

Software-Beschreibung

DS-DRIVE-2-Anhang B

Script
Service-Pack
Parameterliste
Software-Historie



Industriestraße 1
D-71397 Leutenbach 3 - Nellmersbach
☎ 07195/9283-0
Fax 07195/928329
email info@unitek-online.de
Http// www.unitek-online.de

Script

Textdateien für Test- und Einstellprogramme. (Name.uts)

Die Script -Datei wird mit einem Text-Editor erstellt und als Script.uts Datei abgespeichert. Beim jedem Öffnen wird die Datei übersetzt.

Programm-Änderungen

Programm oder Makro in Text-Editor laden.

Änderungen vornehmen und unter gleichem oder geänderten Datei-Namen abspeichern.

Geändertes Programm mit doppelklick starten.

Script ausführen

Drive-Fenster schließen.

(Drive und Script schließen sich aus.)

Regler-Versorgungsspannungen ein, Freigabe geschlossen.

Ordner Script öffnen.

Script Datei auswählen und mit *uts_com1* oder *uts_com2* öffnen.

Die Datei wird sofort gestartet und ausgeführt.

UniTeKScript (.uts) Format A script is interpreted from the first line. Line between routine NAME ...end routine are only interpreted when called. Variables are allocated on reference and have GLOBAL scope. At present "for" loops may not be nested			
Simplified language definition			
Funktion	Befehl	Beispiel	Erklärung
ROUTINE	call NAME	call lesen	Routine aufrufen
	routine NAME	routine lesen	Anfang der Routine
	STATEMENTS		Befehle in der Routine
	return		Routine verlassen
	end routine		Ende der Routine
SCHLEIFE	for VAR = VAL	for v1 = 0 V1 < 10	
	COND STATEMENT	inc v1	
	break		
	endfor		Ende for-Schleife
DECLARATION	var name INDEX(array)	table[N]	N gibt die Zeigerpositon (Zeile) in dem Variablen-Array
	var name {= VAR}	counter = 99	Zähler-Startpunkt wird auf den Wert gesetzt
DECIMAL	0...9	1234 // -1234	Ganze Zahlen mit Vorzeichen
HEX	0..9,A..F, a..f	02Face	Hexadecimal
NAME	A..Z, a..z, 0..9, _	Myname1// la_la	Buchstaben, Zahlen, Unterstrich
INDEX	[SVAR]	[0]// [10]// [v10]// [myindex]	Index zur Variablen
	empty		

Funktion	Befehl	Beispiel	Erklärung
VAR	debug	debug	Variable mit oder ohne Index Variablen-Bezeichner plus Nummer (v10) Ergebnis-Variable plus Nummer (r10) Variable mit Namensbezeichnung
	NAME INDEX	Index// table[]// lala	
VAL	0xHEX	0x8000	Wert in Hexadezimal immer mit 0x...
	DECIMAL	1234 -1234	Dezimalwerte Ganzzahlig mit Vorzeichen
	VAR	x	Wert der Variablen
OPR	!=	IST != 100	ungleich
	<>	IST <> SOLL	ungleich
	=	REGID_.. = -100	gleich
	==		gleich
	<	0x8000<Soll	kleiner als
	>		größer als
	<=		gleich kleiner als
	>=		gleich größer als
	a<	100 a< Soll	Absolutwert kleiner als
	a>	-100 a> Ist	Absolutwert größer als
	a<=		Absolutwert gleich kleiner als
a>=		Absolutwert gleich größer als	
UNIOP	-		minus
	+		plus
	~		1s compliment
	!		© nicht
	neg		minus
	not		© nicht
	abs		Absolutwert (ohne Vorzeichen)
BINOP	+		addition
	-		subtraktion
	*		multiplikation
	/		division
	%		modulo
	&		bitweise und
DCOM	\a	alarm, beep	Direkte Commandos für die Anzeige Am Ende der Programmzeile erfolgt ein Zeilenwechsel. Mit dem Commando no newline wird der Zeilenwechsel unterdrückt und die Anzeigevariablen in eine Zeile geschrieben. Da die Zeichen % und & als Befehle benutzt werden müssen diese für die Text-Anzeige mit einem schrägstrich geschrieben werden.
	\r	carriage return	
	\n	newline	
	\t	tab	
	\+	no newline	
	\%	percent sign	
\&	and sign		

