

Inhaltsverzeichnis

1.	Einleitung	3
2.	INDEL-INI-Datei	3
	2.1 Einträge	3
3.	Verdrahtung der seriellen Schnittstelle RS 232	8
4.	Indel Debugger (ID.EXE u. IDR.EXE)	8
	4.1 Einleitung	8
	4.2 Arbeitsweise des Debuggers	8
	4.3 *.ini Datei	9
	4.4 Task-Fenster	9
	4.5 Assembler-Fenster	11
	4.6 Unterschied ID.EXE und IDR.EXE	12
	4.7 Modem-Betrieb	13
	4.8 Verdrahtung 2k-SIO - Modem	13
	4.9 Kurzreferenz	14
5.	INDEL Transfer (Trans.exe)	16
	5.1 Einträge in der *.ini Datei	16
	5.2 Fehlermeldungen von Trans	16
6.	Scandep	18
7.	Arbeiten mit Windows NT	19
8.	INDEX	20

1. Einleitung

Das INDEL-Tools Angebot umfasst zur Zeit die Programme Debugger (ID und IDR), Transfer (Trans), Config, Show und den Makro-Assembler (MSI).

Weiter existieren noch kleinere Hilfs- und Dienstprogramme. Nachfolgend finden Sie eine Dokumentation der Themen: INI-Datei, Debugger (ID und IDR) und Transfer (Trans). Zum Config und Show Tool finden Sie in den Peripherie-Karten-Dokumentationen eine Beschreibung.

2. INDEL-INI-Datei

Alle INDEL-Tools, ausgenommen MSI, beziehen die notwendigen Informationen über das Zielsystem aus einer *.ini-Konfigurationsdatei. Die *.ini-Datei kann als Übergabeparameter übergeben werden.

z. B. Trans demo.ini

Wird das Tool ohne Übergabeparameter gestartet, verwendet es die INDEL.ini Datei im aktuellen Verzeichnis.

Der Aufbau einer solchen Datei lehnt sich an die von Windows bekannten '.INI'-Datei Strukturen an. Einer Überschrift (Applicationname) folgen sogenannte Schlüsselworte (Keynames), welche die einzelnen Konfigurationspunkte beschreiben :

```
[Application1]
Keyname1=...
Keyname2=...

[Application2]
Keyname1=...
...
```

2.1 Einträge

[Target]

System=

Definiert das zu behandelnde Zielsystem.

PCMASTER - das Zielsystem ist ein PC-Master

IPS-32 - das Zielsystem ist ein INDEL 19"-Rack

Default: PCMASTER

[PCMaster]

Address=

Angabe der Adresse, auf der sich der PC-Master befindet (Drehschalterwerte), z. B. CA00.

Default: D000

ConfigFile=

Name und Pfad der Konfigurationsdatei, die mit Hilfe von CONFIG.EXE erstellt wurde, z. B. c:\Project\test.pcm.

Default: CONFIG.PCM

WarmBoot=

NO

- das Zielsystem wird in jedem Fall zuerst initialisiert und anschliessend mit Software befruchtet

YES

- das Zielsystem wird nur dann initialisiert und mit Software befruchtet, wenn es nicht bereits läuft oder den Geist aufgegeben hat

Default: NO

EnableTime= NO - Evtl. gebrauchte Time-Befehle liefern ein falsches Ergebnis
 YES - Im PCMaster stehen PC-Zeit und PC-Datum über die Standard-Time-Befehle zur Verfügung.
 Hinweis: Diese Option bezieht sich immer auf alle im PC installierten PCMaster. Der TSR-Treiber holt sich die jeweiligen Adressen von SET PCMASTER = Eintrag in Autoexec.bat.

FloatingPointValues= NO: - die Werte im DPR werden im üblichen Festkommaformat dargestellt
 YES: - die Werte werden im Fließkommaformat (floating-point) dargestellt. (Diese Option steht nur in Verbindung mit einem INFO-Master zur Verfügung.)

[IPS-32]

Baudrate= Baudrate für den Datentransfer PC -> IPS-32 Rack
 2400 2400 Baud
 9600 9600 Baud
 19200 19200 Baud
 38400 38400 Baud

DataBits= Anzahl Data-Bits pro BYTE.
 7 7-Bit
 8 8-Bit

Stop-Bits= 1 1 Stop-Bit
 2 2 Stop-Bit

Parity= no no Parity
 even even Parity
 odd odd Parity

Retries= Anzahl Versuche bei Übertragungsfehler bis Bildschirm-Meldung.
 5 5 Versuche

Timeout= Wartezeit in ms bis Retry. Normalerweise wird dieser Eintrag nicht benötigt, da die optimale Timeoutzeit aufgrund der aktuellen Baudrate berechnet wird.

SlaveNumber= Slave-Nummer von IPS-32 Rack
 1 Slave-Nummer 1

Port= PC/AT Schnittstellen Nummer
 COM1 erste Schnittstelle
 COM2 zweite Schnittstelle

[Trans]

SystemSoftware= Name und Pfad der Systemsoftware, z.B.c:\pcmaster\pcm.hex
 Default: PCM.HEX

SystemOffset=	Hier kann ein Downloadoffset angegeben werden (nur bei Target=IPS-32). Der Offset wird als Wortadresse in Hex angegeben. Default: 0
SystemDownload=	NO - die Systemsoftware wird nicht ins Zielsystem geladen YES - die Systemsoftware wird ins Zielsystem geladen Default: YES
SystemVerify=	NO - es findet kein Vergleich zwischen Source und Zielcode statt YES - Source und Zielcode werden miteinander verglichen und evtl. Fehler angezeigt. Default: NO
SystemAutostart=	NO - das Betriebssystem wird gestartet und sogleich auf HALT gesetzt (für eingefleischte Indel Freaks entspricht dies dem 'Init-Halt' mit dem Utility) YES - das Betriebssystem wird normal gestartet Default: YES
DownLoad=	NO - die evtl. unter [ProjectFiles] angegebenen Dateien werden nicht automatisch ins Zielsystem geladen YES - die evtl. unter [ProjectFiles] angegebenen Dateien werden ins Zielsystem geladen Default: NO
Verify=	NO - es findet kein Vergleich zwischen Source und Zielcode statt YES - Source und Zielcode werden miteinander verglichen und evtl. Fehler angezeigt. Default: NO
Autostart=	NO - der Monitortask wird gestartet und sogleich auf HALT gesetzt YES - der Monitortask wird gestartet Default: NO
[Debug]	
TabSize=	Tabulatorzeichen (09) werden in Dateien zu TabSize Leerzeichen erweitert. Default: 8
maxInputs=	Definiert die maximale Anzahl Eingänge, die ein Input-Fenster verwaltet. Der Wert wird auf ein Vielfaches von 16 gerundet und kann 4096 nicht überschreiten. Default: 256
maxOutputs=	Wie „maxInputs“, aber für die Ausgänge.
maxFlags=	Wie „maxInputs“, aber für die Flags.
AutoTaskWndClose=	YES: - das Fenster eines nicht mehr existierenden Tasks wird automatisch gelöscht. NO: - das Fenster wird nicht automatisch gelöscht. Default: YES

