

CoDeSys LFZ und Library

**Dok-Rev. 1.4 vom 30.04.2003
Software-Rev. 2.3 vom 01.03.2003**

Inhaltsverzeichnis

1	Urheberrecht und Haftung	3
1.1	Handhabung	3
1.2	Erklärung	3
2	Installation von CoDeSys	4
2.1	Installation der Oberfläche	4
2.2	Die Datei CODESYS.INI	4
3	CoDeIEP Laufzeitsystem	5
3.1	Starten des Laufzeitsystemes unter RTOS-UH	5
3.2	Prinzipielle Funktionsweise des Laufzeitsystemes	5
3.2.1	Laden eines Anwenderprogrammes	5
3.2.2	Ablauf eines Anwenderprogrammes	5
3.3	Optionen des CoDeIEP's Laufzeitsystems	6
3.4	Fehlermeldungen des CoDeIEP's Laufzeitsystems	8
4	Retain-Variablen	10
4.1	Anlegen von Retain-Variablen	10
4.1.1	Beispiel für Retain-Variablen	10
5	Librarys	11
5.1	I/O-Lib	11
6	Problembehandlung	15

Revisionsliste:

Rev.	Datum	Na.	Änderung
1.0	28.06.2001	Ko	Erstellung
1.1	03.08.2001	Ko	Hinweis auf FPU Einstellungen, Retain-Variablen ergänzt
1.2	11.01.2002	Ha	Installation Hilfesystem detailliert
1.3	30.05.2002	Ko	Kapitel Problembehandlung ergänzt
1.4	23.05.2003	Kr	Update auf CoDeSys V2.3

1 Urheberrecht und Haftung

Alle Rechte an diesen Unterlagen liegen bei der IEP GmbH, Langenhagen.

Die Vervielfältigung, auch auszugsweise, ist nur mit unserer ausdrücklichen schriftlichen Genehmigung zulässig.

In Verbindung mit dem Kauf von Software erwirbt der Käufer einfaches, nicht übertragbares Nutzungsrecht. Dieses Recht zur Nutzung bezieht sich ausschließlich darauf, daß dieses Produkt auf oder in Zusammenhang mit jeweils **einem** Computer zu benutzen ist. Das Erstellen einer Kopie ist ausschließlich zu Archivierungszwecken unter Aufsicht des Käufers oder seines Beauftragten zulässig. Der Käufer haftet für Schäden, die sich aus der Verletzung seiner Sorgfaltspflicht ergeben, z.B. bei unautorisiertem Kopieren, unberechtigter Weitergabe der Software usw.. Der Käufer gibt mit dem Erwerb der Software seine Zustimmung zu den genannten Bedingungen. Bei unlizensiertem Kopieren muß vorbehaltlich einer endgültigen juristischen Klärung von Diebstahl ausgegangen werden. Dies gilt ebenso für Dokumentation und Software, die durch Modifikation aus Unterlagen und Programmen von IEP hervorgegangen ist, gleichgültig, ob die Änderungen als geringfügig oder erheblich anzusehen sind.

Eine Haftung seitens IEP für Schäden, die auf den Gebrauch von Software, Hardware oder Benutzung dieses Manuskriptes zurückzuführen sind, wird ausdrücklich ausgeschlossen, auch für den Fall fehlerhafter Software oder irrtümlicher Angaben.

Das Einverständnis des Käufers oder Nutzers für den Haftungsausschluß gilt mit dem Kauf und der Nutzung der Software und dieser Unterlagen als erteilt.

1.1 Handhabung

Lesen Sie bitte zuerst sorgfältig diese Dokumentation bevor Sie anfangen zu programmieren. Sie sparen Zeit und vermeiden Probleme.

1.2 Erklärung

Wir behalten uns das Recht vor, Änderungen, die einer Verbesserung der Schaltung oder des Produktes dienen, ohne besondere Hinweise vorzunehmen. Trotz sorgfältiger Kontrolle kann für die Richtigkeit der hier gegebenen Daten, Schaltpläne, Programme und Beschreibungen keine Haftung übernommen werden. Die Eignung des Produktes für einen bestimmten Einsatzzweck wird nicht zugesichert.