COND	VAL OPR VAL	Soll1 > 100	Wert Soll1 größer als 100
	if COND STATEMENT	if Ist1 >1000 inc REG1	Wenn Ist1 größer als 1000 dann incremente Reg1
TEXT	T%wVAR	T%4v1 T%tab[v3]	Anzeige vom Text plus Dezimal- Wert der Variablen
	T&wVAR	T&5v0 T&max[j]	Anzeige vom Text plus Hexadezimal- Wert der Variablen
	T plain text	La la la	
	Befehl	Beispiel	Erklärung
STATEMENT	; comment	; this is	Kommentar, keine Programmzeile
	:labelname	:startpoint	Sprungadresse
	goto :labelname	goto :startpoint	direkter Sprung auf den Labelnamen
	sleep VAL	sleep 2000	akt.Zustand bleibt erhalten für 2s
	speed VAL	speed 2000 speed -2000	Dezimalwerte für Drehzahl
	write id VAL	write 0x2c v5	überschreibe den Wert in id von VAL
	read id VAL	read REGID_.. Soll	lese den Wert vonVAL nach id
	result VAL		Werte von TMS nach r0 lesen
	abs VAR		bildet Absolutwert (kein Vorzeichen)
	neg VAR		negiert den Wert
	and VAR VAL	and REGID_.. Soll2	REGID ist gleich REGID plus Soll2
	or VAR VAL		
	xor VAR VAL		
	dec VAR [VAL-1]	dec Schleife1 100	reduziere Variable Schleife1 um 100
	inc VAR [VAL-1]	inc Schleife2 10	erhöhe Variable Schleife2 um 10
	mul VAR [VAL-1]	mul Soll1 100	multipliziere Soll1 mit 100
	div VAR [VAL-1]	div Soll2 10	dividiere Soll2 durch 10
	mod VAR [VAL-1]		
	let VAR [=] [VAL=0]	let Ist3 temp7+500/y	lade Ist3 von temp7+500/y
	key VAR msg	key v0 (Esc =abbrechen)	Eingabe Tastaturbefehle
SCREEN	exit TEXT		Anzeige Abbruch-Text
	error TEXT		Anzeige Fehler-Text
	msg TEXT	msg speed=%v1	Anzeige speed und den Wert von v1
	if COND STATEMENT	if v1 >1000 inc v2	
OSZILLOSKOP			
	osc_source N VAR	osc_source 1 REGID_	Kanalnummer, aufgezeichnete VAR
	osc_chan CHANNEL	osc_chan channels	Anzahl der Kanäle
	osc_sampels N	osc_sampels 500	Aufzeichnungspunkte (alle Kanäle)
	osc_time N	osc_time 1000	
	osc_trig VAR	osc-trig REGID_	Triggerbedingung
	osc_run		Triggerstart
	osc_result		Aufzeichnungsergebnis
	let channels N	let channels 5	Oszilloskop- Kanäle
	trig edge_equal		

Beispiel Drehzahl-Testprogramm

Erklärung

speed-test.uts	
.include includes\uts.inc	;Dateiname
line	;Aufruf Macro-Bibliothek (uts.inc)
kern_status	;Linie zeichnen
err_bitmap1 :on_error	;Macro Statusabfrage
	;Macro Fehlerabfrage
let channels 2	;lade Channels mit 2 Kanäle vom Oszilloskop
let samples 500	;lade samples mit 500 Aufzeichnungspunkte
let skip 100	;lade skip mit 100 Aufzeichnungen
mul skip channels	;multipliziere
	;Macros Oszilloskopfunktionen (osz_.....)
osc_source 1 REGID_SPEED_IST	;setze Messwert Kanal 1 = Drehzahl-Istwert
osc_source 2 REGID_I_SOLL	;setze Messwert Kanal 1 = Drehzahl-Istwert
osc_chan channels	;setze Oszilloskop 2 Kanäle
osc_time 1000	;setze Ablenkzeit auf 1000ms
	;REGID_..... sind Parameter im Regler DS..
read REGID_I_MAXPLUS i_max	;lese Parameter von I_MAXPLUS nach i_max
msg lmax=%i_max	;zeige Wert von I_max
line	;Zeichne Linie
write REGID_SPEED_MAXPLUS 32766	;überschreibe Parameter SPEED_MAXPLUS mit 32766
for v1 = 2500 v1=10000 inc v1 2500	;start Schleife von 2500 bis 10000 in Schritten von 2500
msg speed step %v1	;zeige Wert von v1
; osc_easy1 REGID_SPEED_IST	;Kommentar
; REGID_SPEED_REF edge_equal v1 1000	;Kommentar
osc_trig REGID_SPEED_REF edge_equal v1	;Oszilloskop Triggerbedingung
osc_run	;Oszilloskop starten(sendet Einstellungen nach TMS)
speed v1	;Drehzahl von v1
sleep 1000	;Zustand bleibt für 1000ms
speed 0	;Drehzahl Null
sleep 1000	;Zustand bleibt für 1000ms
osc_result samples	;Osz-Werte werden aufgenommen und abgespeichert
let v0 skip	;Lade v0 mit dem Wert von skip (100)
inc v0 0	;Startpunkt gleich wert von v0 (100)
result_stat v0 samples channels	;Macro Ergebnis
msg speed ist avg=%result_avg min=%result_min max=%result_max	;zeigt Rechenergebnisse
let v0 skip	;Lade v0 mit dem Wert von skip (100)
inc v0 1	;Startpunkt gleich wert von v0 plus 1 (101)
result_stat v0 samples channels	;Macro Ergebnis
let v0 result_max	;lade v0 von Macro result_max
sub v0 result_min	;subtrahiere von v0 Macro result_min
mul v0 100	;multipliziere v0 mit 100
div v0 i_max	;dividiere v0 durch i_max
msg current avg=%result_avg min=%result_min max=%result_max buzz=%v0\%	;zeigt Rechenergebnisse
line	;Linie zeichnen
endfor	;Ende der Schleife
exit	;Programmende
:on_error	;Fehlermeldung

