

Plattform μ E-Geräte

eStudio Version 2.91

The screenshot displays the eStudio software interface, which is used for developing and simulating microcontroller-based systems. The interface is divided into several main sections:

- Project Management (Left):** A tree view showing the project structure, including folders for 'Beispiel Ordner', 'AWL_EXAMPLE (PRG)', 'KOP_EXAMPLE (PRG)', and 'ST_EXAMPLE (PRG)'. A table below the tree lists 'Name', 'Datenpunkte', 'Programme', and 'Ressourcen'.
- Code Editor (Right):** A window titled 'CodeSys - Bsp01f.prj*' showing the source code for a function block 'FB0_EXAMPLE (FUN-FUP)'. The code includes variable declarations and logic for handling inputs and outputs. Below the code, there is a ladder logic diagram for 'KOP_EXAMPLE (PRG-KOP)' and another code window for 'AWL_EXAMPLE (FB-AWL)' containing assembly-like instructions such as 'LD r1', 'SIN', 'MUL 1000.0', 'ST sinus', 'LD r1', 'COS', and 'MUL 1000.0'.
- Simulation (Bottom):** A simulated HMI panel for a control system. It features a temperature display showing '22.5°C' with a target of 'Soll: 25°C', a date display '12.01.2012', and several control buttons for 'Musik', 'Licht', 'Jalousie', and 'Heizung'. There are also indicators for system status and a lock icon.

elrest Automationssysteme GmbH
Leibnizstraße 10
73230 Kirchheim unter Teck
Germany
Telefon: +49 (0) 7021 / 92025-0
www.elrest.de

elrest®

Inhaltsverzeichnis

Inhaltsverzeichnis	2
1 Umgang mit den μ E-Geräten	5
1.1 Was ist μ E	5
1.2 <i>BETRIEBS</i> - Variationen	5
1.3 GeräteEinstellungen	6
1.4 Bootvorgang	6
1.5 Update der Geräte	7
1.5.1 Update der Firmware	7
1.5.2 Update der Runtime über CF	10
1.6 Backup aus dem CompactFlash	13
1.6.1 Update der Runtime aus CoDeSys	14
1.6.2 Update Visioweb	14
1.7 Verwendung des geräteinternen EEprom's	15
1.8 Fernwartung mit Telnet	17
1.8.1 Allgemeines zu Telnet	17
1.9 Telnet-Befehle	19
1.9.1 Telnet Befehl : help	19
1.9.2 Telnet Befehl : state	20
1.9.3 Telnet Befehl : date	20
1.9.4 Telnet Befehl : option	21
1.9.5 Telnet Befehl : save	27
1.9.6 Telnet Befehl : load	28
1.9.7 Telnet Befehl : rs232	29
Telnet Befehl : can	30
1.9.8 Telnet Befehl : setdef	34
1.9.9 Telnet Befehl : password	34
1.9.10 Telnet Befehl : setmod	35
1.9.11 Telnet Befehl : setip	35
1.9.12 Telnet Befehl : subnet	36
1.9.13 Telnet Befehl : gateway	37
1.9.14 Telnet Befehl : setdhcp	38
1.9.15 Telnet Befehl : cf	38
1.9.16 Telnet Befehl : threads	39
1.9.17 Telnet Befehl : res	40

1.9.18	Telnet Befehl : xtest.....	40
1.9.19	Telnet Befehl : hwstate	40
1.9.20	Telnet Befehl : setCE.....	40
1.9.21	Telnet Befehl : clearMT	40
1.9.22	Telnet Befehl : linkstats	40
1.9.23	Telnet Befehl : sockets	41
1.9.24	Telnet Befehl : memstats	41
1.9.25	Telnet Befehl : vfssync	42
1.9.26	Telnet Befehl : X – Befehle (Analogkonfiguration P303)	42
1.9.27	Telnet Befehl : profibus.....	46
1.10	Anwendungen	47
1.11	Datenaustausch	48
1.11.1	Autoscan-Verbindung vorbereiten	48
1.11.2	Eine ftp-Verbindung herstellen	48
1.11.3	Eine Telnet-Verbindung herstellen.....	48
1.12	Betriebssystem Einstellungen	49
1.12.1	Speicheraufteilung.....	49
1.12.2	Kalibrierung des Touches	49
1.12.3	Hinweise zum FileSystem	49
1.13	Preemptives Multitasking.....	51
1.13.1	Task Konfiguration	51
1.13.2	Empfohlene Task Konfiguration.....	54
1.13.3	Task Konfiguration im ungestörten Betrieb	56
1.13.4	Task Konfiguration mit Echtzeitfehlern	56
1.13.5	Unterbrechung von Task's.....	58
2	Kommunikation.....	58
2.1	RS232	58
2.1.1	Low-Level RS232 oder RS485	58
2.1.2	RTU-Modbus über RS232 oder RS485	58
2.2	CAN	58
2.2.1	Low-Level CAN	58
2.2.2	ElaCAN	59
2.2.3	ESB.....	59
2.2.4	CANopen von 3S.....	60
2.2.5	Änderung unter 3S Steuerungskonfiguration	60

2.2.6	Erweiterter Retainspeicher 512kb.....	62
3	Applikationen.....	63
4	Benchmark.....	65
4.1	Threading.....	65
5	Support.....	69
6	Historie.....	70

1 Umgang mit den μ E-Geräten

Folgende Schnittstellen werden bei μ E-Geräten von elrest unterstützt:



Serielle Schnittstellen RS232 und/oder RS485. Diese können mit beliebigen UART Protokollen oder dem vorbereiteten Protokoll RTU-Modbus betrieben werden.

CAN

Offenes Feldbusinterface zu beliebigen CAN Protokollen, wie beispielsweise Truck-Norm J1939-based.

CANopen

CANopen ist ein verbreitetes Layer7 Protokoll für die Automatisierung.

ESB

Elrest Systembus, ein auf CAN basiertes selbstkonfigurierender Systembus.

Ethernet

Unter Ethernet kann TCP-Modbus als UDP oder TCP verwendet werden. Weiter können auf Basis von Socket Funktionen weitere Protokolle ergänzt werden.

1.1 Was ist μ E

μ E ist ein eigenes, von Elrest geschriebenes Betriebssystem für CPU167- Prozessoren.

Es unterstützt folgende Komponenten:

- Plattformen- ElaDesign mit CoDeSys V2.3
- Kommunikationswege:
 - Ethernet, TCP- und UDP Modbus
 - RS232, RS485 mit RTU-Modbus
 - Can,ESB (ElrestSystemBus), CanOpen
 - Profibus

1.2 BETRIEBS- Variationen

Die Geräte lassen sich mit den 3 folgenden Kombinationen betreiben:

Eladesign

z.B. als weiteres Display zu einer vorhandenen PLC. Die Kommunikation erfolgt hierbei z.B. automatisch über TCP-Modbus

ElaDesign mit CoDeSys

Visualisierung, Datenpunkte, Rezepte und Kommunikationswege über ElaDesignAblaufsteuerung über CoDeSys

CoDeSys

Ablaufsteuerung über CoDeSys, keine HMI oder Bildschirmausgabe über CoDeSys –Draw-Befehle.

Am Gängigsten ist hierbei Variante Nr2: ElaDesign mit CoDeSys

1.3 GeräteEinstellungen

Grundlegende Geräteeinstellungen können je nach Gerätetyp über einen oder allen nachfolgende Möglichkeiten vorgenommen werden.

1.) Servicemenü (Geräte mit Display)

Nach betätigen des Servicetaster von 2 Sekunden oder Schalterstellung „Servicemode“ erhalten sie ein Servicemenü im Display. Dies kann nach Geräteart stark variieren. Übliche Einstellmöglichkeiten sind:

- IP-/Subnet-/ und Gatewayadresse
- Baudrate der Seriellen Schnittstelle
- Baudrate und Knotennummer des CAN
- Löschen der Resource, des IEC_Code (CoDeSys Programm), Persist.bin (Daten)
- Kalibrieren des Touch
- Helligkeit/Kontrast

2.) Telnet (Geräte mit Ethernetanschluss)

Wie unter 1.) nur deutlich umfangreicher.

Siehe hierzu nachfolgendes Kapitel „Telneteinstellungen“. Typische Defaultadresse: IP 192.168.1.254

3.) Serielle Schnittstelle

Hierzu wird durch aktivieren des Servicemode 1.) die serielle Schnittstelle in den Debug-Mode geschaltet. Diese erlaubt z.B. mit Hyperterminal über RS232 direkt mit dem Betriebssystem zu kommunizieren. Die Befehle sind die gleichen wie bei Telnet.

Typische Defaulteinstellungen der RS:

Bits pro Sekunde:

Datenbits:

Parität:

Stoppbits:

Flusssteuerung:

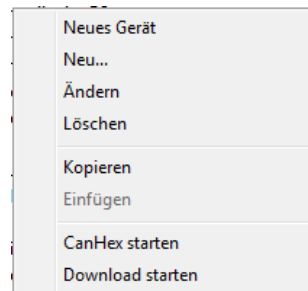
1.4 Bootvorgang

Nach dem Anlegen der Versorgungsspannung erscheint während der Boot- & Selbsttestphase für ca. 2 Sekunden kein Bild bzw. ein Startbildschirm.

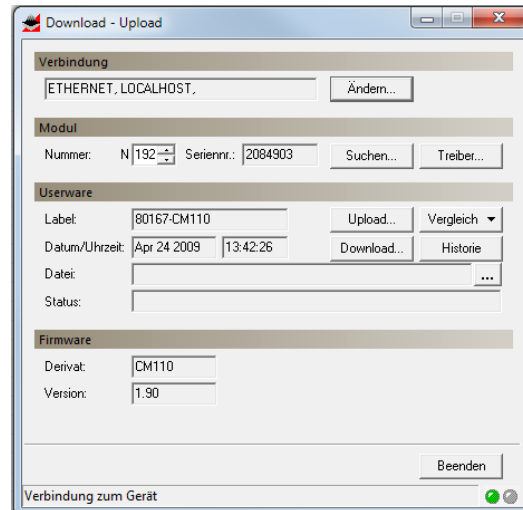
1.5 Update der Geräte

1.5.1 Update der Firmware

Aus der eStudio Projektverwaltung kann der Download gestartet werden.

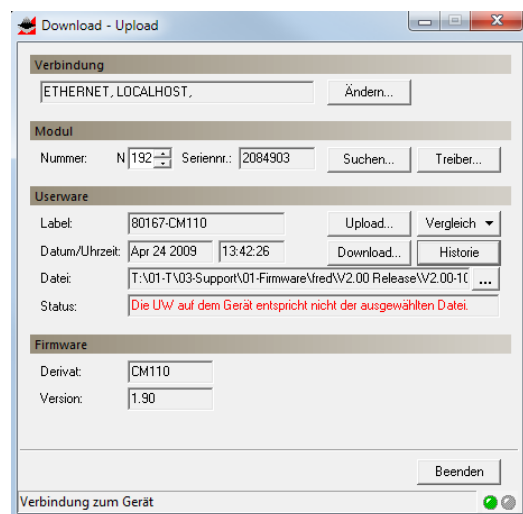


Hierzu wird das Download Tool gestartet werden:

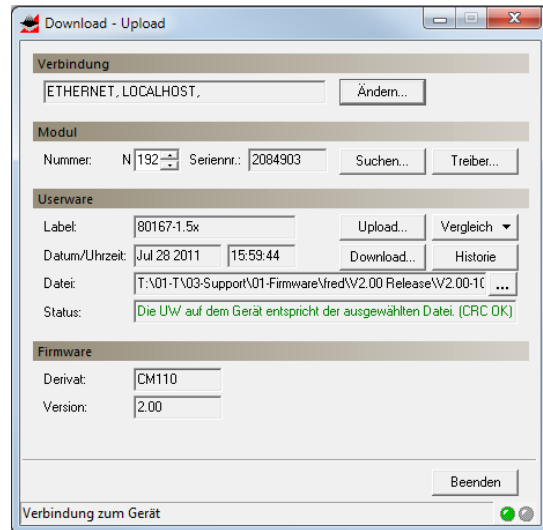


Mit dem Knopf „...“ kann die gewünschte Firmwareversion ausgewählt werden.

Mit dem Knopf „Download“ wird der Vorgang gestartet. Der Hinweis besagt, dass die aktuelle UW=Firm- und Userware auf dem Gerät nicht mit der neu Ausgewählten übereinstimmt.



Nach einem erfolgreichen Download erscheint:

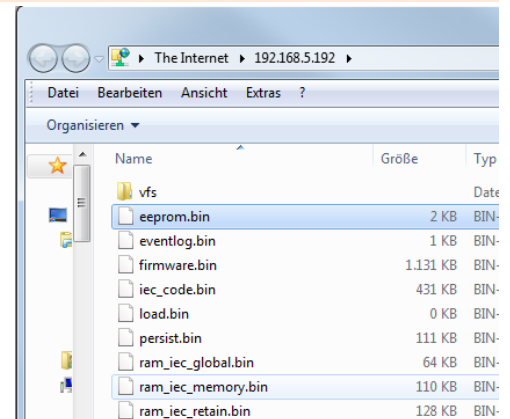


Damit der Betrieb nach dem Firmware Download wieder gegeben ist, müssen zuvor die Daten auf dem Gerät gesichert werden!

Folgende Dateien müssen mit ftp:

gesichert werden:

- **eeprom.bin**
darf nicht gesichert werden, siehe Kapitel eeprom.
- **eventlog.bin**
Hier werden alle gespeicherten Alarm- und Ereignismeldungen abgelegt. Falls diese für Sie wichtig sind muss diese Datei gesichert werden.
- **firmware.bin**
Diese Datei wird durch den Firmwaredownload automatisch upgedatet.
- **iec_code.bin**
Falls Sie die CoDeSys Applikation kopieren möchten, müssen Sie diese Datei zusammen mit ram_iec_global.bin, ram_iec_memory.bin und ram_iec_retain.bin sichern.
- **persit.bin** und ggf **persit1.bin** (ab FW 2.00)
In dieser Datei werden alle MERKER (%M) Datenpunkte unabhängig ob diese als CONSTANT, RETAIN oder PERSIT projiziert wurden gespeichert.
Sichern Sie diese Datei und löschen Sie anschließend diese Datei auf dem Gerät.
Von CoDeSys auf kann das Speichern mit folgenden Programmzeilen einmalig ausgeführt werden:
`psUW := UwGetStructPointer(FALSE);`
`M5_PERSIST := M5_PERSIST + 1;`
`psUW^.nSaveParameter := 1;`
Damit die Speicherung in jedem Durchlauf ausgeführt wird, muss mindestens ein als PERSIST konfigurierter Datenpunkt geändert werden.



- **resource.bin**
Diese Datei ist die durch Resgen erzeugte Kompilation der Visualisierungselemente.

1.5.2 Update der Runtime über CF

Diese Funktionalität ist bei folgenden Geräten verfügbar:

- IT1.2 ab FW 1.55-7
- P305 ab FW 1.55-7
- Combo ab FW 1.60



1.5.2.1 Ablauf

Im Telnet muss unter Option die CompactFlash „Enable“ aktiviert sein“. Die CompactFlash ist jetzt betriebsbereit.

Wenn dieser Parameter gesetzt ist erscheint ein zweiter Parameter: Enable CF-FW. Diese Einstellung muss für die automatische Update- Backup Funktion auch auf Enable gesetzt werden.

- Im laufenden Betrieb kann die CF ein- und ausgesteckt werden.
- Die Daten der CF erscheinen im Unterverzeichnis „cf0“, falls mit einem Explorer auf das Gerät zugegriffen wird. z.B. FTP-Explorer, Telnet.

```
Telnet 192.168.1.240
$fred/>oo
Options : 0.no Option 1.DI08/I01 2.RS232/RS [COM-2k] 4.COM-8k
Clear Eventlogger : 6...clear eventlog SRAM
TCP-prio higher as OB:1 : 9...I0.1.21 (higher) [3.4.5] (equal) -> (<)
Enable CompactFlash : 10...set value I0.11 ->'enable'
Start delay UDP+CodeSys: 14...I0...91 (<)
Modbus0-Off_1=TCP_2=UDP: 20 -> 2
Exist persist.bin file : 21...set value I0.11 ->'yes'
Open empty Mask first : 22...set value I0.11 ->'yes'
Count of adu sampling : 23...set value I1.1001 -> 1
Resource (resource.bin) -> 00240000 001C0000 (1792kB flash)
IEC code (iec_code.bin) -> 00120000 000E0000 ( 896kB flash)
Persist Memory (persist.bin) -> 00000000 000E0000 ( 544kB flash)
CodeSys - Code (iec_code.bin) -> 00920000 000E0000 ( 896kB sram)
CodeSys - Global (ram_iec_global.bin) -> 008C0000 00060000 ( 384kB sram)
CodeSys - Memory (ram_iec_memory.bin) -> 00824000 00008000 ( 41.5 sram)
CodeSys - Input -> 00c00000 00002100 ( 8kB sram)
CodeSys - Output -> 00822100 00002100 ( 8kB sram)
CodeSys - Retain (ram_iec_retain.bin) -> 00600000 000D0000 ( 832kB sram)
$fred/>_
```

```
$combo/>oo
Options : [no Option]
Clear Eventlogger : 6...clear eventlog SRAM
Key repeatabili : 7...set value I0.11 ->'on'
Key beeper (0-off) : 8...set value I0...2551 -> (<2)
TCP-prio higher as OB:0 : 9...I0.4.21 (higher) [3.4.5] (equal) -> (<0)
Enable CompactFlash : 10...set value I0.11 ->'enable'
Trace enable : 13...I<0>.11 (<0)
Start delay UDP+CodeSys: 14...I0...91 (<0)
Modbus0-Off_1=TCP_2=UDP: 20 -> 2
Modbus Port Read (502): 21 -> 502 (client+server)
Modbus Port Write (502): 22 -> 502 (client)
Open empty Mask first : 25...set value I1.1001 ->'on'
Count of adu sampling : 27...set value I0.11 ->'disable'
Resource (resource.bin) -> 00210000 00170000 (1422kB flash)
RAMdisk aktiv : 22...set value I0.11 ->'off'
ESB slave aktiv : 32...set value I0.11 ->'off'
IEC code (iec_code.bin) -> 00380000 00080000 ( 512kB flash)
Persist Memory (persit.bin) -> 00180000 00020000 ( 128kB flash)
$combo/>
```

1.5.2.2 Über Telnet

Telnet Befehl „cf“ :

Dieses Menü erscheint nur wenn die CF-Karte gesteckt ist und der Befehl CF ausgeführt wird.

```
$fred/>ccff
>CF: - dump: arguments <sector> <count> or <command> with command below zero
> command -1: automatic FW update
> command -2: quick format CF
> command -3: erase CF completely
> - 62720 clusters, 4 sec/clus, 128459560 bytes
> - FAT: 0: from 21, 245 sectors, 1: from 116, 245 sectors,
> - root dir from 20B, 32 sectors, first data sector: 22B
>DOS: 0 open files
$fred/>
```

Telnet Befehl cf -1 :

Es wird geprüft, ob bereits ein Unterverzeichnis auf dem CompactFlash existiert. :

<ftp://<IP-adresse>/cf0/CONTROL/<Gerätederivat>/backup/>

Wenn kein Verzeichnis vorhanden ist, wird das Verzeichnis mit folgende Dateien angelegt. Vorausgesetzt die jeweilige Datei hat Inhalt. Sonst wird die Datei nicht angelegt.