2 Installation von CoDeSys

2.1 Installation der Oberfläche

Auf der CD befindet sich das Programm `setup.exe`, das den weiteren Ablauf der Installation automatisiert. Ist die Autostart-Funktion des CD-Laufwerkes aktiv, brauchen Sie nichts zu tun, außer nach dem Einlegen der CD kurz warten.

Es werden automatisch die für den Dongle nötigen Dateien auf Ihrem Computer installiert. Der Dongle ist nötig, damit Sie Verbindung zum RTOS-UH Zielsystem aufnehmen können. Ohne Dongle läuft CoDeIEP im Demo-Mode, damit ist nur die Simulation des Programmes auf dem PC möglich, aber keine Verbindung zum Zielsystem.

Im weiteren Verlauf der Installation ist das Installationsverzeichnis anzugeben oder das vorgeschlagene Verzeichnis zu übernehmen (z.B. "C:\Programme\IEP").

Alle weiteren Einzelheiten entnehmen Sie bitte den Handbuch, dieses wird im Installationsverzeichnis unter Dokumenten abgelegt.

2.2 Die Datei CODESYS.INI

Die Datei `CODESYS.INI` liegt im `INSTALLIONS`-Verzeichnis. In ihr können die folgenden Einträge mit einem Texteditor (z.B. Notepad) angelegt bzw. geändert werden:

UpdateRate=200

Update Rate der Oberfläche für das Monitoring der Variablen. Die Zeit wird in msec angegeben. Der Default-Wert ist 200 msec. Die Oberfläche holt die Variablen vom Laufzeitsystem ab, d.h. wenn die Zeit zu kurz eingestellt wird, kann die Update Rate nicht eingehalten werden, es treten aber keine weiteren negativen Effekte auf.

TASK=YES

Bei IEC 61131 Multi-Tasking-Anwendungen wird automatisch die Task `CODESYS_EVENT_WATCH` generiert. Diese Task führt die physikalischen I/O's für alle IEC 61131 Tasks durch und kümmert sich um die Überwachung der IEC 61131 Events.

DefaultWaitTime=6600

Bei der Kommunikation ist dies der Wert für das Timeout der Oberfläche, d.h. die Oberfläche hat während der Timeout-Zeit keine Kommunikation mit dem Laufzeitsystem durchführen können. Die Zeit wird in msec angegeben. Der Default-Wert beträgt 6600 msec. Ist das Time-Out abgelaufen, führt die Oberfläche einen Logout durch.

Bei der Benutzung von großen `KONFIG.INI`-Dateien ist ein Wert von 10000 angebracht, da es sonst beim ersten Download des Projektes zu Logouts wegen Zeitüberschreitungen kommen kann.

Alle anderen Einträge werden automatisch generiert und dürfen nicht verändert werden.



3 CoDeIEP Laufzeitsystem

3.1 Starten des Laufzeitesystems unter RTOS-UH

Das Laufzeitsystem wird mit dem Bedienbefehl CODESYS gestartet. Die hierbei möglichen Optionen sind in Kapitel 3.3 beschrieben.

3.2 Prinzipielle Funktionsweise des Laufzeitesystems

Das Laufzeitsystem dient der Kommunikation zwischen der Programmieroberfläche auf dem PC und dem PLC-Anwendungsprogramm unter RTOS-UH. Um seine Aufgabe erfüllen zu können, bildet es unter RTOS-UH mehrere Tasks:

SPS_PRG	exekutiert das Anwendungsprogramm.
TCP_CODESYS	Kommunikator für TCP-Verbindungen.
ComCode	exekutiert Befehle der Programmieroberfläche.
<i>DebugTasks</i>	dienen Debug- und Traceaufgaben.

3.2.1 Laden eines Anwenderprogrammes

Ein vom Anwender geschriebenes Programm wird folgendermaßen behandelt: Die Task ComCode nimmt die Daten der Oberfläche entgegen und bereitet sie für die Ausführung vor. Dabei werden auch die physikalischen I/O-Adressen und Accessvariablen aufgelöst. Danach wird der Task SPS_PRG (bei einer Single Task Anwendung) der Funktionspointer des Anwenderprogrammes übergeben und von dieser der Code zyklisch ausgeführt.