Beispiel Macros für die Oszilloskopfunktionen

osz.inc

```
var edge_rising      0
var edge_falling    1
var edge_auto       2
var edge_equal      3
var edge_not_equal  4

.macro osc_time us
let v20 %1
sub v20 150
div v20 150
if v20 0 let v20 0 write REGID_CAPTURE_SKIP v20
.endmacro

.macro osc_chan n(1..8)
let v20 %1
dec v20 write REGID_CAPTURE_CHAN v20
.endmacro

.macro osc_source n(1..8) id
let v20 %1
dec v20
mul v20 0x100
add v20 %2
write REGID_CAPTURE_SOURCE v20
.endmacro

.macro osc_trig id edge level
write REGID_CAPTURE_TRIGSOURCE %1
write REGID_CAPTURE_TRIGEDGE %2
write REGID_CAPTURE_TRIGLEVEL %3
.endmacro

.macro osc_source1 id1
osc_chan 1
osc_source 1 %1
.endmacro

.macro osc_run
write REGID_CAPTURE_RUN 1
.endmacro

.macro osc_easy1 id1 idT edge level time
osc_source1 %1
osc_trig %2 %3 %4
osc_time %5
osc_run
.endmacro

.macro osc_result n
:osc_result++
write REGID_CAPTURE_READ 0x4000
result 1
if r0 %1 goto :osc_result—
write REGID_CAPTURE_READ %1
result %1
.endmacro
```

Service-Pack1

Zugriff auf geschützte Parameter

Regedit bei Windows eintragen

Ablauf:

Tastenfeld Start	anklicken	Fenster Startfeld öffnet sich
Befehl Ausführen	wählen und anklicken	Fenster Ausführen öffnet sich.

In der Eingabezeile hinter Öffnen **regedit** eintragen und mit OK bestätigen

Fenster **Registrieungs-Editor** öffnet sich.
Pull-Down-Menü öffnet sich.

In der Menüleiste **Registrierung** anklicken

Registrierungsdatei importieren wählen und anklicken

Fenster Registrierungsdatei importieren öffnet sich.

Datei **regedit unitek.reg** suchen und wählen.
Bei Dateiname erscheint regedit unitek.reg

Tastenfeld **Öffnen** anklicken

Meldefenster Registrierungs Editor öffnet sich und meldet den Eintrag

Mit OK bestätigen.

Test:

DRIVE anwählen und öffnen

In der Menüleiste **View** anklicken

Pull-Down-Menü öffnet sich

Vor **All** muß ein Auswahlhaken sein, wenn nicht **All** wählen und anklicken

In der Symbolleiste **C** anklicken

Der Auswahlhaken erscheint
Das Fenster **Current** öffnet sich

Die Eingabefelder müssen hell sein. Die Parameterwerte können überschrieben werden.

Nach Eingabe der Parameter:

wieder im Menü **View**

die Auswahl **All** anklicken

Die **Current-Parameter** sind dann wieder schreibgeschützt.

Neue Software-Version laden (Software Update)

Kein Software-Update mit WINDOWS XP !!!

Regelgerät (UniTek Dsxx, BAMO-D3) RS232 und Computer COM1 oder COM2 mit dem Schnittstellen-Kabel RS232 verbinden.

Regelgerät mit der Hilfsspannung 24V= versorgen. Freigabe-Eingang offen.

