (core) in m
i	gear ratio (motor / load)

  If the calculated value n_{max} can not be handled by the motor, the maximum motor speed from motor name plate is to be set into 20.02 (invers value in 20.01). In this case the maximum line speed can not be reached with minimum diameter.

  30.16 Please account for overspeed value also a preset of 10% (default value). Check the higher value of e.g. 115..120%, that also of motor datas and mechanical point of view.

- S04 2. Macro assistant
  Special hints for winder application:
  Deselect here digital inputs, which are used for winder application control
- S05 3. Autotuning field current controller (→ follow the instructions)
- S06 4. Autotuning armature current controller (→ follow the instructions)

- S07 5. Speed feedback assistant (→ follow the instructions)
- S08 6. Autotuning speed controller (→ follow the instructions)
  Set the step response inside the range of Slow and Normal
- S09 7. Field weakening assistant (→ follow the instructions)
- S10 Advanced: Activate all built in options, like field bus adapter an additional IOs.
- S11 It is recommended to save the parameters settings, before continuing with winder application software!
  Please write down also the values of parameters 24.03 and 24.09.
- A01 Switch off the electronic supply of DCS800.
  Plug in the MemoryCard SDCS-MEM-8.
  Switch on the electronic supply of DCS800 again.
- A02 By using the **MemoryCard with ready loaded application** please go to point A03

  Loading application by using DWL (DriveWindow light)
  - Take care that the PC is connected via COM-1 port to the DCS800.
  - Set DWL in Offline mode.
  - Select the Tools menu and then “CoDeSys Application Download”
  - “Select…” the desired application program file (extension PRG).
  - “Send” it to the DCS800.
  - After finishing “Close” the current “ABB Download Tool” window.

  Loading application by using ControlBuilder DCS800
  (use instruction concerning the training)
- A03 **Activate** the application by using parameter 16.06 with “Enable application”.
- A04 Close DriveWindow (light).
  A restart of the tool is necessary to get also the parameters of the winder application.
  A restart of the DCS800 is necessary to get also the parameters via COM-8 board to DriveWindow.

- W01 **Open DriveWindow (light).**

- W02 **Preset the following parameters out of the DCS800 firmware:**

23.12	WinCtrlMode	=	SpeedActWin
26.01	TorqSel	=	Add
26.03	TorqSelMod	=	Fix
90.02	DsetXVal2	=	6101
90.03	DsetXVal3	=	6401
90.04	DsetXplus2val1	=	0
90.05	DsetXplus2val2	=	0

Preset of the following values (written down in step S11):

61.14	KpDiaMin	=	value of 24.03
61.15	TnDiaMin	=	value of 24.09

- W03 **Calculate the diameter relation:**

  → $65.01 = MinCoreDia = \frac{D\min}{D\max} \cdot 100\% =$ _____

  Only if the calculated value n_{max} (see step S03) is higher than the maximum motor speed, then parameter 65.02 is to be set higher than 65.01:

  → $65.02 = D_{V\max} = \frac{v\max}{\pi \cdot n_{\max\ Motor}} \cdot i \cdot \frac{100\%}{D_{\max}} =$ _____

- W04 **Winder control and presetting:**
  Please set also the digital inputs as desired:

60.01	WinderMode	=	**Ind.Tension**
60.02	SelReWinder	=	your selection
60.03	SelWinDir	=	your selection
60.04	SelWinderOn	=	your selection
61.02	SelVelRef	=	your selection
61.08	SelJog1	=	your selection
61.09	SelJog2	=	your selection
65.08	SelDiaInit	=	your selection
65.09	DiaInit1	=	value of 65.01

- W05 **Measure loss compensation parameters with only core load.**
  Set the diameter to minimum value. Set also the parameter 64.39 TqSpeedCtrl = Yes, so that the loss compensation will also work in velocity controlled mode. It is easier concerning polarities to measure in the condition with rewinder (=ON) and the WinDir (winder direction = NotUsed). In this case also the local reference can be used, which value is to be set in rpm.

  For this test it is necessary to measure with warm conditions of the mechanic! After warm-up the speed range should be split into 7 supporting points and the actual motor torque is accommodated in this points.

