

sFWxx

INHALT

<i>Historie</i>	2
<i>Allgemeines</i>	2
<i>CoDeSys Strukturen</i>	2
SFW.....	3
T_PD.....	8
tEEPROM.....	8
tMASTER_EEPROM.....	9
tSLAVE_EEPROM.....	17
tUSER_EEPROM.....	18
uFW_CLC.....	19
<i>CoDeSys Funktionen</i>	22
FwDATE2Day.....	22
FwDATE2Month.....	24
FwDATE2Year.....	24
FwDATE2Hour.....	25
FwDATE2Minute.....	25
FwDATE2Second.....	26
FwGetWeekday.....	26
FwGetDATE_TIME.....	27
FwDATE_TIME2DATE_AND_TIME.....	27
FwDDMMYYYY2DATE.....	28
FwHHMMSS2TIME.....	28
FwLED (Ab Version 2).....	29
FwLoad (Ab Version 2).....	30
FwSave (Ab Version 2).....	30
FwGetStructPointer.....	31
FwCFAvailable (Ab Version 5).....	31
FwCFRestore (Ab Version 5).....	32
FwCFSave (Ab Version 5).....	33
FwGetFreeMemoryspaceOfCF (Ab Version 3).....	35
FwGetFreeMem (Ab Version 14 / Nur bei Windows CE Geräten).....	35
FwGetFreeDiskSpace (Ab Version 14 / Nur bei Windows CE Geräten).....	35
FwTurnDisplayOn (Ab Version 4).....	36
FwGetCRC16.....	36
FwGetCOMINStructPointer (Ab Version 6).....	37
FwGetCOMOUTStructPointer (Ab Version 6).....	37
TECHNISCHE UNTERSTÜTZUNG.....	38

Historie

Version	Datum	Autor	Inhalte
V2.81.0	03.05.2005	Hitzelberger	Neuanlage
V2.81.1	02.06.2005	Hitzelberger	Die vorhanden Funktionen ergänzt
V2.81.2	20.10.2005	Prodana	Die vorhanden Funktionen ergänzt
V2.81.3	26.10.2005	G.Schauer	Ergänzung der sFW Felder
V2.82.0	27.10.2006	Prodana	CF Funktionalität
V2.82.1	02.11.2006	Essig	Korrekturen
V2.82	27.07.2007	Kramer	Stand zu Release V2.82
V2.83	15.12.2008	Prodana	Stand zu Release V2.83
V2.84	29.11.2010	Nething	Stand zu Release V2.84
V2.91.1	21.06.2012	Essig	Stand zu Release V2.91
V2.91.2	10.06.2014	N. Nething	Fokusrechteck für Drehgeber
V2.91.3	27.04.2018	N. Nething	tMASTER_EEPROM: byModbusTimeout, bLogIDinList, wExtFlags sFW: Gerätetemperatur, CAN-Busabschluß, Artikelnummer, Soft-und Hardarelabel, Überarbeitung, Bitparameter
V2.91.4	31.08.2018	F.Nething	Neue Funktionen FwGetFreeMem und FwGetFreeDiskSpace
V2.91.5	05.08.2018	Essig	Zusätzliche Optionen bei FwLoad und FwSave

Allgemeines



Diese Bibliothek ist nur unter dem Programmiersystemen CoDeSys verfügbar.
In dieser Bibliothek sind grundsätzliche firmwarenahe Funktionen enthalten.

Die Funktionen korrelieren mit Telnetbefehlen.
Lesen Sie hierzu auch die Dokumente „Plattform_CE_DE.pdf“ und „Plattform_mE_DE.pdf“ das Kapitel „Telnetbefehle“ um konkrete Beispiele zu erhalten.

CoDeSys Strukturen

Es stehen folgende Strukturen zur Verfügung:

- sFW
- T_PD
- tEEPROM
- tMASTER_EEPROM
- tSLAVE_EEPROM
- tUSER_EEPROM
- uFW_CLC

SFW

Diese Struktur enthält allgemeine Firmwareeinstellungen. Hier sFW13.lib

TYPE sFW :

STRUCT

```

    pOfThisStruct           : DWORD;
    nSizeOfFWStruct        : DWORD;
    nVersionOfFirmware      : DWORD;
    szFirmwareLabel         : STRING(31);
    szFirmwareDate          : STRING(15);
    szFirmwareTime          : STRING(11);
    psUW                    : DWORD;
    nActualState            : DWORD;
    nActualTime             : DWORD;
    nActualDate             : DWORD;
    uRemanent               : tEEPROM;
    nCrcOfEEProm           : INT;
    nOUIOfSlave             : ARRAY [0..63] OF DWORD;
    nOrderOfSlave           : ARRAY [0..63] OF BYTE;
    nTimeoutOfSlave         : ARRAY [0..63] OF BYTE;
    nTypeOfSlave            : ARRAY [0..63] OF BYTE;
    nFWVersionOfSlave       : ARRAY [0..63] OF INT;
    nHWVersionOfSlave       : ARRAY [0..63] OF INT;
    nPdoInitState           : BYTE;
    nSdoInitState           : BYTE;
    nUsedPdos               : ARRAY [0..63] OF BYTE;
    nUsedPdoTab             : ARRAY [0..63] OF INT;
    nPdoToSend              : ARRAY [0..63] OF BYTE;
    aPdo                    : ARRAY [0..511] OF T_PD;
    nUsedSdos               : ARRAY [0..63] OF BYTE;
    nUsedSdoTab             : ARRAY [0..63] OF INT;
    nSdoWriteCommand        : BYTE;
    nSdoWriteSlaveNo        : BYTE;
    nSdoWriteIndex          : INT;
    db4SdoWriteData         : DWORD;
    aSdo                    : ARRAY [0..8191] OF DWORD;
    nSemaClockSetup         : INT;
    nSemaCheckEEPROM        : INT;
    nSema1000ms             : INT;
    bSaveFTPFiles           : BYTE;
    bSwitchUWRunning        : BYTE;
    nCountOfOb14            : INT;
    nLastPressedKey         : INT;
    nProcessorType          : BYTE;
    bDownloadRunning        : BYTE;
    nIECTimeInMs            : DWORD;
    nPhysicalAddressInTarget : DWORD;
    bDownloadOfFirmware     : BYTE;
    biButtonPresent         : BYTE;
    bESBConfigurationFailedPowerUp : BYTE;

```

<i>bESBConfigurationFailedRunning</i>	: BYTE;
<i>nTriggerDynPdo</i>	: ARRAY [0..63] OF DWORD;
<i>BeeperTimer</i>	: BYTE;
<i>BeeperManual</i>	: BYTE;
<i>bFirmwareDownloadRunning</i>	: BYTE;
<i>bDebugMode</i>	: BYTE;
<i>sCLC</i>	: uFW_CLC;
<i>nHexSwitch</i>	: BYTE;
<i>nDummyByte</i>	: BYTE;
<i>LCDOffTimer</i>	: INT;
<i>EthernetBlockingTelegramCntr</i>	: INT;
<i>EthernetBlockingTelegramActive</i>	: BYTE ;
<i>DPSOapClients</i>	: BYTE;
<i>nCurrentFWChecksumWithoutHistory</i>	: DWORD;
<i>nTempOfDevice</i>	: DWORD;
<i>nEditStyle</i>	: BYTE;
<i>nLastBrightness</i>	: BYTE;
<i>byNormalMode</i>	: BYTE;
<i>bNumLock</i>	: BYTE;
<i>nESBCmd</i>	: BYTE;
<i>nDummyBytes0</i>	: BYTE;
<i>nSemaCheckEEPROM2</i>	: INT;
<i>nTypeOfSlaveCE</i>	: ARRAY [0..63] OF BYTE;
(* type of CE-Units, plugged onto CMxxx or CSxxx:	
1:CE001;2:CE100;3:CE101;4:CE150;5:CE151;6:CE152 *)	
<i>nAoutSensorType</i>	: ARRAY [0..15] OF BYTE;
<i>nLastIP_gateway</i>	: DWORD;
<i>nLastIP_subnet</i>	: DWORD;
<i>nLastIP_address</i>	: DWORD;
<i>nLastDHCPactivated</i>	: BYTE;
<i>nLastKeybBacklightBrightness</i>	: BYTE;
<i>nLastContrast</i>	: BYTE;
<i>nDummy4BytesAlign1</i>	: BYTE;
<i>nThreadCount</i>	: ARRAY [0..15] OF DWORD;
(* Count Thread callings::Index 0: ob0, Index 1: ob1, Index 2 ob2, ... *)	
<i>nDevMaxTemp</i>	: INT; (* Max. device temperature in °C *)
<i>nDevMinTemp</i>	: INT; (* Min. device temperature in °C *)
<i>bCAN1Termination</i>	: BYTE;(* CAN1 termination *)
<i>aDummy1</i>	: ARRAY[0..2] OF BYTE;
<i>szArticleNumberDevice</i>	: STRING(19);(* elrest article number of device *)
<i>nSoftwareChangeStatusDevice</i>	: BYTE;(* software change status / version of devices' software
package *)	
<i>nHardwareChangeStatusDevice</i>	: BYTE; (* hardware change status of device *)
<i>END_STRUCT</i>	
<i>END_TYPE</i>	