Am Regelgerät müssen die LED `VCC` und `RDY` leuchten. LED `RUN` muß dunkel sein.



Vom Internet

Software-Download von **Unitek-online.de (.com)**

.zip Datei auf ausgewählten Ordner Speichern

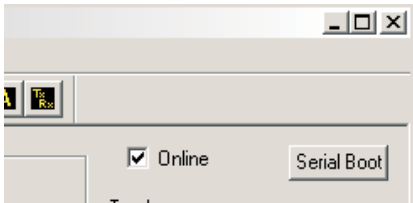
und dekomprimieren.

aufrufen und öffnen.

Ordner **DRIVE-SOFTWARE** aufrufen und

Software-v36)

öffnen.



Datei `Drive.exe` anklicken.

Auf dem Bildschirm erscheint das

Fenster

(Window) `DRIVE` mit der

Titelzeile `Unbenannt Drive`

Von Compact Disk CD

CD `Software-Vxx` einlegen und mit Windows-

Explorer oder gleichartigem Programm starten.

Software-Vxx

(zB.



Tastenfeld `Serial Boot` anklicken und

DRIVE-Programm schließen.

(Tastenfeld **X** anklicken)

Bei `Software-Vxx` die Batch-Datei `UPDATE_1.BAT`

oder `UPDATE_2.BAT` anklicken.

(Je nachdem ob die Schnittstelle COM1 oder COM2

mit dem Regler verbunden ist)

Das Lade-Fenster wird geöffnet und das

Ladeprogramm startet selbsttätig.



Achtung:

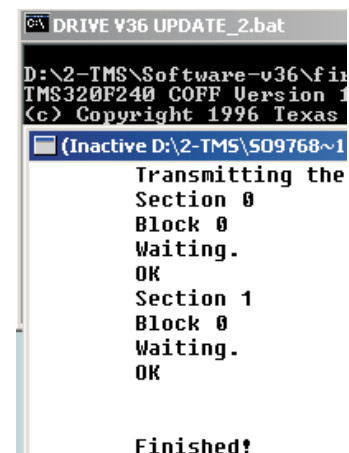
Während das Ladeprogramm läuft,

keinerlei andere Funktion mit dem PC ausführen !!

Wenn das Programm abbricht:

Schnittstelle kontrollieren, Hilfsspannung aus und einschalten und nochmals von vorne beginnen.

Bei korrektem Programmablauf endet es mit der letzten Zeile mit `Finished!`.



Die neue Software ist in das Regelgerät (Unitek DS,BAMO-D3) geladen!

Ladefenster schließen. (Tastenfeld **X** anklicken)

Am Reglergerät die 24V= Hilfsspannung Aus- und Einschalten. (globaler Reset)

Am PC auf den Ordner `Software-Vxx` zurückgehen und die Datei `Drive.exe` durch anklicken starten.

Auf dem Bildschirm erscheint das Fenster (Window) `DRIVE` mit der Titelzeile **Unbenannt Drive**.

Das Tastenfeld Information `i` in der Informationsleiste anklicken.

Das Informationsfenster öffnet sich und im Feld `Firmware-Version` steht die neue Software-Nummer.

Direktes Auslesen und Eingeben der Parameterwerte

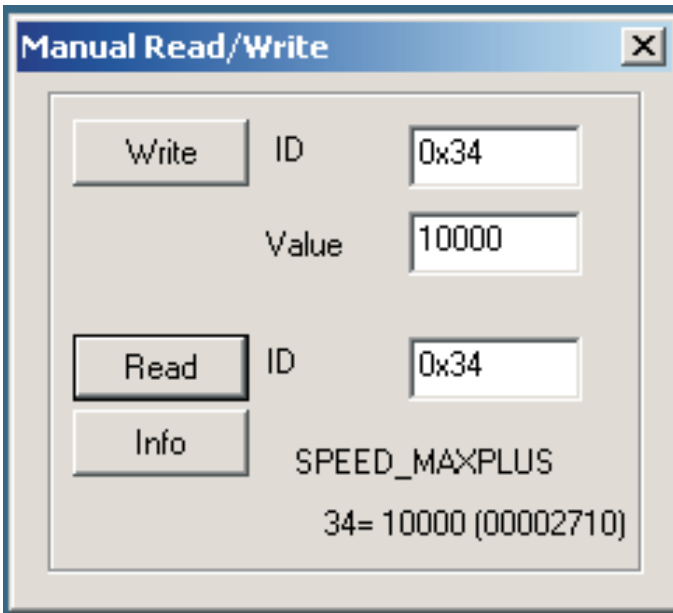
Feld für manuelle Parameter-eingabe und -abfrage öffnen.