  During measuring the actual speed can be read out of parameter 64.19 TqCpSpIn. This value is in percent, same unit as speed depending parameters of TqCpSpx.
  The torque value of parameter 2.09 can be used as values of TqCpTqx.
  Note: Measuring of these points should be done without oscillation of torque or speed.

The following values show the speed depending supporting points as default.

- 64.32 TqCpSp6 = 100,0% 64.33 TqCpTq6 = measured value
- 64.30 TqCpSp5 = 80,0% 64.31 TqCpTq5 = measured value
- 64.28 TqCpSp4 = 60,0% 64.29 TqCpTq4 = measured value
- 64.26 TqCpSp3 = 40,0% 64.27 TqCpTq3 = measured value
- 64.24 TqCpSp2 = 20,0% 64.25 TqCpTq2 = measured value
- 64.22 TqCpSp1 = 0,5% 64.23 TqCpTq1 = measured value
- 64.20 TqCpSp0 = 0,0% 64.21 TqCpTq0 = measured value (adhesion)

- W06 **Acceleration: Derivation of velocity reference**

The derivative signal can either created out of the velocity reference or can be taken as value depending on the superior ramp generator.

- **… either derivative velocity signal**
  The output ScaledVal of the velocity reference function block is the default input source. This assumes that the winder drive gets a ramped reference.

  During commissioning this ramped reference is mostly not free available. So the accelerating adjustment can also be done by using the DCS800 internal ramp generator, if the shortest ramp time is known.
- Therefore set the selector via parameter 64.02 SelDeriIn to “Spec 2”, which connects the input to the signal SpeedRampOut (2.32).
- Please take care:
  For the next steps the minimum diameter has continuously be set.
  If no other values known, start with 64.03 AccFilT= 40ms and 64.04 AccTD = 100ms.
- Let the reference ramp up and down.
  Observe the wave form of output 64.20 AccFiltOut. If necessary do an adaptation with the parameters 64.03 and 64.04.
- For increasing the amplitude to the necessary value of ±30000 as highest value for the shortest ramp time, please use the parameters 64.05 AccMult and 64.06 AccDiv for adaptation.
  Take care that a regular value will not generate a value higher than ±30000.
- **… ramp depending value**
  By using a derivative value of a superior ramp generator set the following parameters:
  64.02 SelDeriIn = ? select the input
  64.04 AccTD = 0 function block will not do a derivation
- For increasing the amplitude to the necessary value of ±30000 as highest value for the shortest ramp time, please use the parameters 64.05 AccMult and 64.06 AccDiv for adaptation.
  Take care that a regular value will not generate a value higher than ±30000.

- W07 **Accelerating compensation of winder mechanics**
  Let the reference ramp up and down.
  Observe the output of speed controller, parameter 2.09, and write the values of ramping up and down into parameter 64.13 and 64.14:
  64.13 AccMecPosSca = value of 2.09 during ramping up
  64.14 AccMecNegSca = value of 2.09 during ramping down.

  Stop and switch off the winder drive.

- W08 **Big coil needed**
  For the next steps the coil (reel) with the maximum values of this winder drive has to be mounted.
  If the values (diameter, width) is lower, than please set the correct value into the parameters.
  64.17 AccCoiWidth = 100% or the real mounted percental width
  65.09 DiaInit1 = 100% or the real mounted percental diameter

  Take care that the new diameter init value is set and take care that the web is fixed, because the coil will turn up to maximum line speed.

- W09 **Tuning of speed controller with maximum diameter**
  If parameter 64.39 TqSpeedCtrl is set to No, the adjustment can be done without losses and mechanical accelerating torque.
  Let the winder run and increase carefully the speed up to the maximum **line speed**, not the maximum motor speed. Measure the line speed on the surface of the coil.

  Adjust the speed controller, which works as line speed controller. The following values are used:
  61.16 KpDiaMax = value of proportional part of controller
  61.17 TnDiaMax = value of integral part of controller

  Please note:
  - The filter of actual speed is increased depending on diameter.
  - The controller with this settings are used while e.g. jogging and E-Stop.