Elemente

pOfThisStruct	Nur für internen Gebrauch
nSizeOfFWStruct	Nur für internen Gebrauch
nVersionOfFirmware	Version der aktuellen Firmware
szFirmwareLabel	Firmware als Zeichenkette
szFirmwareDate	Erstellungsdatum der Firmware als Zeichenkette
szFirmwareTime	Erstellungszeit der Firmware als Zeichenkette
psUW	Nur für internen Gebrauch

nActualState	Nur für internen Gebrauch
nActualTime	Verwenden Sie stattdessen aus der CoDeSys Bibliothek SysRTCLib.lib die Funktion SysRTCGetTime ()
nActualDate	Verwenden Sie stattdessen aus der CoDeSys Bibliothek SysRTCLib.lib die Funktion SysRTCGetTime ()
uRemanent	Nur für internen Gebrauch
nCrcOfEEProm	Nur für internen Gebrauch
nOUIOfSlave	Siehe hierzu Kommunikation_ESB_DE.pdf
nOrderOfSlave	Siehe hierzu Kommunikation_ESB_DE.pdf
nTimeoutOfSlave	Siehe hierzu Kommunikation_ESB_DE.pdf
nTypeOfSlave	Siehe hierzu Kommunikation_ESB_DE.pdf
nFWVersionOfSlave	Siehe hierzu Kommunikation_ESB_DE.pdf
nHWVersionOfSlave	Siehe hierzu Kommunikation_ESB_DE.pdf
nPdoInitState	Nur für internen Gebrauch
nSdoInitState	Nur für internen Gebrauch
nUsedPdos	Nur für internen Gebrauch
nUsedPdoTab	Nur für internen Gebrauch
nPdoToSend	Nur für internen Gebrauch
aPdo	Nur für internen Gebrauch
nUsedSdos	Nur für internen Gebrauch
nUsedSdoTab	Nur für internen Gebrauch
nSdoWriteCommand	Nur für internen Gebrauch
nSdoWriteSlaveNo	Nur für internen Gebrauch
nSdoWriteIndex	Nur für internen Gebrauch
db4SdoWriteData	Nur für internen Gebrauch
aSdo	Nur für internen Gebrauch
nSemaClockSetup	Nur für internen Gebrauch
nSemaCheckEEPROM	Nur für internen Gebrauch
nSema1000ms	Nur für internen Gebrauch
bSaveFTPFile	Verwenden Sie stattdessen aus der CoDeSys Bibliothek sFWxx.lib die Funktion FwSave ()
bSwitchUWRRunning	Nur für internen Gebrauch
nCountOfOb14	Nur für internen Gebrauch
nLastPressedKey	Verwenden Sie stattdessen aus der CoDeSys Bibliothek Pultxx.lib die Funktion KeyboardActualKey()
nProcessorType	Nur für internen Gebrauch
bDownloadRunning	Nur für internen Gebrauch
nIECTimeInMs	Nur für internen Gebrauch
nPhysicalAddressInTarget	Nur für internen Gebrauch
bDownloadOfFirmware	Nur für internen Gebrauch
biButtonPresent	Nur für internen Gebrauch
bESBConfigurationFailedPowerU	Siehe hierzu Kommunikation_ESB_DE.pdf
bESBConfigurationFailedRunning	Siehe hierzu Kommunikation_ESB_DE.pdf
nTriggerDynPdo	Nur für internen Gebrauch
BeeperTimer	Nur für internen Gebrauch
BeeperManual	Nur für internen Gebrauch

bFirmwareDownloadRunning	Nicht ändern; TRUE beim Firmware-Update.
bDebugMode	Unbenutzt
sCLC	Nur für internen Gebrauch
nHexSwitch	Nur für internen Gebrauch
nDummyByte	Nur für internen Gebrauch
LCDOffTimer	Nur für internen Gebrauch
EthernetBlockingTelegramCntr	Telnet memstat -Befehl, eth-t
EthernetBlockingTelegramActive	Telnet memstat -Befehl, eth-t
DPSoapClients	Die aktuelle Anzahl der SOAP-Clients, die eine SOAPStart aber noch keine SOAPStop Anforderung gesandt haben.
nCurrentFWChecksumWithoutHistory	Nur für internen Gebrauch
nTempOfDevice	Nur für internen Gebrauch
nEditStyle	Telnet Option 34: Edit Style [<0>-standard,1-4keys,2-6keys]
nLastBrightness	Nur für internen Gebrauch
byNormalMode	unbenutzt
bNumLock	Nur für internen Gebrauch
nESBCmd	Nur für internen Gebrauch
nDummyBytes0	unbenutzt
nSemaCheckEEPROM2	unbenutzt
nTypeOfSlaveCE	Nur für internen Gebrauch
nAoutSensorType	Nur für internen Gebrauch
nLastIP_gateway	Letzte gesetzte sMasterEEProm.nMyIP_gateway
nLastIP_subnet	letzte gesetzte sMasterEEProm. nMyIP_subnet
nLastIP_address	letzte gesetzte sMasterEEProm.nMyIP_address
nLastDHCPactivated	letzte gesetzte sMasterEEProm.nDHCPactivated
nLastKeybBacklightBrightness	letzte gesetzte sMasterEEProm.nKeybBacklightBrightness
nLastContrast	Letzte gesetzte sMasterEEProm.nLCDContrast
nThreadCount	unbenutzt
nDevMaxTemp	Maximale Gerätetemperatur in °C
nDevMinTemp	Minimale Gerätetemperatur in °C
bCAN1Termination	CAN1 Termination aktiv (0-off, 1-on); CAN Option 36 Hinweis: Termination von CAN0 über sFW über z.B.: psFW^.uRemanent.sSlaveEEProm.OptionFlag.5 = FALSE; CAN0/CAN1 Termination über SystemParameter.lib. z.B. SetSystemParameter(SP_CAN0_TERMINATION, 1);
szArticleNumberDevice	Artikelnummer (aus EEPROM)
nSoftwareChangeStatusDevice	Software change status (siehe Befehl ,state')(aus EEPROM)
nHardwareChangeStatusDevice	Hardware change status (siehe Befehl ,state')(aus EEPROM)

T_PD

Diese Struktur enthält .

```

TYPE T_PD:
STRUCT
    PdIn          : DWORD;
    PdOut         : DWORD;
END_STRUCT
END_TYPE
    
```

Elemente

- PdIn** Nur für internen Gebrauch
- PdOut** Nur für internen Gebrauch

tEEPROM

Diese Struktur enthält .

```

TYPE tEEPROM:
STRUCT
(*      EEPROM          : ARRAY [0..1023] OF INT;
*)
    sUserEEprom        : tUSER_EEPROM;
    gap                : ARRAY [0..45] OF INT;
    sMasterEEprom      : tMASTER_EEPROM;
    sSlaveEEprom       : tSLAVE_EEPROM;
END_STRUCT
END_TYPE
    
```

Elemente

- sUserEEprom** Frei vom Kunden nutzbarer im EEprom gespeicherter Wertebereich.
- gap** Nur für internen Gebrauch
- sMasterEEprom** Siehe Strukturbeschreibung
- sSlaveEEprom** Siehe Strukturbeschreibung

tMASTER_EEPROM

Diese Struktur enthält .