In der Menüleiste **'View'** anklicken

Zeile **'Manual Entry'** anklicken

Pull-down Menü öffnet sich.

Fenster **Manual Read/Write** öffnet sich



Parameterwert schreiben

Adresse

Wert (Inhalt)

Parameterwert lesen

Adresse

Parameter-Bezeichnung

Wert(Inhalt)

Parameter lesen:

In das Eingabefeld Read-ID die Parameter-Adresse eingeben.

Tastenfeld **'Read'** anklicken

Unterhalb vom eingabefeld erscheint die Parameter-Bezeichnung und der Inhalt des Parameters.

Tastenfeld **'Info'** anklicken

Die im PC vorhandenen Parameterwerte werden angezeigt.

Parameter schreiben:

In das Eingabefeld Write-ID die Parameter-Adresse eingeben.

Im Eingabefeld Value den Wert für den gewählten Parameter eingeben.

Tastenfeld **'Write'** anklicken

Mit Parameter lesen kontrollieren ob der Wert richtig übernommen wurde.

Achtung: Schreibgeschützte Werte können nur gelesen werden.

Parameter-Funktion	Adresse	Parameter-Funktion	Adresse
var REGID_ASSIGN_WRITE_ADDRESS	0x00	var REGID_MPOS3_IST	0x48
var REGID_TO_WRITE_ADDRESS	0x01	var REGID_MPOS1_MOD	0x49
var REGID_ASSIGN_READ_ADDRESS	0x02	var REGID_MPOS2_MOD	0x4a
var REGID_FROM_READ_ADDRESS	0x03	var REGID_MPOS3_MOD	0x4b
var REGID_USER_KEY	0x04	frei	0x4c
frei	0x05	var REGID_MOTOR_I_MAX	0x4d
frei	0x06	var REGID_MOTOR_I_DAUER	0x4e
frei	0x07	var REGID_MOTOR_POLE	0x4f
frei	0x08	var REGID_SPEED_FENSTPOSICONTROL	0x50
frei	0x09	var REGID_MODE	0x51
frei	0x0a	var REGID_PWM_MIN	0x52
frei	0x0b	var REGID_PWM_MAX	0x53
frei	0x0c	var REGID_I1_IST	0x54
frei	0x0d	var REGID_I2_IST	0x55
frei	0x0e	var REGID_I3_IST	0x56
frei	0x0f	var REGID_MPOS_ROTORSTATE	0x57
var REGID_CAPTURE_CHAN	0x10	var REGID_I_ISTANDMASK	0x58
var REGID_CAPTURE_CHANOFFSET	0x11	var REGID_MOTOR_RPMMAX	0x59
var REGID_CAPTURE_TRIGLEVEL	0x12	var REGID_KERN_OPTIONS	0x5a
var REGID_CAPTURE_TRIGEDGE	0x13	var REGID_SPEED_KS	0x5b
var REGID_CAPTURE_TRIGSOURCE	0x14	var REGID_ROTOR	0x5c
var REGID_CAPTURE_SOURCE	0x15	frei	0x5d
var REGID_CAPTURE_SKIP	0x16	var REGID_SPEED_ISTFILTERWGT	0x5e
var REGID_CAPTURE_READ	0x17	var REGID_KERN_SEMAPHORE	0x5f
var REGID_CAPTURE_RUN	0x18	frei	0x60
var REGID_PWM_OFFSET	0x19	frei	0x61
var REGID_LOOKUP_TEMP	0x1a	frei	0x62
var REGID_KERN_VERSION	0x1b	frei	0x63
var REGID_I1_KP	0x1c	frei	0x64
var REGID_I1_KI	0x1d	frei	0x65
var REGID_I1_KD	0x1e	frei	0x66
var REGID_I1_ISTOFFSET	0x1f	frei	0x67
var REGID_I_IST	0x20	var REGID_CAN_ID_RX	0x68
var REGID_I_SOLLOFFSET	0x21	var REGID_CAN_ID_TX	0x69
var REGID_I_REF	0x22	var REGID_POS_KP	0x6a
var REGID_I_ERR	0x23	var REGID_POS_KI	0x6b
var REGID_I_MAXPLUS	0x24	var REGID_POS_KD	0x6c
var REGID_I_DELTAMAXPLUS	0x25	var REGID_POS_IST	0x6d
var REGID_I_SOLL	0x26	var REGID_POS_DEST	0x6e
var REGID_I1_REF	0x27	var REGID_PWM_DELTAMAX	0x6f
var REGID_I2_REF	0x28	var REGID_POS_ERR	0x70
var REGID_I3_REF	0x29	var REGID_POS_ERRSUMMAX	0x71
var REGID_I_DCCORR	0x2a	var REGID_POS_OFFSET	0x72
var REGID_I_ERRSUMMAX	0x2b	var REGID_FORE_CAN_BTR	0x73
var REGID_SPEED_KP	0x2c	var REGID_POS_ZEROCAPTURE	0x74
var REGID_SPEED_KI	0x2d	var REGID_POS_REFRESOEDGE	0x75
var REGID_SPEED_KD	0x2e	var REGID_SPEED_TOREF	0x76
var REGID_AIN1_OFFSET	0x2f	var REGID_SPEED_FROMREF	0x77
var REGID_SPEED_IST	0x30	var REGID_CAN_REF_START	0x78
var REGID_SPEED_CMD	0x31	var REGID_POS_WINDOW	0x79
var REGID_SPEED_REF	0x32	var REGID_POP_PRESET	0x7a
var REGID_SPEED_ERR	0x33	frei	0x7b
var REGID_SPEED_MAXPLUS	0x34	frei	0x7c
var REGID_SPEED_DELTAMAXACC	0x35	frei	0x7d
var REGID_COMMAND_SOURCE	0x36	frei	0x7e
var REGID_SPEED_COUNTMAX	0x37	frei	0x7f
var REGID_I1_ERR	0x38	var REGID_FUN_BOOT	0x80
var REGID_I2_ERR	0x39	var REGID_FUN_KERNUPDAT	0x81
var REGID_I3_ERR	0x3a	var REGID_FUN_EXIT	0x82
var REGID_SPEED_ERRSUMMAX	0x3b	var REGID_FUN_PARAREAD	0x83
var REGID_SYSTEM_TIMERMS	0x3c	var REGID_FUN_PARAWRITE	0x84
var REGID_READ	0x3d	var REGID_FUN_INT	0x85
var REGID_SET_INDEX	0x3e	var REGID_FUN_CANWRITE	0x86
var REGID_WRITE_INDEX	0x3f	var REGID_FUN_SPIEEPREAD	0x87
var REGID_KERN_STATUS	0x40	var REGID_FUN_SPIRESOREAD	0x88
var REGID_INCR_DELTA	0x41	var REGID_FUN_SPIADCREAD	0x89
var REGID_MPOS1_IST	0x42	var REGID_FUN_DATAMEMREAD	0x8a
var REGID_MPOS2_IST	0x43	var REGID_FUN_PROGMEMREAD	0x8b
var REGID_MPOS_ISTOFFSETA	0x44	var REGID_FUN_DACWRITE	0x8c
var REGID_MPOS_ISTOFFSET_SPD	0x45	var REGID_FUN_ERRREPORT	0x8d
frei	0x46	var REGID_FUN_ERRCANCEL	0x8e
var REGID_MPOS_IST60	0x47	var REGID_ERR_BITMAP1	0x8f

