

i386EX System- Handbuch

ModuNORM^â

Änderungsnachweis

Änderung:	Mutationsstand:	Datum / Visum:
Erstausgabe	900330A	20.04.95 / AS
Interrupts, Konfiguration PC-386VGA	900330B	02.02.96 / BT
Konfiguration PC-386CAN, CPU-386CAN/VGA	900330C	10.12.96 / AS
Konfiguration CPU-386CAN, Fehlerkorrektur, IDOS Update	900330D	10.02.97 / AS
Memory-Mapping PC-386CAN/VGA	900330E	25.08.98 / SE
System-ROM Update	900330F	13.04.99 / MA

© Copyright 1996 durch:

ModuNORM GmbH
CH-8840 Einsiedeln
Switzerland

CoDeSys ist Warenzeichen von 3S GmbH
ModuNORM® ist Warenzeichen von ModuNORM GmbH
QVis ist Warenzeichen der Kinz Elektronik
RTXDOS und IBIOS sind Warenzeichen von Technosoftware AG
VxWorks ist Warenzeichen von WindRiver Systems
WINbloc ist Warenzeichen von Weidmüller GmbH & Co.
Windows CE ist Warenzeichen von Microsoft

Inhaltsverzeichnis**Seite:**

1.	Einleitung.....	4
1.1	Die ModuNORM® Produktstrategie.....	4
1.2	Das PC/104i System.....	4
1.3	Das I/O-System.....	6
2.	Übersicht.....	7
2.1	System-ROM Paket.....	7
2.2	System-Diskette.....	7
2.3	System-EPROM.....	9
3.	Inbetriebnahme.....	10
3.1	Installation der System-Diskette.....	10
3.2	CPU-386CAN.....	11
3.3	CPU-386VGA mit EVA-Board.....	12
3.4	CPU-386VGA ohne EVA-Board.....	12
3.5	PC-386CAN.....	13
3.6	PC-386VGA.....	13
3.7	PC(CPU)-386CAN/VGA.....	14
4.	System Übersicht.....	15
4.1	Memory-Mapping CPU-386CAN.....	15
4.2	Memory-Mapping CPU-386VGA.....	16
4.3	Memory-Mapping PC-386CAN.....	17
4.4	Memory-Mapping PC-386VGA.....	18
4.5	Memory-Mapping PC(CPU)-386CAN/VGA.....	19
4.5	Interrupt Struktur.....	20
5.	Industrial-BIOS.....	22
5.1	System Konfiguration.....	22
5.2	User Konfiguration.....	23
5.3	IBIOS.BIN Generierung.....	28
6.	Industrial-DOS.....	29
6.1	User Konfiguration.....	30
6.2	Konfigurations Beispiele.....	32
6.3	IDOS.BIN Generierung.....	33
7.	VGA-BIOS.....	34
8.	Locale-Drives.....	34
9.	System-ROM.....	35
10.	Systemerweiterungen.....	36
10.1	PC-Terminal.....	36
10.2	Tastaturkontroller 81C51SL.....	37
10.3	Flash-File.....	38
11.	Konfigurations Beispiele.....	40
12.	Tools.....	41
13.	Versionstabelle.....	43

1. Einleitung

1.1 Die ModuNORM® Produktstrategie

Der Erfolg der ModuNORM® Produktstrategie, dokumentiert durch die Vielzahl der daraus entstandenen Module, beruht auf der Realisierung von modularen funktionsbezogenen Baugruppen. Auf den kompakten steckbaren CPU-Modulen sind jeweils alle für eine stand-alone Anwendung notwendigen Funktionselemente wie MCU, RAM, ROM, RTC, Schnittstellen sowie lokale I/O's enthalten. Damit stehen für Embedded Control Anwendungen optimierte Rechnerkerne in hochintegrierter Ausführung als auf anwendungsspezifische Baugruppen aufsteckbare SMD-Module zur Verfügung.

Dank der Konzentration der komplexen Rechnerkerne auf steckbare CPU-Module kann der jeweils anwendungsspezifisch ausgeführte Grundprint in aller Regel in weniger komplexer Technik und damit kostengünstig gefertigt werden. Die standardisierten CPU-Module dagegen profitieren von der hochautomatisierten Fertigung in grossen Stückzahlen. Sämtliche ModuNORM® Baugruppen für den Europäischen Markt werden in unserem Werk in der Schweiz hergestellt.