TYPE tMASTER_EEPROM:

STRUCT

nFocusRectangleData	:DWORD; (* Contains the settings for the focus rectangle. Format: 16#tsBBGGRR where t (4 Bits) =Thickness, s(4Bits)=Line style(0=solid, 1=dot,2=dash,3=dashdot,4=dashdotdot) , BBGGRR is the RGB color. *)
nTouchMarkData	:DWORD;(* Contains the settings for the touch mark. Format: 16#rrBBGGRR where BBGGRR is the RGB color and rr the radius. MSB in rr is used to enable/disable the markf *)
usStepperEnableBaseboard	: WORD; (* Used by CE130.lib *)
usStepperEnableExtension	: WORD; (* Used by CE130.lib *)
byFINCfBaseExtension	: BYTE; (* Used by CE130.lib *)
byReserved0	: BYTE;
usReserved1	: WORD;
usReserved2	: WORD;
byReserved1	: BYTE;
byReserved2	: BYTE;
usReserved3	: WORD;
byReserved3	: BYTE;
byReserved4	: BYTE;
nRemotePanelNodeIDMin	: BYTE;
nRemotePanelNodeIDMax	: BYTE;
nRebootCounterCExx	: WORD;
nTaskWatchdogTimeOut	: BYTE;
nSRAMWaitStates	: BYTE;
nEditStyle	: BYTE;
nKeybBacklightBrightness	: BYTE;
byCan0Distance	: BYTE;
byCan1Distance	: BYTE;
nLogFlags1	: DWORD;
nLogOutput	: INT;
nMyHWVersion1	: BYTE;
nMyHWVersion0	: BYTE;
nMyType	: DWORD;
nEETypeOfSlaveCE	: ARRAY [0..63] OF BYTE;
nUnused	: BYTE; Use internal Keyboard ein (Wie Telnet o 27 1) psfW^uRemanent.sMasterEEProm.nUnused := psfW^uRemanent.sMasterEEProm.nUnused OR 2;
nCAN1ReinitTick	: BYTE;
nCAN0ReinitTick	: BYTE;
ESBActivated	: BYTE; (* 0 = no ESB, 1=CAN0, 2=CAN1, only for CE-Panel *)
nModbusPortRD	: INT;
nModbusPortWR	: INT;
nPERSISTENT_CODE_BEGIN	: DWORD;(* reserved, do not change , set by image *)
nPERSISTENT_CODE_LEN	: DWORD;(* reserved, do not change , set by image *)
nRESSOURCE_CODE_BEGIN	: DWORD;(* reserved, do not change , set by image *)
nRESSOURCE_CODE_LEN	: DWORD;(* Telnet Option 30 *)
nIEC61000_CODE_BEGIN	: DWORD;(* reserved, do not change , set by image *)
nIEC61000_CODE_LEN	: DWORD;(* reserved, do not change , set by image *)
db4Can1MaskRegister	: DWORD;(* reserved, do not change, set by image *)
db4Can1BaseAdr	: DWORD;(* reserved, do not change, set by image *)
nRemotePanelQueueSize	: INT;(* Queue size for Remote-Panel *)
byKeyMatrixIndex	: BYTE;(* Telnet Option 33 *)
nCAN1BaudrateIndex	: BYTE;(* Index for CAN1-Baudrate *)
nCAN1Baudrate	: WORD; (* CAN1 Baud [10,20,50,100,<123>,125,250,500] *)
bCAN129bit	: BYTE; (* CAN1 29bit *)
bElCAN1Active	: BYTE; (* CAN1 native, CAUTION regarde bCAN1open1Active *)
nCAN1ClientNo	: WORD; (* CAN1 NodeID (My Module) *)
nCAN1IntermodulMaster	: WORD; (* CAN1 Intermod Master *)

nCAN1BaseSendID	: DWORD;	(* CAN1 Send Id *)
nCAN1BaseRecvID	: DWORD;	(* CAN1 Recv Id *)
nCAN1BaseIntermodulID	: DWORD;	(* CAN1 Intermod Id *)
nCAN1openClientNo	: BYTE;	(* CAN1 NodeID (My Module) *)
bCAN1open1Active	: BYTE;	(* CAN1 CANOpen Active/Node *)
db4CAN0MaskRegister	: DWORD;	(* reserved, do not change, set by image *)
db4CAN0BaseAdr	: DWORD;	(* reserved, do not change, set by image *)
nDisplayBrightness	: BYTE;	(* Brightness of display backlights *)
bRemotePanelCANPort	: BYTE;	(* Remote Panel CAN0/1 *)
bRemotePanels	: BYTE;	(* Remote Panel Srv/Client *)
nNoOfADUAverageElements	: BYTE;	(* Count of adu sampling *)
bOpenEmptyMaskFirst	: BYTE;	(* Open empty Mask first *)
bSavePersistFile	: BYTE;	(* Exist persist.bin file *)
bRemotePanelCount	: WORD;	(* every 'bRemotePanelCount' a delay of 'bRemotePanelDelay' will be inserted *)
bRemotePanelIDs	: ARRAY [0..3] OF BYTE;	(* Panel ID of RemotePanel *)
bRemotePanelDelay	: DWORD;	(* multiple of 0.1ms *)
nPassword	: ARRAY [0..15] OF DWORD;	
FredFwFilesDate	: ARRAY [0..15] OF INT;	
FredFwFilesTime	: ARRAY [0..15] OF INT;	
bOpenKeyPadWithEnter	: BYTE;	
nDisplayShutOffBright	: BYTE;	
nLCDShutOffTime	: INT;	
nCAN0openClientNo	: BYTE;	
bCAN0openActive	: BYTE;	
nRs232Baudrate	: DWORD;	
nLCDContrast	: BYTE;	
nUDPStartupDelay	: BYTE;	
nTouchBeeperTimer	: BYTE;	
bEditWithTouchKeyPad	: BYTE;	
nPrimaryLanguage	: INT;	
nKeyboardBeeperTimer	: BYTE;	
bKeyboardRepetiermode	: BYTE;	
nCAN0Baudrate	: INT;	
bCAN029bit	: BYTE;	
bElaCAN0Active	: BYTE;	
nCAN0ClientNo	: INT;	
nCAN0IntermodulMaster	: INT;	
nCAN0BaseSendID	: DWORD;	
nCAN0BaseRecvID	: DWORD;	
nCAN0BaseIntermodulID	: DWORD;	
nTaskSwitchTicks	: ARRAY [0..5] OF INT;	
byESBConfig	: BYTE;	
EnableCF	: BYTE;	(* Bit 0: = Telnet Option 10; Bit 1: Telnet Option 27)
bSocketType	: BYTE;	
nCoDeSysCOMPPort	: BYTE;	
nScreenSample	: ARRAY [0..5] OF INT;	
nCalibrationValid	: INT;	
nMyIP_gateway_PPP	: DWORD;	
nMyIP_gateway	: DWORD;	
nMyIP_subnet_PPP	: DWORD;	
nMyIP_subnet	: DWORD;	
nMyIP_address_PPP	: DWORD;	

```
nReserved04           : STRING(15);
szUserPassword        : STRING(15);
nRebootCounter        : INT;

wExtFlags              : WORD;
nFirmwareChecksumWithHistory : DWORD;
nFirmwareChecksumWithoutHistory : DWORD;
nEEPROMValid          : DWORD;
nDHCPactivated        : BYTE;
nReserved03           : BYTE;
nMyIP_address         : DWORD;
nMyLogicalModulNumber : INT;
byModbusTimeout       : BYTE;
bLogIDinList          : BYTE; (* LogID in Alarmlist ein- / ausschalten (0/1) *)
nReserved02           : DWORD;
nReserved01           : DWORD;
db4MACID              : DWORD;
nCountOfStoreFTP      : INT;
END_STRUCT
END_TYPE
```