3.2.2 Ablauf eines Anwenderprogrammes

Das Anwendungsprogramm wird immer in der folgenden Reihenfolge abgearbeitet:

- Einlesen der Eingänge (%I x Adressen und Accessvariablen)
- Setzen von Forcewerten
- Bearbeitung des Anwenderkodes
- Setzen der Forcewerte
- Ausgeben der Ausgänge (%Q x und Accessvariablen)

Bei einer Single-Task-Anwendung – ohne IEC 61131 Task – wird die Task SPS_PRG und damit Ihr Anwendungsprogramm mit maximaler Geschwindigkeit abgearbeitet. Es kann keine Zykluszeit vorgegeben werden. Ist eine bestimmte Zykluszeit erwünscht, muss in der Oberfläche eine IEC 61131 Task gebildet werden (siehe Taskkonfiguration). Hier kann eine Zykluszeit vorgegeben werden. Die kleinste einstellbare Zykluszeit ist 0,001 Sekunden (1 ms).

Da während der Abarbeitung des SPS-Zyklus parallel die Kommunikation mit der Oberfläche stattfindet, sind dargestellte Monitorwerte in der Oberfläche nicht immer zyklussynchron.

Nur im Sonderfall FLOW-CONTROL sind die Monitorwerte immer die des letzten durchlaufenen Zykluses.

Bei einer IEC 61131 Multitaskanwendung werden die einzelnen IEC 61131-Programme auf eigenständige Tasks im RTOS-UH abgebildet, sodass der Anwender – wie bei anderen Programmpackage-

ten (PEARL, CREST-C) auch – selbst für Datenkonsistenz sorgen muss. Die Tasks werden entsprechend ihrer Priorität vom System lauffähig gemacht, so dass eine höher priorisierte Task eine niedriger priorisierte **jederzeit** unterbrechen kann.

In einer IEC 61131 Multitasking-Anwendung werden die physikalischen I/O's von der zusätzlichen Task CODESYS_EVENT_WATCH betreut. Diese hat u.a. die Aufgabe, die 61131-Events zur Triggerrung der Starts der IEC 61131-Tasks zu verwalten. Damit diese Task automatisch bei jedem Projekt generiert wird, ist in der im WINDOW's-Verzeichnis liegenden Datei CODEIEP.INI der Eintrag TASK=YES zu ergänzen.

3.3 Optionen des CoDeIEP's Laufzeitsystems

Das CodeSys-Laufzeitsystem wird mit dem Bedienbefehl CODESYS gestartet.

Folgende Optionen sind möglich:

-A

Autostart des PLC_PRG ausschalten. Ihr Anwenderprogramm wird nach dem Einschalten **nicht** gestartet.

-B

Bootet das IEC 61131-Anwendungsprogramm aus dem EPROM. Um diese Option nutzen zu können, benötigen Sie ein spezielles Laufzeitsystem. Der Programmcode wird – nachdem er ausgetestet ist – auf dem PC vorbereitet und kann dann ins EPROM/FLASH gebrannt werden. Das Laufzeitsystem durchsucht beim Start die entsprechenden Bereiche. Findet es ein Programm im EPROM, so wird es aus dem EPROM heraus exekutiert, d.h. es wird kein Platz im RAM für den Programmcode belegt. Allerdings kann das Programm auch nicht mehr Online gesichert werden, es muß immer auf dem PC behandelt werden.

-C

Diese Option muß angegeben werden, wenn Sie mit der C-Code Generierung arbeiten, normalerweise bei PowerPC-Prozessoren.

-D [=xx]

Setzen des Debuglevel. Weglassen des Parameter oder D=0 disabled die Debugausgaben. Diesen Parameter sollten Sie nur setzen, wenn Sie von uns die entsprechende Anweisung haben.

-E

Ist diese Option gesetzt, werden Ausgaben des Laufzeitsystems über die Pipe /VO/LOGFILE ausgegeben. Normalerweise (ohne Umlenkung) werden Meldungen auf der Schnittstelle ausgegeben, von der das Laufzeitsystem gestartet wurde. Im Regelfall ist dies die /A1/.