- [firmware.bin](#)
- [iec_code.bin](#)
- [persist.bin](#)
- [ram_iec_global.bin](#)
- [ram_iec_memory.bin](#)

ram_iec_retain.bin
resource.bin
eeeprom.bin

Falls ein erneutes Backup gemacht wird, muss der Ordner umbenannt werden. Sonst ist ein erneutes Backup nicht möglich.

Telnet Befehl `cf -2` :

Die CF Karte wird formatiert. Der Backup Ordner ist somit gelöscht.

<ftp://<IP-adresse>/cf0/CONTROL/<Gerätederivat>/backup/>



Die CF-Karte darf nur an der Control Steuerung gelöscht werden z.B. über Ftp. Beim Löschen an einem PC, muss die CF Karte neu mit FAT16 formatiert werden.

Telnet Befehl `cf -3` :

Die CF Karte wird gelöscht. Dieser Menüpunkt ist ausschließlich für den Supervisor und somit passwortgeschützt.

ab Version 2.00.08 ohne Passwortschutz

Telnet Befehl `cf -4` :

Mit diesem Befehl kann der Inhalt der CompactFlash angezeigt werden.

```
$combo/>cf -4
>Checking clusters: cluster 0 at sector 022E
combo      (COMBO-1  ) scan dir.  230,  32 cls.
.          (.      )
..         (..     )
CM110     (CM110  ) scan dir.  250,  32 cls.
.          (.      )
..         (..     )
UPDATE    (UPDATE ) scan dir.  270,  32 cls.
.          (.      )
..         (..     )
firmware.bin (FIRMWAREBIN) file 290, 2305 cls.
iec_code.bin (IEC_CODEBIN) file  B91,  24 cls.
ram_iec_global.bin (RAM_IEC_BIN) file  E49, 128 cls.
ram_iec_memory.bin (RAM_IE-1BIN) file  C29, 220 cls.
ram_iec_retain.bin (RAM_IE-2BIN) file  D05, 256 cls.
eeeprom.bin (EEPROM BIN) file  E05,   4 cls.
resource.bin (RESOURCEBIN) file  E09,  57 cls.

files: 7
BACKUP      (BACKUP  ) scan dir.  E42,  32 cls.
.          (.      )
..         (..     )
firmware.bin (FIRMWAREBIN) file  E62, 2305 cls.
iec_code.bin (IEC_CODEBIN) file 1763,  24 cls.
ram_iec_global.bin (RAM_IEC_BIN) file 177B, 128 cls.
ram_iec_memory.bin (RAM_IE-1BIN) file 17FB, 220 cls.
ram_iec_retain.bin (RAM_IE-2BIN) file 18D7, 256 cls.
eeeprom.bin (EEPROM BIN) file 18D7,   4 cls.
resource.bin (RESOURCEBIN) file 19DB,  57 cls.

files: 7

$combo/>
```

1.5.2.3 Update aus dem CompactFlash

Die Steuerungen „IT1.2“, „P305 und combo“ befindet sich in der **Serviceschalterstellung „RUN“**. Ist die CF Karte gesteckt, leuchtet die LED „CF“ (LED11). Ist die CF Karte nicht gesteckt, blinkt die LED „CF“ (LED11) innerhalb der ersten 5 Sekunden und ist anschließend aus.

Die IT1.2,P305 sowie combo wird neu gestartet. durch einen „Reset“ oder eine Spannungsunterbrechung.

Nach dem Reset ist im Hyperterminal folgende Ausgabe zu sehen (Siehe Bild).

Jetzt muss innerhalb von 120 Sekunden aber frühestens nach 20 Sekunden :

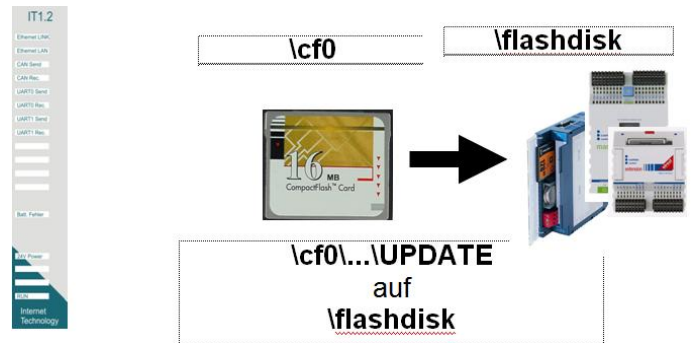
1. die CompactFlash gesteckt werden.
2. während dem BACKUP in das Verzeichnis **ftp://<IP-
adresse>/cf0/CONTROL/<Geräterderivat>/
BACKUP/** bzw. UPDATE aus dem Verzeichnis **ftp://<IP-
adresse>/cf0/CONTROL/<Geräterderivat>/
UPDATE/**auf die interne Flashdisk blinkt die LED „CF“ (LED11).
3. Anschließend wird ein Neustart durchgeführt.
4. Der komplette Vorgang ist abgeschlossen, wenn die LED „CF“ (LED11) permanent leuchtet.

Wichtig:

Ein Update wird nur dann gestartet, wenn ein Update Verzeichnis existiert.

**ftp://<IP-
adresse>/cf0/CONTROL/<Geräterderivat>/UPDAT
E/**

Folgende Dateien werden nun aus dem Update angelegt.



```
>dosFwUpdate: plug CF in at least 120s
>CF: "TOSHIBA THNCF128MCA" 122MB/2KB Cache
>dosFwUpdate: backup/update from "\cf0\.\UPDATE" in progress...
>FW: current 089A5793 (Jan 11 2006 12:27:19)
>FW: saving "/cf0/FRED/IT12/BACKUP/firmware.bin"
>UW: saving "/cf0/FRED/IT12/BACKUP/iecode.bin"
>UW: saving "/cf0/FRED/IT12/BACKUP/ram_iec_global.bin"
>UW: saving "/cf0/FRED/IT12/BACKUP/ram_iec_memory.bin"
>UW: saving "/cf0/FRED/IT12/BACKUP/ram_iec_retain.bin"
>UW: saving "/cf0/FRED/IT12/BACKUP/persist.bin"
>UW: saving "/cf0/FRED/IT12/BACKUP/eprom.bin"
>UW: saving "/cf0/FRED/IT12/BACKUP/resource.bin"
>FW/UW saving successful

>restoring from "/cf0/FRED/IT12/UPDATE/iecode.bin"->Stop Target ...
iecode.bin: opened, writing...
>restoring from "/cf0/FRED/IT12/UPDATE/ram_iec_global.bin"->Stop Target ...
ram_iec_global.bin: opened, writing...
>restoring from "/cf0/FRED/IT12/UPDATE/ram_iec_memory.bin"->Stop Target ...
ram_iec_memory.bin: opened, writing...
>restoring from "/cf0/FRED/IT12/UPDATE/ram_iec_retain.bin"->Stop Target ...
ram_iec_retain.bin: opened, writing...
>restoring from "/cf0/FRED/IT12/UPDATE/eprom.bin"->Stop Target ...
eprom.bin: opened, writing...
>restoring from "/cf0/FRED/IT12/UPDATE/resource.bin"->Stop Target ...
resource.bin: opened, writing...
>FW/UW restoring successful

>restoring from "/cf0/FRED/IT12/UPDATE/firmware.bin"->Stop Target ...
firmware.bin: opened, writing...
Load target image (IT12 CODESYS on IT12 CODESYS)
> > K E E P   P O W E R   O N < < <
Length 1043409, Check...
save to FLASH >KEEP POWER<
FlashStoreFirmware
```

firmware.bin
iec_code.bin
persist.bin
ram_iec_global.bin
ram_iec_memory.bin
ram_iec_retain.bin
resource.bin
eprom.bin

Gleichzeitig wird ein Backup-Ordner angelegt in dem die vorherigen Dateien stehen.

[ftp://<IP-
adresse>/cf0/CONTROL/<Gerätederivat>/backup
/](#)

firmware.bin
iec_code.bin
persist.bin
ram_iec_global.bin
ram_iec_memory.bin
ram_iec_retain.bin
resource.bin
eprom.bin

1.6 Backup aus dem CompactFlash

Die Steuerungen „IT1.2“, „P305“ und combo befindet sich in **Serviceschalterstellung „STOP“**.

Ist die CF Karte gesteckt, leuchtet die LED „CF“ (LED11).

Ist die CF Karte nicht gesteckt, blinkt die LED „CF“ (LED11) innerhalb den ersten 5 Sekunden und ist anschließend aus.

Jetzt muss innerhalb von 120 Sekunden aber frühestens nach 20 Sekunden :

die CompactFlash gesteckt werden.

während dem UPDATE aus dem Verzeichnis **ftp://<IP-**

adresse>/cf0/CONTROL/<Gerätederivat>/BACKUP/auf die interne Flashdisk blinkt die LED „CF“ (LED11).

Anschließend wird ein Neustart durchgeführt.

Der komplette Vorgang ist abgeschlossen, wenn die LED „CF“ (LED11) permanent leuchtet.

Anschließend muss der Serviceschalter von STOP nach RUN betätigt werden.

Schreib- Lesezugriff über CoDeSys

Um über CoDeSys auf die CF- Karte zugreifen zu können müssen Sie die Bibliothek

- SysLibFile.lib

einbinden. Dies Bibliothek umfasst im aktuellen Stand folgende Funktionen.

- SysFileOpen();
- SysFileClose();
- SysFileRead();
- SysFileWrite();

Alle übrigen Funktionen sind zum Stand 1.6.2006 noch nicht in die Firmwaretargets der jeweiligen Geräte implementiert und somit ohne Funktion.

1.6.1 Update der Runtime aus CoDeSys

Vorbereitung siehe ‚Update beim Windows-Start‘

Um aus CoDeSys heraus ein Update der Runtime durchzuführen verwenden Sie die Funktion FwCFRestore mit dem Parameter 1.

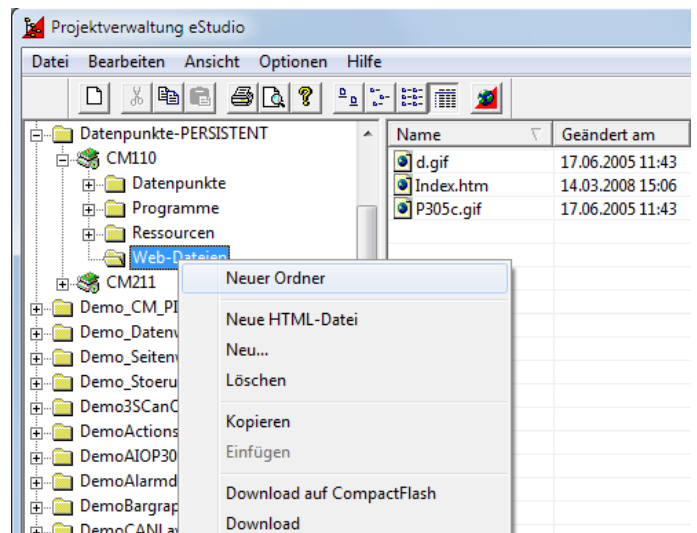
```
FwCFRestore(1) oder FwCFRestore(RUNTIME)
```

Diese Funktion befindet sich in der Bibliothek sFW05.

1.6.2 Update Visioweb

Die Web Dateien sind im Ordner „Web-Dateien“ abgelegt.

Mit dem Menüpunkt „Download“ können die Dateien auf das Gerät übertragen werden.



Die entsprechenden Web-Dateien können Sie aus dem Projekt: „DemoWebVisualisierung“ mit Kopieren und Einfügen in Ihr aktuelles Projekt kopieren.

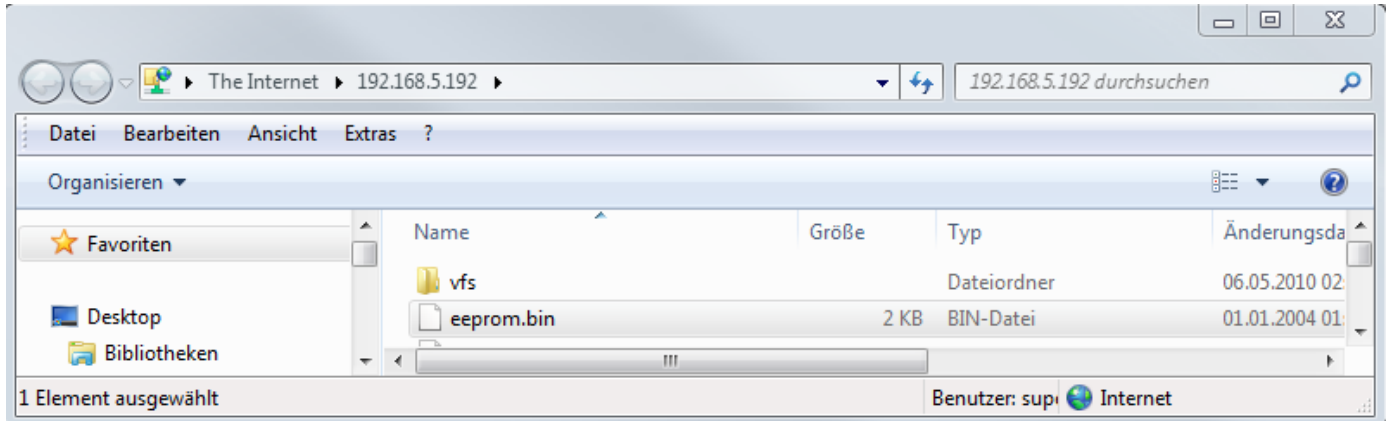
1.7 Verwendung des geräteinternen EEPROM's



Daten im EEPROM dürfen maximal 100.000 mal geschrieben werden. Sie können im Gegensatz zum Flash einzeln geschrieben werden.

Es dürfen niemals die eeprom.bin Dateien von Geräte getauscht werden! Darin gespeichert sind gerätespezifische Daten wie Seriennummern, IP-Nummer, u.v.m.

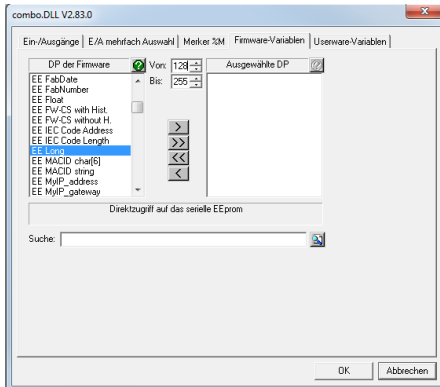
Jedes Gerät besitzt ein 2kByte großes EEPROM. Diese kann unter ftp angesehen werden:



Zu Sicherungszwecken, können Sie diese Datei für ein Gerät einer Seriennummer auf dem PC speichern. Von diesen 2kByte stehen für anwendungsspezifische Daten nur $128 * 4 \text{ Bytes} = 512 \text{ Bytes}$ (DWORD=long oder REAL=float) für die Kundenapplikation zur Verfügung.

Um auf diese Werte zugreifen zu können sind folgende Möglichkeiten gegeben:

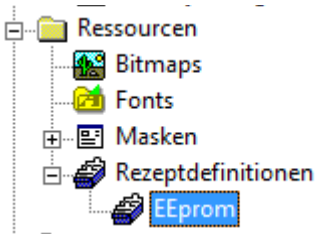
- Unter eStudio können bei den Gerätedatenpunkte ausgewählt werden:



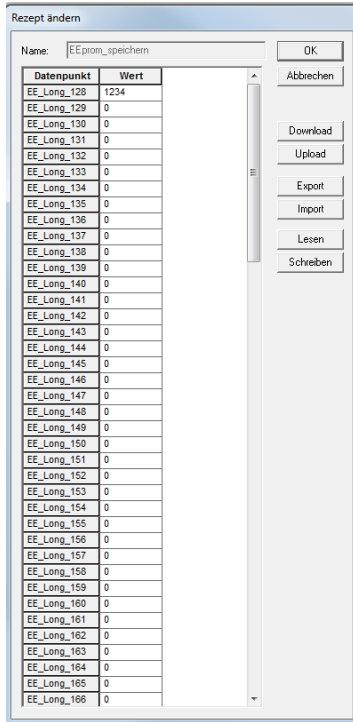
Maximal dürfen „EE long“ oder „EE Float“ im Indexbereich von 128 ... 255 verwendet werden.

- Innerhalb von CoDeSys kann über die Struktur sFW direkt zugegriffen werden `pEE_Long_128 := ADR(psFW^.uRemanent.sUserEEProm.aUserFreeSpace[256]);` Achtung die Adressen von aUserFreeSpace sind WORD, deshalb muss bei pEE_Long_128 die Adresse umgerechnet werden.

Wenn man diese EEPROM Daten ins Projekt kopieren möchte, kann man eine Rezept anlegen:



und alle projizierten EEprom Werte :



mit der Taste „LESEN“ aus dem Gerät kopieren, bzw mit „SCHREIBEN“ auf ein anderes Gerät übertragen.

1.8 Fernwartung mit Telnet

1.8.1 Allgemeines zu Telnet

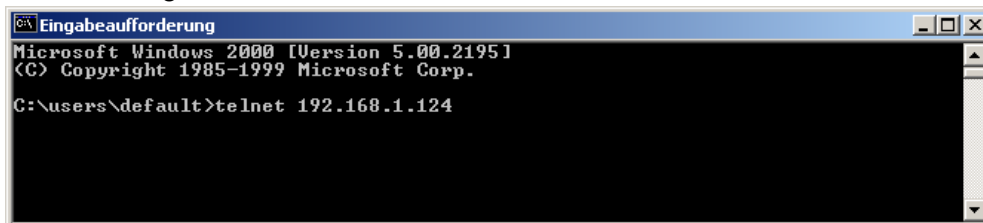
Telnet ist dazu gedacht, einen fernen Rechner im Internet so zu bedienen, als säße man direkt davor. Telnet ist damit eine einfache Lösung für Teleworker.

Telnet ist vor allem für Unix-Systeme gedacht. Es erlaubt das betriebssystemeigene **login** (Anmelden) eines Benutzers an einem ans Internet angeschlossenen Host-Rechner in Form eines **rlogin** (remote login). Das Anmelden ist nur möglich, wenn Sie den User-ID und das Passwort kennen, d.h. auf dem angewählten Host-Rechner als Benutzer eingetragen sind. Nach dem Einwählen erhalten Sie eine Unix-Shell (Eingabeaufforderung) und können auf dem entfernten Rechner Betriebssystembefehle eingeben, Programme starten usw.

Das Programm erlaubt es, vom eigenen PC Geräte aus der Serie visio oder combo zu konfigurieren. Für die Konfiguration dieser Geräte sind Kenntnisse über die Befehle notwendig, die im Folgenden beschrieben sind.