Elemente

nFocusRectangleData	Bit 31..28 Telnet Option 47: Focus Rectangle Thickness Bit 27 -24 Telnet Option 45: Focus Rectangle Style [0=Solid, 1=Dot, 2=Dash, 3=DashDot, 4=DashDotDot] Bit 0 - 23 Telnet Option 46: Focus Rectangle Color
nTouchMarkData	Bit 31 Telnet Option 42: Touch Mark On/Off Bit 24 – 30 Telnet Option 43: Touch Mark Radius Bit 0 – 23 Telnet Option 44: Touch Mark Color
nRemotePanelNodeIDMin	Telnet CAN Option 33: Remote Panel NodeID Min
nRemotePanelNodeIDMax	Telnet CAN Option 34: Remote Panel NodeID Max
nRebootCounterCExx	Nur für internen Gebrauch
nTaskWatchdogTimeOut	Telnet Thread Option 10: Task Watchdog Timeout
nSRAMWaitStates	Telnet Option 35: SRAM Waitstate [0,<1>]
nEditStyle	Telnet Option 34: Edit Style [<0>-standard,1-4keys,2-6keys]
nKeybBacklightBrightness	Nur für internen Gebrauch
byCan0Distance	Unbenutzt
byCan1Distance	Unbenutzt
nLogFlags1	Siehe OptionFlag, Bit 10.
nLogOutput	Nur für internen Gebrauch
nMyHWVersion1	Nur für internen Gebrauch
nMyHWVersion0	Nur für internen Gebrauch
nMyType	Nur für internen Gebrauch
nEETypeOfSlaveCE	Nur für internen Gebrauch
nCAN1ReinitTick	Unbenutzt
nCAN0ReinitTick	Unbenutzt
ESBActivated	Telnet CAN Option 1: ESB functionality
nModbusPortRD	Telnet Option 21: Modbus Port Read <502>
nModbusPortWR	Telnet Option 22: Modbus Port Write <502>
nPERSISTENT_CODE_BEGIN	Telnet Option-Befehl -> Persist Memory
nPERSISTENT_CODE_LEN	Telnet Option-Befehl -> Persist Memory
nRESSOURCE_CODE_BEGIN	Telnet Option-Befehl -> Resource (Resource.bin)
nRESSOURCE_CODE_LEN	Telnet Option-Befehl -> Resource (Resource.bin)
nIEC61000_CODE_BEGIN	Telnet Option-Befehl -> >IEC code (iec_code.bin)
nIEC61000_CODE_LEN	Telnet Option-Befehl -> IEC code (iec_code.bin)
db4Can1MaskRegister	Nur für internen Gebrauch
db4Can1BaseAdr	Nur für internen Gebrauch
nRemotePanelQueueSize	Telnet CAN Option 32: Remote Panel Queue Size
byKeyMatrixIndex	Telnet Option 33: Active Key Matrix
nCAN1BaudrateIndex	Telnet CAN Option 13: CAN1 Baud
nCAN1Baudrate	Baudrate des CAN1 Interfaces. Telnet CAN Option 13 : CAN1 Baud
bCAN129bit	TRUE bedeutet 29bit (Extendet Frame Format) des CAN1 Interfaces FALSE bedeutet 11bit (Standard Frame Format) des CAN1 Interfaces Telnet CAN Option 14: CAN1 Extended (29bit)
bElaCAN1Active	ElaCAN Protokoll wird aktiviert am CAN1 Interfaces Telnet CAN Option 17: CAN1 ElaCAN Active
nCAN1ClientNo	Logische Node Nummer des CAN1 Interfaces Telnet CAN Option 15: CAN1 NodeID (My Module)

nCAN1IntermodulMaster	Nur in Verbindung mit ElaCAN Protokoll, siehe hierzu Kommunikation_CAN_DE.pdf Telnet CAN Option 21: CAN1 Intermod Master								
nCAN1BaseSendID	Nur in Verbindung mit ElaCAN Protokoll, siehe hierzu Kommunikation_CAN_DE.pdf Telnet CAN Option 18: CAN1 Send Id								
nCAN1BaseRecvID	Nur in Verbindung mit ElaCAN Protokoll, siehe hierzu Kommunikation_CAN_DE.pdf Telnet CAN Option 19: CAN1 Recv Id								
nCAN1BaseIntermodulID	Nur in Verbindung mit ElaCAN Protokoll, siehe hierzu Kommunikation_CAN_DE.pdf Telnet CAN Option 20: CAN1 Intermod Id								
nCAN1openClientNo	Logische CANopen Node Nummer des CAN1 Interfaces								
bCAN1openActive	Freischaltung der CANopen Funktionalität des CAN1 Interfaces Telnet CAN Option 16: CAN1 CANopen Active/Node								
db4Can0MaskRegister	Nur für internen Gebrauch								
db4Can0BaseAdr	Nur für internen Gebrauch								
nDisplayBrightness	Telnet Option 26: Brightness Display								
bRemotePanelCANPort	CAN-Port; Telnet CAN Option 31								
bRemotePanels	Remote Panel Typ – Telnet CAN Option 24								
	<table border="1"> <thead> <tr> <th>Wert</th> <th>Bedeutung</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>kein</td> </tr> <tr> <td>1, 11</td> <td>Server; 1 alte Version</td> </tr> <tr> <td>2, 12</td> <td>Client; 2 alte Version</td> </tr> </tbody> </table>	Wert	Bedeutung	0	kein	1, 11	Server; 1 alte Version	2, 12	Client; 2 alte Version
Wert	Bedeutung								
0	kein								
1, 11	Server; 1 alte Version								
2, 12	Client; 2 alte Version								
nNoOfADUAverageElements	Telnet Option 25: Count of adu sampling								
bOpenEmptyMaskFirst	Bit 0: Telnet Option 24: Open empty Mask first Bit 1: Telnet Option 36: Hour glass by open mask								
bSavePersitFile	Telnet Option 23: Exist persist.bin file								
bRemotePanelCount	Telnet CAN Option 26: Remote Panel Count Frame								
bRemotePanelIDs	ID der momentan angeschlossenen Panels								
bRemotePanelDelay	$i = i * 0.1$ ms, Pausezeit jede bRemotePanelCount CAN-Frames. Telnet CAN Option 25								
nPassword	Nur für internen Gebrauch								
FredFwFilesDate	Nur für internen Gebrauch								
FredFwFilesTime	Nur für internen Gebrauch								
bOpenKeyPadWithEnter	Telnet Option 18: Open key pad with ENTER								
nDisplayShutOffBright	Telnet Option 17: ShutOff Brightness								
nLCDShutOffTime	Telnet Option 16: ShutOff Backlight								
nCAN0openClientNo	Logische CANopen Node Nummer des CAN0 Interfaces								
bCAN0openActive	Freischaltung der CANopen Funktionalität des CAN0 Interfaces Telnet Option 5: CAN0 CANopen Active/Node								
nRs232Baudrate	Telnet RS232 Option 4: RS232 Baudrate: [600...<38400>...57600] Baudrate des RS232 COM0 Interfaces								
nLCDContrast	Telnet Option 15: Contrast LCD								
nUDPStartupDelay	Telnet Option 14: Start delay UDP+CoDeSys								
nTouchBeeperTimer	Telnet Option 12: Touch beeper								
bEditWithTouchKeyPad	Telnet Option 11: Touch used for edit								