Zur Zeit sind folgende ModuNORM® CPU-Module verfügbar:

CPU-6303	CPU-Modul mit Hitachi MCU HD6303
CPU-180	CPU-Modul mit Hitachi MCU HD64180
CPU-RTX	CPU-Modul mit Harris CPU RTX200X
CPU-332	CPU-Modul mit Motorola MCU MC68332
EVA-332	Evaluations- und Prototypenboard zu CPU-332
CPU-V25	CPU-Modul mit NEC MCU V25
CPU-386CAN	CPU-Modul mit Intel MCU i386EX und dual CAN Controller
CPU-386CAN/VGA	CPU-Modul mit Intel MCU i386EX, dual CAN und LCD Controller
CPU-1100CAN/LCD	CPU-Modul mit Intel MCU SA1100, dual CAN und LCD Controller
CPU-386VGA	CPU-Modul mit Intel MCU i386EX und LCD-Controller
EVA-386	Evaluations- und Prototypenboard zu CPU-386VGA
CPU-166	CPU-Modul mit Siemens MCU SAB80C166
EVA-166	Evaluations- und Prototypenboard zu CPU-166
CPU-167CAN	CPU-Modul mit Siemens MCU SAB80C167 CAN
EVA-167	Evaluations- und Prototypenboard zu CPU-167
GLCD	Controller für Grafik-LCD
CLCD	Controller für S/W-, Graustufen- und Farb-LCD

1.2 Das PC/104i System

Das PC/104i System ist eine modulare industrietaugliche Steuerung als absolut offenes System, welches ausschliesslich auf internationalen Normen basiert. Die Normen werden dabei konsequent auf allen Hardware-Baugruppen, Software-Modulen sowie deren Schnittstellen eingehalten. Dank der PC/104 Hardware-Norm, dem Softwarestandard der PC-Welt, sowie der Einhaltung der geltenden EMV-Normen entspricht die ModuNORM® PC/104i Steuerung ausschliesslich internationalen Normen. Dadurch ist es möglich, dieses normierte System mittels beliebigen Modulen der weltweit verschiedensten Hersteller zu kombinieren.

Dem Anwender steht damit ein normiertes, offenes, modulares und industrietaugliches Steuerungssystem zur Verfügung. Da dieses offene System auf Initialkosten wie Backplanes verzichtet, lassen sich massgeschneiderte Lösungen für Embedded Control Anwendungen kostenoptimal realisieren. Die absolute Offenheit erlaubt es, nicht nur anwendungsspezifische Peripheriemodule, sondern selbst anwendungs-optimierte Rechnermodule zu realisieren, ohne die Norm zu verlassen.

Die Baugruppen dieses Systems sind auf der kompakten Fläche von 96 x 90 mm² (PC/104 Formfaktor) aufgebaut und über den 104-poligen PC/AT-Bus stackartig miteinander verbunden. Alle PC/104i Module weisen ausschliesslich codierte und verriegelbare industrietaugliche Steckverbinder auf, und lassen sich in ein sehr kompaktes, modulares und EMV-gerechtes Gehäuse integrieren. Es werden drei verschiedene Einbauvarianten unterstützt.

DIN-Schienen Montage:

Das PC/104i Gehäuse unterstützt dank seiner kompakten Bauform das Aufschnappen auf die DIN-Schiene in optimale Weise. Mit einer Länge von 105 mm und einer Breite von 99 mm sind die Aussenabmessungen nur unwesentlich grösser als diejenigen des PC/104 Formats. Die Bauhöhe richtet sich dabei nach der Anzahl der aufgesteckten Module. Durch die Verwendung einer entsprechenden Basis-Baugruppe lassen sich die verschiedensten Klemmenmodule direkt ankoppeln.

Front-Einbau:

Alle PC/104i Bedieneinheiten erlauben den Einbau in eine beliebige Schaltschrankfront mit Schutzklasse IP65. Sie enthalten eine hinterleuchtete grafikfähige LCD-Anzeige sowie eine industrietaugliche Tastatur mit Kurzhubtasten und/oder Touch-screen. Da die Rechereinheit jeweils als PC/104i Modul direkt auf der Rückseite aufgesteckt ist, lässt sich die Rechnerleistung durch die Wahl der entsprechenden Rechereinheit auf einfachste Weise den Anforderungen anpassen. Durch Aufstecken von weiteren PC/104i Peripherie-Modulen können die Bedieneinheiten kostenoptimal zu kompakten, industrietauglichen Kleinststeuerungen ausgebaut werden.