- W10 **Set parameter 61.18 KpWinMode**
  The value of this parameter is only responsible during window control, which is activated while running in tension controlled mode.
  A preset value could be:
  61.18 KpWinMode = 5 times of value of 61.16 KpDiaMax

- W11 **Accelerating compensation of the coil**
  The diameter and the width are still to be set to the current values as described in point W08. Set also parameter 64.39 TqSpeedCtrl to Yes.

Calculate according the following equation:

$$64.15 = 64.16 = \frac{\rho \cdot W \cdot \pi}{D_{max} \cdot} \cdot \left(D_{max}{}^4 - D_{min}{}^4\right) \cdot \frac{1}{i} \cdot \frac{dv}{dt} \cdot \frac{100\%}{T_{Motor}}$$

Velocity $dv \left[\frac{m}{s}\right]$ Strip density $\rho \left[\frac{kg}{m^3}\right]$ Acceleration time $dt \ [s]$

Diameter $D \ [m]$ Web width $W \ [m]$ Gear ratio $i \ [-]$

Nominal motor torque $T_{Motor} \ [Nm]$

Set the reference ramp up and down.
Observe the output of speed controller, parameter 2.09. If necessary adjust the parameters 64.15 and 64.16 so, that the speed controller output stays near zero.:

64.15 AccCoiPosSca = value of 2.09 during ramping up
64.16 AccCoiNegSca = value of 2.09 during ramping down.

Finally stop and switch off the winder drive.

- W12 **Diameter calculation**

The calculated signal will be ramped with following parameters>

65.04 TrampWDir = ramp time of the expected output slope

65.05 TrampWcDir = ramp time of counter expected output slope

If a clue is needed, please use the following equation:

$$65.04 = 0.4 \cdot t_{Ramp} \qquad t_{Ramp} \approx \frac{D_{max}{}^2 \cdot \pi}{v \cdot 2 \cdot \delta}$$

$$65.05 = 0.8 \cdot t_{Ramp}$$

Velocity $v \left[\frac{m}{s}\right]$ Thickness $\delta \ [m]$ Diameter $D \ [m]$

Perhaps these ramp values have to be modified during running under production.

- W13 **Torque scaling**

  The tension reference (also called tension set point) has to be scaled. This is done with parameter 66.03 TorqueScale

$$66.03 = \frac{F_{max} \cdot D_{max}}{2 \cdot i} \cdot \frac{100\%}{T_{Motor}}$$

  Tension F [N] Diameter D [m] Gear ratio i [-]

  Nominal motor torque T_{Motor} [Nm]

- W14 **Set back parameters**
  A few parameters are set to other values for supporting this commissioning.
  Set them back:

64.02 SelDeriIn	= Spec 1	(default)	or as desired; see W06
64.39 TqSpeedCtrl	= No	(default)	or as desired

- W15 **Further commissioning**
  Up to now it has been a basic commissioning.
  Further settings are to be done yet, if desired.

Control Bits (Example)

The following table shows one possibility of controlling this winder application.
In some cases it could be necessary to take other combinations.

Commands	ControlWord:Bit	New start	Switch On	Jogging 1	Jogging 2	Run w/o winding	Stop w/o winding	**Winding**	E-stop	Reset
ON	7.04:00	0	1	1	1	1	1	**1**	1	0
E-off	7.04:01	1	1	1	1	1	1	**1**	1	1
E-stop	7.04:02	1	1	1	1	1	1	**1**	0	1
RUN	7.04:03	0	0	1	1	1	0	**1**	1	0
RpOutZero	7.04:04	0/1	0/1	1	1	1	1	**1**	1	0/1
RpHold	7.04:05	0/1	0/1	1	1	1	1	**1**	1	0/1
RpInZero	7.04:06	0/1	0/1	1	1	1	1	**1**	1	0/1
Reset	7.04:07	0/1	0	0	0	0	0	**0**	0	1
Remote	7.04:10	1	1	1	1	1	1	**1**	1	1
WinOn	7.12:02	0/1	0	0	0	0	0	**1**	0/1	0/1
WinJog1	7.12:08	0	0	1	0	0	0	**0**	0/1	0/1
WinJog2	7.12:09	0	0	0	1	0	0	**0**	0/1	0/1
DiaInit	7.12:03	1	0/1	0/1	0/1	0/1	0/1	**0**	0/1	0/1
Status / Mode		Off	Main cont. on	Velocity control int. reference		Velocity control ext. reference	Velocity control reference = 0	Tension control	Stopping as selected	Reset faults