WatchCaseSensitiv=	YES: - bei den überwachten Ausdrücken (Watch-Fenster) wird auf Gross-/Kleinschreibung geachtet. NO: - keine Unterscheidung zwischen Gross- und Kleinschreibung. Default: NO
SourceFileTrace=	YES: - es wird bei einem Halt immer der aktuelle Quelltext geladen oder aktiviert. NO: - der aktuelle Quelltext muss selbst geladen oder aktiviert werden (mit „View => Task Source „)
EpromRetries=	Definiert die Anzahl Versuche zum Brennen einer EPROM-Zelle beim Menüpunkt „Burn EPROM“, wird aber gleichzeitig auch bei „Dump Memory“ und „Clear Memory“ als Versuchszähler verwendet.
MemoryFileName=	Voreingestellter Dateiname für das Schreiben eines Memory-Dumps in eine Datei. Default Dateiname: mx
NumberOfValues=	Voreingestellte Anzahl Werte, die von einem Memory-Dump in eine Datei geschrieben werden.
ShowAsmWndCPUHalt=	YES: - das Assembler-Fenster wird bei einem CPU-Stop automatisch geladen oder aktiviert. NO: - das Fenster wird nicht automatisch geladen oder aktiviert. Default: YES
	Zum Auffinden von Traps im Makro-Code kann es störend oder gar unnötig sein, wenn jedesmal beim Auftreten eines Traps das gesamte Assembler-Fenster mit Source geladen wird. Schalten Sie in diesem Fall diese Option aus, damit Sie im Makro-Code die trapverursachende Stelle sehen.
AsmSystemLoad=	YES: - das „System.lst“ wird bei einem CPU-Stop in diesem Modul geladen NO: - das „System.lst“ wird nicht automatisch geladen Default: YES
	Das gesamte Betriebssystem ISM X.XX befindet sich in einer Datei dem sogenannten „System.lst“. Weil diese Datei eine überdurchschnittliche Grösse von über 500kB aufweist, dauert ein Ladevorgang dementsprechend lang, was dann wiederum als störend empfunden werden kann. Schalten Sie diese Option aus, wenn Sie nicht explizit im System debuggen wollen.
[ProjectFiles]	
FILE1=	Hier werden die Projektdateien eingetragen, die von TRANS.EXE ins Zielsystem geladen werden, bzw. die dem Debugger bekannt sein sollen. Hinter '=' kann ein Downloadoffset (in hex) angegeben werden, da ja bekanntlich mit dem ISM-Compiler MSI.EXE nur Compile im Adressbereich von 0..FFFF erzeugt werden können.

0 - Das File wird in den Bereich 00'0000...00'FFFF geladen.
10000 - Das File wird in den Bereich 01'0000...01'FFFF geladen.
Default : 0

[Config]

P_INPBAS= Zeiger auf Uebergabebereich InputBase,
Default: \$2C0000

P_OUTBAS= Zeiger auf Uebergabebereich OutputBase,
Default: \$2C0020

P_IOCDOK= Zeiger auf Statusuebergabebereich IO-Karten,
Default: \$2C001E

P_ANABAS= Zeiger auf Uebergabebereich analoge Base,
Default: \$2C0100

P_ANADEF= Zeiger auf analoge Kanaldefinition,
Default: \$2C0300

P_ANACFG= Zeiger auf Definition analoge Karten,
Default: \$2C0720

P_AXSBAS= Zeiger auf Achsenuebergabebereich,
Default: \$2C0500

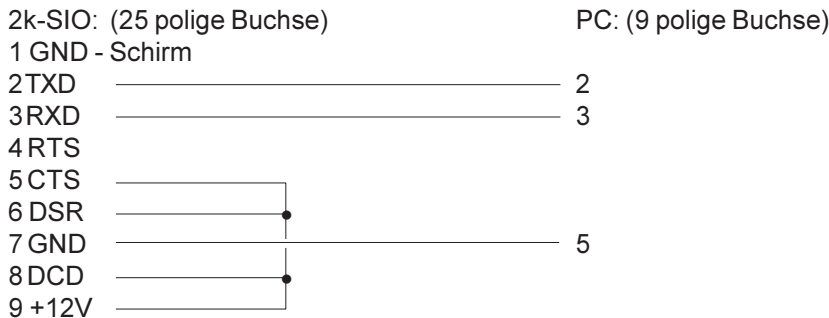
P_NROFAC= Zeiger auf Anzahl analoge Karten,
Default: \$2C0700

P_NROFDC= Zeiger auf Anzahl digitale IO-Karten,
Default: \$2C0701

P_SPCCFG= Zeiger auf Spezialkartenbereich,
Default: \$2C0760

3. Verdrahtung der seriellen Schnittstelle RS 232 (nur bei Rackbetrieb)

Um einen einwandfreien Betrieb mit den INDEL-Tools zu gewährleisten, sollten Sie die nachfolgende Verdrahtungsempfehlung beachten:



4. Indel Debugger (ID.EXE u. IDR.EXE)

4.1 Einleitung

Der Indel Debugger ist ein leistungsstarkes Tool zur Fehlerdiagnose, Fehlerbehebung und Verifizierung Ihrer Programme. Der Debugger lässt sich sowohl mit PC-, INFO-Master als auch mit dem Rack betreiben und verhält sich dabei vollständig konsistent.

Source-Level Debugging ist auf Makro- als auch auf Mikro-Ebene möglich. Mikrounterprogrammaufrufe aus dem Makro, sogenannte JEX, REX, CXP oder RCXP, können direkt aus dem Makro in den Mikro-Code verfolgt werden. Zur Überprüfung Ihrer Variablen, Register oder Speicherstellen stehen Watch-, Register- oder Dump-Fenster zur Verfügung, die Ihnen Ihre Werte im gewünschten Format anzeigen können. Das Modul und Revisionsystem vom Betriebssystem wird im Debugger vollständig unterstützt.