nPrimaryLanguage	Nur für internen Gebrauch
nKeyboardBeeperTimer	Siehe in Platform_XX_DE.pdf: Key beeper („option 8“):
bKeyboardRepetiermode	Siehe in Platform_XX_DE.pdf: Key repeatabili. („option 7“):
nCAN0Baudrate	Baudrate des CAN0 Interfaces Telnet CAN Option 2: CAN0 Baud
bCAN029bit	TRUE bedeutet 29bit (Extendet Frame Format) des CAN0 Interfaces FALSE bedeutet 11bit (Standard Frame Format) des CAN0 Interfaces Telnet CAN Option 3: CAN0 Extended (29bit) [0, 1]
bElaCAN0Active	ElaCAN Protokoll wird aktiviert am CAN0 Interfaces Telnet CAN Option 6: CAN0 ElaCAN Active
nCAN0ClientNo	Logische Node Nummer des CAN0 Interfaces Telnet CAN Option 4: CAN0 NodeID (My Module)
nCAN0IntermodulMaster	Nur in Verbindung mit ElaCAN Protokoll, siehe hierzu Kommunikation_CAN_DE.pdf Telnet CAN Option 10: CAN0 Intermod Master
nCAN0BaseSendID	Nur in Verbindung mit ElaCAN Protokoll, siehe hierzu Kommunikation_CAN_DE.pdf Telnet CAN Option 7: CAN0 Send Id
nCAN0BaseRecvID	Nur in Verbindung mit ElaCAN Protokoll, siehe hierzu Kommunikation_CAN_DE.pdf Telnet CAN Option 8: CAN0 Recv Id
nCAN0BaseIntermodulID	Nur in Verbindung mit ElaCAN Protokoll, siehe hierzu Kommunikation_CAN_DE.pdf Telnet CAN Option 9: CAN0 Intermod Id
nTaskSwitchTicks	Zykluszeit der Threads: ob0, ob1, ob2, TCPIP und ESB. Telnet Befehl: threads.
byESBConfig	Bit7..4 = Priority TCP-IP, Telnet Option 9: TCP-Prio higher as OB: X Bit 3..0 =0 = ESB Check OUI, 1=ESB Check Typ, 2=ESB No Check siehe Telnet CAN Option 2 wenn die ESB-Funktionalität aktiv ist: Save ESB-Configuration
bEnableCF	Bit 0 = Telnet Option 10: Enable CompactFlash Bit 1 = Telnet Option 27: Enable CF-FW
bSocketType	Telnet Option 20: Modbus0=Off,1=TCP,2=UDP: 20
nCoDeSysCOMPort	Telnet RS232 Option 3: RS232 COMx for CoDeSys: [0=disable,1=COM0]
nScreenSample	Nur für internen Gebrauch
nCalibrationValid	Nur für internen Gebrauch
nMyIP_gateway_PPP	Gateway - Adresse des PPP-Netzwerkes, siehe Index[1]
nMyIP_gateway	Gateway - Adresse des Ethernet-Netzwerkes, siehe Index[0]
nMyIP_subnet_PPP	Subnetmask des PPP-Netzwerkes, siehe Index[1]
nMyIP_subnet	Subnetmask des Ethernet-Netzwerkes, siehe Index[0]
nMyIP_address_PPP	IP - Adresse des PPP-Netzwerkes, siehe Index[1]
nReserved04	Nur für internen Gebrauch
szUserPassword	User Kennwort
nRebootCounter	Nur für internen Gebrauch
wExtFlags	Telnet Option 48: ‚Save DataMonitor on CF [0,<1>]‘ Mit dieser Option kann festgelegt werden ob Datamonitor / Datenlogger in das Standardverzeichnis oder die CF/SD Karte speichert.

nFirmwareChecksumWithHistory	Nur für internen Gebrauch
nFirmwareChecksumWithoutHistory	Nur für internen Gebrauch
nEEPROMValid	Nur für internen Gebrauch
nDHCPactivated	Aktivieren der DHCP Funktionalität.
nReserved03	Reserviert; nicht verwenden.
nMyIP_address	IP - Adresse des Ethernet-Netzwerkes, siehe Index[0]
nMyLogicalModulNumber	Logische Node-Nummer des Gerätes. Diese dient zur automatischen Identifikation der Geräte beim Autoscan. Telnet Befehl: „ setmod “
byModbusTimeout	Telnet Option 49: 'Modbus TCP Timeout [ms]' Value x => (x+1)*10 ms
bLogIDinList	Telnet Option 50: 'LogID in Alarm List [0, <1>]' In der Alarm-/Ereignisliste kann in einer zusätzlichen Spalte die logische ID des Meldungstexts angezeigt werden.
nReserved02	Reserviert; nicht verwenden
nReserved01	Reserviert; nicht verwenden.
db4MACID	Die niederwertigen 3 Bytes der eingestellten MAC Adresse. Die höherwertigen 3 Bytes sind festgelegt auf „00-07-7E“. Telnet Befehl setmac .
nCountOfStoreFTP	Reserviert; nicht verwenden.

tSLAVE_EEPROM

Diese Struktur enthält .

TYPE tSLAVE_EEPROM:

STRUCT

```

nFWVersionOfSlave      : ARRAY [0..61] OF INT;
OptionFlag             : INT;
feed                   : INT;
nOUIOfSlave            : ARRAY [0..63] OF DWORD;
nTypeOfSlave           : ARRAY [0..63] OF BYTE;

```

END_STRUCT

END_TYPE

Elemente

NFWVersionOfSlave Siehe hierzu Kommunikation_ESB_DE.pdf

OptionFlag

Bit	Bedeutung
0	aktiviert ein Erweiterungs-Modul mit Ein- bzw. Ausgängen; siehe Telnet Option 1
1	aktiviert ein Erweiterungs-Modul mit RS232 bzw. RS485; siehe Telnet Option 2
2	aktiviert die COM0 Schnittstelle für PPP; siehe Telnet RS232 Option 1: RS232 COM0 as PPP [0,1]
3	aktiviert die COM1 Schnittstelle für PPP; siehe Telnet RS232 Option 6: RS232 COM1 as PPP [0,1]
4	aktiviert die COM0 Schnittstelle für Debugausgaben (im run mode); siehe Telnet RS232 Option 2: RS232 COM0 as DEBUG [0,1]
5	schaltet den CAN-Abschlusswiderstand zu; siehe Telnet CAN Option 0
6	muss 0 sein; nur für interne Gebrauch
7	Linkstats (siehe help diagnostics) werden zyklisch alle 10 Minuten aufgerufen und mit den Ausgaben von Telnet im RAM Speicher abgelegt. Die Speicherung in die Datei eventlog.bin erfolgt durch die Funktion EventLoggerStoring aus der Bibliothek event01.lib. Siehe Telnet Option 13: Trace Enable [0,1]
8	unbenutzt
9	Wenn die Bits 0, 1 und 9 gesetzt sind, dann werden die COM-Module mit 2 kByte Speicher aktiviert; Siehe Telnet Option 3 Wenn die Bits 0, 1 gesetzt sind und Bit 9 nicht, dann werden die COM-Module mit 8 kByte Speicher aktiviert; Siehe Telnet Option 4
10	werden wichtige Profibus-Informationen während dem Bootvorgang ausgegeben; siehe Telnet Profibus Option 1: Set profibus diagnostic ; Bit 1 in nlogFlags1 muss auch gesetzt werden.
11	unbenutzt
12	unbenutzt
13	unbenutzt

14	Aktiviert wird das VFS-Verzeichnis (Virtual File System) wie eine Standardspeichermedium. Damit Daten remanent abgelegt werden ist kein Fixieren mehr nötig. Sonst sind bei ausgeschaltetet Ramdisk eine der nachfolgenden Aktionen nötig: Codesys-> FwSave(2); Telnet -> VFSSYNC Siehe Telnet Option 31: RAMdisk aktiv
15	unbenutzt

nOUIOfSlave Siehe hierzu Kommunikation_ESB_DE.pdf
nTypeOfSlave Siehe hierzu Kommunikation_ESB_DE.pdf

tUSER_EEPROM

Diese Struktur enthält .

```
TYPE tUSER_EEPROM:
STRUCT
    aUserFreeSpace                                : ARRAY [0..511] OF INT;
END_STRUCT
END_TYPE
```

Elemente

aUserFreeSpace Frei vom Kunden nutzbarer im EEprom gespeicherter Wertebereich.



Bei Geräten mit kalibrierbaren analogen Kanälen werden in diesem Bereich die Kalibrierwerte abgelegt, so dass dieser Bereich nicht nutzbar ist

uFW_CLC

Diese Struktur enthält .