-F [x]=file_name

Setzen des SaveFileNamen, es sollte immer der komplette Pfad angegeben werden. Hier wird das aktuelle Projekt zum Wiederanlauf nach einem Reset gespeichert. Für x sind Werte von 1 bis 3 zulässig.

x entfällt: Bei einem SAVE Programm von der Oberfläche aus wird das aktuell laufende Programm in die angegebene Datei gerettet. Während des Retten des Programms ist keine Kommunikation mit dem Laufzeitsystem möglich. Der Status der Operation – hat geklappt oder es ist ein Fehler aufgetreten – wird an die Oberfläche gemeldet.

x=1 Wie ohne Option, zusätzlich wird der RTOS-UH-Bedienbefehl SAVE_IEC1131 aufgerufen. Diesen Befehl können Sie als Anwender programmieren, um beim Retten noch weitere Aktionen ausführen zu lassen. Es wird gewartet, bis der Bedienbefehl abgearbeitet wurde – Sie sollten also keinen Dauerläufer programmieren – und das Ergebnis des Befehls wird an die Oberfläche gemeldet, d.h. nur wenn das Retten und der Bedienbefehl erfolgreich waren, bekommt die Oberfläche OK gemeldet.

x=2 Wie ohne Option, aber die Sicherung erfolgt in einer extra Task, so dass weiterhin Kommunikation möglich ist. Allerdings erhält die Oberfläche keinen Status.

x=3 Wie x=2, zusätzlich wird der RTOS-UH-Bedienbefehl SAVE_IEC1131 aufgerufen. Es wird kein Status an die Oberfläche zurückgemeldet.

-I [=xxx]

Setzen des Time Outs für die Kommunikation. Findet in der angegebenen Zeit keine Kommunikation mit der Oberfläche statt, geht das Laufzeitsystem davon aus, dass die Verbindung unterbrochen ist. Die Zeit wird in msec angegeben. Der Default-Wert beträgt 10 Sec.

Bei 0 oder negativem Wert ist das Timeout abgeschaltet. Es ist sinnvoll, die Werte für das Time Out in der CODEIEP . INI auf den gleichen Wert zu setzen

Nach Ablauf des Time Out wird ein automatischer Logout durchgeführt.

-K= *file_name*

Setzen des KONFIG . INI Filenamen, es sollte immer der komplette Pfad angegeben werden.

-P [=prio]

Es wird die Priorität für das SPS_PRG (Single-Task IEC 61131 Programm) umgesetzt. Der Defaultwert für die Priorität ist 255.

-Q [=xxx]

Bei einem Single-Task IEC 61131 Programm ist das Programm normalerweise ein Endlosläufer und wird mit der maximal möglichen Geschwindigkeit abgearbeitet. Mit dieser Option kann eine Zykluszeit (in msec) vorgegeben werden, die natürlich länger als ein Programmumlauf sein sollte. Für den Rest der Zeit wird das Programm dann ausgesetzt. Der Default-Wert ist 0 sec.

-R [=xxx]

Mit dieser Option wird ein Time Out gesetzt, um ggf. einen Dauerläufer beenden zu können. Die Zeit wird in msec angegeben. Der Default-Wert beträgt 2 Sec. Nach Ablauf der Zeit wird eins der folgenden Kommandos – die von der Oberfläche abgesetzt werden können – ausgeführt: Stop, Reset, Online-Change, Download oder Flow-Control.

-S [=xx]

Angabe über welche serielle Schnittstelle mit der Oberfläche kommuniziert werden soll. Die Default-Schnittstelle ist /A2. Normalerweise (ohne Umlenkung) werden Meldungen auf der Schnittstelle ausgegeben, von der das Laufzeitsystem gestartet wurde. Im Regelfall ist dies die /A1/. Achten Sie bitte darauf, dass (Fehler-) Meldungen auf einer anderen als der Kommunikationsschnittstelle mit der Oberfläche ausgegeben werden! Kommunikationsstörungen oder Verbindungsabbrüche sind sonst unvermeidlich. Sind Sie an den Meldungen nicht interessiert, kann das Laufzeitsystem mit 0 /NIL/ CODESYS ... gestartet werden, die Meldungen landen dann im Null-Kanal. Ebenso können Sie mit 0 /Ax CODESYS ... die Meldungen auf einer anderen seriellen Schnittstelle ausgeben.