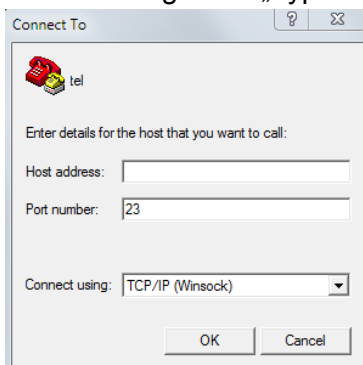
Sie können eine Telnet Verbindung aufbauen:

- Mit dem Programm „telnet.exe“.



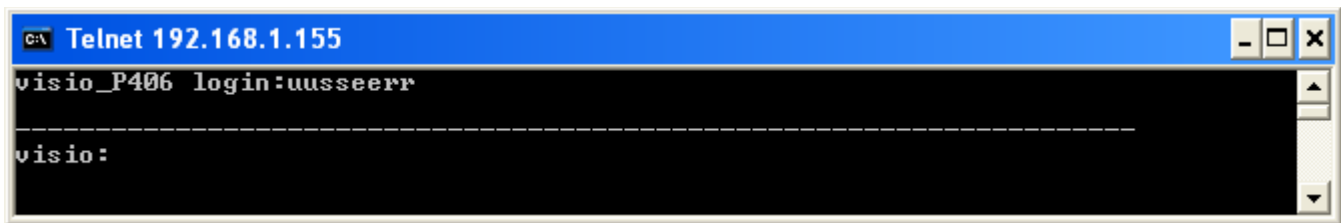
Hinweis : ist bei Win 7 nicht mehr im Standardlieferumfang.

- Mit dem Programm „hyperterminal.exe“. Dabei wählen Sie die Option:



- Oder falls Sie im Zielsystem die Option RS232-DEBUG aktiviert haben, können Sie über die serielle Schnittstelle mit jedem Terminprogramm (hyperterminal, Tera Term, o.ä.) gleichermaßen verwenden.

Außer bei der seriellen Verbindung erfolgt zunächst ein Login:



```
C:\ Telnet 192.168.1.155
visio_P406 login:uusseerr
-----
visio:
```

Bestätigen Sie dies durch Eingabe von “user” bzw. “supervisor”.

Mit dem Befehl : **help** kann man alle verfügbaren Befehle auflisten.



Tipp

Zur Übersichtlichkeit haben Kommandos ein Untermenü. Durch Eingabe des Kommandos erscheint das Untermenü oder bei einem Befehl die Information.

Alle Kommandos können durch deren Anfangsbuchstaben ausgeführt werden, solange die Zuordnung eindeutig ist.



Tipp

Programmierhinweis:

Viele Telneteinstellungen können in CoDeSys gelesen und geschrieben werden. D.h. die CoDeSys- Initialisierungsroutine kann so geschrieben werden, dass ihre Systemeinstellungen in CoDeSys vorgenommen werden können.

In Folge finden Sie Codebeispiele die sich jeweils auf das Lesen beziehen. Um den Wert schreiben zu können, tauschen Sie einfach die Zuweisung:

(*Lesern*) nDisplayBrightness :=psFW^.uRemanent.sMasterEEProm.nDisplayBrightness;

(*Schreiben*) psFW^.uRemanent.sMasterEEProm.nDisplayBrightness:= nDisplayBrightness;

WICHTIG:

Bevor sie erstmalig mit psFW (PointerStructureFirmWare) und psUW (PointerStructureUserWare) arbeiten können, muss der Pointer initialisiert werden.

Durch das Initialisieren wird CoDeSys der Beginn des Speicherbereiches der FW und UW bekannt gegeben.



WARNUNG bei Pointern:

Findet diese Zuweisung nicht, oder fehlerhaft statt, ist der Zugriff ins System willkürlich. Das Ändern des Passwortes kann für das Überschreiben des Betriebssystems verantwortlich sein.

Vergewissern Sie sich daher UNBEDINGT, dass alle Pointerinitialisierungen korrekt sind.

```

PLC_PRG (PRG-ST)
0001 PROGRAM PLC_PRG
0002 VAR
0003   psFW : POINTER TO sFW:=0;  (* Zeiger auf sie Startadresse der Firmwaredatenpunkte *)
0004   psUW : POINTER TO sUW:=0;  (* Zeiger auf sie Startadresse der Userwaredatenpunkte *)
0005 END_VAR
0006
0007
0008
0001 IF psFW = 0 OR psUW = 0 THEN
0002   psFW:= FwGetStructPointer(0 (*nDummy*));
0003   psUW:= UwGetStructPointer(0 (*nDummy*));
0004   RETURN;  (* Nachfolgendet Code darf nur ausgeführt werden, wenn für psFW/psUW eine Adresse zugewiesen wurde *)
0005 END_IF;
0006

```

Im Anschluss haben Sie Online Zugriff auf sämtliche Datenpunkte:

```

psFW = <00810000>
  psFW^
    pOfThisStruct = 8454144
    nSizeOfFWStruct = 43834
    nVersionOfFirmware = 131078
    szFirmwareLabel = 'CM100'
    szFirmwareDate = 'Sep 08 2010'
    szFirmwareTime = '17:35:57'
    psUW = 8514338
    ...

```



Alle Kommandos können durch deren Anfangsbuchstaben ausgeführt werden, solange die Zuordnung eindeutig ist. Es ist auf Groß- / Kleinschreibung zu achten.



Der Umfang sowie der Inhalt der Befehle variiert mit dem jeweiligen Gerät und der Softwareversion der Runtime. Eine eindeutige Beschreibung für Befehle ist deshalb nur im Gerätehandbuch ersichtlich.

1.9 Telnet-Befehle

1.9.1 Telnet Befehl : help

Weitere Hilfen finden Sie mit dem Befehl “help diagnostic”.

```

$visio_P205V/>help diagnostic
SNMP Station: diagnostic commands:
  logging - set logging options

```

Durch Eingabe der Kommandos kann folgende Funktion ausgeführt werden:

logging (Untermenü) : Funktionen zur Protokollierung

1.9.2 Telnet Befehl : state

Statusanzeige des Gerätes:

```
$visio_P205V/>state
>visio_P205V
>Runtime-Version      : 2.03-01 Aug 11 2010 18:57:11
>CE-Version          : 6.16 13.07.2010
>CoDeSys CAN
>Module 191=run, CAN:terminated 125kBaud
>192.168.5.191, subnet 255.255.255.0, gatew. 0.0.0.0
>MAC=00-07-7E-00-57-B7 fOK SN=-1737075662 Date 25.07.2001 01:07:48
```

Zeile 1: Gerätetyp

Zeile 2: Runtime- Version

Zeile 2: Softwaretyp

Zeile 3: Modulnummer, CAN Abschlußwiderstand aktiviert, CAN Übertragungsrate

Zeile 4: Ethernet IP-Adresse

Zeile 5: MAC-ID und Systemdatum

1.9.3 Telnet Befehl : date

Lesen und Ändern der SystemUhr / SystemDatum

Beispiel:

```
CM211/>date
>Date/Time: 2.1.2008 07:12
CM211/>date 15.12.2010 17:00
>Date/Time: 15.12.2010 17:00
```

1.9.4 Telnet Befehl : option

Ändern von gerätespezifische Optionen:

```
Clear Eventlogger      : 6...clear eventlog SRAM
Key repeatabili.      : 7...set value [<0>,1] ->"off"
TCP-prio higher as OB:2: 9...[0,1,<2>] (higher) [3,4,5] (equal) -> (2)
Enable CompactFlash   : 10...set value [0,<1>] ->"disable"
Touch used for edit    : 11...set value [0,<1>] ->"yes"
Trace enable          : 13...[<0>,1] (0)
Start delay UDP+CoDeSys: 14...[<0>...9] (0)
ShutOff Backlight     : 16...set value [<0>=off...32767s] -> (0)
ShutOff Brightness    : 17...[0..<30>..255] (30)
Open key pad with ENTER: 18...set value [<0>,1] ->"no"
Modbus0=Off,1=TCP,2=UDP: 20 -> 2
Modbus Port Read <502>: 21 -> 502 (client+server)
Modbus Port Write <502>: 22 -> 502 (client)
Open empty Mask first : 24...set value [<0>,1] ->"no"
Count of adu sampling : 25...set value [1..<16>..100] -> 16
Brightness Display    : 26...[0..<200>..255] (200)
Resource              (resource.bin) -> 00210000 00170000 (1472kB flash)
RAMdisk aktiv        : 31...set value [<0>,1] ->"off"
Edit Style           : 34...set value [<0>-standard,1-4keys,2-6keys] ->"2"
SRAM Waitstate       : 35...set value [<0>,1] ->"no"
Hour glass by open mask: 36...set value [0,<1>] ->"yes"
RTU: Use Coils for Bits: 37...set value [<0>,1] ->"no"
RTU Timeout          : 38...set value [ms] -> 200
RTU Frame Spacing    : 39...set value [ms] -> 10
IEC code             (iec_code.bin) -> 00380000 00080000 ( 512kB flash)
Persist Memory       (persit.bin) -> 00140000 00020000 ( 128kB flash)
Persist 2 Memory     (persit2.bin) -> 00160000 00020000 ( 128kB flash)
```

Option:

Die Optionen 1-4 stehen für die möglichen Erweiterungs-Module zu Verfügung („option 0-4“).

```

Telnet 192.168.5.214
$visio/>oo 22
Options : 0.no Option 1.DIO8/IO1 [RS232/RS] 3.COM-2k 4.COM-8k
Clear Eventlogger : 6...clear eventlog SRAM
Key repeatabili. : 7...set value [<0>,1] ->"off"
Key beeper (<0=off) : 8...set value [0..<2>..255] -> <2>
TCP-prio higher as OB:2 : 9...[0,1,<2>] <higher> [3,4,5] <equal> -> <2>
Enable CompactFlash : 10...set value [0,<1>] ->"enable"
Touch used for edit : 11...set value [0,<1>] ->"yes"
Touch beeper(<0=off) : 12...set value [0..<3>..255] -> <3>
Trace enable : 13...[<0>,1] <0>
Start delay UDP+CoDeSys: 14...[<0>..9] <0>
Contrast LCD : 15...[0..255] <200>
ShutOff Backlight : 16...set value [<0>=off...32767s] -> <0>
ShutOff Brightness : 17...[0..<30>..255] <30>
Open key pad with ENTER: 18...set value [<0>,1] ->"no"
Start calibration : 19
Modbus0=Off,1=TCP,2=UDP: 20 -> 2
Modbus Port Read <502>: 21 -> 502 <client+server>
Modbus Port Write <502>: 22 -> 502 <client>
Open empty Mask first : 24...set value [<0>,1] ->"no"
Count of adu sampling : 25...set value [1..<16>..100] -> 16
Brightness Display : 26...[0..<200>..255] <200>
Enable CF-FW : 27...set value [<0>,1] ->"enable"
Resource <resource.bin>: 30...set value -> 00210000 00170000 <1472kB flash>
Edit Style : 34...set value [<0>-standard,1-4keys,2-6keys] ->"0"
SRAM Waitstate : 35...set value [0,<1>] ->"yes"
Hour glass by open mask: 36...set value [0,<1>] ->"yes"
IEC code <iec_code.bin> -> 00380000 00080000 < 512kB flash>
Persist Memory <persit.bin> -> 00180000 00020000 < 128kB flash>
$visio/>_

```

- option 1. DIO8/IO1 : aktiviert ein Erweiterungs-Modul mit Ein- bzw. Ausgängen
- option 2. RS232/RS : aktiviert ein Erweiterungs-Modul mit RS232 bzw. RS485
- option 3. DPx/COS-2k : aktiviert COM-Module mit 2 kByte Speicher.
- option 4. DPx/COx-8k : aktiviert COM-Module mit 8 kByte Speicher.

- **Clear Eventlogger** („option 6“):

Löscht die im RAM Speicher abgelegten Einträge des Eventloggers.

- **Key repeatabili.** („option 7“):

Wenn diese Option aktiviert ist, werden die Tasten von Bedienpulten zyklisch für die Dauer der Betätigung ermittelt. Bei deaktivierter Option werden nur die Zustandsänderungen der Tasten ermittelt.

(*O 7 Byte*)

[bKeyboardRepetiermode:=psFW^.uRemanent.sMasterEEprom.bKeyboardRepetiermode;](#)

- **Key beeper** („option 8“):

Diese Option legt fest, ob und wie lange (Faktor 10 ms) die interne Hupe bei Betätigung eines Tasters eingeschaltet wird.

(*O 8 Byte*)

[nKeyboardBeeperTimer:=psFW^.uRemanent.sMasterEEprom.nKeyboardBeeperTimer;](#)

- **TCP-prio higher as OB:x : („option 9“):**

Festlegung der Priorität der Ethernetkommunikation bzgl. des Aufrufs von den Threads ob2, ob1 und ob0:

- (2) => - Die Ausführung des TCP-IP Threads erfolgt in einer höheren Priorität als ob2.
- Die Ethernetkommunikation kann die Echtzeitfähigkeit von ob2 beeinflussen.
- Somit ist das Debuggen von CoDeSys über Ethernet möglich.
- (1) => - Die Ausführung des TCP-IP Threads erfolgt in einer höheren Priorität als ob1.
- Somit ist eine harte Echtzeitfähigkeit von ob2 gewährleistet.
- Es ist kein Debuggen von CoDeSys über Ethernet möglich !
- (0) => - Die Ausführung des TCP-IP Threads erfolgt in einer höheren Priorität als ob0.
- Somit ist eine harte Echtzeitfähigkeit von ob2 und ob1 gewährleistet.
- Es ist kein Debuggen von CoDeSys über Ethernet möglich !

Festlegung der Priorität der Ethernetkommunikation bzgl. des Aufrufs von den Threads ob2, ob1 und ob0:

- (2) => - Die Ausführung des TCP-IP Threads erfolgt in einer höheren Priorität als ob2.
- Die Ethernetkommunikation kann die Echtzeitfähigkeit von ob2 beeinflussen.
- Somit ist das Debuggen von CoDeSys über Ethernet möglich.
- (1) => - Die Ausführung des TCP-IP Threads erfolgt in einer höheren Priorität als ob1.
- Somit ist eine harte Echtzeitfähigkeit von ob2 gewährleistet.
- Es ist kein Debuggen von CoDeSys über Ethernet möglich !
- (0) => - Die Ausführung des TCP-IP Threads erfolgt in einer höheren Priorität als ob0.
- Somit ist eine harte Echtzeitfähigkeit von ob2 und ob1 gewährleistet.
- Es ist kein Debuggen von CoDeSys über Ethernet möglich!

- **Enable CompactFlash („option 10“):**

Aktivierung der Compact Flash Karte. Diese Option kann nur geschaltet werden, wenn das Gerät über eine CF- Karte verfügt.

(*O 10 Byte*) `bEnableCF:=psFW^.uRemanent.sMasterEEProm.bEnableCF;`

- **Touch used for edit („option 11“):**

Wenn diese Option aktiviert ist, wird bei einem Touch-Ereignis auf ein Eingabefeld die Software-Tastatur für die Eingabe geöffnet.

(*O 11 Byte*)

`bEditWithTouchKeyPad:=psFW^.uRemanent.sMasterEEProm.bEditWithTouchKeyPad;`

- **Touch beeper („option 12“):**

Diese Option legt fest, ob und wie lange (Faktor 10 ms) die interne Hupe bei Betätigung des Touch eingeschaltet wird.

(*O 12 Byte*) `nTouchBeeperTimer:=psFW^.uRemanent.sMasterEEProm.nTouchBeeperTimer;`

- **Trace enable („option 13“):**

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.

- **Start delay UDP+CoDeSys** („option 14“):
Verzögerung für den Start der Kommunikation und CoDeSys.
Beispiel CoDeSys
(*O 14 Byte*) nUDPStartupDelay:=nDisplayShutOffBright:=psFW^.uRemanent.sMasterEEprom.nUDPStartupDelay;
- **Contrast LCD** („option 15“):
Kontrast des Displays
- **ShutOff Backlight** („option 16“):
Dauer nach der letzten Tasten- bzw. Touchbetätigung bis in den Stromsparmodus gewechselt wird.
Das Gerät wechselt in den Normalbetrieb, wenn im Stromsparmodus eine Tasten- bzw. Touchbetätigung erfolgt. Diese Tasten- bzw. Touchbetätigung führt keine weiteren Ereignisse aus und dient ausschließlich zur Wiederherstellung des Normalbetriebs.
Beispiel CoDeSys
(*O 16 INT*) nLCDShutOffTime:=psFW^.uRemanent.sMasterEEprom.nLCDShutOffTime;
- **ShutOff Brightness** („option 17“):
Helligkeit des Displays im Stromsparmodus
Beispiel CoDeSys
(*O 17 Byte*) nDisplayShutOffBright:=psFW^.uRemanent.sMasterEEprom.nDisplayShutOffBright
- **Open key pad with ENTER** („option 18“):
Wenn diese Option aktiviert ist, wird bei einem Eingabefeld bei Betätigung von ENTER die Software-Tastatur für die Eingabe geöffnet.
Beispiel CoDeSys
(*O 18 Byte*)
bOpenKeyPadWithEnter:=psFW^.uRemanent.sMasterEEprom.bOpenKeyPadWithEnter;
- **Start calibration** („option 19“):
Startet die Touch-Kalibrierung im Service-Mode
- **Modbus0=Off,1=TCP,2=UDP** („option 20“):
Legt fest, ob der integrierte Modbus Client/Server aktiv ist und mit welchem Protokoll erarbeitet.
- **Modbus Port Read** („option 21“):
Legt den Port für Leseoperationen des integrierten Modbus Client/Server fest
- **Modbus Port Write** („option 22“):
Legt den Port für Schreiboperationen des integrierten Modbus Client fest
- **Exist persist.bin File** („option 23“):
Entscheidet, ob auf dem Gerät die Datei persist.bin vorhanden sein soll. Bei nicht aktivierter Datei kann bei bestimmten Geräten eine Speicherplatzvergrößerung für den IEC-Code erreicht werden.
- **Close KeyPad after inp.** („option 24“):
Mit dieser Option kann festgelegt werden, ob sich das Eingabefeld nach einer Eingabe schließen soll oder nicht.
- **Count of adu sampling** („option 25“):
Definiert die Anzahl der Stützstellen, aus denen der Mittelwert der analogen Eingangswerte gebildet wird (bei geräteinternen analogen Einheiten, wie z.B. COM.IO1, CM1xx, ...).