TYPE uFW_CLC:

STRUCT

SetPoint	: ARRAY [0..15] OF REAL;	(* Set point	Sollwert *)
Kp1	: ARRAY [0..15] OF REAL;	(* Proportional Band Xp1	Xp Proportionalband = 1/Kp *)
Tn1	: ARRAY [0..15] OF REAL;	(* Integral Action Time Tn1	Tn Nachstellzeit / Integrierzeit *)
Tv1	: ARRAY [0..15] OF REAL;	(* Derivate Action Time Tv1	Tv Vorhaltezeit / Differentialzeit *)
F	: ARRAY [0..15] OF DWORD;	(* PWM frequency	PWM Schaltfrequenz [1/min] *)
Ymax	: ARRAY [0..15] OF REAL;	(* max. Effective Controller Output Y	Stellgroessenbegrenzung *)
CurrentValue	: ARRAY [0..15] OF REAL;	(* Current Value	Istwert *)
Yout	: ARRAY [0..15] OF REAL;	(* Effective Controller Output Y	Stellgröße *)
T	: ARRAY [0..15] OF REAL;	(* Cycle Time T	Abtastzeit T *)
Ymanual	: ARRAY [0..15] OF REAL;	(* Manual Controller Output	Manuelle Stellgröße *)
ScalingFactor	: ARRAY [0..15] OF REAL;	(* Scaling Factor *)	
Xsh2	: ARRAY [0..15] OF REAL;	(* Neutral zone 3point Xsh2 *)	
Tmin1	: ARRAY [0..15] OF REAL;	(* Minimum Pulse Time Tmin1-TMmin	Mindestimpulslänge bei 3PS *)
Tm	: ARRAY [0..15] OF REAL;	(* Motor Response Time Tm	TT Stellzeit des Motorantriebes *)
UnUsed1	: ARRAY [0..15] OF REAL;	(* *)	
UnUsed2	: ARRAY [0..15] OF REAL;	(* *)	
UnUsed3	: ARRAY [0..15] OF REAL;	(* *)	
OutBit1	: ARRAY [0..15] OF REAL;	(* *)	
OutBit2	: ARRAY [0..15] OF REAL;	(* *)	
ControllerMode	: ARRAY [0..15] OF REAL;	(* Controller mode	Reglertyp *)
SensorType	: ARRAY [0..15] OF REAL;	(* Sensor Type	Fühlerart *)
ActualCurrentValue	: ARRAY [0..15] OF REAL;	(* Current Value	Istwert *)
UnUsed4	: ARRAY [0..15] OF REAL;	(* *)	
UnUsed5	: ARRAY [0..15] OF REAL;	(* *)	
UnUsed6	: ARRAY [0..15] OF REAL;	(* *)	
UnUsed7	: ARRAY [0..15] OF REAL;	(* *)	
UnUsed8	: ARRAY [0..15] OF REAL;	(* *)	
Former	: ARRAY [0..15] OF REAL;	(*Fuehrungsformer *)	
Ymin	: ARRAY [0..15] OF REAL;	(* min. Effective Controller Output Y	Stellgroessenbegrenzung *)
ScalingOffset	: ARRAY [0..15] OF REAL;	(* Scaling Offset *)	
Xsh1	: ARRAY [0..15] OF REAL;	(* Neutral Zone 3point Xsh1	Totband *)
UnUsed9	: ARRAY [0..15] OF REAL;	(* *)	
Xsd1	: ARRAY [0..15] OF REAL;	(* Switching difference of signaller Xsd1	Xsd1 = Schaltdifferenz / bei
elrest Hysterese *)			
UnUsed10	: ARRAY [0..15] OF REAL;	(* *)	
UnUsed11	: ARRAY [0..15] OF REAL;	(* *)	
Kp2	: ARRAY [0..15] OF REAL;	(* Proportional Band Xp2 *)	
Tn2	: ARRAY [0..15] OF REAL;	(* Integral Action Time Tn2 *)	
Tv2	: ARRAY [0..15] OF REAL;	(* Derivate Action Time Tv2 *)	

```

nDummy0      : ARRAY [0..15] OF REAL;  (* *)
nDummy1      : ARRAY [0..15] OF REAL;  (* *)
nDummy2      : ARRAY [0..15] OF REAL;  (* *)
nDummy3      : ARRAY [0..15] OF REAL;  (* *)
nDummy4      : ARRAY [0..15] OF REAL;  (* *)
nDummy5      : ARRAY [0..15] OF REAL;  (* *)
nDummy6      : ARRAY [0..15] OF REAL;  (* *)
nDummy7      : ARRAY [0..15] OF REAL;  (* *)
nDummy8      : ARRAY [0..15] OF REAL;  (* *)
nDummy9      : ARRAY [0..15] OF REAL;  (* *)
nDummy10     : ARRAY [0..15] OF REAL;  (* *)
nDummy11     : ARRAY [0..15] OF REAL;  (* *)
(* Data      : ARRAY [0..799] OF DWORD;*)
END_STRUCT
END_TYPE

```

Elemente

SetPoint	Nur für internen Gebrauch
Kp1	Nur für internen Gebrauch
Tn1	Nur für internen Gebrauch
Tv1	Nur für internen Gebrauch
F	Nur für internen Gebrauch
Ymax	Nur für internen Gebrauch
CurrentValue	Nur für internen Gebrauch
Yout	Nur für internen Gebrauch
T	Nur für internen Gebrauch
Ymanual	Nur für internen Gebrauch
ScalingFactor	Nur für internen Gebrauch
Xsh2	Nur für internen Gebrauch
Tmin1	Nur für internen Gebrauch
Tm	Nur für internen Gebrauch
UnUsed1	Nur für internen Gebrauch
UnUsed2	Nur für internen Gebrauch
UnUsed3	Nur für internen Gebrauch
OutBit1	Nur für internen Gebrauch
OutBit2	Nur für internen Gebrauch
ControllerMode	Nur für internen Gebrauch
SensorType	Nur für internen Gebrauch
ActualCurrentValue	Nur für internen Gebrauch
UnUsed4	Nur für internen Gebrauch
UnUsed5	Nur für internen Gebrauch
UnUsed6	Nur für internen Gebrauch
UnUsed7	Nur für internen Gebrauch
UnUsed8	Nur für internen Gebrauch
Former	Nur für internen Gebrauch
Ymin	Nur für internen Gebrauch
ScalingOffset	Nur für internen Gebrauch
Xsh1	Nur für internen Gebrauch
UnUsed9	Nur für internen Gebrauch
Xsd1	Nur für internen Gebrauch
UnUsed10	Nur für internen Gebrauch
UnUsed11	Nur für internen Gebrauch
Kp2	Nur für internen Gebrauch
Tn2	Nur für internen Gebrauch
Tv2	Nur für internen Gebrauch

nDummy0	Nur für internen Gebrauch
nDummy1	Nur für internen Gebrauch
nDummy2	Nur für internen Gebrauch
nDummy3	Nur für internen Gebrauch
nDummy4	Nur für internen Gebrauch
nDummy5	Nur für internen Gebrauch
nDummy6	Nur für internen Gebrauch
nDummy7	Nur für internen Gebrauch
nDummy8	Nur für internen Gebrauch
nDummy9	Nur für internen Gebrauch
nDummy10	Nur für internen Gebrauch
nDummy11t	Nur für internen Gebrauch

CoDeSys Funktionen

Es stehen folgende Funktionen zur Verfügung:

- *FwDATE2Day*
- *FwDATE2Month*
- *FwDATE2Year*
- *FwTIME2HOUR*
- *FwTIME2MINUTE*
- *FwTIME2SECOND*
- *FwGetWeekday*
- *FwGetDATE_TIME*
- *FwDATE_TIME2DATE_AND_TIME*
- *FwDDMMYYYY2DATE*
- *FwHHMMSS2TIME*
- *FwGetCRC16*
- *FwLED (Ab Version 2)*
- *FwLoad (Ab Version 2)*
- *FwSave (Ab Version 2)*
- *FwGetStructPointer*
- *FwGetFreeMemoryspaceOfCF (Ab Version 3)*
- *FwGetFreeMem (Ab Version 14)*
- *FwGetFreeDiskSpace(Ab Version 14)*
- *FwTurnDisplayOn (Ab Version 4)*
- *FwCFAvailable (Ab Version 5)*
- *FwCFRestore (Ab Version 5)*
- *FwCFSave (Ab Version 5)*
- *FwGetCOMINStructPointer*
- *FwGetCOMOUTStructPointer*
-

FwDATE2Day



ST

FUNCTION FwDATE2Day : BYTE

VAR_INPUT

DateIn : DATE;

END_VAR

VAR

END_VAR

Diese Funktion erzeugt aus einer Variablen vom Typ DATE die Information um welchen Tag des Monats es sich handelt.

ST - Beispiel:

Day := FwDATE2Day(DateIn);

FwDATE2Month



ST

*FUNCTION FwDATE2Month : BYTE**VAR_INPUT**DateIn : DATE;**END_VAR**VAR**END_VAR*

Diese Funktion erzeugt aus einer Variablen vom Typ DATE die Information, um welchen Monat des Jahres es sich handelt.