4.2 Arbeitsweise des Debuggers

Der Debugger bezieht sich wie jedes INDEL-Tool auf eine zum Zielsystem zugehörige *.ini-Datei, derer er die benötigten Informationen entnimmt und wo er in seiner Arbeitsweise beeinflusst werden kann. Die *.ini Datei kann als Kommandozeilenparameter übergeben werden. Wird kein Parameter übergeben, sucht der Debugger im aktuellen Verzeichnis nach einer Indel.ini-Datei und verwendet diese.

Beim Hochstarten sucht der Debugger alle Task-Dateien, die unter [ProjectFiles] in der *.ini-Datei vereinbart sind, und berechnet ihre Start- und Endadressen. Dies ermöglicht ihm, jedem Task aufgrund seines Makro-Programm-counters ein Listing zuzuordnen. Weiter wird, wenn noch keine gültige Desktopdatei (id.dsk) besteht, die gesamte globale Variablen-tabelle hochgeladen. Diese wird für das Modul-Revisionsystem sowie für die Funktion „Step into external procedure“, direktes Debuggen von Assemblerprozeduren, die vom Makro aufgerufen werden, benötigt. Die globale Variablen-tabelle wird beim Verlassen des Debuggers in der Desktopdatei abgespeichert und muss bei einem erneuten Hochstarten nicht mehr hochgeladen werden. Es besteht ein spezieller Menüpunkt „Load Asm Def“ zur Auffrischung dieser Tabelle, falls dies erforderlich ist. Der Debugger überprüft aber in jedem Fall beim Hochstarten die *.ind Datei und lädt bei einer Änderung dieser Datei die Tabelle wieder hoch. Wenn er keine *.ind Datei findet, wird mit einer Warnung darauf hingewiesen. Ein Mikro-Debuggen ist in diesem Fall nur ohne Source-Level-Debugging möglich.

Im normalen Debugging-Betrieb wird jedes offene Fenster mit der unter dem Menüpunkt „Refreshrate“ eingestellten periodischen Auffrischungszeit neu dargestellt. Die Auffrischungszeit kann als ganzzahliger Sekundenwert gewählt oder auf 0 gesetzt werden, was einer maximalen Auffrischungsrate entspricht. Die Auffrischung kann aber auch vollständig ausgeschaltet werden, was manchmal sinnvoll sein kann, wenn über die serielle Schnittstelle mit dem Rack gearbeitet wird. Beim Rackbetrieb sollte die Auffrischungszeit auf ein vernünftiges Mittelmaß eingestellt werden, damit der Debugger eine brauchbare Reaktionszeit aufweist und dennoch Werte anzeigt, die genügend genau sind.

4.3 *.ini Datei

Der Debugger verwendet die Einträge unter [Target], [PCMaster] oder [IPS-32], und weiter existiert unter dem Eintrag [Debug] ein spezieller Abschnitt, in dem verschiedene Optionen des Debuggers eingestellt werden können. Für Einzelheiten zu den erwähnten Punkten betrachten Sie den Abschnitt INDEL-INI-Datei.

4.4 Task-Fenster

The screenshot shows the INDEL-Tools debugger interface. At the top, there is a menu bar with options: Run, Memory, View, Breakpoint, Search, Window, Options. The current task is DEMO_60, and the refresh rate is 42713 I/sec. The Task-Fenster (Task Window) is active, showing the following content:

```

Tasks : | 0 1 2 3 4 5 6 7 8 9 | 0 1 2 3 4 5 6 7 8 9 | 0 1 2 3 4 5 6 7 8 9 | 0 1
[ ] = halted Task 0 : DEMO_60

[ ] DEMO_60
8007 GGP @TX_TIM,TIM_POI ; Get Glob
;----- Starte P10 -----
; Vorsicht: HI_TSK muss mit :D angegeben werden
300B S_LOOP: RCXP HI_TSK:D,10,EXQ_DCP ; Starte H
8010
;----- 1ms Timer -----
; Der 1ms SystemTimer läuft rückwärts !
25:11

[ ] Watches
@TIM_POI: 00(R12) FFFEE064 (double hex)
E_TIM: R14 B29D (word hex)

Reg
R10 020F
R11 0000
R12 03EA
R13 0016
R14 B29D
R15 FFFF
R16 0000
R17 0000

F7 Single Step Alt-F7 Step into EXP F8 Halt F9 Go Alt-F10 Local Menu

```

Ein Task-Fenster können Sie laden oder aktivieren, indem Sie beim Task-Balken auf die entsprechende Task-Nummer klicken oder über den Menüpunkt „View => Task“ und die darauf folgende Task-Liste einen Task auswählen. Ein Task-Fenster besteht aus einem Rahmenfenster, das alle zum Task gehörenden Fenster beherbergt. Ein Task-Fenster kann ein Task-Control-Fenster, ein Watch-Fenster und ein oder mehrere Quelltext-, Register- und Memory-Dump-Fenster beinhalten. Weiter können noch ein Inputs-, Outputs und Flags-Fenster sowie Textdateien zur Betrachtung geladen werden.

Das Task-Fenster repräsentiert in diesem Sinne für den Anwender eine Schnittstelle zur Veranschaulichung aller taskspezifischen Daten sowie zum aktiven Eingriff in den Prozess.

4.4.1 Task-Control-Fenster

Das Task-Control-Fenster beinhaltet die Task-Register: RNR, MPC, HTW, TIM, SEC und ABA. Für die Bedeutung dieser Register schlagen Sie bitte den Abschnitt Task-Kontroll Register in der INDEL-Betriebssystem-Dokumentation nach. Das Task-Control-Fenster kann über den Menüpunkt „View => Task control“ geladen werden.

4.4.2 Task-Quelltext-Fenster

Ein Task-Quelltext-Fenster können Sie über den Menüpunkt „View => Task source“ laden oder aktivieren, oder wenn Sie die Option SourceFileTrace nicht ausgeschaltet haben, wird der Debugger bei einem Halt des Tasks automatisch den richtigen Quelltext laden oder aktivieren. Mit „F2“ oder dem Menüpunkt „Breakpoint => Toggle Breakpoint“ können Sie an beliebiger Stelle im Quelltext einen Breakpoint setzen. Falls Sie an einer Stelle einen Breakpoint setzen, wo sich kein ausführbarer Makro-Code befindet, wird sie der Debugger mit einer Fehlermeldung darauf hinweisen. Beachten Sie, dass im Makro immer nur ein Breakpoint pro Task möglich ist. Ueber „Alt-F10“ oder die rechte Maustaste können Sie zwischen einer ausführlichen Darstellung mit Makro-Code oder einer gekürzten Darstellung umschalten.