Parameter-Funktion	Adresse	Parameter-Funktion	Adresse
var REGID_TORQUE-CMD	0x90	var REGID_CAN_TIMEOUT	0xd0
var REGID-POS-SOLL	0x91	var REGID_VAR1	0xd1
var REGID_CAN_ERRRESETREQ	0x92	var REGID_VAR2	0xd2
var REGID_CAN_ERRWRITETIME	0x93	var REGID_VAR3	0xd3
var REGID_CAN_READOVERRUN	0x94	var REGID_VAR4	0xd4
var REGID_CAN_COUNTREAD	0x95	var REGID_AIN1	0xd5
var REGID_CAN_COUNTWRITE	0x96	var REGID_AIN2	0xd6
var REGID_CAN_COUNTREJ	0x97	var REGID_AIN2_OFFSET	0xd7
var REGID_SYSTEM_TIMERUS	0x98	var REGID_LOGIC_READ_BITS	0xd8
var REGID_FORE_ERRSYNCD	0x99	var REGID_KERN_1_200PC	0xd9
var REGID_TEMP	0x9a	frei	0xda
var REGID_DIGIO_IN1	0x9b	frei	0xdb
var REGID_DIGIO_IN2	0x9c	frei	0xdc
var REGID_DIGIO_OUT1	0x9d	frei	0xdd
var REGID_DIGIO_OUT2	0x9e	frei	0xde
var REGID_KERN_ERROR	0x9f	frei	0xdf
var REGID_FREIGABE	0xa0	var REGID_O_DOU1	0xe0
var REGID_BALLST-COUNT	0xa1	var REGID_O_DOU2	0xe1
var REGID_ERROR_WARN	0xa2	var REGID_O_BT	0xe2
var REGID_ERROR_FATAL	0xa3	var REGID_O_GO	0xe3
var REGID_GEBER	0xa4	var REGID_I_END1	0xe4
var REGID_GEBER_RPT	0xa5	var REGID_I_END2	0xe5
var REGID_GEBER_INCPERREV	0xa6	var REGID_I_DIN1	0xe6
var REGID_GEBER_POLE	0xa7	var REGID_I_DIN2	0xe7
var REGID_IIST	0xa8	var REGID_I_FRG	0xe8
var REGID_I1_ADC	0xa9	var REGID_I_FAULT	0xe9
var REGID_I2_ADC	0xaa	var REGID_I_BALLAST	0xea
var REGID_LOGIC_HZ	0xab	var REGID_I_UESP	0xeb
var REGID_PWM1	0xac	var REGID_I_LOS	0xec
var REGID_PWM2	0xad	var REGID_SPEED_DELTAMAXDEC	0xed
var REGID_PWM3	0xae	var REGID_IIST_100PC	0xee
var REGID_TIMER_DELTA	0xaf	var REGID_O_NOFAULT	0xef
var REGID_FUN_SERIALBOOT	0xb0	var REGID_N_DAUER_MAX	0xf0
var REGID_I_ERR_MAX	0xb1	var REGID_USER_T_BRAKE	0xf1
var REGID_MPOS_SPEEDFACTOR	0xb2	var REGID_VO_BRAKE	0xf2
var REGID_FUN_HACK1	0xb3	var REGID_VO_ICNS	0xf3
var REGID_HACK1	0xb4	var REGID_VO_TOLER	0xf4
var REGID_MPOS_ISTSPIKES	0xb5	frei	0xf5
var REGID_MPOS_ISTSHADOW	0xb6	frei	0xf6
var REGID_TEMP1_PTR	0xb7	frei	0xf7
var REGID_TEMP1_PTR_INDIR	0xb8	frei REGID_ASCII_USER	0xf8
var REGID_TEMP2_PTR	0xb9	frei REGID_ASCII_WR_EEP	0xf9
var REGID_TEMP2_PTR_INDIR	0xba	frei REGID_ASCII_RD_EEP	0xfa
var REGID_MOTOR_INDUKTIVITAET	0xbb	frei	0xfb
var REGID_MOTOR_INNENWIDERSTAND	0xbc	frei	0xfc
var REGID_MOTOR_TRAEGHEIT	0xbd	frei	0xfd
var REGID_LOGIC_END1	0xbe	frei	0xfe
var REGID_LOGIC_END2	0xbf	frei	0xff
var REGID_LOGIC_DIN1	0xc0		
var REGID_LOGIC_DIN2	0xc1		
var REGID_LOGIC_DOUT1	0xc2		
var REGID_LOGIC_DOUT2	0xc3		
var REGID_REGLER_I_MAX-PC	0xc4		
var REGID_REGLER_I_DAUER-PC	0xc5		
var REGID_REGLER_I_GERAET	0xc6		
var REGID_SPEED_DELTAMAXSTOP	0xc7		
var REGID_SPEED_RPM_MAX	0xc8		
var REGID_POSI_KP	0xc9		
var REGID_POSI_KI	0xca		
var REGID_POSI_KD	0xcb		
var REGID_POSI_ERR	0xcc		
var REGID_POSI_ERRSUM_MAX	0xcd		
frei	0xce		
var REGID_POSI_KY	0xcf		