ST - Beispiel:

Month := FwDATE2Month(DateIn);

FwDATE2Year



ST

*FUNCTION FwDATE2Year : WORD**VAR_INPUT**DateIn : DATE;**END_VAR**VAR**END_VAR*

Diese Funktion erzeugt aus einer Variablen vom Typ DATE die Information, um welches Jahr es sich handelt.

ST - Beispiel:

Year := FwDATE2Year(DateIn);

FwDATE2Hour



ST

```
FUNCTION FwTIME2Hour : BYTE  
VAR_INPUT  
    TimeIn : TIME;  
END_VAR  
VAR  
END_VAR
```

Diese Funktion erzeugt aus einer Variablen vom Typ TIME die Information, um welche Stunde des Tages es sich handelt.

ST - Beispiel:

```
Hour := FwTime2Hour(TimeIn);
```

FwDATE2Minute



ST

```
FUNCTION FwTIME2Minute : BYTE  
VAR_INPUT  
    TimeIn : TIME;  
END_VAR  
VAR  
END_VAR
```

Diese Funktion erzeugt aus einer Variablen vom Typ TIME die Information, um welche Minute der Stunde es sich handelt.

ST - Beispiel:

```
Minute := FwTime2Minute(TimeIn);
```

FwDATE2Second



ST

```

FUNCTION FwTIME2Second : BYTE
VAR_INPUT
    Timeln : TIME;
END_VAR
VAR
END_VAR

```

Diese Funktion erzeugt aus einer Variablen vom Typ TIME die Information, um welche Sekunde einer Minute es sich handelt.

ST - Beispiel:

```

Second := FwTime2Second(Timeln);

```

FwGetWeekday



ST

```

FUNCTION FwGetWeekday : BYTE
VAR_INPUT
    Day : BYTE;
    Month : BYTE;
    Year : WORD;
END_VAR
VAR
END_VAR

```

Diese Funktion erzeugt aus einem vorgegebene Datum, bestehend aus Day, Month, Year, den zugehörigen Wochentag.

ST - Beispiel:

```

WeekDay := FwGetWeekday(1, 3, 2005);

```

FwGetDATE_TIME



ST

```

FUNCTION FwGetDATE_TIME : BOOL
VAR_INPUT
    DateAndTimeIn : DATE_AND_TIME;
    pDate : POINTER TO DATE;
    pTime : POINTER TO TIME;
END_VAR
VAR
END_VAR

```

Diese Funktion erzeugt aus einer vorgegebenen Variable vom Typ DATE_AND_TIME die Datumsinformationen in eine Variable vom Typ DATE und die Uhrzeitinformationen in eine Variable vom Typ TIME.

ST - Beispiel:

FwDATE_TIME2DATE_AND_TIME



ST

```

FUNCTION FwDATE_TIME2DATE_AND_TIME :
DATE_AND_TIME
VAR_INPUT
    DateIn : DATE;
    TimeIn : TIME;
END_VAR
VAR
END_VAR

```

Diese Funktion wandelt den Inhalt von zwei Variablen vom Typ DATE und TIME in den Inhalt einer Variablen vom Typ DATE_AND_TIME um.

ST - Beispiel:

FwDDMMYYYY2DATE



ST

```

FUNCTION FwDDMMYYYY2DATE : DATE
VAR_INPUT
    Day : BYTE;
    Month : BYTE;
    Year : WORD;
END_VAR
VAR
END_VAR

```

Diese Funktion wandelt den Inhalt von drei Variablen Day, Month, Year in den Inhalt einer Variablen vom Typ DATE um.

ST - Beispiel:

FwHHMMSS2TIME



ST

```

FUNCTION FwHHMMSS2TIME : TIME
VAR_INPUT
    Hour : BYTE;
    Minute : BYTE;
    Second : BYTE;
    Millisecond : WORD;
END_VAR
VAR
END_VAR

```

Diese Funktion wandelt den Inhalt von vier Variablen Hour, Minute, Second, Millisecond in den Inhalt einer Variablen vom Typ TIME um.

ST - Beispiel:

FwLED (Ab Version 2)

ST

*FUNCTION FwLED : BOOL**VAR_INPUT**nLEDNo : BYTE; (* 0..19 equal LEDs on front *)**bOn : BOOL; (* bOn = TRUE / FALSE *)**END_VAR**VAR**END_VAR*

Diese Funktion können anwendungsspezifische LEDs (falls auf dem jeweiligen Gerät verfügbar) ein bzw. ausgeschaltet werden.

ST - Beispiel:

FwLoad (Ab Version 2)



ST

```

FUNCTION FwLoad : BOOL
VAR_INPUT
    nMode :WORD;    (* 1=Persist.bin, see eLoadOption *)
END_VAR
VAR
END_VAR
END_VAR
TYPE eLoadOption :
(
    LOAD_PERSIST_BIN      := 16#01,    (* load all values from persist.bin or persist2.bin to sram0 *)
    LOAD_DEFAULT          := 16#08    (* load default values for all data points (From Version 14) *)
);
END_TYPE

```

Diese Funktion dient zum Zurückholen definierter Daten aus dem Flash.

ST - Beispiel:

FwSave (Ab Version 2)



ST

```

FUNCTION FwSave : BOOL
VAR_INPUT
    nMode :WORD;    (* single or multiple used bits : Bit0=Persist.bin,
                    Bit1=Eventlog.bin, Bit2=VFS Files, see eSaveOption *)
END_VAR
VAR
END_VAR
END_VAR
TYPE eSaveOption :
(
    SAVE_PERSIST_BIN      := 16#01,    (* save persist.bin, persist2.bin file to internal flash0 disk *)
    SAVE_EVENTLOG_BIN     := 16#02,    (* save eventlog.bin file to internal flash0 disk *)
    SAVE_VFS_FILES        := 16#04,    (* save all file in folder 'vfs' to internal flash0 disk *)
    SAVE_IBUTTON          := 16#08,    (* special option, save to serial cable iButton storage *)
    SAVE_PERSIST_WITHOUT_CHECK := 16#10 (* save data point without checking for changes. Combine
    this Option with SAVE_PERSIST_BIN *)
);
END_TYPE

```

Diese Funktion dient zum Abspeichern definierter Daten in das Flash.

ST - Beispiel:

FwGetStructPointer



ST

```

FUNCTION FwGetStructPointer : DWORD
VAR_INPUT
    nDummy : BOOL;
END_VAR
VAR
END_VAR

```

Programmbeispiel:

(* Variablendeklaration *)

```

VAR
    psFW : POINTER TO sFW:=0;(* Zeiger auf sie Startadresse der Firmwaredatenpunkte *)
END_VAR

```

(* Code *)

```

IF psFW = 0 THEN
    psFW:= FwGetStructPointer(0 (*nDummy*));
END_IF;

```

Diese Funktion gibt die Adresse auf die Firmware-Struktur sFW zurück.
Hiermit kann ein flexibler Zugriff auf die Firmware-Variablen ermöglicht werden

FwCFAvailable (Ab Version 5)



ST

```

FUNCTION FwCFAvailable : BOOL
VAR_INPUT
    nMode : WORD;
END_VAR
VAR
END_VAR

```

(* unused *)

Diese Funktion gibt TRUE zurück wenn eine Compact Flash Karte, bzw die SD- Karte vorhanden ist.