Geräte-Einbau:

Zum direkten Einbau in Geräte kann eine PC/104i Baugruppe auf einen anwendungsspezifischen Grundprint aufgesteckt werden. Dank der Konzentration der komplexen Schaltungsteile auf steckbare PC/104i Module kann die Grundplatine in aller Regel in weniger komplexer Technik und damit kostengünstig gefertigt werden. Die Herstellkosten der standardisierten PC/104i Module dagegen profitieren von der hochautomatisierten Fertigung in grossen Stückzahlen. Ausserdem lassen sich optional angebotene Zusätze wie Schnittstellen, Speicher oder I/O's bei Bedarf einfach aufstecken.

Zur Zeit sind folgende ModuNORM® PC-Module verfügbar:

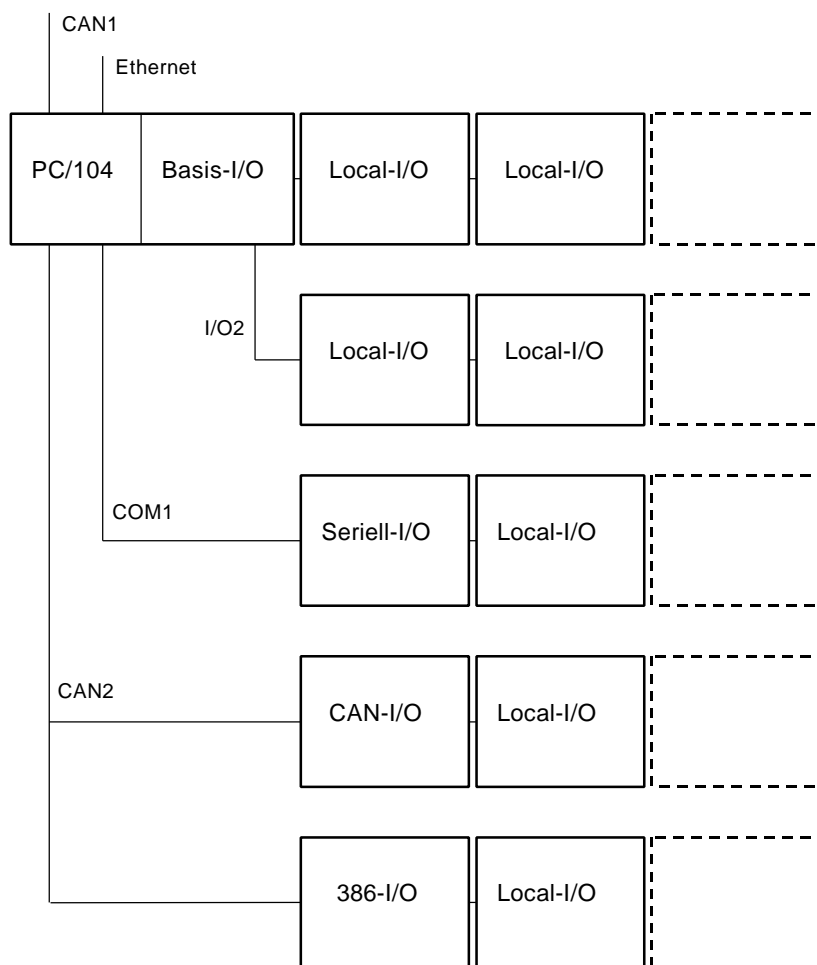
PC-386CAN	PC/104i CPU mit Intel i386EX, zwei COM und zwei CAN Schnittstellen
PC-1100CAN/NET	PC/104i CPU mit Intel SA1100, drei COM, zwei CAN und Ethernet Schnittstelle
PC-386VGA	PC/104i CPU mit Intel i386EX, zwei COM und LCD Schnittstelle
PC-1100LCD/NET	PC/104i CPU mit Intel SA1100, drei COM, LCD und Ethernet Schnittstelle
PC-386CAN/VGA	PC/104i CPU mit Intel i386EX, zwei COM, zwei CAN und LCD Schnittstelle
PC-1100CAN/LCD/NET	PC/104i CPU mit Intel SA1100, drei COM, CAN, LCD und Ethernet Schnittstelle
DP-386CAN	PC/104i Adapter mit Intel i386EX und zwei CAN Schnittstellen
DP-386COM	PC/104i Adapter mit Intel i386EX und zwei High-speed COM Schnittstellen
PC-Adapter Dual-CAN	PC/104i Adapter mit zwei CAN Schnittstellen
PC-Adapter Quad-COM	PC/104i Adapter mit vier COM Schnittstellen
PC-Adapter Ethernet	PC/104i Adapter mit Ethernet 10Base2 Schnittstelle
PC-Adapter PCMCIA	PC/104i Adapter mit PCMCIA Einschub für PC-Cards bis Typ 3
PC-I/O Digital	PC/104i Adapter mit je 16 digitalen Ein- und Ausgängen 24 Volt
PC-Basis WINbloc	PC/104i Basis Speisung mit Anschluss für WINbloc Reihenklemmen I/O's
PC-Basis Terminal	PC/104i Basis Speisung mit Anschluss für Matrix-Tastatur
PC-Handy	Handheld-PC mit Grafik-LCD, Tastatur, CAN und COM Schnittstellen
PC-Terminal Mini	PC/104i Bedieneinheit mit Grafik-LCD, Tastatur, CAN und COM Schnittstellen
PC-Terminal	PC/104i Bedieneinheit mit ¼ VGA s/w oder Farb-STN LCD-Anzeige
PC-Terminal Touch	PC/104i Bedieneinheit mit ¼ VGA s/w LCD-Anzeige und Touch-screen
PC-Terminal VGA	PC/104i Bedieneinheit mit VGA Farb-TFT LCD-Anzeige und Touch-screen