Appendix A - Application handling

Identification

Parameter 4.03

With running winder application, the parameter 4.03 shows:

4.03 ApplicName = **DEABBDC-WiIT**

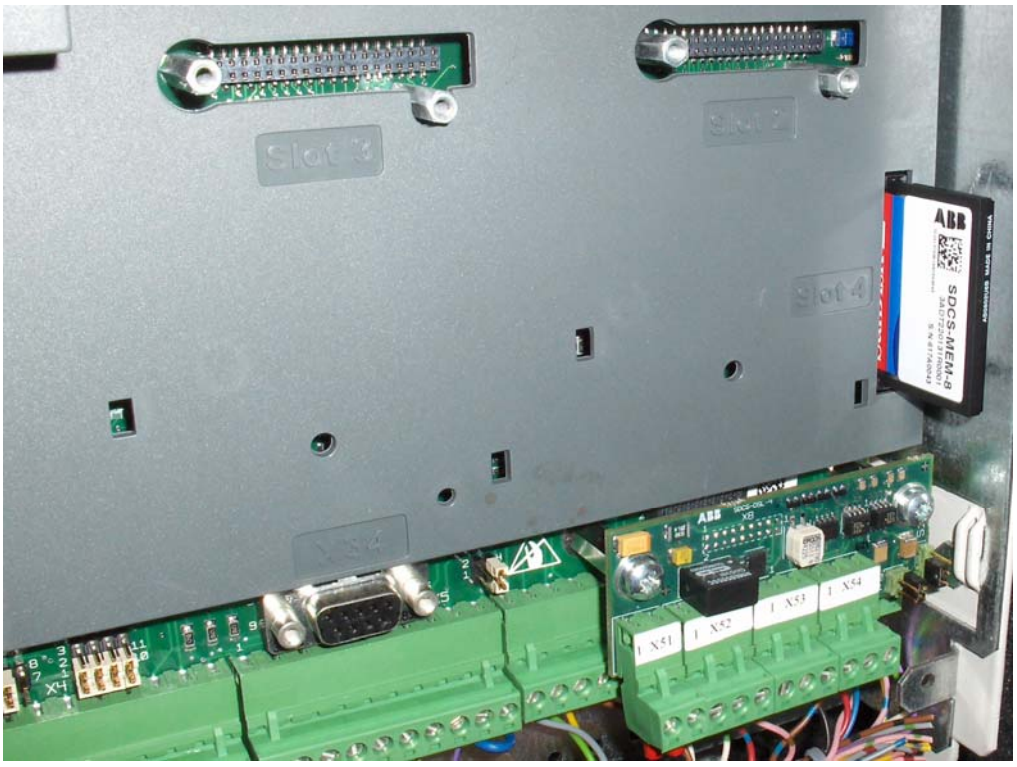
Parameter 4.12

With running winder application, the parameter 4.12 shows:

4.12 ApplicVer = **1.1**

Installation

Installation of the SDCS-MEM8 (memory card) is possible in slot 4 of the CON-4 board (see picture).
Installation or changing memory cards is only allowed if the electronic supply is switched off.



Enable / disable application

The application can be enabled and disabled by using parameter 16.06:

- Enable application (activate application program)
- Disable application (deactivate application program)

After the activation of the memory card, all application parameters will be visible and active.

ABB Automation Products
Wallstadter Straße 59
68526 Ladenburg • Germany
Tel: +49 (0) 62 03-71-0
Fax:+49 (0) 62 03-71-7609
www.abb.com/motors&drives

Ident. No.: 3ADW000308R0201 Rev. B
05_2008

308R0201A8210000