2002 / 2003

```
;
; ~~~~
; 32 - Position calibrating using serial input start
; 33 - Position Window & Can REF_Start
; 34 - software FRG_OFF
; - new data page (128 word limit), kern_var2
; 35 - BTB with undervoltage (UESP) determined by software switch
; =====> ccstudio
; 36 - can id's standardized
; 37 - CAN auto feedback increased from 4 to 8
; - UBM / UESP per options bit
; 38 - CAN auto feedback bug fix - false 0 length response removed
; 39 - delayed FRG off variable
; 40,41 - Latched fault output
; 42 - Motor temperature now readable (Error when temp > 140 (>3v))
; 43 - Tickdown and Filter_I_Soll changes (time given in seconds:external variable
; _N_dauer_max)
; 44 - using SJA1000 hardware rx-filter (still in BasicCAN mode)
; 45 - Tickdown and Filter_I_Soll now uses SPEED_count_max
; 46 - Bug fix - initialisation of *PTR1, *PTR2
; 47 - allow analog sign-magnatude SPEED_soll
; - analog Ain1*Ain2 now with a shift
; - SPEED_count_max minmax(2-20)
; 48 - MOTOR_pole minmax(2-40)
; 49 - Brake logic
; - took out speed_ist_filter_ref
; 50 - allow MPOS offset also with INCR - rotor sync now in MPOS_ist_offset_r
; 51 - weighted Speed_ist_filter
; - Speed ist read quad / speed reverse removed
; - ballast trip / error in foreground
```