1.3 Das I/O-System

Das I/O-System ergänzt die ModuNORM® PC-104i Steuerung zu einer Klemmen SPS, welche Einheiten mit bis zu 240 digitalen Ein- und 240 digitalen Ausgängen kombiniert. Sie besteht aus einer PC/104i Basisbaugruppe sowie direkt anreihbaren Klemmenmodulen. Alle Ein- und Ausgänge sind einzeln galvanisch getrennt auf Federzugklemmen geführt. Deren logische Zustände werden über LED's angezeigt. Alle Ausgänge sind Kurzschluss und Überlastsicher. Eine Abschaltung infolge Überstrom- bzw. Übertemperatur oder offene Last infolge Drahtbruch wird signalisiert.

Die 16 digitalen Ein- und Ausgänge jedes Klemmenmoduls sind in zwei getrennt gespeisten 8-er Gruppen angeordnet. Das Anreihen der Klemmenmodule erfolgt über einen seriellen I/O-Bus, welcher das Einlesen aller Eingänge in weniger als 2 ms erlaubt.

Auf der PC/104i Basisbaugruppe PC-Basis I/O sind bereits 16 digitale Ein- und Ausgänge integriert. Je vier dieser Ein- und Ausgänge sind mit kurzen Ansprechzeiten realisiert. Ausserdem können schnelle lokale Verknüpfungen in einem ladbaren PAL realisiert werden.



Zur Zeit sind folgende ModuNORM® I/O-Module verfügbar:

PC-Basis I/O Digital
Local-I/O Digital
Seriell-I/O Digital
CAN-I/O Digital
386-I/O Digital
I/O-Tester

PC/104i Basis Speisung mit je 16 digitalen Ein- und Ausgängen
anreihbares Erweiterungsmodul mit je 16 digitalen Ein- und Ausgängen
über serielle Schnittstelle anschliessbares Klemmenmodul
über CAN anschliessbares Klemmenmodul mit je 16 digitalen Ein- und Ausgängen
Stand-alone Klemmen SPS mit je 16 digitalen Ein- und Ausgängen
steckbares Testmodul mit Schaltern zur Simulation der digitalen Eingänge

2. Übersicht

Diese Beschreibung gilt für folgende ModuNORM® Programme:

ab Artikel-Nr:	Benennung:	Bemerkung:
001210K	System-Diskette i386EX	3½"

Achtung:

Wir behalten uns Änderungen zur Verbesserung unserer Produkte ausdrücklich vor. Dies trifft vor allem auf Maskenänderungen der verwendeten Controller zu, welche ohne Vorankündigung in die Serienprodukte einfließen können.

2.1 System-ROM Paket

Das System-ROM Paket für die ModuNORM® PC/104 Module mit dem i386EX enthält:

1 Stück	i386EX System-Handbuch	Artikel-Nr. 90033	(Dieses Handbuch)
1 Stück	System-Diskette 3½ Zoll	Artikel-Nr. 00121	

2.2 System-Diskette

Die System-Diskette enthält alle notwendigen Files, um auf Ihrem PC anwendungsspezifische System-(Flash)EPROM's zu erzeugen.