4.4.3 Task-Register-Fenster

Das Task-Register stellt alle anwenderspezifischen Register dar. Es kann mehrmals in einem Task-Fenster über den Menüpunkt „View => Task registers“ geladen werden und mit „Alt-F10“ oder der rechten Maustaste in einem lokalen Menü in seinen Darstellungseigenschaften verändert werden.

4.4.4 Task-Watch-Fenster

Im Watch-Fenster lassen sich anwenderspezifische Ausdrücke überwachen. Ueber den Menüpunkt „Task watches => Add watch“ können Sie einen neuen Ausdruck hinzufügen. Mit „Task watches => Delete watch“ löschen Sie den selektierten Ausdruck, mit „Task watches => Edit watch“ editieren Sie den selektierten Ausdruck, und mit „Task watches => Remove all watches“ löschen Sie alle Ausdrücke. Mit dem lokalen Menü, das Sie über „Alt-F10“ oder die rechte Maustaste aktivieren, können Sie die Darstellungsart des selektierten Ausdruckes nachträglich ändern.

4.4.5 Task-Memory-Dump-Fenster

Mit dem Memory-Dump-Fenster können Sie einen beliebigen Speicherblock in dem von Ihnen gewünschten Format betrachten.

Wählen Sie dazu den Menüpunkt „View => Dump“ und anschliessend das gewünschte Darstellungsformat. Es sind mehrere Memory-Dump-Fenster in einem Task-Fenster möglich. Beachten Sie, dass in einem Task-Fenster die Adressen immer Wort orientiert sind.

Im lokalen Menü stehen Ihnen folgende Funktionen zur Verfügung:

- Goto: wechselt zu einer neuen Basisadresse
- Display as: ändert die Darstellungsart
- Follow: nützlich zum Verfolgen von Pointern. Es wird der Wert, an dem der Cursor momentan steht, als neue Basisadresse verwendet.
- Write to file: die Daten werden in eine Datei geschrieben

4.5 Assembler-Fenster

```

≡ Run Memory View Breakpoint Search Window Options I/sec : 0
Tasks : | 0 1 2 3 4 5 6 7 8 9 | 0 1 2 3 4 5 6 7 8 9 | 0 1 2 3 4 5 6 7 8 9 | 0 1
[ ]= halted Assembler c:\info6.0\user\INIT.lst [ ]
CPU
R0 001FDC00
R1 0000FA40
R2 002C07DC
R3 0000FF10
R4 0000132A
R5 0001003C
R6 0000800B
R7 001FE000
SB 001FA000
FP 002C0000
SP0 001FDBEC
SP1 FFFFFFFF
IBS 001FA000
MOD 0000
PSR 0840
PC 00000049

MOVQD P_FLG,R0 ; set pointer to flag
MOVQD 0,0(R0) ; clear flags 0..31
MOVQD 0,4(R0) ; clear flags 32..63
MOVQD 0,8(R0) ; clear flags 64..95
MOVQD 0,12(R0) ; clear flags 96..127
CXP INIT_MODULES ; init all other modu
CXP SYSIN ; start OS (never com

FPU
F0 +1.00000000 E+04
F1 +1.00000007 E-02
F2 **,***** E+**
F3 **,***** E+**

Dump (Module NMI)
SB:000000 0000 0000 0000 0000
SB:000008 0000 0000 0000 0000
SB:000010 0000 0000 0000 0000
SB:000018 0000 0000 0000 0000

F7 Single Step Alt-F7 Step into EXP F8 Halt F9 Go Alt-F10 Local Menu

```

Das Assembler-Fenster ist das Äquivalente zum Task-Fenster auf Mikro-Code Ebene. Manche Menüpunkte beziehungsweise Funktionen stehen sowohl beim Task-Fenster als auch beim Assembler-Fenster mit konvertierter Bedeutung, auf den Mikro-Code bezogen, zur Verfügung. Ein Assembler-Fenster wird über den Menüpunkt „View => Asm“ geladen oder aktiviert. Beim Erzeugen des Assembler-Rahmenfensters wird voreingestellt ein CPU- und ein Quelltext-Fenster geladen. Weiter können zusätzlich weitere Quelltext-Fenster, mehrere Memory-Dump-Fenster, ein FPU-Fenster und beim Rackbetrieb ein ICU-Fenster geladen werden.

4.5.1 Assembler-CPU-Fenster

Das CPU-Fenster beinhaltet alle CPU-Register sowie die Flag-Bits. Im lokalen Menü kann die Darstellung des Programm-Counters (PC) und des Modul-Registers zwischen relativ und absolut umgeschaltet werden. In der absoluten Darstellung wird in beiden Register-Feldern der effektive CPU-Registerwert angezeigt. In der relativen Darstellung werden die beiden Werte umgerechnet, und Sie sehen im Modul-Register-Feld die momentane Modulnummer und das Programm-Counter-Feld zeigt den Programm-Counter relativ zum Modul an. Das CPU-Fenster ist über den Menüpunkt „View => Asm CPU“ ladbar.

4.5.2 Assembler-Quelltext-Fenster

Sie können ein Assembler-Quelltext-Fenster explizit über den Menüpunkt „View => Asm source“ laden, oder wenn die CPU stoppt, wird automatisch der Quelltext des aktuellen Moduls geladen. Das automatische Laden ist über die Option „SourceFileTrace“ in der *.ini Datei ein- und ausschaltbar. Sie können in den Quelltexten mehrere Breakpoints mit „F2“ oder über den Menüpunkt „Toggle Breakpoint“ setzen. Falls an der aktuellen Cursorposition ein Breakpoint-Setzen unmöglich ist, wird Sie der Debugger mit einer Fehlermeldung darauf hinweisen. Mit „Alt-F10“ oder der rechten Maustaste kann zwischen einer ausführlichen Darstellung mit Mikro-Code und einer gekürzten Version umgeschaltet werden.

4.5.3 Assembler-Memory-Dump-Fenster

Um Variablen und Speicherblöcke zu untersuchen, steht Ihnen das Assembler-Memory-Dump-Fenster zur Verfügung. Es unterstützt die CPU-spezifischen Adressierungsarten (SB, PC, FP relativ). Beachten Sie, dass im Assembler-Fenster die Adressen Byte-orientiert sind. Im lokalen Menü stehen Ihnen die selben Funktionen wie im Task-Memory-Dump-Fenster zur Verfügung:

- Goto: wechselt zu einer neuen Basisadresse
- Display as: ändert die Darstellungsart
- Follow: nützlich zum Verfolgen von Pointern. Es wird der Wert, an dem der Cursor momentan steht, als neue Basisadresse verwendet.
- Write to file: die Daten werden in eine Datei geschrieben

4.5.4 Assembler-FPU-Fenster

Es stellt die acht Floating-Point-Register und das Status-Register der FPU dar.