-T [=port]

Portnummer für die TCP/IP Kommunikation. Als Default-Port ist 12345. Bitte beachten Sie, dass die Portnummer > 4096 sein muss.

-W [=xxxx]

Mit diesem Parameter kann die Zykluszeit Ihres Programmes überwacht werden. Ohne diesen Parameter wird die Zykluszeit nicht überwacht. Die Zeit wird in msec angegeben. Der Default-Wert ist 5 sec. Wird die Zykluszeit überschritten führt das Laufzeitsystem einen Notstop durch, d.h. alle Ausgänge werden auf 0 gesetzt und alle zu diesem Laufzeitsystem gehörenden Tasks werden gestoppt. Bitte beachten Sie, dass diese Option nicht in allen Laufzeitsystemen verfügbar ist.

-Z

Die dynamische Prioritätsänderung der Kommunikationstask wird ausschaltet, d.h. die Kommunikationstask läuft fest auf der Priorität 20. Ohne die Angabe dieses Parameters setzt sich die Kommunikationstask in der Priorität um eine Stufe über die höchste für dieses Laufzeitsystem konfigurierte Anwender-Task. Sie kann damit von keiner Anwender-Task dieses Laufzeitsystems behindert werden, so dass eine Kommunikation immer möglich ist. Im Gegenzug sind natürlich Schwankungen im Timing der IEC 61131-Tasks möglich, wenn kommuniziert wird.

3.4 Fehlermeldungen des CoDeIEP's Laufzeitsystems

Die folgenden Meldungen und Fehlermeldungen kann das Laufzeitsystem ausgeben. Sie werden entweder auf der seriellen Schnittstelle, über die das Laufzeitsystem gestartet wurde, oder eine entsprechend umgelenkte Schnittstelle, ausgegeben. Alle Fehlermeldungen beginnen mit einer Nummer, anhand derer eine weitere Auswertung erfolgen kann:

Meldungen:

- 1** Start des Laufzeitsystems
- 2** Stop des Laufzeitsystems
- 3** AutoProgram load and go
- 4** AutoProgram not loaded
- 5** AutoProgramFile saved
- 6** AutoProgramFile not saved
- 7** Overflow Cyclus
- 8** Auto reload Data modified

File I/O

- 10** Configuration filename wrong
- 11** File not readable
- 12** AutoProgramFile destroyed
- 13** AutoProgramFile found
- 14** File not found
- 16** Info-Block nicht gefunden
- 17** Info-Block zerstört
- 18** Info-Block falsche Version
- 19** Datenfile gelöscht

Allgemeine Fehlermeldungen

- 28** FEHLER: CPU.NR nicht gefunden
- 29** FEHLER: Fehler beim Öffnen
- 30** FEHLER: Fehler beim Lesen der CPU-Adressen
- 31** FEHLER: Label nicht gefunden
- 32** FEHLER: Tabellenkopf fehlerhaft
- 33** FEHLER: Hashtabellenüberlauf
- 34** FEHLER: Label falscher Typ
- 35** FEHLER: Hashtabellenkopf falsch

Fatale Fehler mit Abbruch der Aktion

- 40** FATAL No Memory for :xxxxx
- 41** ERROR
- 42** INFO PLC_IO ignored (BUSERR)
- 43** FATAL ACCESS index out of range
- 44** FATAL Task Variable not Found
- 48** FATAL External Funktion yyyy not found
- 49** FATAL Len of POU>256 KB

Kommunikationsfehler

- 50** ERROR TCP-Buffer overrun
- 51** ERROR TCP-Send Error
- 52** ERROR SER-Error

Fatale Fehler bei Options

- 53** FATAL Device yyy not found
- 54** FATAL Wrong Portnumber
- 55** FATAL Wrong Prio<0, WatchTimeOut<0, CycleTime<0
- 59** INFO Ignore Option yyy