- **Brightness Display** („option 26“):
Mit dieser Option kann die Helligkeit des Displays im Normalbetrieb eingestellt werden.
(*O 26 Byte*) nDisplayBrightness
:=psFW^.uRemanent.sMasterEEprom.nDisplayBrightness;
- **Enable CF-FW** („option 27“):
Wird die automatische Update-Backup Funktion (de)aktiviert.
- **Keyb. Backl. Brightness** („option 28“):
Mit dieser Option kann die Helligkeit der Tastaturhintergrundbeleuchtung eingestellt werden.
- **Resource (resource.bin)** („option 30“):
- **RAMdisk aktiv** : („option“ 31...set value [0,1] ->"off"):
Aktiviert, wird das VFS-Verzeichnis (Virtual File System) zu einem Standardspeichermedium.
Damit Daten remanent abgelegt werden, ist kein Fixieren mehr nötig.
Sonst sind bei ausgeschaltetet Ramdisk eine der nachfolgenden Aktionen nötig:
Codesys-> FwSave(2);
Telnnet -> VFSSYNC
- **Active Key Matrix** : 33...set value [0-n]
- **Edit Style : 34 set value [[0]-standard, 1-4keys, 2-6keys]**
[0] - die Standardbedienung der Maskenelemente über 8 Tasten des Panels: LINKS; RECHTS, AUF; AB, +, -, OK, ESC
[1] - die Bedienung der Maskenelemente erfolgt über 4 Tasten des Panels: LINKS, RECHTS, OK, ESC
Für die Eingabe mit Hilfe eines Inkrementalgebers vorgesehen.
[2] - die Bedienung der Maskenelemente erfolgt über 6 Tasten des Panels: LINKS, RECHTS, OK, ESC, AUF, AB
Nach dem Start befindet sich das Gerät im Navigiermodus. Hier kann mit den Tasten Auf, Ab, Links und Rechts durch die Bedienelemente navigiert werden. Ausgenommen davon sind die Elemente Taster und Funktionstaster. Die Tasten Auf und Links bzw. Ab und Rechts haben dabei jeweils dieselbe Funktion. Mit der Taste ENTER wechselt man in den Editiermodus. Mit ENTER wird der Editiermodus mit Übernahme der Änderung, mit ESC ohne Übernahme der Änderung wieder verlassen.
Der Editiermodus ist bei den einzelnen Bedienelementen visuell erkennbar. Im Editiermodus haben die Tasten folgende Funktion:

Eingabefeld

Links, Rechts Bewegung des Cursors nach links bzw. rechts (Bei Ziffernweiser Eingabe)
Auf, Ab Inkrementieren bzw. dekrementieren des Werts bzw. der entsprechenden Stelle
OK Übernahme der Änderung und verlassen des Editiermodus
ESC Verlassen des Editiermodus ohne Änderung

Combobox

Links, Rechts Ohne Funktion
Auf, Ab Auswahl des Eintrags. Scrollen wenn der Eintrag nicht sichtbar ist.
OK Übernahme der Änderung und verlassen des Editiermodus
ESC Verlassen des Editiermodus ohne Änderung

Listbox

Links, Rechts Ohne Funktion

Auf, Ab Auswahl des Eintrags. Scrollen wenn der Eintrag nicht sichtbar ist.
OK Übernahme der Änderung und verlassen des Editiermodus
ESC Verlassen des Editiermodus ohne Änderung

Liste

Links, Rechts Ohne Funktion

Auf, Ab Auswahl des Eintrags. Scrollen wenn der Eintrag nicht sichtbar ist.
OK Verlassen des Editiermodus. Auswahl bleibt erhalten
ESC Verlassen des Editiermodus. Auswahl bleibt erhalten

Numerische Eingabe

Mit den Tasten Auf und Ab kann der Wert jeweils um 1 inkrementiert bzw. dekrementiert werden. Das Vorzeichen kann dadurch verändert werden, dass die erste Ziffer über 0 hinaus um 1 dekrementiert wird. Nach dem Neustart des Gerätes befindet es sich nicht im numerischen Eingabemodus.

Alphanumerische Eingabe

Mit Links, Rechts kann der Cursor auf die gewünschte Stelle bewegt werden. Mit den Tasten Auf und Ab kann dann ein Zeichen aus folgender Sequenz ausgewählt werden: A-Z, a-z, Sonderzeichen, Leerzeichen, 0-9. Wird das Ende der Sequenz erreicht wird wieder von vorne begonnen.

- **SRAM Waitstate** („option 35“): Default:= OFF;
Fügt für den Speicherzugriff einen Waitstate ein. Dadurch wird das Gerät ca. 20% langsamer. Dies ist erforderlich, wenn Sie die Geräte bei maximaler Gerätetemperatur, siehe Datenblatt, betreiben möchten. Falls die Umgebungstemperatur 10°C unterhalb der maximaler Gerätetemperatur liegt, kann diese Option mit OFF gewählt werden.
- **Hour glass by open mask** („option 36“):
Darstellung einer Sanduhr am Bildschirm zwischen Aufruf einer nicht geladenen Maske ihrer tatsächlichen Anzeige
- **Resource Begin Address** :
Anfangsadresse der Ressource (in hexadezimaler Schreibweise)
- **Resource Length** :
Länge der Ressource (in hexadezimaler Schreibweise)
- **IEC code Begin Address** :
Anfangsadresse des IEC-Codebereichs (in hexadezimaler Schreibweise)
- **IEC code Length** :
Länge der IEC-Codebereichs (in hexadezimaler Schreibweise)
- **Persist Begin Address** :
Anfangsadresse des Persist-Bereichs (in hexadezimaler Schreibweise)
- **Persist Length** :
Länge der Persist-Bereichs (in hexadezimaler Schreibweise)
- **RTU: Use Coils for Bits** („option 37“):
Bei Aktivierung werden Bits als Coils angefragt, d.h. mit Funktionscode 02 statt 03 Default:=No
- **RTU Timeout** („option 38“):

Default:=200ms

Zeit nach welcher eine neue Anfrage auf weitere Register durch den Master gestellt wird, wenn der Slave nicht antwortet.

- **RTU Frame Spacing** („option 39“):

Default:=10ms

Minimalzeit zwischen 2 Anfragen.

Ist diese Zeit zu klein gewählt, wird das Register erneut angefragt, obwohl der Slave aktuell im Begriff ist zu antworten.

Der Wert sollte mindestens das Doppelte der Zykluszeit des anzufragenden Slaves betragen.



Eine Korrektur von fehlerhaften FAT- Einträgen ist in den IT1.2, P305, CM1xx-Geräten nicht möglich. Dazu muss die CF in einem PC per "chkdsk d: /F" überprüft werden. Nach Umstellen der "Option 9" ist in den meisten Fällen ein Netzreset erforderlich, weil TCP/IP dann nicht mehr ordnungsgemäß abgearbeitet wird.

Wird während einer Schreiboperation auf dem CF die Spannung abgeschaltet, kann das Dateisystem zerstört werden. Hier soll von der Applikation ein Shutdown- Schalter oder Button eingeführt werden, mit dem vor dem Ausschalten der Spannung alle (im Schreibmodus) geöffneten Dateien (per SysLibFile-Funktionen) geschlossen werden.

1.9.5 Telnet Befehl : save

Speichern von Daten in Dateien:

```
C:\ Telnet 192.168.1.155
$visio/>ssaaavvee
save 'persist.bin' : 2  #=0 at 00.00.0000 00:00:00
save 'eventlog.bin' : 3
$visio/>
```

save 'persist.bin' (save 2) :

Speichert die als ‚persistent‘ definierten Datenpunkte in die Datei ‚persist.bin‘.

Mit Speicherung der persistent Daten werden auch die retain Daten gespeichert, wenn sich mind. ein aktueller Wert der persistent Daten von den gespeicherten Werten unterscheidet (siehe auch Befehl load). Die persistent Daten können auch durch das Setzen der Userware-Variablen saveParameter = 1 gespeichert werden.

Es wird zusätzlich die Anzahl der Speichervorgänge, sowie Uhrzeit und Datum des letzten Speichervorgangs angezeigt.

save 'eventlog.bin' (save 3) :

Speichert die im RAM abgelegten Einträge des Eventloggers in die Datei ‚eventlog.bin‘.

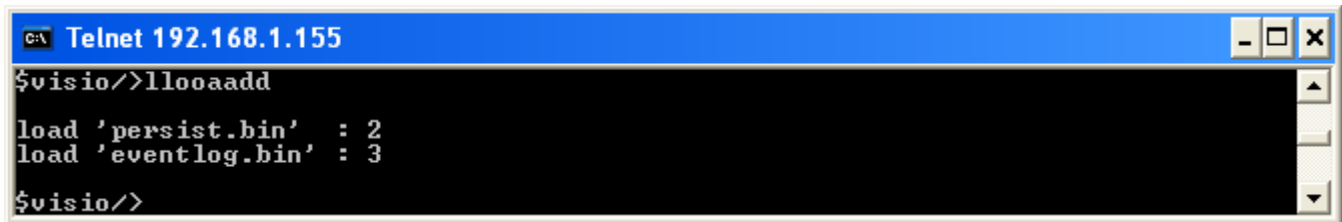
Unter CoDeSys steht hier der Befehl FwSave(nMode) der Bibliothek sFwXX.lib zur Verfügung.

Beispiel:

(* Flashen der Retain- und Persistent- Daten *) FwSave(16#11);

1.9.6 Telnet Befehl : load

Laden von Daten aus Dateien:



```
C:\ Telnet 192.168.1.155
$visio/>lloadadd
load 'persist.bin' : 2
load 'eventlog.bin' : 3
$visio/>
```

load 'persist.bin' (load 2) :

Lädt persistente Daten (Defaultwerte für “mit gespeichertem Parameter (PERSISTENT)” und “mit batteriegepuffertem Wert (RETAIN)” definierte Datenpunkte) aus der Datei ,persist.bin' (siehe auch Befehl save).

load 'eventlog.bin' (load 3) :

Lädt die in der Datei 'eventlog.bin' gespeicherten Einträge des Eventloggers ins RAM. Existiert die Datei nicht werden die Einträge im RAM gelöscht.



(* Laden der zuvor gespeicherten Retain- & Pers.- Daten *)

FwLoad(1);

ST

1.9.7 Telnet Befehl : rs232

```
$combo/>rs
RS232 COM0 Rx= 28,Tx=4017,Overrun=0
          38400 bit/s, Mode 0,Data 8,Stop 1, Parity 0
RS232 COM1 Rx= 0,Tx= 0,Overrun=0
          0 bit/s, Mode 0,Data 0,Stop 0, Parity 0
-----
RS232 COM0 as DEBUG      : 2...set value [0,1] ->"on"
RS232 COMx for CoDeSys  : 3...set value [0=disable,1] -> 0 = disable
RS232 COM0 Baudrate     : 4...set value [600...<38400>...57600] -> 38400
RS232 RTS Off=0/On=1   : 5 -> 0n
```

RS232 COM0 as Debug:

Die erste Serielle Schnittstelle wird zur Debugausgabe verwendet. Schalten sie diese Option ab, wenn sie die Serielle Schnittstelle in CoDeSys benutzen wollen. Im Sonderfall Betriebsart Servicemode ist diese Option eingeschalten und die PLC-Code Abarbeitung gestoppt.

RS232 CPMx for CoDeSys:

Wahl der Schnittstelle die für den Download und Debugging von CoDeSys verwendet wird. Möchten Sie die Serielle Schnittstelle aus CoDeSys heraus verwenden, so muß diese Option abgeschaltet sein.

RS232 COM0 Baudrate:

Die Defaultbaudrate für die Debug-Ausgabe.

Bitte beachten sie, dass die tatsächliche Baudrate beim Öffnen des Comports aus CoDeSys gesetzt wird.

RS232 RTS off/On

Aktiviert / deaktiviert den Hardware- Handshake (Nur an voll belegten RS232-Schnittstellen)

Telnet Befehl : can

Ändern von Parametern und Diagnose der CAN-Schnittstellen:

```
Telnet 192.168.1.126
$visio/>ccaann

CAN0                               : Remote | Normal | Queue | Overrun |
CAN0 MSG01/02<send obj.>:         0 |    6366 |    0 |    0 |
CAN0 MSG15  <recv obj.>:         0 |    0 |    0 |    0 |
CAN0 User   :                     0 |    0 |    0 |    0 |
CAN1
CAN1 MSG17/18<send obj.>:         0 |    0 |    0 |    0 |
CAN1 MSG31  <recv obj.>:         0 |    0 |    0 |    0 |
CAN1 User   :                     0 |    0 |    0 |    0 |

ESB functionality                 : 1...set value [0,1] <off>
CAN0 Baud                          : 2...set value [10,20,50,100,<123>,125,250,500] <500>
CAN0 extended <29bit>              : 3...set value [<0>,1] <0>
CAN0 NodeID <My Module>           : 4...set value [62] <62>
CAN0 CANopen Active/Node:         : 5...set value [0,<1>] <0> on 0 Node
CAN0 ElaCAN Active                 : 6...set value [0,<1>] <1>
CAN0 Send Id                       : 7...set value [0x200] <0x200>
CAN0 Recv Id                       : 8...set value [0x240] <0x240>
CAN0 Intermod Id                   : 9...set value [0x280] <0x280>
CAN0 Intermod Master               :10...set value [0x00] <0>
CAN0 Termination                   :13...set value [0,1] ->"on"

CAN1 Baud                          :14...set value [10,20,50,100,<123>,125,250,500] <125>
CAN1 extended <29bit>              :15...set value [<0>,1] <0>
CAN1 NodeID <My Module>           :16...set value [62] <62>
CAN1 CANopen Active/Node:         :17...set value [0,<1>] <0> on 0 Node
CAN1 ElaCAN Active                 :18...set value [0,<1>] <1>
CAN1 Send Id                       :19...set value [0x200] <0x200>
CAN1 Recv Id                       :20...set value [0x240] <0x240>
CAN1 Intermod Id                   :21...set value [0x280] <0x280>
CAN1 Intermod Master               :22...set value [0x00] <0>

$visio/>
```

Statistik

Zeigt die Anzahl empfangener und gesendeter Telegramme der CAN-Schnittstellen an.

In der Statistik Queue/Overrun darf es zu keinem Überlauf kommen.

Falls dies dennoch auftritt, müssen die Thread-Zeiten angepasst werden.

CAN- Monitor („can 0“):

Hinweis: Diese Monitoring-Ausgabe ist als serielle Ausgabe nicht verfügbar. Bitte verwenden Sie Telnet.

Scrollen mit Tasten +/-

```
CAN 0 : show receiving CAN frames, press <+>,<-> for page or <CR>

CAN=Bus ok(00000000), 125[kBits/s], 11bit normal, MaxIndex=0256, Free=      3076 from      18480[kB]

CAN0      |receive      0+RTR      0      || 1082.728|transmit      0+RTR      0

Timestamp |Index|ID  |L|E|R|Data      ||Timestamp |Index|ID  |L|E|R|Data

0.000|>000<|0000|0| | | || 0.000|>000<|0000|0| | |
0.000| 001 |0000|0| | | || 0.000| 001 |0000|0| | |
0.000| 002 |0000|0| | | || 0.000| 002 |0000|0| | |
0.000| 003 |0000|0| | | || 0.000| 003 |0000|0| | |
0.000| 004 |0000|0| | | || 0.000| 004 |0000|0| | |
0.000| 005 |0000|0| | | || 0.000| 005 |0000|0| | |
0.000| 006 |0000|0| | | || 0.000| 006 |0000|0| | |
```

```

0.000| 007 |0000|0| | | || 0.000| 007 |0000|0| | |
0.000| 008 |0000|0| | | || 0.000| 008 |0000|0| | |
0.000| 009 |0000|0| | | || 0.000| 009 |0000|0| | |
0.000| 010 |0000|0| | | || 0.000| 010 |0000|0| | |
0.000| 011 |0000|0| | | || 0.000| 011 |0000|0| | |
0.000| 012 |0000|0| | | || 0.000| 012 |0000|0| | |
0.000| 013 |0000|0| | | || 0.000| 013 |0000|0| | |
0.000| 014 |0000|0| | | || 0.000| 014 |0000|0| | |
0.000| 015 |0000|0| | | || 0.000| 015 |0000|0| | |

```

```
CAN=Busoff(00000001), 125[kBits/s], 11bit normal, MaxIndex=0256, Free= 3076 from 18480[kB]
```

```
CAN1 |receive 0+RTR 0 || 1082.740|transmit 0+RTR 0
```

Timestamp	Index	ID	L	E	R	Data	Timestamp	Index	ID	L	E	R	Data
0.000	>000	<	0000	0			0.000	>000	<	0000	0		
0.000	001	0000	0				0.000	001	0000	0			
0.000	002	0000	0				0.000	002	0000	0			
0.000	003	0000	0				0.000	003	0000	0			
0.000	004	0000	0				0.000	004	0000	0			
0.000	005	0000	0				0.000	005	0000	0			
0.000	006	0000	0				0.000	006	0000	0			
0.000	007	0000	0				0.000	007	0000	0			
0.000	008	0000	0				0.000	008	0000	0			
0.000	009	0000	0				0.000	009	0000	0			
0.000	010	0000	0				0.000	010	0000	0			
0.000	011	0000	0				0.000	011	0000	0			
0.000	012	0000	0				0.000	012	0000	0			
0.000	013	0000	0				0.000	013	0000	0			
0.000	014	0000	0				0.000	014	0000	0			
0.000	015	0000	0				0.000	015	0000	0			

- **ESB functionality** („can 1“): Default:1

Mit diesem Kommando kann die ESB-Funktionalität auf der CAN-Schnittstelle 0 an bzw. ausgeschaltet werden. Schaltet zwischen CAN- und ESB-Funktionalität um: 0 : CAN 1 : ESB
Beispiel in CoDeSys;

```
(*C 1 Byte*) bESBActivated:=psFW^.uRemanent.sMasterEEProm.bESBActivated;
```

Ist die ESB-Funktionalität nicht eingeschaltet, sind folgende weiteren Optionen für beide CAN-Schnittstellen verfügbar. (Ist sie eingeschaltet nur für CAN 1.)

- **CANn Baud** („can 2 / 14“) Default:125

Auswahl der Baudrate für die jeweilige CAN-Schnittstelle

Zu beachten: 123 kBaud werden für CoDeSys-Programmierung nicht unterstützt.