4.5.5 Assembler-ICU-Fenster

Zur Darstellung der ICU-Register.

4.6 Unterschied ID.EXE und IDR.EXE

Funktionell gibt es keinen Unterschied zwischen ID und IDR. Es sind exakt die gleichen Programme, aber mit dem Unterschied, dass der ID im Protected-Mode arbeitet und dabei den vollen Erweiterungsspeicher nutzen kann, während der IDR im konventionellen Real-Mode läuft und damit an die DOS-Speicherbarriere gebunden ist. Der Grund, warum weiterhin ein Real-Mode Debugger angeboten wird, ist der, dass die Abarbeitung von Interrupt-Routinen im Real-Mode einiges schneller ist als im Protected-Mode. Falls Sie mit dem Rack arbeiten und öfters mit Kommunikationsproblemen geplagt werden, gibt es zwei Möglichkeiten zur Lösung des Problems:

- reduzieren der Baudrate
- verwenden des IDR's

Ihr PC kann in diesem Fall den Interrupt ausgelöst von der seriellen Schnittstelle nicht in der erforderlichen Zeit abarbeiten. Wie bereits erwähnt, ist der IDR an die DOS-Speicherbarriere gebunden, was zur Folge hat, dass nicht beliebig viele Fenster geöffnet werden können. Ein bisschen Optimierungsspielraum ist aber noch vorhanden, indem Sie versuchen konventionellen DOS-Speicher freizugeben (möglichst viele Treiber in den oberen Bereich laden oder nicht gebrauchte entfernen).

Beim PC-Master-Betrieb ist dieses Problem nicht vorhanden, und Sie können immer mit dem ID arbeiten. Wenn trotz dem Einsatz des IDR's (höchste Baudrate 38k4 sollte möglich sein) immer noch Kommunikationsprobleme auftreten, liegt oft ein Problem in der Konfiguration des PC's vor, z.B. nicht standardmässige Belegung von höherpriorisierten Interrupts oder Benützung von speziellen Disk-Cache Programmen. Es ist aber auch möglich, dass das Problem bei der Verdrahtung der seriellen Schnittstelle liegt. Beachten Sie dazu den Abschnitt „Verdrahtung der seriellen Schnittstelle“.

4.7 Modem-Betrieb

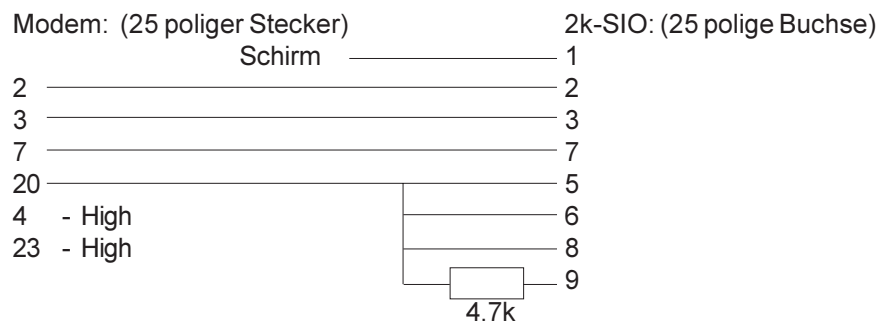
Da im Rack-Betrieb der Debugger über die serielle Schnittstelle mit dem Rack kommuniziert, ist es denkbar, ein sogenanntes Remote-Debugging über PC-Modem-Telefon-Modem-Rack zu betreiben. Für diese Betriebsart ist es von Vorteil, gewisse Punkte zu beachten. Neue Highspeed-Modems mit Übertragungsraten von über 14400 bps können ihre volle Leistungsfähigkeit erst entfalten, wenn Sie mittlere bis grössere Datenblöcke übertragen. Leider ist die Struktur des INDEL-Protokolls für diesen Fall nicht gerade optimal. Es wird synchron gearbeitet, d.h. der Sender erwartet vom Empfänger nach jedem versandten Block ein „ACK“. Dies bewirkt, dass die Senderichtung andauernd gewechselt wird und meist nur kleinere Blöcke übertragen werden. Es ist daher möglich auch bei guter Telefonleitung, dass der Debugger ziemlich träge reagiert. Beachten Sie daher folgende Punkte:

- Erstellen Sie sich eine Desktop-Datei zur Anlage. Es wird dann nicht bei jedem Hochfahren die gesamte Variablen-tabelle hochgeladen, sondern aus der Desktop-Datei entnommen.
- Schliessen Sie nicht mehr gebrauchte Memory-Dump-Fenster.
- Lassen Sie nur die gebrauchten Task- und Asm-Fenster geöffnet.
- Erhöhen Sie die Refreshungs-Zeit, aber beachten Sie, dass die Daten dann nicht mehr die Aktuellsten sein können.

Zur Erstellung dieses Betriebs sind bei den Modems gewisse Einstellungen und Initialisierungen notwendig. Es muss entweder das Modem auf der PC- oder der Rackseite in den Host-Mode geschaltet werden. Dies erreichen Sie bei gewissen Modems durch DIP-Switches; das Modem begibt sich dann per Voreinstellung beim Einschalten in den Host-Mode, oder Sie müssen das Modem mit einem Initialisierungsstring (AT-Kommando, z.B. ATA) in den Host-Mode transferieren. Mit dem Modem auf der Gegenseite muss die Verbindung aufgebaut werden. Falls dies die PC-Seite ist, verwenden Sie mit Vorteil ein Terminalprogramm dazu. Konfigurieren Sie das Terminalprogramm auf die gleiche Baudrate, die Sie beim Debugger benutzen. Wählen Sie die Telefonnummer des Rackmodems, und warten Sie bis die Verbindung aufgebaut ist. Anschliessend können Sie das Terminalprogramm verlassen (aber ohne aufzuhängen) und den Debugger starten. Der Verbindungsaufbau kann auch von der Rackseite durchgeführt werden, indem über die 2K-SIO String-Kommandos (AT&...) an das Modem gesendet werden.