Folgende Funktionen können vom Anwender parametrisiert werden:

- Chip-select Unit des i386EX
- Versorgungsspannungen VPOS, VNEG für die LCD-Anzeige
- Schnittstellendaten der LCD-Anzeige
- Schnittstellendaten der Remote-Console (Terminalprogramm)
- Definition der Devices DEV_A und DEV_B
- Integration von Files in die definierten Devices
- Tastaturkontroller
- PC-Terminal
- Flash-File System

Inhalt der System-Diskette

Directory Flash:

Filename:	Erläuterung:
RTFFLAH.EXE	Flash Device Treiber
RTFUTI.EXE	Flash Formatierungs Programm
RTFFLASH.INI	Konfigurations File für Flash Device Treiber

Directory IBIOS:

Filename:	Erläuterung:
DC_DC.BIN	Industrial BIOS-Modul für DC/DC Spannungen
DS1302.BIN	Industrial BIOS-Modul für RTC
IBIOS.BIN	Industrial BIOS Binary File
IBIOS.INI	Industrial BIOS Konfigurations File (Kopie von aktuellen Projekt)
IBIOSCNF.CNF	Industrial BIOS Konfigurations Datei
IBIOSCNF.EXE	Industrial BIOS Konfigurations Tool
I386EX.BIN	Industrial BIOS Main-Modul
IB_KEY.BIN	Industrial BIOS-Modul für Keyboard Interface
IB_RTX.BIN	Industrial BIOS-Modul für RTX Emulator
KBMATRIX.BIN	Industrial BIOS-Modul für Tastaturmatrix
KB8151.BIN	Industrial BIOS-Modul für Keyboardcontroller 81C51
8x8_128.BIN	BGI 8x8 Zeichensatz

Directory IDOS:

Filename:	Erläuterung:
IDOS.BIN	Industrial DOS Binary File
IDOSCONF.CNF	Industrial DOS Konfigurations Datei
IDOSCONF.EXE	Industrial DOS Konfigurations Tool
IDOS.INI	Industrial DOS Konfigurations File ((Kopie von aktuellen Projekt)

Directory PROJEKT:

Directoy:	Erläuterung:
CPU386C1	Files zum Generieren von 256kByte ROM-File für des Modul CPU-386CAN (Vers. 1)
CPU386C2	Files zum Generieren von 256kByte ROM-File für des Modul CPU-386CAN (Vers. 2)
CPU386C3	Files zum Generieren von 256kByte ROM-File für des Modul CPU-386CAN (Vers. 3)
CPU386C4	Files zum Generieren von 256kByte ROM-File für des Modul CPU-386CAN (Vers. 4)
CPU386C5	Files zum Generieren von 256kByte ROM-File für des Modul CPU-386CAN (Vers. 5)
CPU386V1	Files zum Generieren von 256kByte ROM-File für das Modul CPU-386VGA
CPU386V2	Files zum Generieren von 512kByte ROM-File für das Modul CPU-386VGA
CPU386V3	Files zum Generieren von 256kByte FlashROM-File für das Modul CPU-386VGA
PC386C1	Files zum Generieren von 256kByte PROM-File für das Modul PC-386CAN
PC386C2	Files zum Generieren von 512kByte ROM-File für das Modul PC-386CAN
PC386C3	Files zum Generieren von 256kByte FlashROM-File für das Modul PC-386CAN
PC386CV1	Files zum Gen. von 256kByte FlashROM-File für das Modul PC(CPU)-386CAN/VGA
PC386V1	Files zum Generieren von 256kByte ROM-File für das Modul PC-386VGA
PC386V2	Files zum Generieren von 512kByte ROM-File für das Modul PC-386VGA
PC386V3	Files zum Generieren von 256kByte FlashROM-File für das Modul PC-386VGA
TOUCHCV1	Files zum Gen. von 256kByte FlashROM-F. für Modul PC-TERMINAL TOUCH CAN
TOUCHV1	Files zum Gen. von 256kByte FlashROM-File für das Modul PC-TERMINAL TOUCH

Directory SID:

Filename:	Erläuterung:
SID.EXE	Singel-Task Debugger (PC-Modul)
SID_ERR.*	Singel-Task Debugger Error Text Files in English und Deutsch
SID.HELP.*	Singel-Task Debugger Help Text Files in English und Deutsch
SIDTGPC.EXE	Singel-Task Debugger (Target-Modul)

Directory TOOLS:

Filename:	Erläuterung:
C_REMOTE.EXE	Device-Treiber für PC-Laufwerke als Remote-Drives des PC/104 Moduls
DEVICES.EXE	Zeigt die Konfiguration der IDOS-Devices des System-ROM's an
EDIT.EXE	Einfacher Text Editor
FORMAT.EXE	Formatierungsprogramm für RAM-Devices (FAT)
MEM.EXE	Zeigt die Memorybelegung des System-ROM's an
MERGEVIN.EXE	Konfigurations Tool zur Erzeugung des System-ROM
OUTMODE.EXE	Umschaltung der Anzeige auf PC-Monitor und/oder lokalen LC-Display
RDCON.EXE	Terminalprogramm für Remote-Console
RTFDIR.EXE	Zeigt die Files im erstellten Device (DEV_A oder DEV_B) an
RTFMKDEV.EXE	Konfigurations Tool zur Erzeugung der Device-Files DEV_A und DEV_B