Beispiel in CoDeSys;

```
(*C 2 INT*) nCAN0Baudrate:=psFW^.uRemanent.sMasterEEProm.nCAN0Baudrate;
```

```
(*C 13 INT*) nCAN1Baudrate:=psFW^.uRemanent.sMasterEEProm.nCAN1Baudrate;
```

- **CANx Extended (29bit)** („can 3 / 15“) Default: 0
Länge des CAN-Identifizier (1 = 29 bit, 0 = 11 bit
Beispiel in CoDeSys;
(*C 3 Byte*) `bCAN029bit:=psFW^.uRemanent.sMasterEEprom.bCAN029bit;`
(*C 14 Byte*) `bCAN129bit:=psFW^.uRemanent.sMasterEEprom.bCAN129bit;`
- **CANx NodeID (My Module)** („can 4 und 16“):
a) CANopen Active : Node für CANopen Slave
b) CAN Layer2 Active: Moduladresse für CoDeSys-Programmierung über USB- oder Centronix-Schnittstelle
- **CANx CANopen Active/Node** („can 5 und 17“):
Aktivieren von CoDeSys-CANopen. Der CANopen Node wird direkt über die Konfiguration von CoDeSys eingestellt.
Zu beachten: Für die Programmierung mit CoDeSys muss CANopen aktiviert sein.
Beispiel in CoDeSys;
(*C 5 Byte*) `bCAN0CanOpenActice:= psFW^.uRemanent.sMasterEEprom.bCAN0openActive;`
(*C 17 Byte*) `bCAN1CanOpenActice:= psFW^.uRemanent.sMasterEEprom.bCAN1openActive;`
- **CANx Layer2 Active** („can 6 und 18“):
Aktivieren des elrest CAN Protokolls. Mit aktivieren dieser Option stehen weitere Einstellmöglichkeiten zur Verfügung. Weitere Details finden Sie im Kapitel „Kommunikation CAN“.
- **CANx Send Id** („can 7 und 19“):
Basisadresse für die Send-ID.
- **CANx Recv Id** („can 8 und 20“):
Basisadresse für die Empfangs-ID.
- **CANx Intermod Id** („can 9 und 21“):
Basisadresse für die Intermodul-ID.
- **CANx Intermod Master** („can 10 und 22“):
Umschaltung auf Multi-Master Betrieb.
- **CAN0 Termination** („can 13 <on/off>“) Default: 1
Schaltet den geräteinternen CAN0-Abschlusswiderstand: falls auswählbar
0 : Abschlusswiderstand nicht zugeschaltet
1 : Abschlusswiderstand zugeschaltet
fixed on xxx board (on/off) : wird angezeigt, falls Abschlusswiderstand im Gerät immer/nicht vorhanden.
Beispiel unter CoDeSys:
Unter CoDeSys gibt es keinen separaten Parameter. Hier muss ausgelesen, maskiert und geschrieben werden.


```

26
27 (* Relais für den Abschlusswiderstand rücksetzen, CAN Konfiguration und Neustart erforderlich -----
28 IF bCANTermination = 0 THEN
29     nCopyTemp                := INT_TO_BYTE(psFW^.uRemanent.sSlaveEEProm.OptionFlag);
30     wMaskeOptionFlag         := 2#00100000;
31     nCopyTemp                := nCopyTemp OR wMaskeOptionFlag;
32     psFW^.uRemanent.sSlaveEEProm.OptionFlag := nCopyTemp;
33 ELSE
34     nCopyTemp                := INT_TO_BYTE(psFW^.uRemanent.sSlaveEEProm.OptionFlag);
35     wMaskeOptionFlag         := 2#11011111;
36     nCopyTemp                := nCopyTemp AND wMaskeOptionFlag;
37     psFW^.uRemanent.sSlaveEEProm.OptionFlag := nCopyTemp;
38 END_IF
39
40 ESB := psFW^.uRemanent.sSlaveEEProm.OptionFlag;
41 *)
42

```

Ist die ESB-Funktionalität eingeschaltet sind folgende weiteren Optionen für die CAN-Schnittstellen 0 verfügbar

```

Telnet 192.168.1.126
$visio/>ccaann

CAN1                               : Remote | Normal | Queue | Overrun |
CAN1 MSG17/18<send obj.>:         : 0 | 0 | 0 | 0 |
CAN1 MSG31 <recv obj.>:           : 0 | 0 | 0 | 0 |
CAN1 User                          : 0 | 0 | 0 | 0 |

ESB functionality                   : 1...set value [0,1] <CAN0>
Save ESB-Configuration             : 2...compare [<0>:OUI,1:TYP,2:NOT] -> TYP
Restart ESB-Config.                : 3
SDOs of module n                   : 4

#   actual ---ESB modules--- saved <2>  ESB-OK
#  type order OUI             type      OUI
1  CS100 1  3025408          CS100   3025408
2  CS100 1  3418624          CS100   3418624

CAN1 Baud                          :14...set value [10,20,50,100,<123>,125,250,500] <125>
CAN1 extended <29bit>              :15...set value [<0>,1] <1>
CAN1 NodeID <My Module>            :16...set value [62] <62>
CAN1 CANopen Active/Node:17...set value [0,<1>] <1> on 0 Node
CAN1 ElaCAN Active                 :18...set value [0,<1>] <0>

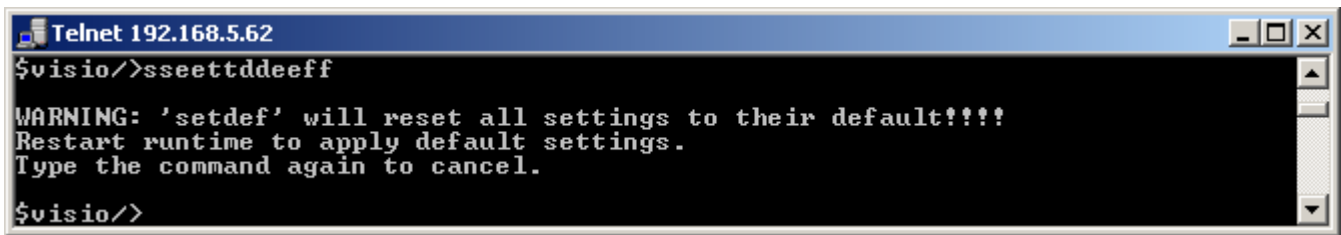
$visio/>_

```

- **Save ESB-Configuration („can 2“):**
Speichert die aktuelle Gerätekonfiguration im seriellen EEPROM und legt fest, ob und nach welchem Kriterium (OUI, Typ) diese Konfiguration überprüft werden soll.
- **Restart ESB-Config. („can 3“):**
Einlesen der aktuellen Gerätekonfiguration am ESB.
- **SDOs of module n („can 4“):**
nur zur internen Verwendung
- **Diagnose:**
zeigt den aktuellen Status des ESB (weitere Informationen hierzu siehe „Kommunikation ESB“)

1.9.8 *Telnet Befehl : setdef*

Setzt alle Einstellungen auf ihre Standardwerte zurück.



```
Telnet 192.168.5.62
$visio/>>sseettddeeff
WARNING: 'setdef' will reset all settings to their default!!!!
Restart runtime to apply default settings.
Type the command again to cancel.
$visio/>>
```

Die Werte werden erst nach einem Neustart der Runtime zurückgesetzt. Sollen die Werte doch nicht zurückgesetzt werden kann der Befehl noch einmal eingegeben werden.

1.9.9 *Telnet Befehl : password*

Der Befehl

password <eigenes password>

setzt nach einem Neustart das Passwort für Telnet.

Künftig ist es nur noch möglich, sich unter Telnet mit diesem Passwort anzumelden. Die FTP-Funktionalität bleibt davon unberührt.

Beispiel:

```
visio_P205V login:user
user
User name ok, need password
```

```
-----
telnet:
$visio_P205V/>>
```

Löschen des Passwort:

Nach erneuter Anmeldung unter Telnet den Befehl

*Password **

eingeben.

1.9.10 Telnet Befehl : setmod

Ändern der Moduladresse:

```
Telnet 192.168.5.62
$visio/>sseettmoodd 220000
WARNING: 'setmod' will kill all current elrest connections!!!!
```

```
Telnet 192.168.5.62
$visio/>ssttaatee

>visio_P406 1.72-2 Jul 11 2007 15:49:06
>ElaDevice ElaOnline ESB
>Module 200=serware, CAN:terminated 500kBaud
192.168.5.62, subnet 255.255.255.0, gatew. 192.168.5.1, DHCP
>MAC=00-50-C2-3B-89-52 nOK SN=-1 Date 12.07.0007 17:18:36
$visio/>
```

1.9.11 Telnet Befehl : setip

Ändern der IP-Adresse:

```
Telnet 192.168.5.221
$visio/>sseettiipp 119922..116688..55..222222
replacing 192.168.5.221 with 192.168.5.222
$visio/>
```

Die IP-Adresse wird nur übernommen wenn DHCP abgeschaltet ist. Telnet muss anschliessend mit der geänderten IP-Adresse neu gestartet werden.

Beispiel CoDeSys:



ST

```
IF xREAD then

    MP_IP_Adress[0]:=BYTE_OF_DWORD(psFW^.nLastIP_address,0);
    MP_IP_Adress[1]:=BYTE_OF_DWORD(psFW^.nLastIP_address,1);
    MP_IP_Adress[2]:=BYTE_OF_DWORD(psFW^.nLastIP_address,2);
    MP_IP_Adress[3]:=BYTE_OF_DWORD(psFW^.nLastIP_address,3);
    xREAD:=FALSE;

END_IF;

IF xWRITE THEN
    dwTemp:=USINT_TO_DWORD(MP_IP_Adress[0]);
    dwTemp:=dwTemp OR (SHL(USINT_TO_DWORD(MP_IP_Adress[1]),8));
    dwTemp:=dwTemp OR (SHL(USINT_TO_DWORD(MP_IP_Adress[2]),16));
    dwTemp:=dwTemp OR (SHL(USINT_TO_DWORD(MP_IP_Adress[3]),24));
    psFW^.uRemanent.sMasterEEprom.nMyIP_address:=dwTemp;
    xWRITE:=FALSE;
END_IF;
```

1.9.12 Telnet Befehl : subnet

Ändern der Subnet-Maske:

```
C:\ Telnet 192.168.1.155
$visio/>ssuubbnneett
usage: X.X.X.X
WARNING: this will kill all current net connections!
$visio/>
```

Beispiel CoDeSys:



ST

```
IF xREAD then

    MP_IP_Subnet[0]:=BYTE_OF_DWORD(psFW^.nLastIP_subnet,0);
    MP_IP_Subnet[1]:=BYTE_OF_DWORD(psFW^.nLastIP_subnet,1);
    MP_IP_Subnet[2]:=BYTE_OF_DWORD(psFW^.nLastIP_subnet,2);
    MP_IP_Subnet[3]:=BYTE_OF_DWORD(psFW^.nLastIP_subnet,3);

    XREAD:=FALSE;
    END_IF;

IF xWRITE THEN

    dwTemp:=USINT_TO_DWORD(MP_IP_Subnet[0]);
    dwTemp:=dwTemp OR (SHL(USINT_TO_DWORD(MP_IP_Subnet[1]),8));
    dwTemp:=dwTemp OR (SHL(USINT_TO_DWORD(MP_IP_Subnet[2]),16));
    dwTemp:=dwTemp OR (SHL(USINT_TO_DWORD(MP_IP_Subnet[3]),24));
    psFW^.uRemanent.sMasterEEProm.nMyIP_Subnet:=dwTemp;
    xWRITE:=FALSE;
    END_IF;
```

1.9.13 Telnet Befehl : gateway

Ändern der Gateway-Adresse:

```
C:\> Telnet 192.168.1.155
$visio/>ggaatteewwaayy
usage: X.X.X.X
WARNING: this will kill all current net connections!
$visio/>
```

1.9.14 Telnet Befehl : setdhcp

Schaltet DHCP auf dem Gerät ein oder aus.



```
Telnet 192.168.5.62
$visio/>> ssettdhccpp 00
DHCP reset
$visio/>>
```

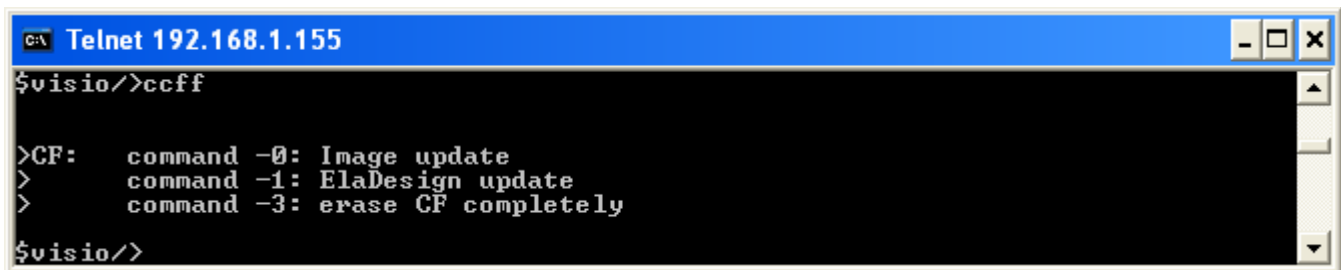
Beispiel CoDe3Sys;

(*setdhcp byte*)

nDHCPactivated:=psFW^.uRemanent.sMasterEEprom.nDHCPactivated;

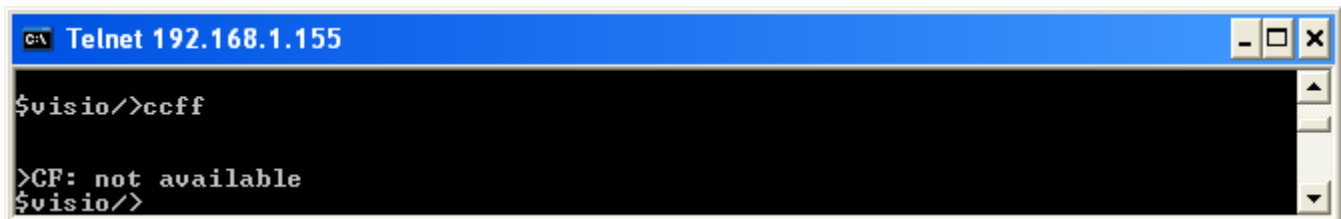
1.9.15 Telnet Befehl : cf

Siehe Kapitel „Update-Funktionalität“



```
Telnet 192.168.1.155
$visio/>> ccff

>CF:  command -0: Image update
>      command -1: ElaDesign update
>      command -3: erase CF completely
$visio/>>
```



```
Telnet 192.168.1.155
$visio/>> ccff

>CF: not available
$visio/>>
```

Ist keine CF-Karte eingelegt, so erscheint „CF: not available“

1.9.16 Telnet Befehl : threads

Anzeigen und Einstellen von Thread-Zeiten:

<T>

Sie erhalten einen Screenshot über die Systemzeiten.

```
$combo/>t
Threads:      ob0      ob1      ob2 (CO)  TcpIp      CAN0      CAN1
Period /ms   endless  150      30        35         10        10
Max           2         1         0         41         0         0
Gap           0         132      29        35         5         6
Count        307      40289   4826     41701     14887    14887
```

```
Task Watchdog Timeout :10...set value [10..100]s,<0>=OFF -> (0 s)
```

Im oberen Teil der Anzeige werden die Zykluszeiten der einzelnen Tasks angezeigt

Im unteren Teil wird die Anzahl der Durchläufe und die Abarbeitungszeit der einzelnen Tasks angezeigt.

Zykluszeiten und Abarbeitungszeiten werden jeweils zwischen zwei Aufrufen des Kommandos gemessen.

Threads

Name des System- oder projektierten CoDeSysthreads

Period/ms

Angabe der eingestellten Zykluszeit

Max

Zyklusdauer in ms

Count:

Zähler , Zyklen insgesamt, seit dem letzten aufruf des Befehl <T>

<T [Spaltennummer] xyz>

Einstellen der Zykluszeit

Beispiel zum Screenshot oben:

<T 1 150> [ENTER] → ändern der Zykluszeit von OB1 auf 150ms

<T 2 30> [ENTER] → ändern der Zykluszeit von OB2 auf 150ms

<T 3 35> [ENTER] → ändern der Zykluszeit von Ethernet auf 150ms

<T 4 10> [ENTER] → ändern der Zykluszeit des ersten CanController auf 10ms

<T 5 10> [ENTER] → ändern der Zykluszeit des zweiten CanController auf 10ms

1.9.17 Telnet Befehl : res

Startet das Gerät neu.



```
Telnet 192.168.5.222
visio:rreesett
Reseting target...
$visio/>
```

1.9.18 Telnet Befehl : xtest

Nur für interne Verwendung.

1.9.19 Telnet Befehl : hwstate

Der Hardwarestatus zeigt den Zustand der analogen Ein- und Ausgänge (Verfügbar nur bei combo-CM1xx).

```
$combo/>hwstate
>CM110/CE152 1.60-25 Dec 11 2006 08:43:28 HW-V2.0
>-----
>Device Temp: 35 C
>Sensor Settings:
> Channel 0: no Sensor
> Channel 1: no Sensor
> Channel 2: no Sensor
> Channel 3: no Sensor
>-----
>CE-Unit: combo-CE152 plugged in (HW 0.1)
>Analog inputs:
> Channel 8: off
> Channel 9: off
> Channel 10: off
> Channel 11: off
>Analog outputs:
> Channel 8: off
> Channel 9: off
```

1.9.20 Telnet Befehl : setCE

1.9.21 Telnet Befehl : clearMT

1.9.22 Telnet Befehl : linkstats

Empfangene und gesendete Telegramme seit Neustart des Gerätes:


```
telnet 212
fred:linkstats
>eth(Rx 6410<1/s> Tx 6472<1/s>>
>eth(Arp<24 Skip 703> Ip<6386 Skip 1634> NoIp 8752)
$fred/>_
```

Rx 6410(1/s): 6410 empfangene Telegramme seit Neustart, im Durchschnitt 1/Sekunde.

Ts 6472(1/s): 6472 gesendete Telegramme seit Neustart, im Durchschnitt 1/Sekunde.

Arp(24 Skip 703) : **A**dress **r**esolution **p**rotokoll mit 24 für das Gerät bestimmte Telegramme und 703 nicht für das Gerät bestimmte Telegramme.

IP(6386 Skip 1634) : IP-Telegramme (dto.)

NoIP 8752 : NoIP-Telegramme (dto.)

1.9.23 Telnet Befehl : sockets

Geöffnete Sockets des Gerätes:

```
telnet 212
$fred/>sockets
socket fhost ports opts state rxbytes txbytes snd_una snd_nxt
8550ea 192.168. 1.215 23/1286 100 ESTABLISHED 0 113 3206608 3206610
8687fc 0. 0. 0. 502/ 0 102 LISTEN 0 0 0 0
868920 0. 0. 0. 23/ 0 102 LISTEN 0 0 0 0
868a44 0. 0. 0. 21/ 0 102 LISTEN 0 0 0 0
868b68 0. 0. 0. 80/ 0 102 LISTEN 0 0 0 0
done
$fred/>_
```

1.9.24 Telnet Befehl : memstats

RAM Speicherbereiche:

```
$combo>mmeem
mh_free: 138.439 (352) kB
mh_blocks: 2, max=137.906 kB
z_free: 214.871 (256) kB
z_blocks: 3, max=214.73 kB
np_totfree: 754 (1024)
np_allocs: 274, fails=0, frees=4
stack used: 0 (8192) Bytes
bigfreeq: 39, lilfreeq 1, ethrecvqinp 40, total 80
eth-t/o: 1 (run)
```

mh_free : freier systeminterner RAM Speicher im Heap. In Klammer der gesamte zur Verfügung stehende Speicher.

mh_blocks : Anzahl der Blöcke (zusammenhängender Speicherbereich). In Klammer die maximal mögliche Blockgröße.

z_free : freier RAM Speicher für den Anwenderbereich, z. B. ElaDesign. In Klammer der gesamte zur Verfügung stehende Speicher.

z_blocks : siehe mh_blocks.

np_totfree : ausschließlich für interne Verwendung

np_allocs : ausschließlich für interne Verwendung

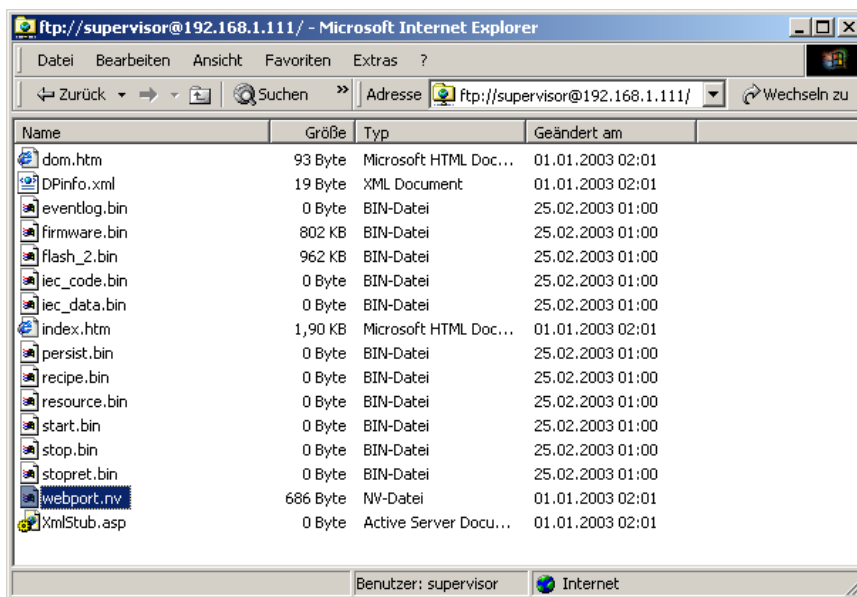
stack used : ausschließlich für interne Verwendung

bigfreeq : ausschließlich für interne Verwendung

eth-t/o :

1.9.25 Telnet Befehl : vfssync

Für Änderungen der Subnetmaske oder des Gateways muss die Datei webport.nv angepasst werden. Hierfür ist die Datei webport.nv über einen Browser und ftp-Zugriff auf eine lokale Festplatte zu kopieren.