Treten des öfteren Kommunikationsfehler auf, dann sollten Sie die Timeout-Zeit erhöhen. Diese können Sie in der *.ini-Datei setzen (siehe Abschnitt *INDEL.INI Datei). Der Debugger errechnet anhand der Baudrate eine standardmässige Timeout-Zeit, die im normalen Betrieb ohne Modem vollaufgenügt. Durch die Verwendung von Modems und eventuellen langen Telefonleitungen können weitere Verzögerungen entstehen. Es kann kein allgemeingültiger Wert angegeben werden, aber starten Sie einmal mit 2000ms und erhöhen Sie den Wert sukzessive in 500ms Schritten, bis die Kommunikationsfehler eliminiert sind.

4.8 Verdrahtung 2k-SIO - Modem



4.9 Kurzreferenz

DOS

- shell: - öffnet eine DOS-Shell => schliessen und zurück mit 'EXIT'
- Quit: - Debugger schliessen
- About: - zeigt Version des Debuggers sowie weitere Informationen über das Zielsystem an

Run

- Download: - lädt einen Task, der über ein Dialogfenster ausgewählt wird, in den Master
- Download all - lädt alle vorhandenen Tasks in den Master
- Verify: - verifiziert Ziel- und Quellcode
- Verify all: - wie Verify, jedoch für alle Tasks
- Execute: - startet einen Task, der Task kann aus einer Liste ausgewählt werden, oder es kann ein Task auf einer bestimmten Adresse (fixed address) gestartet werden, weitere Optionen wie „fixed number“ bestimmen die Tasknummer oder „and halt“ stoppt den Task auf dem ersten Befehl.
- Halt: - im Taskfenster: stoppt den Task; im Asm-Fenster: stoppt die CPU. Beachten Sie, dass im Taskfenster lediglich ein Task gestoppt wird und die anderen Tasks unabhängig davon weiterarbeiten, während im Asm-Fenster die CPU gestoppt wird und somit das gesamte System bzw. die ganze Anlage steht
- Go: - im Taskfenster startet den Task; im Asm-Fenster: startet die CPU
- Single Step: - führt eine Instruktion aus; im Task-Fenster eine Makroinstruktion und im Asm-Fenster eine Mikroinstruktion. Ein Single Step stoppt einen laufenden Task oder eine laufende CPU => siehe somit Auswirkungen bei „Halt“
- Step into EXP - direktes Debuggen eines Mikrounterprogrammaufrufs aus dem Makro. Die CPU wird auf der ersten Mikro-Instruktion im Mikrounterprogramm gestoppt.
- Set M(PC) to cursor: - setzt den Makroprogrammcounter bzw. Mikroprogrammcounter auf die Cursorposition in der Task bzw. Assembler-Source.
- Kill: - killt den aktivierten Task; er muss nachher explizit wieder mit „Execute“ gestartet werden
- Kill all: - wie Kill, jedoch für alle Tasks
- Reset target: - setzt das Zielsystem zurück und startet es wieder
- Reset and halt: - setzt das Zielsystem zurück, startet es wieder und stoppt es auf der ersten Instruktion. Bei aktiviertem Asm-Fenster wird im ersten Modul (Init-Modul) nach der Grundinitialisierung gestoppt, andernfalls beim ersten Makro-Befehl im ersten Task (Task 0).
- Memory - Dump: - kopiert einen Speicherblock, öffnet einen Dialog zur Eingabe der Start- und Zieladresse, Wortadressen
- Clear: - löscht einen Speicherblock mit einem Wert, öffnet einen Dialog zur Eingabe von Start-, Endadresse und Löschwert
- Burn EPROM: - erscheint nur beim Rackbetrieb, zum Brennen von EPROM's, öffnet einen Dialog zur Eingabe der Quelladresse (Daten) und der Zieladresse (EPROM)
- View - Task: - öffnet oder selektiert ein Taskfenster, der Task wird über ein Dialogfenster ausgewählt
- Task registers: - öffnet ein Registerfenster zum aktivierten Task
- Task source: - explizites Laden eines Task-Quelltextes
- Task control: - öffnet oder aktiviert das Task-Control-Fenster im aktiven Taskfenster
- Task overview: - öffnet ein Übersichtsfenster über alle Tasks und deren Status
- Task watches-add watch: - fügt einen neuen zu überwachenden Ausdruck hinzu

delete watch:	- löscht den selektierten Ausdruck
edit watch:	- editiert den selektierten Ausdruck
remove all watches:	- löscht alle überwachten Ausdrücke
Asm:	- öffnet oder selektiert das Asm-Fenster
Asm source:	- explizites Laden eines Assembler-Quelltextes
Asm CPU:	- öffnet oder aktiviert das CPU-Fenster
Asm FPU:	- öffnet oder aktiviert das FPU-Fenster (Floating-point-unit)
Asm ICU:	- öffnet oder aktiviert das ICU-Fenster (Interrupt-control-unit), nur bei Rackbetrieb
Dump:	- öffnet ein Memory-Dump-Fenster, wortadressiert im Taskfenster und byteadressiert im Asm-Fenster
Output base:	- öffnet ein Fenster zur Darstellung der Zustände der digitalen Ausgänge
Input base:	- wie „Output base“, aber für die digitalen Eingänge
Flag base:	- wie „Output base“, aber für die Flags
File:	- öffnet eine Textdatei
Breakpoint	
Toggle Breakpoint:	- setzt oder löscht einen Breakpoint innerhalb eines Task- oder Assembler Quelltextes
Search	
Find:	- Sucht den Text ab der Cursorposition nach der angegebenen Zeichenfolge ab
Search again:	- wiederholt die Suche ab der Cursorposition
Window	
Size/Move:	- zur Veränderung der Fenstergrösse und -position mit der Tastatur
Zoom:	- vergrössert das Fenster auf die volle Grösse
Next:	- aktiviert das nächste Fenster
Previous:	- aktiviert das vorhergehende Fenster
Close:	- schliesst das aktive Fenster
Close Task or Asm:	- schliesst das aktive Task- oder Asm-Fenster
Refresh all windows:	- bewirkt, dass sich alle Fenster neu darstellen
Options	
Screen Size:	- schaltet zwischen 25 und 43/50 Zeilen Darstellung um
Refresh Rate:	- setzen der Auffrischungsrate oder vollständiges deaktivieren der Auffrischung
Load Asm Def:	- explizites Laden der globalen Variablen-tabelle. Sie enthält Informationen zum Asm-Modul und Revisionsystem. Sollten diese Informationen nicht mehr dem neuesten Stand entsprechen, können Sie hier den Debugger zu einer expliziten Auffrischung veranlassen

5. INDEL Transfer (Trans.exe)

Das Transfer Tool transferiert Daten vom PC in das jeweilige Zielsystem. Mit ihm laden Sie ihr Zielsystem mit der Betriebssystemsoftware, mit Ihren Tasks, mit Ihrer Konfigurationsdatei und initialisieren und starten es. Das Trans verhält sich, egal welches Zielsystem sie benutzen, ob PCMaster, INFO-Master oder Rack, vollständig konsistent.