4 Retain-Variablen

Retain-Variablen sind Variablen, die spannungsausfallsicher gespeichert sind. Dazu muß Ihre Hardware einen batteriegepufferten RAM-Bereich zur Verfügung stellen, der nicht von einem anderen Programm oder vom Betriebssystem genutzt wird. Damit stehen die Werte der Retain-Variablen auch nach einem Ab- und Einschalten des Gerätes wieder zur Verfügung. Die Initialisierung für Retain-Variablen wird genau einmal nach dem Download des Programmes durchgeführt. Auch ein Reset/Neustart des Programmes führt nicht zu einer erneuten Initialisierung.

4.1 Anlegen von Retain-Variablen

In der Oberfläche muss unter *Ressourcen->Zielsystemeinstellungen->Speicheraufteilung* für den zunutzenden Speicherbereich (dieser sollte ausserhalb der RTOS-UH Speicherverwaltung liegen) eine Startadresse eingegeben werden.

In diesem Bereich werden vom CoDeIEP die Retain-Variablen in der Reihenfolge ihres Auftretens beim Übersetzen abgelegt. Wenn Sie also Wert auf eine bestimmte Ablage der Retain-Variablen legen, müssen Sie diese am besten in einer Struktur ablegen.

4.1.1 Beispiel für Retain-Variablen

In der Variablen-Definition wird das Schlüsselwort RETAIN angehängt. Alle Variablen in diesem Bereich werden im batteriegepufferten RAM abgelegt. In diesem Beispiel wurden die Retain-Variablen als globale Variablen abgelegt. Sie können aber ebenso lokale Retain-Variablen anlegen.



The screenshot shows the CoDeIEP software interface with the title bar 'CoDeIEP - TESTRETAIN.pro - [Globale_Variablen]'. The menu bar includes 'Datei', 'Bearbeiten', 'Projekt', 'Einfügen', 'Extras', 'Online', 'Fenster', and 'Hilfe'. The toolbar contains various icons for file operations and development. The left sidebar shows a tree view with 'Ressourcen' expanded to 'Globale Variablen', where 'Globale_Variablen' is selected. The main editor area displays the following code:

```
0001 VAR_GLOBAL
0002 END_VAR
0003
0004 VAR_GLOBAL RETAIN
0005   Far : ARRAY[ 1..128 ] OF INT ;
0006 END_VAR
0007
0008
0009
0010
0011
0012
0013
```

5 Librarys

5.1 I/O-Lib

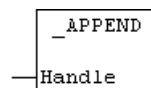
Die I/O-Library stellt Funktionen zum Zugriff auf Dateien bzw. serielle Schnittstellen zur Verfügung:

_APPEND

Es wird auf das Ende der Datei positioniert.

Eingang: Handle: DWORD

Ausgang: Status: DINT

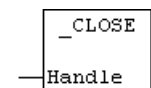


_CLOSE

Eine Datei bzw. eine geöffnete Schnittstelle wird geschlossen.

Eingang: Handle: DWORD

Ausgang: Status: DINT

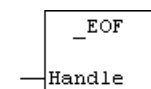


_EOF

Testet, ob ein End Of File aufgetreten ist.

Eingang: Handle: DWORD

Ausgang: Status: DINT

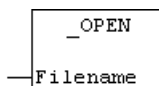


_OPEN

Die Funktion `_OPEN` öffnet eine Datei und liefert ein `Handle` für die weiteren Operationen zurück. Ist der Rückgabewert `Handle=0`, so ist beim Öffnen ein Fehler aufgetreten.

Eingang: Filename: STRING(64)

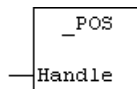
Ausgang: Status: DWORD



_POS

Es wird die aktuelle Position des Schreib-/Lesezeigers der Datei zurückgeliefert.

Eingang: Handle: DWORD
Ausgang: Position: DINT



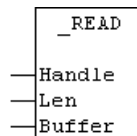
_READ : DINT

Die Funktion `_READ` liest Daten aus einer Datei bzw. von einer seriellen Schnittstelle o.ä. Ist nach Aufruf der Funktion $Len \geq 0$ so ist es die Anzahl der gelesenen Bytes, bei $Len \geq 32768$ ein Fehlercode (siehe RTOS-UH Handbuch Kapitel 5.7.2 "Die Funktion ST...").