Directory VGA:

Filename:	Erläuterung:
62x5.BIN	VGABIOS Binary Hilfsfile
62x5.INI	VGABIOS Konfigurations File
VGABIOS.BIN	VGABIOS Binary File von Cirrus-Logic
OEMSI.EXE	VGABIOS Konfigurations Tool
PATCH.EXE	VGABIOS Hilfs-Konfigurations Tool
001310A.BIN	Konfiguriertes VGABIOS Binary-File
001320A.BIN	Konfiguriertes VGABIOS Binary-File
001320B.BIN	Konfiguriertes VGABIOS Binary-File
001330A.BIN	Konfiguriertes VGABIOS Binary-File
001340A.BIN	Konfiguriertes VGABIOS Binary-File
001480A.BIN	Konfiguriertes VGABIOS Binary-File
OEMDATA.TXT	VGABIOS Text File

2.3 System-EPROM

Mit Hilfe der System-ROM Diskette können auf einfache Weise System EPROM's (EVN + ODD) für die ModuNORM® PC/104 Module mit dem i386EX erzeugt werden. Mit diesen EPROM's kann eine Inbetriebnahme gemäss Kapitel 3 erfolgen.

Das System-ROM beinhaltet folgende Funktionen:

- IBIOS** Das Industrial Basic Input/Output System IBIOS von Technosoftware initialisiert die Hardware nach dem Einschalten und stellt während des Betriebes die Interrupt Funktionen zur Verfügung.
- VGABIOS** Das VGA-BIOS von Cirrus Logic initialisiert den VGA-Controller GD6235 bzw. GD6245 gemäss dem konfigurierten LCD-Display.
- IDOS** Das Industrial Disc Operating System IDOS von Technosoftware ist kompatibel zum Industrie-Standard. Damit kann auf der MPU i386EX so gearbeitet werden wie auf einem PC.
- DEV_A** Anwendungsspezifisches Device-File enthaltend:
- autoexec.bat
 - config.sys
 - sidtgp.exe
 - c_remote.exe
 - devices.exe
 - mem.exe

3. Inbetriebnahme

Da die ModuNORM® PC/104 Module mit dem i386EX in der Regel über kein on-board Interface für eine PC-Tastatur verfügen, können keine Eingaben über eine direkt angeschlossene Tastatur erfolgen. Die Funktionalität einer Konsole (Tastatur und Bildschirm) kann jedoch auf einen PC ausgelagert werden. Dazu ist das PC/104 Modul über eine serielle RS232 Schnittstelle mit einem PC zu verbinden, und ein entsprechendes Terminalprogramm auf dem PC zu starten. Über dieselbe serielle Schnittstelle wird dem PC/104 Modul auch der Zugriff auf Remote Drives, das Floppy-Laufwerk und die Harddisk des angeschlossenen PC ermöglicht.

Vorsicht:

Diese Baugruppen enthalten Bauelemente, welche auf statische Entladungen empfindlich sind. Um eine Beschädigung der Baugruppen zu vermeiden, sind die entsprechenden Vorschriften zur Verpackung und Handhabung unbedingt zu beachten.

Der Einbau der Baugruppen in Geräte hat unter Berücksichtigung sämtlicher in den Destinationenländern anwendbarer Normen und Vorschriften zu erfolgen. Entsprechende Massnahmen zur Erfüllung solcher Anforderungen (z.B. betreffend EMV, EMB, usw.) sind durch den Hersteller dieser Geräte zu treffen.

Achtung:

Diese Baugruppen werden unter Verwendung von hochintegrierter SMD Technologie gefertigt. Eine mechanische Belastung der Bauelemente ist in keinem Falle zulässig. Um eine Beschädigung der Baugruppe zu vermeiden, ist beim Einstecken oder Ausziehen von Speicherbausteinen eine Belastung der Lötstellen der DIL-Stecksockel in SMD Technologie unbedingt zu verhindern. Ein entsprechendes Ausziehwerkzeug ist lieferbar.

Um die Anzahl der Steckzyklen der DIL-Stecksockel in der Entwicklungsphase auf ein Minimum zu beschränken, ist ein entsprechend optimierter ROM-Simulator lieferbar. Er ersetzt die aufwendigen Programmier- und Löschkzyklen von EPROM's durch komfortablen Download ab einer seriellen Schnittstelle des PC. Dadurch kann nicht nur die Qualität gesteigert, sondern auch die Entwicklungszeit verkürzt werden.