ST - Beispiel:

FwCFRestore (Ab Version 5)



ST

```

FUNCTION FwCFRestore : BOOL
VAR_INPUT
    nMode :WORD; (* used only by the WinCE devices; specifies what should be updated => see
eCFBackupUpdateType enumeration type*)
END_VAR
VAR
END_VAR
TYPE eCFBackupUpdateType :
(
    RUNTIME           := 16#01,
    IMAGE             := 16#02
);
END_TYPE

```

bei µE Geräten:

Mit dieser Funktion kann man die folgenden Dateien mit denen aus dem Verzeichnis './cf0/FRED/UPDATE' aktualisieren:

- lec_code.bin – das CoDeSys Programm
- ram_iec_global.bin, ram_iec_memory.bin, ram_iec_retain.bin
- persist.bin
- eeprom.bin
- resource.bin - Ressource
- Firmware.bin – Image

bei CE Geräten:

Mit dieser Funktion kann man das CE-Image und die ElaDesign-Runtime Dateien mit denen aus dem Verzeichnis ‚\VOLUME_NAME\Control\DEVICE_TYPE\Update\UPDATE_TYPE‘ aktualisieren:

- DEVICE_TYPE.dat – CE-Image
- Default.* – CoDeSys Dateien
- Eeprom.bin
- ElaDesignCE.exe, ElaDesignCoDeSysCE.exe – ElaDesign-Runtime
- ElaLibCE.dll
- eventlog.bin
- persist.bin
- resource.res - Ressource
- keypad*.xml – Tastatur Konfigurationen

VOLUME_NAME – Verzeichnisname der Compact Flash Karte

DEVICE_TYPE – Die erste Zeile aus der Datei: \FlashDisk\SysExtras\System\Device.inf

UPDATE_TYPE – ElaDesign oder Image

FwCFSave (Ab Version 5)



ST

FUNCTION FwCFSave : BOOL

VAR_INPUT

nMode :WORD; (used only by the WinCE devices; specifies what should be backedup => see eCFBackupUpdateType enumeration type*)*

END_VAR

VAR

END_VAR

TYPE eCFBackupUpdateType :

(

RUNTIME := 16#01,

IMAGE := 16#02

);

END_TYPE

bei μ E Geraten:

Mit dieser Funktion kann man die folgenden Dateien im Verzeichnis ‚cf0/FRED/BACKUP‘ speichern:

- lec_code.bin – das CoDeSys Programm
- ram_iec_global.bin, ram_iec_memory.bin, ram_iec_retain.bin
- persist.bin
- eeprom.bin
- resource.bin - Ressource
- Firmware.bin – Image

bei CE Geraten:

Mit dieser Funktion kann man das CE-Image und die ElaDesign-Runtime Dateien im Verzeichnis ‚\VOLUME_NAME\Control\DEVICE_TYPE\Backup\UPDATE_TYPE‘ speichern:

- DEVICE_TYPE.dat – CE-Image
- Default.* – CoDeSys Dateien
- Eeprom.bin
- ElaDesignCE.exe, ElaDesignCoDeSysCE.exe – ElaDesign-Runtime
- ElaLibCE.dll
- eventlog.bin
- persist.bin
- resource.res - Ressource
- keypad*.xml – Tastatur Konfigurationen

VOLUME_NAME – Verzeichnisname der Compact Flash Karte

DEVICE_TYPE – Die erste Zeile aus der Datei: \FlashDisk\SysExtras\System\Device.inf.

UPDATE_TYPE – ElaDesign oder Image

FwGetFreeMemoryspaceOfCF (Ab Version 3)



ST

```
(* FwGetFreeMemoryspaceOfCF gets the free kBytes of an CF, it returns 0 if CF is not plugged *)
(* CAUTION : this function read the cluster of the CF, is not allowed while writing and take his time *)
(* CAUTION : call this function only from lowest priority task 'ob0' *)
FUNCTION FwGetFreeMemoryspaceOfCF : DWORD      (* return kBytes of free CF space *)
VAR_INPUT
    nMode :WORD;                               (* dummy, without functionality *)
END_VAR
VAR
END_VAR
```

Diese Funktion gibt den freien Speicherbereich (in kByte) einer vorhandenen Compact Flash Karte zurück.

FwGetFreeMem (Ab Version 14 / Nur bei Windows CE Geräten)



ST

```
FUNCTION FwGetFreeMem : DWORD
VAR_INPUT
    bDummy :BOOL;                               (* unused *)
END_VAR
VAR
END_VAR
```

Diese Funktion gibt die Größe des freien Speichers in KByte zurück bzw. 16#FFFFFFFF falls sie fehlschlägt.

ST - Beispiel:

```
(* Variablendeklaration *)
VAR
    dwFreeMem : DWORD;
END_VAR

(* Code *)
dwFreeMem := FwGetFreeMem(TRUE);
```

FwGetFreeDiskSpace (Ab Version 14 / Nur bei Windows CE Geräten)



ST

```
FUNCTION FwGetFreeDiskSpace : DWORD
VAR_INPUT
    sVolume :STRING;                             (* Volume name e.g. 'flashdisk' *)
END_VAR
VAR
END_VAR
```

Diese Funktion gibt die Größe des freien Speichers auf dem Laufwerk sVolume in KByte zurück bzw. 16#FFFFFFFF falls sie fehlschlägt.

ST - Beispiel:

```
(* Variablendeklaration *)
VAR
    dwFreeDiskSpace : DWORD;
END_VAR

(* Code *)
dwFreeDiskSpace := FwGetDiskSpace('flashdisk');
IF dwFreeDiskSpace = 16#FFFFFFFF THEN
    (* Something went wrong *);
END_IF
```

FwTurnDisplayOn (Ab Version 4)



ST

```
(* FwTurnDisplayOn turns the backlight on or off *)
FUNCTION FwTurnDisplayOn : BOOL
    VAR_INPUT
        bOn : BOOL; (* bOn: TRUE = on, FALSE = off *)
    END_VAR
    VAR
    END_VAR
END_VAR
```

Diese Funktion schaltet die Hintergrundbeleuchtung ein oder aus.

FwGetCRC16



ST

```
FUNCTION FwTurnDisplayOn : BOOL
    VAR_INPUT
        pData : POINTER TO BYTE; (* pointer to check sum datas *)
        nCount : WORD; (* lenth of datas in bytes *)
    END_VAR
    VAR
    END_VAR
END_VAR
```

FwGetCOMINStructPointer (Ab Version 6)



ST

FUNCTION FwGetCOMINStructPointer: DWORD

VAR_INPUT

nCardIndex : WORD;END_VAR

VAR

END_VAR

FwGetCOMOUTStructPointer (Ab Version 6)



ST

FUNCTION FwGetCOMOUTStructPointer: DWORD

VAR_INPUT

nCardIndex : WORD;END_VAR END_VAR

VAR

END_VAR

Technische Unterstützung

Hotline

Zusätzliche Hilfe und Informationen bietet Ihnen unsere Hotline in den folgenden Zeiten :

Mo - Do : von 8³⁰- 12⁰⁰ und 13⁰⁰ bis 16³⁰ Uhr
Fr : von 8³⁰- 12⁰⁰ Uhr

Außerhalb dieser Zeiten erreichen Sie uns jederzeit per Email oder Fax.

Telefon: ++49 (0) 7021 / 92025-33
Telefax: ++49 (0) 7021 / 92025-29
Email: hotline@elrest.de

Schulung und Workshop

Gerne bieten wir Ihnen Schulungen oder projektbezogene Workshops zu allen Elrest Produkten an.

Für weitere Informationen wenden Sie sich bitte an unsere Vertriebsabteilung.

Telefon: ++49 (0) 7021 / 92025-0
Telefax: ++49 (0) 7021 / 92025-29
Email: vertrieb@elrest.de

© 2010 elrest Automationssysteme GmbH. Alle Rechte vorbehalten.

Die in diesem Dokument enthaltenen Informationen können ohne Vorankündigung geändert werden und stellen keine Verpflichtung seitens elrest Automationssysteme GmbH dar.

Die Software und/oder Datenbanken, die in diesem Dokument beschrieben sind, werden unter einer Lizenzvereinbarung und einer Geheimhaltungsvereinbarung zur Verfügung gestellt. Die Software und/oder Datenbanken dürfen nur nach Maßgabe der Bedingungen der Vereinbarung benutzt oder kopiert werden.

Es ist rechtswidrig, die Software auf ein anderes Medium zu kopieren, soweit das nicht ausdrücklich in der Lizenz- oder Geheimhaltungsvereinbarung erlaubt wird. Ohne ausdrückliche schriftliche Erlaubnis der elrest Automationssysteme GmbH dürfen weder dieses Handbuch noch Teile davon für irgendwelche Zwecke in irgendeiner Form mit irgendwelchen Mitteln, elektronisch oder mechanisch, mittels Fotokopie oder Aufzeichnung reproduziert oder übertragen werden.

Abbildungen und Beschreibungen sowie Abmessungen und technische Daten entsprechen den Gegebenheiten oder Absichten zum Zeitpunkt des Druckes dieses Prospektes. Änderungen jeder Art, insbesondere soweit sie sich aus technischem Fortschritt, wirtschaftlicher Ausführung oder ähnlichem ergeben, bleiben vorbehalten. Die externe Verschaltung der Geräte erfolgt in Eigenverantwortung.