Beachten Sie bitte: Beim Lesen von einer seriellen Schnittstelle kann Ihr Programm blockiert werden, wenn Sie nicht vom `/CXX`-Port lesen.



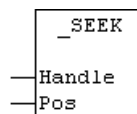
Eingang: Handle: DWORD
Len: DINT
Buffer: POINTER TO STRING
Ausgang: Status: DINT



_SEEK : DINT

Der Schreib-/Lesezeiger der Datei wird auf die Position `Pos` gesetzt.

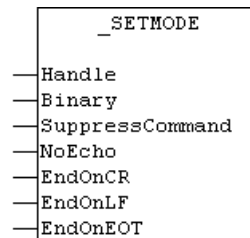
Eingang: Handle: DWORD
Pos: DINT
Ausgang: Status: DINT



_SETMODE : DINT

Mit der Funktion `_SETMODE` können diverse Parameter für die Funktionen `_READ` und `_WRITE` eingestellt werden. Im wesentlichen wird die Funktionen `_SETMODE` für die Parametrierung beim Umgang mit seriellen Schnittstellen gebraucht. Siehe auch RTOS-UH Handbuch Kapitel 8.3 "Das Communication Element" ff.

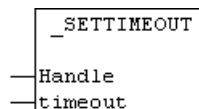
Eingang:	Handle:	DWORD
	Binary:	BOOL
	SupressCommand:	BOOL
	NoEcho:	BOOL
	EndOnCR:	BOOL
	EndOnLF:	BOOL
	EndOnEOT:	BOOL
Ausgang:	Status:	DINT



_SETTIMEOUT : DINT

Mit dieser Funktion kann ein Time Out gesetzt werden. Üblicherweise macht dies nur bei seriellen Schnittstellen Sinn. Die Zeit ist in msec anzugeben. Siehe auch RTOS-UH Handbuch Kapitel 6.1 "Datenstationen Ax, Bx..." ff.

Eingang:	Handle:	DWORD
	TimeOut:	TIME
Ausgang:	Status:	DINT



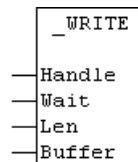
_WRITE

Schreibt Daten in eine Datei, serielle Schnittstelle o.ä. Es werden `Len` Zeichen ausgegeben. Ist `Wait=TRUE` wird gewartet, bis der Schreibvorgang beendet ist. Der Rückgabewert enthält den Status bzw. eine Fehlermeldung. Wird mit `Wait=FALSE` geschrieben, wartet das Programm nicht auf die Beendigung des Schreibvorgangs. Damit ist auch kein Status erhältlich.

Beachten Sie bitte: Bei der Ausgabe mit `Wait` kann Ihr Programm blockiert werden, wenn die Ausgabe ohne `Time Out` läuft.



Eingang:	<code>Handle:</code>	<code>DWORD</code>
	<code>Wait:</code>	<code>BOOL</code>
	<code>Len:</code>	<code>DINT</code>
	<code>Buffer:</code>	<code>POINTER TO STRING</code>
Ausgang:	<code>Status:</code>	<code>DINT</code>



6 Problembehandlung

- **Problem:** Programmdownload bricht immer wieder mit Timeout ab:

Ursache: Bei einem Programmdownload werden binäre Daten übertragen! Dient eine serielle Schnittstelle zur Kommunikation mit dem Zielsystem, findet kein Handshake statt. Das Zielsystem muß also die Daten so schnell entgegen nehmen, wie sie von der Oberfläche geschickt werden! Laufen auf dem Zielsystem weitere Task's, deren Priorität höher ist als die von ComCode, können Daten verloren gehen und ein Download ist nicht möglich.

Abhilfe: Entweder stoppen Sie die höher priorisierten Task's während des Download oder erhöhen die Priorität von ComCode. Unter Umständen können Sie auch die Baudrate der seriellen Schnittstelle herabsetzen, Sie verlängern dann allerdings die Downloadzeit.