5.1 Einträge in der *.ini Datei

Für das Trans-Tool sind die Einträge unter [Target] und [IPS] oder [PCMaster] von Bedeutung. Weiter werden die Einträge unter [Config], wenn mit INFO-Master gearbeitet wird, für die Konfiguration von Systemen verwendet, die über die Möglichkeiten der Standardkonfiguration hinaus gehen. Für ausführlichere Erklärungen zu diesen Punkten wird auf den Abschnitt INI-Datei verwiesen.

5.2 Fehlermeldungen von Trans

29 DUAL PORT RAM R/W ACCESS.ADDRESS: XXXX,VALUE:XXXX

Ursache: Auf der gesetzten Adresse befindet sich kein RAM oder es lässt sich nicht korrekt ansprechen.

Lösung:

- Eintrag "address" unter [PCMaster] überprüfen address = XY00
- X -> Schalterstellung X der PC-Master-Karte
- Y -> Schalterstellung Y der PC-Master-Karte
- Speicherkollision mit einer anderen PC-Komponente

130 PC-MASTER DOESN'T WORK PROPERLY

Ursache: PC-Master-Karte hat einen Defekt.

Lösung: PC-Master-Karte zur Reparatur einsenden.

131 SIMULTANEOUS DPR ACCESS FAILED

Ursache: Das Bustiming ihres PC's entspricht nicht dem Standard.

Lösung: Bitte rufen Sie uns an. (Tel. CH 052 / 44 26 21).

132 PC-MASTER RAM R/W ACCESS.ADDRESS: XXXX,VALUE:XXXX

Ursache: PC-Master-Karte hat einen Defekt.

Lösung: PC-Master-Karte zur Reparatur einsenden.

133 PC-MASTER SYSTEM HORROR WHILE WRITING XXXX TO ADDRESS XXXX**Ursache:** PC-Master-Karte hat einen Defekt.**Lösung:** PC-Master-Karte zur Reparatur einsenden.**134 PC-MASTER DOESN'T START****Ursache:** PC-Master-Karte hat einen Defekt.**Lösung:** PC-Master-Karte zur Reparatur einsenden.**135 CONFIGURATIONFILE XXXX NOT FOUND****Ursache:** Beim Lesen der Konfigurationsdatei ist ein Fehler aufgetreten.**Lösung:** Konfigurationsdateiname richtig angeben**136 UNKNOWN VERSION OF CONFIGURATIONSFILE XXXX****Ursache:** Konfigurationsdatei hat ein unbekanntes Format**Lösung:****1: SYSTEMSOFTWAREFILE *.HEX IS NOT FOUND****Ursache:** Die Systemsoftware wurde nicht gefunden. Überprüfen Sie Pfad und Dateiname in der *.ini Datei**3: COMPILED MAKROFILE *.HEX NOT FOUND****Ursache:** Eine kompilierte Task-Datei wurde nicht gefunden. Überprüfen Sie Pfad und Dateiname unter [ProjectFiles] in der *.ini Datei**4: VERIFY FAILED. ADRESS:XXXX****Ursache:** Ziel- und Quellcode stimmen an der Adresse xxxx nicht überein.**Warnung****138: BATTERY IS LOW****Ursache:** Die Batterie auf der INFO-Master Karte hat eine zu tiefe Spannung oder ist gar nicht vorhanden.**139: CONFIGURATION FILE *.PCM IS INVALID****Ursachen:** - Es wird ein PCMaster mit einer *.pcm Datei konfiguriert, die das Standardlayout überschreitet (INFO-Master)

- Es wird ein PCMaster mit einer F-ADC-Karte konfiguriert.
- Das Layout der *.pcm Datei überschreitet die gesetzten Kartenübergabebereiche unter dem Eintrag [Config] in der *.ini Datei
- Ein INFO-Master wird mit Softwarezählern konfiguriert.

6. Scandep

Dieses Tool soll es Ihnen erleichtern, eigene Betriebssystemvarianten zusammenzustellen. Bis vor kurzer Zeit präsentierte sich ein solches Unterfangen etwa folgendermassen: Man erstellte ein neues Direktory, kopierte alle benötigten Sourcen dort hinein, assemblierte und linkte das Ganze und erhielt dann ein mehr oder weniger lauffähiges System. Bei Aenderungen an einzelnen Sourcen musste entweder diese explizit assembliert und nachher alles gelinkt werden, oder man rief einen Batch auf, der vorsichtshalber einmal den ganzen Haufen neu übersetzte um sicher zu gehen, dass man nichts vergessen hat. Fortgeschrittene User schrieben sich eine eigene Make-Datei, in der die ganzen Abhängigkeiten festgeschrieben waren. Sie mussten dann bei einer Aenderung nur noch das Make-Utility aufrufen und schon war die Suppe gegessen. Genau dort setzt nun das Tool SCANDEP.EXE (scan dependencies) an. Das mühsame zusammenstellen einer Make-Datei gehört ab sofort der Vergangenheit an. Diese Arbeit übernimmt nun Scandep.

Die zentrale Steuerdatei ist das Linker-Direktiven-File *.IND (z.B. INFO.IND oder IPS.IND). In dieser Datei werden ja bekannterweise alle *.OBJ-Dateien eingetragen, die zum Betriebssystem zusammengelinkt werden sollen, d.h. SCANDEP findet in dieser Datei die gesamten Zusammenhänge. Damit nun SCANDEP diese verflixten Sourcen auch findet, muss in der IND-Datei ein zusätzlicher PATH Eintrag gemacht werden. Dieser Eintrag muss folgendermassen aussehen. Das erste Zeichen MUSS ein Semikolon (;) sein (damit der Linker diese Zeile ignoriert). Gerade anschliessend kann man wie unter DOS (im Autoexec.bat) mehrere Pfade angeben, wo SCANDEP nach Sourcen bzw. Tools suchen soll, die nötig sind, um das Betriebssystem zu generieren, z.B.

```
...  
;PATH=h:\indel;c:\info6.0\firmware;c:\info6.0\user;f:\ism6.0  
...
```

- Hinweis:**
- die PATH-Zeile darf nicht länger als 80 Zeichen sein
 - es können mehrere PATH-Zeilen eingefügt werden
 - SCANDEP sucht immer von hinten nach vorn, d.h. im obigen Beispiel wird zuerst auf dem Drive F: im Verzeichnis ISM6.0 nach Sourcen gesucht.