1.9.26 Telnet Befehl : X – Befehle (Analogkonfiguration P303)

Durch Eingabe des Befehles X 5 erhalten Sie eine Übersicht über alle 8 onboard befindenden AIO-Kanäle

```

#: typ adc   dac   input   output   internal
0: 5   22    0   -0.00346 0       0
1: 5   22    0   -0.00346 0       0
2: 5   19    0   -0.00495 0       0
3: 5   23    0   -0.00297 0       0
4: 5   18    0   -0.00544 0       0
5: 5   19    0   -0.00495 0       0
6: 5   18    0   -0.00544 0       0
7: 5   47    0   0.0089   0       0
internalTemperature: 41.3

```

Erläuterung:

Typ: Parametrierung des Kanales gemäß der Enumeration der IO01.lib. Eine Übersicht erhalten Sie auch mit dem Befehl: X 7 4

adc: Analog/Digital Wandler- Wert in Digits (0..4095)

dac: Digital/Analog Wandler- Wert in Digits (0..4095)

inputs: Anzuzeigender Eingangswert

Einstellmöglichkeiten - Übersicht

```

XI0: eXtended I/O
0: channel type e.g. 'x 7 0 <channel> <type>' (for types try 'x 7 4')
1: output value e.g. 'x 7 1 <channel> <value>'
2: calibrate e.g. 'x 7 2 <password> [state]' (only with plug 1)
3: [kp|tn|ymax]: 10U output control parameters
4: sensor types
5: sampling time
6: filter time
7: internal temperature offset
s: write standard values
t: Channel test + password, (only with plug 2: LEDs)
u: Utc fine calibration + password, (only with plug 3: bridges)
v: test 10U outputs + password, (remove plug)
Hint for testing:
plug 1, x 7 2,
plug 2, x 7 t,
plug 3, x 7 u,
no plug, x 7 v

```

1.9.26.1 Telnet Befehl : X 7 ?

Einstellmöglichkeiten - Übersicht

```

XI0: eXtended I/O
0: channel type e.g. 'x 7 0 <channel> <type>' (for types try 'x 7 4')
1: output value e.g. 'x 7 1 <channel> <value>'
2: calibrate e.g. 'x 7 2 <password> [state]' (only with plug 1)
3: [kp|tn|ymax]: 10U output control parameters
4: sensor types
5: sampling time
6: filter time
7: internal temperature offset
s: write standard values
t: Channel test + password, (only with plug 2: LEDs)
u: Utc fine calibration + password, (only with plug 3: bridges)
v: test 10U outputs + password, (remove plug)
Hint for testing:
plug 1, x 7 2,
plug 2, x 7 t,
plug 3, x 7 u,
no plug, x 7 v

```

1.9.26.2 Telnet Befehl : X 7 2

Übersicht über die durchgeführte Kalibrierung

Befehl: X 7 2

Hier erhält man eine Übersicht über die
Eingestellten Parameter und Kalibrierungen.

Die Low- Werte sollten gegen Null tendieren und
keinesfalls dem hi- Wert entsprechen.

Ist dies nicht der Fall, so wurde mangelhaft oder gar
nicht kalibriert.

```
Calibration values:
0: 29 (Uin10U_lo)
1: 1040 (Uin10U_hi)
2: 10 (Iin20mA_lo)
3: 916 (Iin20mA_hi)
6: 1281 (Pt100_3L_lo)
7: 2336 (Pt100_3L_hi)
12: 1013 (TCJ_d_lo)
13: 3731 (TCJ_d_hi)
14: 741 (TCB_d_lo)
15: 3163 (TCB_d_hi)
16: 0 (Temp)
17: 97 (Iout20mA_lo)
18: 2367 (Iout20mA_hi)
21: 15 (SampleTicks)
22: 3 (Filter)
23: 0 (IntTempOffs)
24: 11 (Cal_KTY_lo)
25: 2531 (Cal_KTY_hi)
26: 50 (UoltP)
27: 2 (UoltI)
29: 1 (CalDone)
30: 2 (OffsTcB_0)
31: 6 (OffsTcB_1)
32: 4 (OffsTcB_2)
33: 4 (OffsTcB_3)
34: 1 (OffsTcB_4)
35: 3 (OffsTcB_5)
36: 3 (OffsTcB_6)
37: 0 (OffsTcB_7)
38: 2 (OffsTcJ_0)
39: 5 (OffsTcJ_1)
40: 2 (OffsTcJ_2)
41: 2 (OffsTcJ_3)
42: 5 (OffsTcJ_4)
43: 16 (OffsTcJ_5)
44: 5 (OffsTcJ_6)
45: 8 (OffsTcJ_7)
```

1.9.26.3 Telnet Befehl : X 7 3

Regelparameter für den Spannungsausgang.

Eine Änderung dieser Parameter ist durch den Benutzer nicht vorgesehen

1.9.26.4 Telnet Befehl : X 7 2

Sensor-Type.

Liste aller am P303 verwendbaren Analog E/A-
Arten.

Bitte nehmen Sie diese Einstellung zu CoDeSys-
Programmstart mittels der IO01.lib vor.

```
Sensor types:
0: voltage input, 0/10U,
1: voltage input, 0/70mU
2: voltage input, 0/15mU
3: current input, 0/20mA
4: current input, 4/20mA
5: voltage input, 0/10U, normed
6: current input, 0/20mA, normed
7: current input, 4/20mA, normed
8: resistor input, 0-350 Ohm, 2 wire
9: resistor input, 0-350 Ohm, 3 wire
10: compensation temperature 1
11: compensation temperature 2
12: TC- input, B 250/1820 C
13: TC- input, E -100/800 C
14: TC- input, J -100/1200 C
15: TC- input, K -150/1372 C
16: TC- input, L -150/900 C
17: TC- input, N -150/1300 C
18: TC- input, R -50/1768 C
19: TC- input, S -50/1768 C
20: TC- input, T -150/400 C
21: Pt100 2 wire -100/700 C
25: Ni100 2 wire -100/700 C
26: Ni120 2 wire -100/700 C
31: Pt100 3 wire -100/700 C
35: Ni100 3 wire -100/700 C
36: Ni120 3 wire -100/700 C
52: current output, 0/20mA normed
53: current output, 4/20mA normed
54: voltage output, 0/10U normed
55: disable this channel
65: digital contact current input 5mA
66: digital voltage input 24U
67: digital output 10mA
68: voltage/current source
```

1.9.26.5 Telnet Befehl : X 7 5

Regelparameter für den Spannungsausgang.

Eine Änderung dieser Parameter ist durch den Benutzer nicht vorgesehen.

1.9.26.6 Telnet Befehl : X 7 6

Filter Time

Die Filtertime wird als Exponent von 2 angeben. D.h. bei der Angabe

Filter Time

1 = 2 Messungen

2 = 4 Messungen

3 = 8 Messungen

4 = 16 Messungen

Der Filter wirkt wie eine Mittelwertbildung um Rauschen oder extreme Spitzen zu Filtern.

Beispiel unter CoDeSys

[nNoOfADUAverageElements:=psFW^uRemanent.sMasterEEProm.nNoOfADUAverageElements;](#)

1.9.26.7 Telnet Befehl : X 7 7

Interner Temperatur- Offset.

Der interne Temperatur- Offset sollte immer auf null sein.

Gründe für eine Änderung sind darin begründet, dass z.B. Auf allen Kanälen eine gleich große Abweichung gegenüber einer Referenztemperatur zu beobachten ist.

Bitte Beachten Sie dass Thermofühler eine bestimmte Zeit benötigen, bis Sie korrekt arbeiten. Das Gerät muss hierzu Betriebstemperatur erreicht haben (10- 15 Minuten)

1.9.26.8 Telnet Befehl : X 7 S

Setzt alle, durch den Benutzer änderbaren Werte auf „Default“

1.9.27 Telnet Befehl : profibus

Falls unter Telnet / Hyperterminal die Profibus Option „set profibus diagnostic : 1 -> On“ aktiviert wurde, werden wichtige Informationen während dem Bootvorgang ausgegeben.

```
$fred/>profibus
DPS      COM-DPS U01.058 19.12.03
set profibus diagnostic      : 1 -> On
get life list of slaves      : 2
get config. of slave        : 3 [1...127]
global state field          : 4
task info                    : 5
dump input data              : 6
dump output data             : 7
$fred/>_
```



Ausführliche Angaben zu Profibus finden sie in der Dokumentation Kommunikation Profibus.

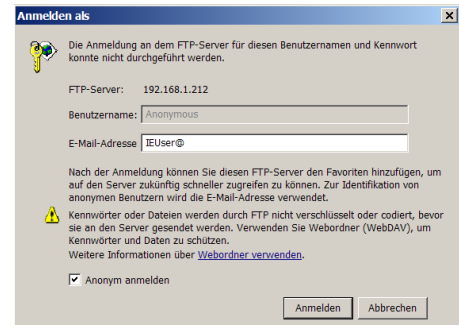
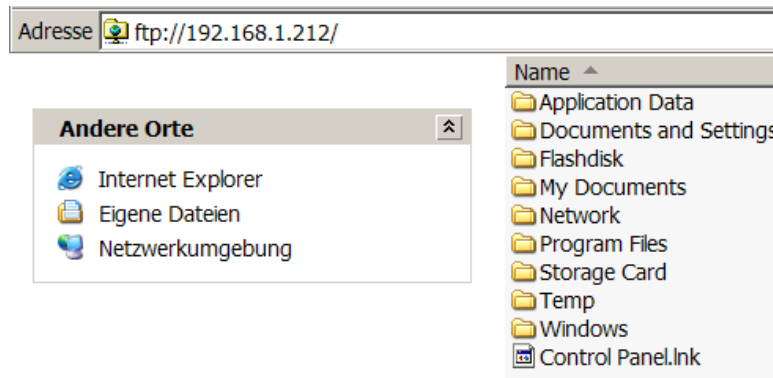
1.10 Anwendungen

Es können unter dem Betriebssystem μ E keine zusätzliche Anwendungen gestartet werden.

1.11 Datenaustausch

1.11.1 Autoscan-Verbindung vorbereiten

1.11.2 Eine ftp-Verbindung herstellen



Anmeldung Anonym

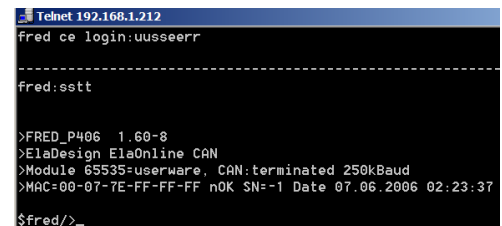
1.11.3 Eine Telnet-Verbindung herstellen

Hierzu muss die Runtime laufen!

Standard-Benutzerpasswort: user

Detaillierte Informationen finden Sie im Kapitel Telnet.

Tipp: Sie können auch mit dem Hyperterminal eine Telnet Verbindung herstellen.



1.12 Betriebssystem Einstellungen

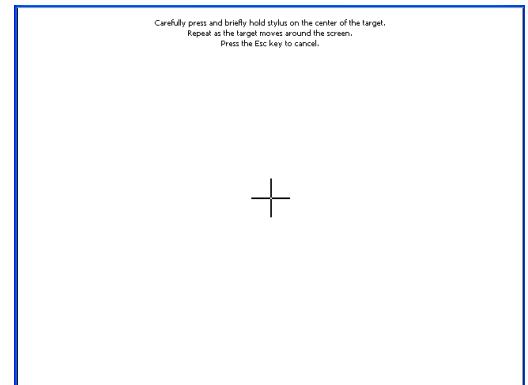
1.12.1 Speicheraufteilung

1.12.2 Kalibrierung des Touches

Der Touchscreen ist im Auslieferungszustand bereits kalibriert.
Auf Wunsch kann der Touch aber erneut kalibriert werden.

Hierzu dient Telnet. Siehe unter „Optionen“ das entsprechende Kommando.

Folgen Sie den Anweisungen, in dem Sie in mehreren Schritten jeweils in die Mitte des Fadenkreuzes tippen. Mit der negativen Flanke, also den losen des Touchstiftes von Kreuz, spring das Kreuz auf seine jeweilige nächste Position.



1.12.3 Hinweise zum FileSystem

Grenzen des FileIO

Das System wurde mit folgenden Parametern erfolgreich verprobt

Anzahl der Dateien:

- 100 Dateien in Root
- 20 Dateien in Verzeichnissen

Dateinamen:

- 8+3 Zeichen Standard
- max. 27 Zeichen + 3 stellige Dateierdung (z.B.: .txt), jedoch bei Direktzugriff in Fremdgerät ist chkdisk /f nötig.
-

Verzeichnisnamen:

- Max. (partizielle) Länge des Verzeichnisnamen: 80 inkl. des Dateinamen mit Endung.

Anzahl der Verzeichnisse:

- 10 Verzeichnisse

Dateisystem:

FAT



- 1.) Nach dem entfernen der Dateien von der CF- Karte MUSS diese formatiert werden, da das PLC- Dateisystem sonst die gelöschten Dateien noch als existent behandelt.
- 2.) Grenzverletzungen unter FileIO

Dateiname zu lang: (mehr als 27 Zeichen)

Das elrest control System kürzt den Dateinamen auf 8+3 Zeichen.

HIERBEI MUSS VOR DEM STECKEN DER KARTE IN FREMDGERÄTEN EINE DER NACHFOLGENDEN SCHRITTE UNBEDINGT ERFOLGEN UM DIE DATENINTEGRITÄT ZU ERHALTEN.

Ausführen des Befehles CF -4 am Gerät unter Telnet

Ausführen von CHKDSK /F unter MSWindows

Verzeichnisnamen zu lang.

Das Filesystem der CF- Karte wird irreversibel geschädigt, die CF- Karte muss neu formatiert werden.

Zu viele Daten gespeichert:

CoDeSys führt Schreibbefehl aus, Dateien sind aber nicht auf Karte vorhanden.

1.13 Preemptives Multitasking

Warum Taskkonfiguration?

An vielen Anlagen ist eine sichere und kontinuierliche Steuer- und Programmarbeit unabdingbar.

Daher werden Aufgaben, wie z.B. eine Betriebsdatenerfassung, das Überprüfen der Systembatterie oder das Aktualisieren der HMI in niederpriorigen Tasks angelegt. Diese Aufgaben sind wenig zeitkritisch und erlauben daher eine Bearbeitung im Hintergrund. Das Hauptprogramm wird somit in der Abarbeitung nicht gestört.

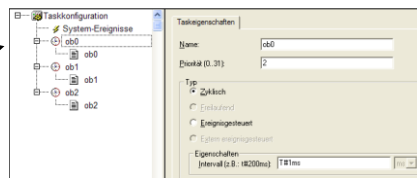
1.13.1 Task Konfiguration

Falls keine Taskkonfiguration unter CoDeSys V2.3 angewählt wurde, wird der PLC_PRG Programmbaustein aus dem Thread „ob2“ gerufen. Die Threadzeiten können dann ausschließlich über Telnet eingestellt werden

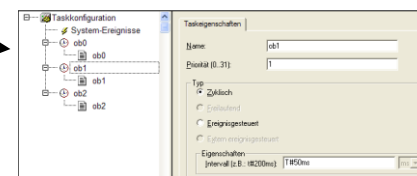
Wird die Taskkonfiguration unter CoDeSys verwendet ist eine fixe Zuordnung gegeben :

Unter μ E-Geräten können bis zu 3 Tasks angelegt werden. Hierbei entspricht 0 der höchsten Priorität, 2 der Niedersten. Beispiel:

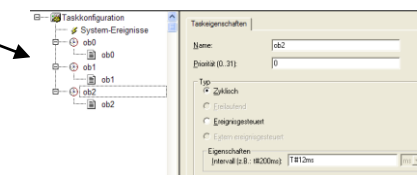
OB0
|
___ Priorität 2



OB1
|
___ Priorität 1



OB2
|
___ Priorität 0



Seit der CoDeSys- Version V2.3.6.x kann der nederpriorie Task (OB0) als freilaufend definiert werden. Dies bedeutet, daß OB0 sowohl Aufgrund seiner Priorität, wie auch seiner Typeeigenschaft immer dann aufgerufen wird, wenn alle anderen Tasks nicht bearbeitet werden.

higher as OB:2

Thread „TCPIP“

Option9 0-1

Thread „ob2“

higher as OB:1

Thread „TCPIP“

Thread „ob1“

higher as OB:0

Thread „TCPIP“

Thread „ob0“

equal as OB:2

Option9 3-5

Thread „ob2“ Thread „TCPIP“

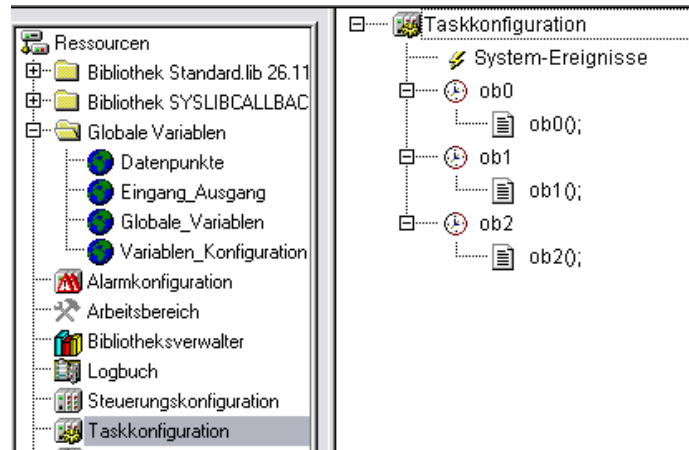
equal as OB:1

Thread „ob1“ Thread „TCPIP“

equal as OB:0

Thread „ob0“ Thread „TCPIP“

Die Threads müssen in CoDeSys wie folgt definiert werden.