2004

- ; ~~~~
- ; 52 jan.12 - ballast trip / error in 150us intr
- ; 53 - analog i/p clipped to +-4095
- ; - ballast_step uses LDPK for REGLER_I_geraet
- ; 54 jan.15 - remove BOOST
- ; jan.16 - adjust POS_hardware on ref. "Z-capture"
- ; 55 jan.20 - PeliCAN / 29bit id's
- ; jan.22 - bug fix - REGID_READ time=0 by can (used to send 32 bit values twice)
- ; 56 feb.05 - eeprom 32bit read and writeable
- ; 57 feb.16 - ref_plus input
- ; feb.23 - adc Ain1 quad (joy-stick)
- ; - bunging most <#if 0>/<.if 0> sections
- ; 58 mar.08 - New sqrt function / Pos_speed rescaling
- ; mar.19 - CAN RXID and TXID changeable on the fly
- ; 59 mar.24 - New mode Torquecontrol
- ; - POS_dest built in
- ; 60 mar.25 - analog Torque [needs new DRIVE]
- ; - intro of O_TOLER
- ; 61 mar.29 - no more auto LEARNSEEDOFFSET / readjust Speed filter sum on change
- ; - llimit no longer affects ldauer
- ; 62 apr.01 - spi_adc if req'd in intr otherwise in fore
- ; 63 apr.05 - mode STOP / ignore semaphore during ref.
- ; 64 apr.06 - BUG! kern_status_REFSTART set differently
- ; - LOGIC_read_bits added
- ; 65 apr.19 - removed calc of IN_TOL to logic
- ; 66 apr.23 - regler_I_max, I_dauer now in fraction of 16384
- ; apr.27 - POS_dest type changed from RW to SP
- ; 67 may.11 - I_max_intern 32 bit intermediate calculation
- ; - INIT_MPOS bug fix clear Assume mpos_ist= 0 and no subtract of possible garbage
- ; 68 may.12 - calibration with speed_0 and Mode_BIT_CAL_OFF
- ; 69 may.15 - Bug fix, now use SPEED_soll(n-1) instead of speed_soll for the reference speed ramp
- ; 70 may.24 - optional softer ramp for reference turnaround
- ; - BUG fix! N_actual now follows N_cmd_ramp exactly across
- ; - zero boundry in both speed and position mode
- ; 71 may.27 - UESP returns analog / digital value depending on option
- ; jun.04 - Speed_max limited to 32767 instead of 32000
- ; 72 jun.08 - Modified_CORR_ADD_ESUM (clipping of I term in PI/PID)
- ; 73 jun.16 - allow position preseting
- ; jun.18 - using I_ist instead of I_soll in I_dauer filter
- ; 74 jun.21 - 32 bit VAR1,2,3,4
- ; - new MINMAX
- ; jun.30 - BUG fix, now use SPEED_soll(n-1) and SPEED_soll(n) for the reference speed ramp
- ; 75 jul.12 - new N filter / modified POS switch
- ; jul.21 - added POS->var1 POS->var2 inputs
- ; 76 jul.28 - new equations for pid factors
- ; - BUG fix in Output logic (from Ver.74)
- ; 77 sep.14 - BUG fix unsigned mult for speed_ist_scale_mult
- ; 78 sep.30 - BUG fix reset mode flag bits on ref start (particularly POSMODE)
- ; 79 oct.06 - Software capture of Pos_ist stored in variable 4
- ; 80 oct.20 - External regen (ballast) cooling time halved
- ; 81 nov.11 =====> win98 -> win2000 (sp4)
- ; - new kp equation for pid's - rpm component now in pos_kp calc; reduced i_kp (code save)

2005

```
;
; 82  jan.12 - remove PWM_min/_max/deltamax from regid list
; 83  jan.17 - remove incr.asm code is now kern.asi (code save)
;     jan.21 - bug fix from ver.73 - clipping also when counting down
;     jan.25 - rearanging lcns clipping
; 84  jan.25 - Bug fix missing access_vars
; 85  jan.27 - added REGID_SPEED_RPM_MAX_INT
; 86  feb.19 - added MASK_STATUS & mode bit TX_TOG_STAT
;           - code optimisation within switch statements
```

KERN_VERSION .set 86