Hat man die IND-Datei fertig editiert, so ruft man einfach z.B.

SCANDEP INFO

auf und schon macht sich Scandep auf die mühsame Dateisuche und erstellt, falls alle Dateien gefunden wurden, ein *.MAK File. In unserem Beispiel INFO.MAK. Das System kann nun mit MAKE generiert werden.

Ab sofort müssen Sie, egal welche Source-Datei sie geändert haben (auch Include-Dateien werden erkannt), nur noch MAKE aufrufen und Ihr System wird neu generiert.

Hinweis: SCANDEP muss in folgenden drei Fällen aufgerufen werden um eine neue *.MAK Datei zu erstellen:

1. Eine *.OBJ-Datei wurde aus dem *.IND File entfernt oder hinzugefügt
2. In einer Source-Datei wurde ein INCLUDE-File entfernt oder
3. Der Pfad-Eintrag im *.IND File wurde geändert.

7. Arbeiten mit Windows NT

Die Indel-Tools TRANS.EXE, ID.EXE, IDR.EXE und SHOW.EXE greifen direkt auf das DualPortRAM des PCMasters zu. Windows NT ist jedoch ein sehr restriktives Betriebssystem und lässt solche 'unschönen' Zugriffe nicht zu. Aus diesem Grund müssen zwei Treiber geladen werden, damit die obigen Tools und auch die INCO.DLL unter Windows NT verwendet werden können :

1. Installieren des Kernel Mode Drivers INCO_MEM.SYS

- kopieren Sie INCO_MEM.SYS nach %WinDir%\SYSTEM32\DRIVERS
- registrieren Sie den Treiber in der Datenbank mit REGINI INCO_MEM.INI

2. Installieren des Virtual Device Drivers INCO_VDD.DLL

- kopieren Sie INCO_VDD.DLL nach %WinDir%
- registrieren Sie den Treiber in der Datenbank mit REGEDT32.EXE unter

HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\VirtualDeviceDrivers

z.B. C:\WIN-NT\INCO_VDD.DLL

3. Setzen der Environment-Variablen PCMASTER.

- erzeugen Sie eine Environment-Variable mit dem Namen PCMASTER
- geben Sie ihr den Wert der Drehschalterstellung auf Ihrem Master

z.B. PCMASTER = CF00

Folgende Revisions werden benötigt um unter Windows NT zu arbeiten :

TRANS.EXE	1.90	oder höher
ID.EXE	2.10	oder höher
IDR.EXE	2.10	oder höher
SHOW.EXE	1.70	oder höher
INCO.DLL	1.40	oder höher

8. INDEX

A

absolut	11
anwenderspezifische Ausdrücke	10
Arbeitsweise des Debuggers	8
AsmSystemLoad	6
Assembler-CPU-Fenster	11
Assembler-Fenster	11
Assembler-FPU-Fenster	12
Assembler-Memory-Dump-Fenster	12
Assembler-Quelltext-Fenster	11
AT-Kommando	13
Auffrischungsrate	15
Auffrischungszeit	9
AutoTaskWndClose	5

B

Betriebssystem	5, 19
Betriebssystemvarianten	18
Breakpoint	10, 15
Breakpoints	11

C

Config	3
ConfigFile	3
CPU-Fenster	15
CPU-Register	11

D

Desktop-Datei	13
Desktopdatei	8
Display as	10, 12
DOS-Shell	14
DownLoad	5
DUAL PORT RAM R/W ACCESS.ADDRESS	16
Dump	15

E

EpromRetries	6
--------------------	---

F

Fehlermeldungen von Trans	16
Festkommaformat	4
Flag-Bits	11
Fliesskommaformat	4
Floating-Point-Register	12
FloatingPointValues	4
Follow	10, 12
FPU-Fenster	15

G

globale Variablen-tabelle	8
Go	14
Goto	10, 12

H

Halt	14
Host-Mode	13

I

ICU-Fenster	15
INDEL Transfer	16
INDEL-INI-Datei	3
Input base	15

K

Kommunikationsfehler	13
Kommunikationsproblemen	12
Konfigurationsdatei	3

M

MAKE	18
Make-Datei	18
Make-Utility	18
maxFlags	5
maxInputs	5
maxOutputs	5
Modem-Betrieb	13
Monitortask	5

O

Output base	15
-------------------	----

P

P_ANABAS	7
P_ANACFG	7
P_ANADEF	7
P_AXSBAS	7
P_INPBAS	7
P_IOCDOC	7
P_NROFDC	7
P_SPCCFG	7
PATH-Zeile	18
PC-MASTER RAM R/W ACCESS.ADDRESS	16
PC-MASTER SYSTEM HORROR WHILE WRITING	17
PC/AT Schnittstellen Nummer	4
ProjectFiles	6
Projektdateien	6

Protected-Mode	12	V	
Q		Verify	5
Quelltext	6	Version des Debuggers	14
R		W	
Rack-Betrieb	13	Watch-Fenster	10
Real-Mode	12	Windows NT	19
Refresh Rate	15	Write to file	10, 12
Refreshrate	9	Z	
relativ	11	Zielsystem	3
Remote-Debugging	13		
Reset and halt	14		
Reset target	14		
S			
SB, PC, FP relativ	12		
Scandep	18		
seriellen Schnittstelle	12		
seriellen Schnittstelle RS 232	8		
Show	3		
ShowAsmWndCPUHalt=	6		
SIMULTANEOUS DPR ACCESS FAILED	16		
Source-Level Debugging	8		
Speicherblock	10		
Speicherblöcke	12		
startet einen Task	14		
Status-Register	12		
Stop-Bits	4		
String-Kommandos	13		
SystemOffset	5		
SystemSoftware	4		
T			
TabSize	5		
Task watches	14		
Task-Control-Fenster	10		
Task-Fenster	9		
Task-Memory-Dump-Fenster	10		
Task-Quelltext-Fenster	10		
Task-Register	10		
Task-Register-Fenster	10		
Task-Watch-Fenster	10		
Taskfenster	14		
Time-Befehle	4		
Timeout	4		
Timeout-Zeit	13		
Timeoutzeit	4		
Transfer Tool	16		
U			
Übergabeparameter	3		
überwachten Ausdrücken	6		
Unterschied ID.EXE und IDR.EXE	12		