Die Threads können unter Telnet mit dem Befehl „thread“ wie folgt in Ihren Zyklen kontrolliert werden

```
$fred/>thread
Threads:      ob0      ob1      ob2(C0)  TcpIp    CAN
Period /ms   endless  100      10       20       10
Max           1        1        0        27       0
Gap           0        28       0        20       51400
Count        49123   282     2818    1409     0
```

1.13.2 Empfohlene Task Konfiguration

Ihre Applikation sollte mit hoher Priorität arbeiten, benötigt aber auch zur Steuerung Schnittstellen. Diese Schnittstellen benötigen eine höhere Priorität als die Applikation.

- CAN
Alleinig der CanController arbeitet auf einer höheren Prio als CoDeSys Prio.
Benötigen sie ausschließlich CAN, so können Sie ihre Applikation unter CoDeSys Prio 0 betreiben
- Ethernet, RS.
Benötigen Sie eine dieser Schnittstellen, empfiehlt es sich die Applikation ab CoDeSys- Prio 0 zu betreiben.
- FileIO
Das Schreiben und Lesen einer Datei wird über das µE- Betriebssystem geregelt und kann daher, je nach Speicherort (Flaskdisk, CF) und Datenmenge unterschiedlich viel Zeit in Anspruch nehmen. Diese Zeit kann äußerst unterschiedlich sein und bis zu mehreren Sekunden dauern. Da weder Ihr Maintask, noch ihr Visutask für mehrere Sekunden angehalten werden darf, sollte der Filetask - um z.B. ein Protokoll zu erstellen – auf Priorität 2 stehen.
Kopieren Sie die benötigten Variablen vom Maintask, um die diesen Task und starten Sie die Bearbeitung an.
Die vom System und der Applikation angelegten Task können sie mit dem Telnetbehehl <Threath> beobachten.
Mehr hierzu finden Sie unter der Rubrik: [Telnet → Threads](#)
- MAINTASK
In Ihrem Maintask steht Ihre Applikation. Diese umfasst die Verarbeitung von Ein-und Ausgängen und die Kommunikation mit diversen anderen Geräten. Da Sie hier auf eine hochpriore Aktualisierung aller Schnittstellen angewiesen sind um Ihre zu verarbeitenden Signale zu erhalten, wählen Sie bitte wie folgt:
Zyklisch, T#XYZms, Prio <1>
- VISUTASK
In den Visutask gehören in der Regel alle Codezeilen, die sich mit der HMI beschäftigen. Hierzu gehören: Abfrage von Tasten, Bildpositionen, Maskennummern etc. weiterhin sind alle Befehle der Pultbibliothek zwingend auf dieser Ebene anzuwenden wie z.B. PultMaskOpen, DrawBitmap... etc..
Wählen Sie daher:
Zyklisch, T#XYZms, Prio <2>

Telnet

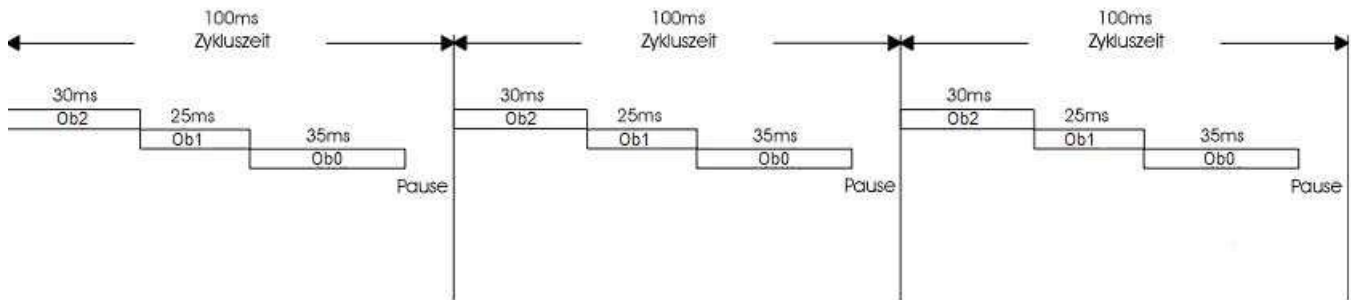
```
CM211/>t
> Memory Free=      25500 from      37856[kB]
> Threads-----|-----|-----Period [ms]-----|----- Duration [ms]-----|Load|-----
>      Set: [no]      |      Count|      Min   Cur.   Max|      Min   Avg   Max| %   |OS-Prio
OB00      endless|      2606|      9.83  9.83  28.95|      0.00  0.52  6.32| 5.2|250
OB01      : 1      10|      2675|      8.94  9.92  28.95|      0.31  0.32  0.77| 3.3| 32
IOs      : 3      10|      2676|      8.97  9.93  28.95|      0.23  0.23  0.45| 2.3| 14
OB1314    1(fixed)|      26759|      0.04  1.00  11.35|      0.01  0.02  0.27| 2.1| 15
CAN Int   interrupt|           0|      0.00  0.00  0.00|      0.00  0.00  0.00| 0.0| 10
CAN0/1    1(fixed)|      26751|      0.05  0.98  11.09|      0.01  0.02  0.17| 2.2| 13
> CoDeSys-Threads-----|-----|-----|-----|-----|-----|-----
DefaultTas      10|      2675|      8.87  9.92  29.02|      0.04  0.05  0.32| 0.5|200

threads 0 = thread info
threads 1 <value> set new value of OB01 = 10 [ms]
threads 3 <value> set new value of IOs = 10 [ms]
threads 10 = show cycle time perodic
threads 11 = disable profiler
```

1.13.3 Task Konfiguration im ungestörten Betrieb

Falls die Zykluszeit, wie im Beispiel 100 ms, größer gewählt wird, als die Summe **der Einzeittaskzeiten** werden die einzelnen Programmbausteine ohne Echtzeitfehler abgearbeitet. Wir sprechen von einem ungestörten Betrieb.

Ungestörter Betrieb

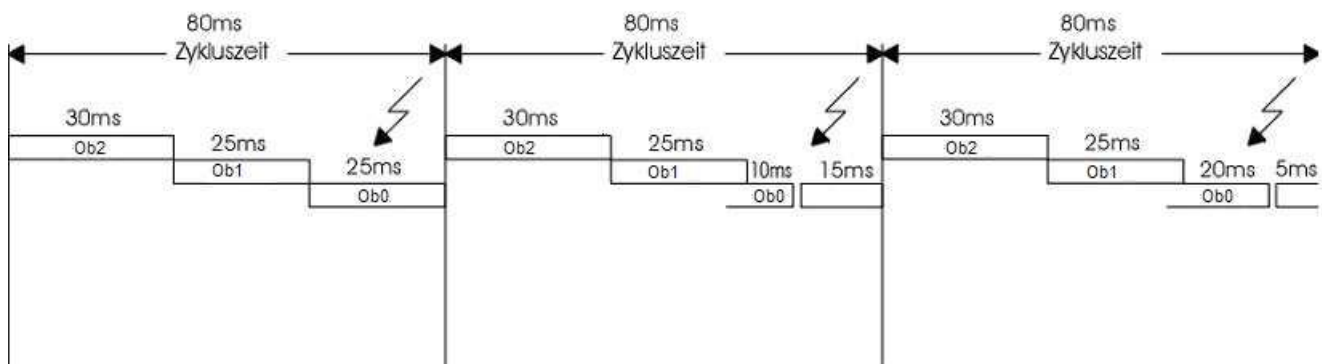


1.13.4 Task Konfiguration mit Echtzeitfehlern

Wenn gegenüber dem Beispiel für den ungestörten Betrieb innerhalb des Programmes ein Thread durch einen wesentlich längeren ersetzt wird oder die Zykluszeit verkürzt wird, kann im Beispiel OB0 nicht vollständig abgearbeitet werden.

Der Thread wird unterbrochen und beim nächsten Durchlauf des Programms an der Unterbrechung fortgesetzt. Nach der Abarbeitung des verbleibenden Ob-Resets wird derselbe Thread erneut gestartet und läuft bis zur Unterbrechung am Ende der Zykluszeit weiter.

Gestörter Betrieb



Auch im gestörten Betrieb ist die Abarbeitung des Ob2 und OB1 gewährleistet.

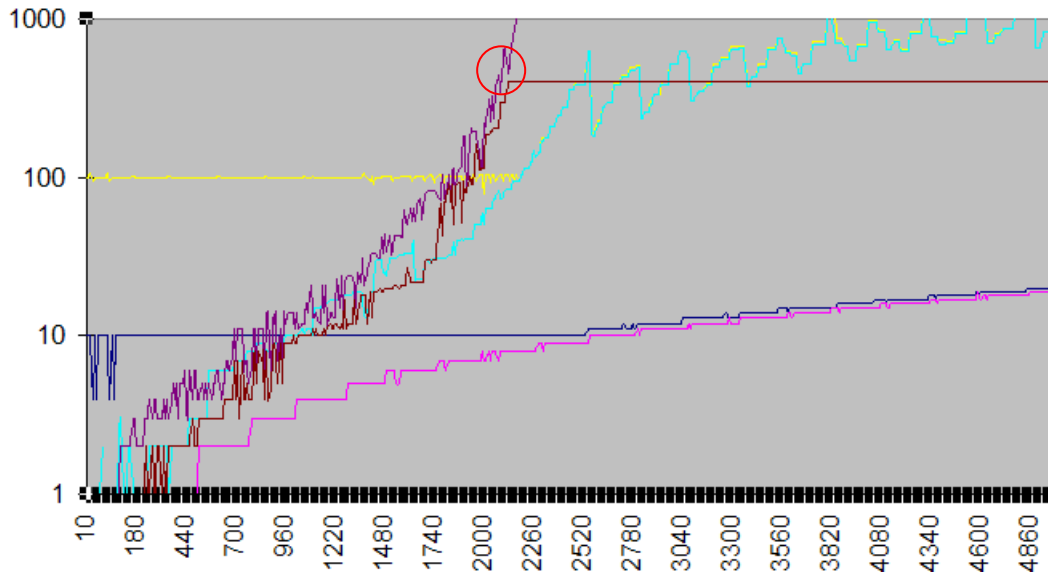
Im vorliegenden Beispiel wird auch der Ob0 abgearbeitet, wenn auch zu einem späteren Zeitpunkt. Wäre jedoch noch ein Ob vorhanden, so würde dieser nie zur Ausführung kommen.



Hilfe bei Echtzeitverletzungen

Als Gegenmaßnahme bei gemeldeten Echtzeitverletzungen kann die Zykluszeit des höher priorien Tasks vergrößert werden.

Threads:	ob0	ob1	ob2(CO)	Tcplp	ESB
Period /ms	endless	100	10 (act 10)	10	10 (act 10)
Max	1	0	0	33	0
Gap	1	72	2	10	6
Count	3884	10	103	103	105



Darstellung der Zykluszeit und –dauer in Abhängigkeit der Programmgröße

Im obigen Bild ist die OB2-Zykluszeit (blaue Linie) so lange stabil auf der eingestellten Zykluszeit 10ms, bis der Code in OB2 so groß ist, dass er für die Abarbeitung ebenfalls 10ms benötigt (violett). Ab diesem Moment steigen bei weiter wachsender Codegröße die OB2- Zykluszeit und die Aufrufdauer gemeinsam an.

Die niedrigeren Tasks OB1/OB0 weisen hierbei ein extremeres Kurvenverhalten auf, da Sie durch den Hochprioreren OB2 entsprechend öfters in ihrer Arbeit unterbrochen werden. Dies drückt sich in einer schnell anwachsenden Abarbeitungsdauer aus (Türkis=OB1, dkviolett:= freilaufender OB0)

Durch einen roten Kreis markiert ist der Zeitpunkt, ab welchem OB0 nicht mehr bedienbar ist, d.h. Die HMI in Form von Visualisierung, Visioweb oder Remotepanel kann nicht mehr aktualisiert werden –OB2 und die Ein- und Ausgänge arbeiten jedoch Hochprior weiter.

Legende:

X-Achse: Rechenoperationen in OB2

Y-Achse: Zeit in ms, beginnend mit OB2:10ms, OB1:100ms, OB0:freilaufend

Blau: OB2 Zyklus

Violett: OB2 Aufrufdauer

Gelb: OB1 Zyklus

hblau: OB1 Aufrufdauer

1.13.5 Unterbrechung von Task's

Der laufende Task kann an jeder Stelle von einem höher prioren Task unterbrochen werden. Die Fortführung des Task's erfolgt zu einem späteren Zeitpunkt an der Stelle, an der die Unterbrechung stattfand.



Da es sich um ein preemptives Multitasking System handelt, werden die Ein- und Ausgänge in einem separaten Thread „IO“ mit höherer Priorität als alle Anwender Threads bearbeitet. **Wenn die Zykluszeit dieses Threads kleiner größer ist als der Anwender Threads, stehen nicht bei jedem Zyklus neu eingelesen Ein- und Ausgangsdaten zu Verfügung.** Dies geschieht unabhängig davon, ob sich das System im ungestörten oder gestörten Betrieb befindet.

2 Kommunikation

2.1 RS232

2.1.1 Low-Level RS232 oder RS485

2.1.2 RTU-Modbus über RS232 oder RS485



Im Dokument „[Kommunikation_RS232_DE.pdf](#)“ werden die einzelnen Möglichkeiten ausführlich beschrieben.

2.2 CAN

2.2.1 Low-Level CAN

Die CAN- Kommunikation erfolgt analog zu den bisherigen VisioControl -Systemen mit CAN03.lib.

Folgendes ist jedoch, zu berücksichtigen, wenn ein Projekt von Can03 auf 04 oder 05 umgestellt wird.

- Der Befehl CanInit() heißt nun CanInitialize();
- Für Firmwarestruktur-Zugriffe muss sfW04 oder höher benutzt werden.
- Bei Firmwarezugriff auf CAN hat sich der Name geändert.

Hintergrund:

War bei vorhergehenden VisioControl- Geräten wie P101/300/305 etc. nur eine vollwertige Can-Schnittstelle vorhanden, so befinden sich auf P400 deren zwei. Dies hatte zur Folge dass die Firmwarestruktur angepasst werden musste.



```
nCheckRecvID := psFW^.uRemanent.sMasterEEProm.nCan0BaseIntermodulID +  
INT_TO_DWORD(psFW^.uRemanent.sMasterEEProm.nCan0ClientNo);
```

ST

2.2.2 *ElaCAN*

Dies ist ein für die ElaCAN - Gerätereihe entwickeltes CAN Protokoll



In den Dokumenten „[Kommunikation_CAN_DE.pdf](#)“ wird dies ausführlich beschrieben.

2.2.3 *ESB*

Dies ist ein auf CANopen basiertes Protokoll, welches um die automatische Erkennung von Slave Teilnehmer und Verwaltung von dynamischen PDOs erweitert wurde.



In den Dokumenten „[Kommunikation_ESB_DE.pdf](#)“ wird dies ausführlich beschrieben.

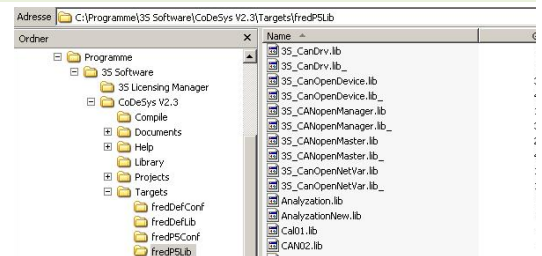
2.2.4 CANopen von 3S

Hier kann in dem Programmiersystemen CoDeSys sowohl CANopen Master als auch CANopen Slave projiziert werden.



In den Dokumenten „[Kommunikation CANopen DE.pdf](#)“ wird dies ausführlich beschrieben.

Um mittels 3S- CanOpen eine Kommunikation herstellen zu können benötigen Sie ältere Bibliotheken. Die aktuellen 3S-Bibliotheken sind fehlerhaft und haben einen kontinuierlichen Verbindungsreset zur Folge. Benützen Sie die im Bild abgelichteten „älteren“ Dateien an Stelle der (hier mit einem Underline markierten) Originaldateien.



Die Kommunikation erfolgt nun wie in anderen Systemen



Bei Verwendung von CanOpen-Bibliotheken aus CoDeSys werden die Aufrufe nach folgendem Schema generiert:

Gibt es keinen Modulparameter des CanMasters UpdateTask dann wird der Aufruf des CanMasters in der alphabetisch ersten Task erzeugt. Ansonsten gibt dieser Modulparameter den Namen der Task an, in der der CanMaster aufgerufen wird.

Die Rx- und Tx-PDOs werden jeweils in der höchsten Priorität aller Tasks, die das PDO referenzieren, aufgerufen.

Wenn PDOs oder der Masteraufruf nach obigem Schema in einer Ereignistask aufgerufen werden, wird eine Warnung ausgegeben.

Um PDO-Aufrufe in eine andere Task zu verlagern, müssen IO-Referenzen in diese Task verschoben werden, um den Masteraufruf zu verschieben, muss der Modulparameter UpdateTask verändert werden, oder eine andere Task über Namensänderung zur alphabetisch Ersten gemacht werden.

Die Aufrufe an die CanDevice-Bibliothek werden alle in der im CanDevice angegebenen Task erzeugt.



Weiterführende Informationen zur Verwendung der CanOpen-Bibliotheken unter CoDeSys sind der Dokumentation „[Kommunikation CANopen DE.pdf](#)“ zu entnehmen..

2.2.5 Änderung unter 3S Steuerungskonfiguration

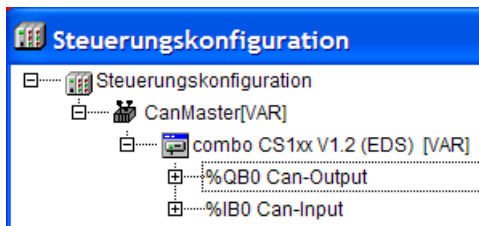


Für Version 2.0 muss die Steuerungskonfiguration neu aufgebaut werden.

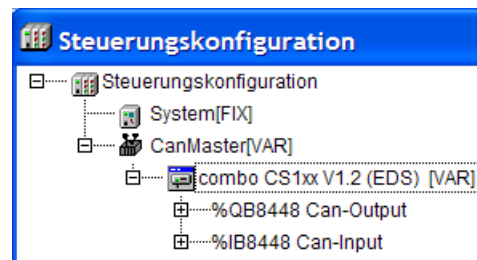
Eine Übernahme aus einer älteren Version ist nicht möglich.

Möchten Sie von V1 auf V2 migrieren, empfiehlt es sich beide Steuerungskonfigurationen mittels 2 Programmen offen zu halten um die V2 nachbilden zu können.

V1 (alt)



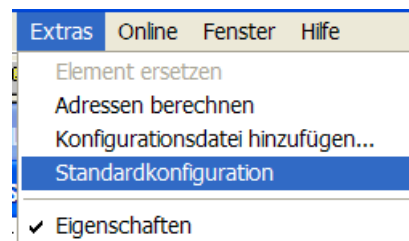
V2 (neu)



Es wird beim einbinden eines CanMaster automatisch ein Block mit System[Fix] eingetragen.
Die Adresse verschiebt sich somit auf 8448dez

Sollte dies nicht automatisch erfolgen, so wählen Sie bitte

Extras → Standardkonfiguration



2.2.6 Erweiterter Retainspeicher 512kb

Ab der Runtimeversion 2.02.xx stehen dem Benutzer unter CoDeSys mehr Retainspeicher zur Verfügung. Hierzu wird...

- 1.) Targetversion: >=Version 2 Stand 12. z.B. P2xx (CP-035, CP-057) V2.3.12 (ARM9) RT >= 2.0
- 2.) Runtime-Version >=2.03-01 Aug 11 2010
- 3.) WinCE >=6.18 oder manuelle Ramtreiberinstallation

Hierzu ist unter Zielsystemeinstellung folgender Eintrag notwendig:

Retain: Statt 1FFF8hex (131.064d) wird der Wert 80000hex (524.288d) eingetragen.

Zielsystem Einstellungen

Konfiguration: P2xx (CP-035, CP-057) V2.3.12 (ARM9) RT >= 2.0

Zielplattform | Speicheraufteilung | **Allgemein** | Netzfunktionen | Visualisierung

	Basis	Größe	Bereich
Code :	<input checked="" type="checkbox"/> Automatisch	16#100000	1
Global :	<input checked="" type="checkbox"/> Automatisch	16#10000 pro Segment	4
Memory :	<input checked="" type="checkbox"/> Automatisch	16#20000	4
Input :	<input checked="" type="checkbox"/> Automatisch	16#3100	4
Output :	<input checked="" type="checkbox"/> Automatisch	16#3100	4
Retain :	<input checked="" type="checkbox"/> Automatisch	16#80000	2

Eigenes Retainsegment

Größe des gesamten Datenspeichers: 16#8000

Maximale Anzahl von Bausteinen: 1024

Maximale Anzahl der Segmente globaler Daten: 16

Voreinstellung OK Abbrechen

3 Applikationen

Dieses Zusatztool ist ausschliesslich für die ElaSIM Programmierung.

Datapoint2C erzeugt eine Datenstruktur aus der Datenpunktliste für die Programmierung in C bzw. als Referenz für einen Zugriff auf die Daten über z.B. ModBus-TCP.

Die Datenpunkte müssen innerhalb eines Projektes in der Projektverwaltung erzeugt worden sein (siehe Kapitel Projektverwaltung). Das System speichert diese Datenpunktinformationen in einer Datei datapkt.set. Die Datenpunkte werden dabei getrennt in „globale selbstdefinierte Datenpunkte“, für indizierte Datenpunkte und für die Verwendung in CoDeSys, und in „selbstdefinierte Datenpunkte“, für Skalare (siehe Kapitel Datenpunkte).

Durch Ausführen des Programmes Datapoint2C.exe wird diese Liste in eine Struktur für weitere Verwendungen umgesetzt. Die Datei Datapoint2C.exe befindet sich im Verzeichnis von ElaSoft\Bin und wird mit zwei Übergabeparametern ausgeführt.

Übergabeparametern 1: Datenpunktdatei datapkt.set

Übergabeparametern 2: Datenstruktur mit beliebiger Bezeichnung

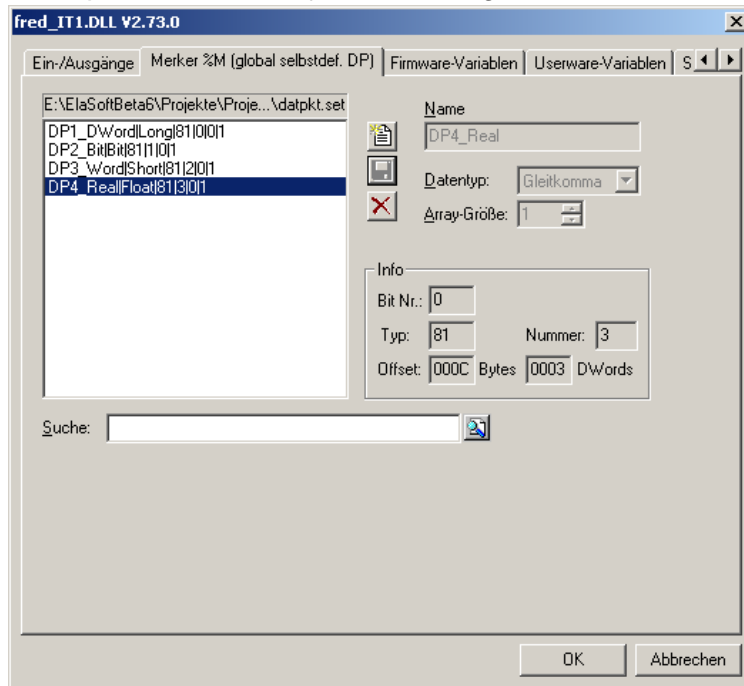
Beispiel:

*D:\Programme\ElaSoft\Bin\Datapoint2C.exe D:\Projekte\ErstesProjekt\ P300\datapkt.set
D:\Datenstrukturstruktur.h*

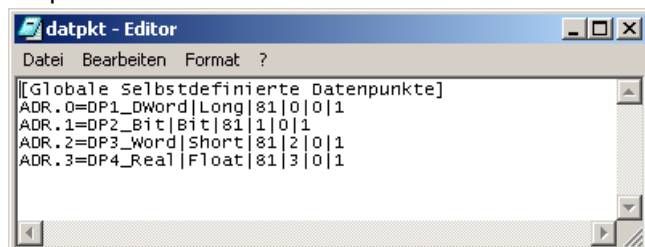
Hinweis:

Die Verzeichnisangaben in den Übergabeparametern dürfen keine Leer- bzw. Sonderzeichen enthalten.

Datenpunkte in der Projektverwaltung:



Datpkt.set:



Datenstruktur.h:

```
Datenstruktur.h - Editor
Datei Bearbeiten Format ?
typedef struct
{
    signed long      DP1_Word; // %MD0(0hex)
    struct
    {
        unsigned char    DP2_Bit : 1;
        unsigned char    nDummy1 : 1;
        unsigned char    nDummy2 : 1;
        unsigned char    nDummy3 : 1;
        unsigned char    nDummy4 : 1;
        unsigned char    nDummy5 : 1;
        unsigned char    nDummy6 : 1;
        unsigned char    nDummy7 : 1;
    } Bits1; // %MD1(1hex)
    struct
    {
        unsigned char    nDummy8 : 1;
        unsigned char    nDummy9 : 1;
        unsigned char    nDummy10 : 1;
        unsigned char    nDummy11 : 1;
        unsigned char    nDummy12 : 1;
        unsigned char    nDummy13 : 1;
        unsigned char    nDummy14 : 1;
        unsigned char    nDummy15 : 1;
    } Bits2; // %MD1(1hex)
    struct
    {
        unsigned char    nDummy16 : 1;
        unsigned char    nDummy17 : 1;
        unsigned char    nDummy18 : 1;
        unsigned char    nDummy19 : 1;
        unsigned char    nDummy20 : 1;
        unsigned char    nDummy21 : 1;
        unsigned char    nDummy22 : 1;
        unsigned char    nDummy23 : 1;
    } Bits3; // %MD1(1hex)
    struct
    {
        unsigned char    nDummy24 : 1;
        unsigned char    nDummy25 : 1;
        unsigned char    nDummy26 : 1;
        unsigned char    nDummy27 : 1;
        unsigned char    nDummy28 : 1;
        unsigned char    nDummy29 : 1;
        unsigned char    nDummy30 : 1;
        unsigned char    nDummy31 : 1;
    } Bits4; // %MD1(1hex)
    signed short      DP3_Word; // %MD2(2hex)
    signed short      nDummy32[1];
    float             DP4_Real; // %MD3(3hex)
} tCstMemory;
tCstMemory _HUGE* psCstMemory = &IO.CstMemory[0];
```

4 Benchmark

4.1 Threading

Globale Variable :



```
VAR_GLOBAL
```

```
    tTime_1: ARRAY[0..2] OF TIME;
```

```
ST      END_VAR
```

Programm im ob0 :



ST

```
VAR
    i: INT;
    rVar: REAL;
END_VAR

CycleTime[0] := TIME_TO_DWORD(TIME()-tTime_1[0]);
tTime_1[0] := TIME();
(* begin *)
FOR i:=0 TO 1000 DO
    rVar := rVar * 2.0;
END_FOR
Digital_Output_00_00 := NOT Digital_Output_00_00;
(* end *)

Run_Time[0] := TIME_TO_DWORD(TIME()-tTime_1[0]);
```

Programm im ob1 :



ST

```
VAR
    i: INT;
    rVar: REAL;
END_VAR

CycleTime[1] := TIME_TO_DWORD(TIME()-tTime_1[1]);
tTime_1[1] := TIME();
(* begin *)
FOR i:=0 TO 200 DO
    rVar := rVar * 2.0;
END_FOR
Digital_Output_00_01 := NOT Digital_Output_00_01;

Run_Time[1] := TIME_TO_DWORD(TIME()-tTime_1[1]);
```

Programm im ob2 :



VAR

i: INT;

ST

rVar: REAL;

END_VAR

```
CycleTime[2] := TIME_TO_DWORD(TIME()-tTime_1[2]);
```

```
tTime_1[2] := TIME();
```

```
(* begin *)
```

```
FOR i:=0 TO 100 DO
```

```
    rVar := rVar * 2.0;
```

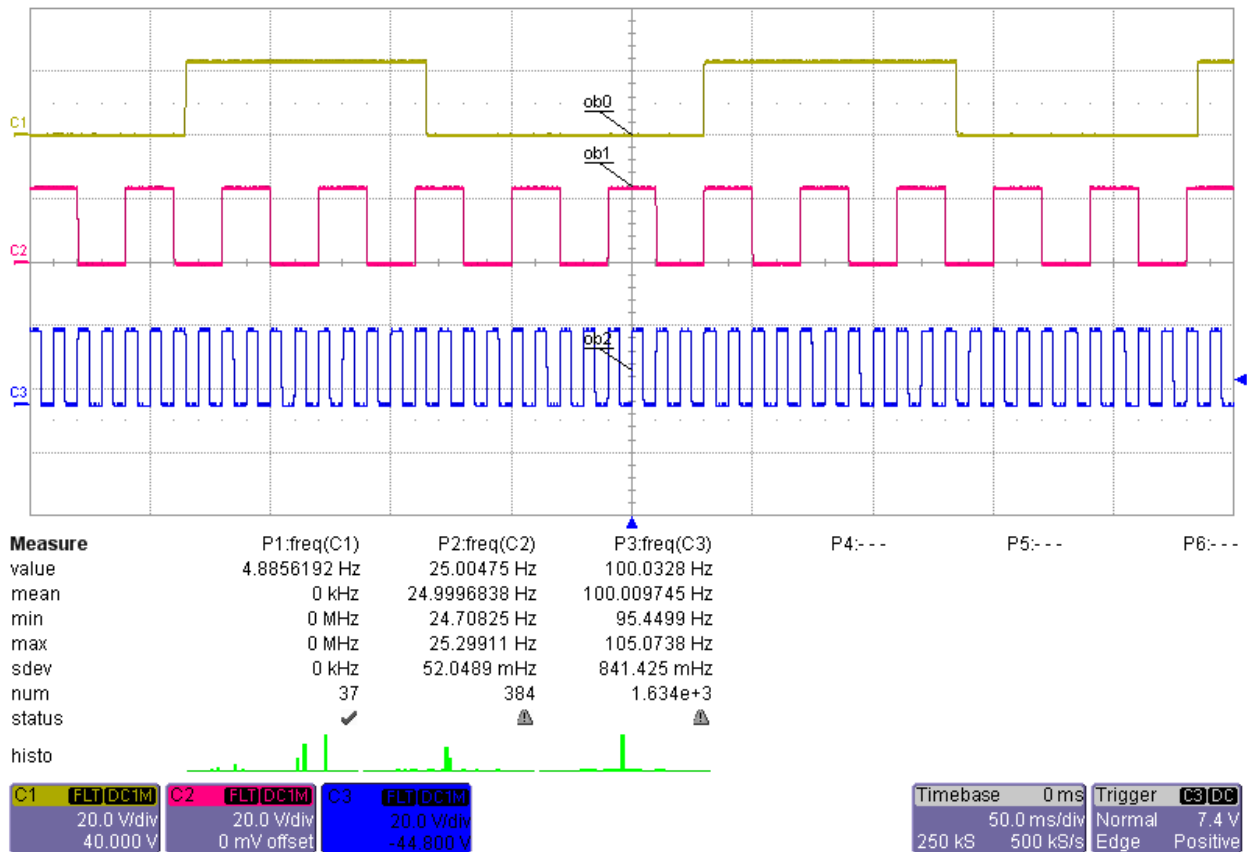
```
END_FOR
```

```
Digital_Output_00_02 := NOT Digital_Output_00_02;
```

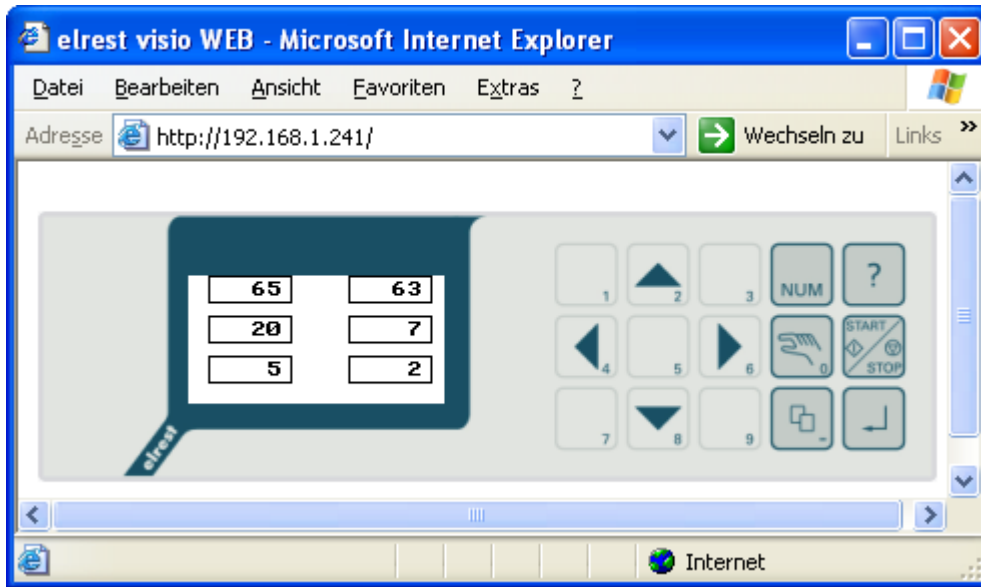
```
(* end *)
```

```
Run_Time[2] := TIME_TO_DWORD(TIME()-tTime_1[2]);
```

Der obige Programmcode wurde durchgeführt :



Mit Webvisualisierung werden die Zykluszeiten (1. Spalte) und Laufzeiten (2. Spalte) ausgegeben für ob0, ob1, ob2 (Zeile 1 .. 3) :



: freilaufend → Runtime → rest CPU ca. 27 %

: 20 ms → Runtime → 33 %

: 5 ms → Runtime → 40 % CPU Last

Thread	Zykluszeit [ms]	Laufzeit [ms] SRAM Waitstate“ : 0 FW 1.72-1	Realtime [Hz] = 500/Zykluszeit	Laufzeit [ms] SRAM Waitstate“ : 1 FW 1.72-1
ob0	freilaufend	60 .. 70 ms	--	100 ms
ob1	20 ms	7 ms	24.5 .. 25.5 Hz	10 ms
ob2	5 ms	2 ms	95..105 Hz	2 ms

5 Support

Hotline

Für zusätzliche Unterstützung und Informationen, können Sie unsere Hotline zu folgenden Zeiten:

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

Außerhalb dieser Zeiten, können Sie uns per e-mail oder fax erreichen:

Telefon: ++49 (0) 7021/92025-33

Fax: ++49 (0) 7021/92025-29

E-mail: hotline@elrest.de

Training und Workshops

Wir bieten Ausbildung oder projekt basierende Workshops zu elrest Produkte an.

Für weitere Informationen, kontaktieren Sie bitte unsere Vertriebsabteilung:

Telefon: ++49 (0) 7021/92025-0

Fax: ++49 (0) 7021/92025-29

E-mail: vertrieb@elrest.de

6 Historie

Datum	Name	Kapitel	Änderung
10.05.2006	Brendle	V2.81.1	Erstellen der Plattform
11.05.2006	Brendle	V2.81.2	Rezepte
11.05.2006	Brendle	V2.81.3	Threads
01.06.2006	Nething	V2.81.3a	Schreiben/Lesen Compactflash
02.06.2006	Brendle	V2.81.4	CF- Karte
21.08.2006	Brendle	V2.81.5	Logo geändert
22.08.2006	Brendle	V2.81.6	Alarme und Variablentypen in Arbeit
05.12.2006	Nething	V2.82.0	Dokumentenkorrektur + Remotepanel
14.12.2006	Brendle	V2.82.7	Überarbeitung der Plattform
14.12.2006	Brendle	V2.82.7	Retain, Persistant Konstant
14.12.2006	Brendle	V2.82.7	Alarme
23.01.2007	Brendle	V2.82.7	CF Einstellungen Enable CF-FW
03.05.2007	Nething	V2.82.10	X- Befehle unter P303
04.07.2007	Kramer	V2.82.14	Benchmark
13.07.2007	Nething	V2.82.14	Taskkonfiguration
26.07.2007	Kramer	V2.82	Stand zu Release V2.82
11.01.2008	Nething	V2.82	Erweiterung Telnetoptionen, Variablen
14.03.2008	Nething	V2.82	Sprachumschaltung
31.03.2008	Prodana	V2.83	Telnet, Option 27, 34, RemotePanel-COB_IDs
12.12.2008	Prodana	V2.83	Telnet, CAN-Optionen 33, 34; RS232 Option 6
19.12.2008	Hitzelberger	V2.83	CAN-Optionen umgearbeitet
27.10.2010	Nething	V2.84	Telnet- Einstellungen, Korrektur Multitask
25.8.2011	Nething	V2.84	Persistant, Telnet
20.02.2012	G. Schauer	V2.91	Aktualisieren und Abgleichen mit CE-Geräten
15.5.2012	Nething	V2.91.1	Gesamt-Überarbeitung, Löschen von ElaDesign-Inhalten
25.6.2012	Nething	V2.91 RC1	Korrektur Telnetbefehle Treath, RS232

© 2012 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.