3.1 Installation der System-Diskette

- 1) Ein Zielverzeichnis (z.B. C:\386EX) auf der Harddisk erzeugen
- 2) Das File 00121xx.EXE von der System-Diskette in das Zielverzeichnis auf der Harddisk kopieren
- 3) In das Zielverzeichnis auf der Harddisk wechseln und 00121xx -d <Enter> eingeben
⇒ alle Files werden jetzt automatisch ausgepackt und im Zielverzeichnis gespeichert
- 4) Das File 00121xx.EXE im Zielverzeichnis auf der Harddisk löschen

3.2 CPU-386CAN

- 1) Serielle Schnittstelle COM1 (bzw. COM2) des eingeschalteten PC's über ein RS232 Null-Modem Schnittstellenkabel mit der Schnittstelle COM2 des ausgeschalteten Moduls CPU-386CAN verbinden.
- 2) System-EPROM's mit Hilfe der System-ROM Diskette erzeugen.
Durch Aufrufen der Datei make.bat im Directoy PROJEKT/CPU386Cx/INI wird das Binary-File ccpx.bin im Directory PROJEKT/CPU386Cx/PROM erzeugt.
- 3) System-EPROM's (EVEN + ODD) auf einem EPROM-Programmiergerät programmieren.
- 4) System-EPROM's auf die DIL-Stecksockel des ausgeschalteten Moduls CPU-386CAN aufstecken (EVEN in Sockel U5 ODD in Sockel U4).
- 5) Das Terminalprogramm RDCON auf dem PC durch Eingabe von **rdcon /c1** (bzw. **rdcon /c2** bei Anschluss an COM2 des PC) starten.
- 6) Speisung des Moduls CPU-386CAN einschalten.

Damit werden dem Modul CPU-386CAN über die serielle RS232 Schnittstelle sowohl die Tastatur, der Bildschirm, als auch die Laufwerke des angeschlossenen PC zur Verfügung gestellt. Die Laufwerke (Remote Drives) des PC/104 Moduls CPU-386CAN (Bestückungsvariante 4) sind wie folgt definiert:

Beschreibung:	Drives auf CPU-386CAN:	Drives auf PC:
Extended Mode ROM Drive (DEV_A)	A:	
Extended Mode FlashROM Drive	B:	
Harddisk auf dem PC	C:	C:
Floppy auf dem PC	D:	A:

3.3 CPU-386VGA mit EVA-Board

- 1) PC/104 Modul CPU-386VGA auf das Evaluationsboard EVA-386 aufstecken.
- 2) serielle Schnittstelle COM1 (bzw. COM2) des eingeschalteten PC's über ein RS232 Null-Modem Schnittstellenkabel mit der Schnittstelle COM2 des ausgeschalteten Evaluationsboards verbinden.
- 3) System-EPROM's mit Hilfe der System-ROM Diskette erzeugen.
Durch Aufrufen der Datei make.bat im Directory PROJEKT/CPU386V1/INI wird das Binary-File cvp1.bin im Directory PROJEKT/CPU386V1/PROM erzeugt.
- 4) System-EPROM's (EVEN + ODD) auf einem EPROM-Programmiergerät programmieren.
- 5) System-EPROM's auf die DIL-Stecksocket des ausgeschalteten PC/104 Moduls CPU-386VGA aufstecken (EVEN in Sockel U5 und ODD in Sockel U4).
- 6) Das Terminalprogramm RDCON auf dem PC durch Eingabe von **rdcon /c1** (bzw. **rdcon /c2** bei Anschluss an COM2 des PC) starten.
- 7) Speisung des Evaluationsboards einschalten.

Damit werden dem PC/104 Modul CPU-386VGA über die serielle RS232 Schnittstelle sowohl die Tastatur, der Bildschirm, als auch die Laufwerke des angeschlossenen PC zur Verfügung gestellt. Die Laufwerke (Remote Drives) des PC/104 Moduls CPU-386VGA sind wie folgt definiert:

Beschreibung:	Drives auf CPU-386VGA:	Drives auf PC:
Extended Mode ROM Drive (DEV_A)	A:	
Harddisk auf dem PC	B:	C:
Floppy auf dem PC	C:	A:

3.4 CPU-386VGA ohne EVA-Board

- 1) serielle Schnittstelle COM1 (bzw. COM2) des eingeschalteten PC's über ein RS232 Null-Modem Schnittstellenkabel mit der Schnittstelle COM2 des ausgeschalteten PC/104 Moduls CPU-386VGA verbinden.
- 2) System-EPROM's mit Hilfe der System-ROM Diskette erzeugen.
Durch Aufrufen der Datei make.bat im Directory PROJEKT/CPU386V1/INI wird das Binary-File cvp1.bin im Directory PROJEKT/CPU386V1/PROM erzeugt.
- 3) System-EPROM's (EVEN + ODD) auf einem EPROM-Programmiergerät programmieren.
- 4) System-EPROMs auf die DIL-Stecksocket des ausgeschalteten PC/104 Moduls CPU-386VGA aufstecken (EVEN in Sockel U5 und ODD in Sockel U4).
- 5) Das Terminalprogramm RDCON auf dem PC durch Eingabe von **rdcon /c1** (bzw. **rdcon /c2** bei Anschluss an COM2 des PC) starten.
- 6) Speisung des PC/104 Moduls CPU-386VGA einschalten.

Damit werden dem PC/104 Modul CPU-386VGA über die serielle RS232 Schnittstelle sowohl die Tastatur, der Bildschirm, als auch die Laufwerke des angeschlossenen PC zur Verfügung gestellt. Die Laufwerke (Remote Drives) des PC/104 Moduls CPU-386EX sind wie folgt definiert:

Beschreibung:	Drives auf CPU-386VGA:	Drives auf PC:
Extended Mode ROM Drive (DEV_A)	A:	
Harddisk auf dem PC	B:	C:
Floppy auf dem PC	C:	A:

3.5 PC-386CAN

- 1) Serielle Schnittstelle COM1 (bzw. COM2) des eingeschalteten PC's über ein RS232 Null-Modem Schnittstellenkabel mit der Schnittstelle COM2 des ausgeschalteten PC/104 Modules PC-386CAN verbinden.
- 2) System-EPROM's mit Hilfe der System-ROM Diskette erstellen.
Durch Aufrufen der Datei make.bat im Directory PROJEKT/PC386C1/INI wird das Binary-File pcp1.bin im Directory PROJEKT/PC386C1/PROM erzeugt.
- 3) System-EPROM's (EVEN + ODD) auf einem EPROM-Programmiergerät programmieren.
- 4) System-EPROM's auf die DIL-Stecksocket des ausgeschalteten PC/104 Moduls PC-386CAN aufstecken (EVEN in Sockel U5 und ODD in Sockel U4).
- 5) Das Terminalprogramm RDCON auf dem PC durch Eingabe von **rdcon /c1** (bzw. **rdcon /c2** bei Anschluss an COM2 des PC) starten.
- 6) Speisung des PC/104 Moduls PC-386CAN einschalten.

Damit werden dem PC/104 Modul PC-386CAN über die serielle RS232 Schnittstelle sowohl die Tastatur, der Bildschirm, als auch die Laufwerke des angeschlossenen PC zur Verfügung gestellt. Die Laufwerke (Remote Drives) des PC/104 Moduls PC-386CAN sind wie folgt definiert:

Beschreibung:	Drives auf PC-386CAN:	Drives auf PC:
Extended Mode ROM Drive (DEV_A)	A:	
Extended Mode SRAM Drive (DEV_B)	B:	
Harddisk auf dem PC	C:	C:
Floppy auf dem PC	D:	A:

3.6 PC-386VGA

- 1) Serielle Schnittstelle COM1 (bzw. COM2) des eingeschalteten PC's über ein RS232 Null-Modem Schnittstellenkabel mit der Schnittstelle COM2 des ausgeschalteten PC/104 Modules PC-386VGA verbinden.
- 2) System-EPROM's mit Hilfe der System-ROM Diskette erzeugen.
Durch Aufrufen der Datei make.bat im Directory PROJEKT/PC386V1/INI wird das Binary-File pvp1.bin im Directory PROJEKT/PC386V1/PROM erzeugt.
- 3) System-EPROM's (EVEN + ODD) auf einem EPROM-Programmiergerät programmieren.
- 4) System-EPROM's auf die DIL-Stecksocket des ausgeschalteten PC/104 Moduls PC-386VGA aufstecken (EVEN in Sockel U5 ODD in Sockel U4).
- 5) Das Terminalprogramm RDCON auf dem PC durch Eingabe von **rdcon /c1** (bzw. **rdcon /c2** bei Anschluss an COM2 des PC) starten.
- 6) Speisung des PC/104 Moduls PC-386VGA einschalten.

Damit werden dem PC/104 Modul PC-386VGA über die serielle RS232 Schnittstelle sowohl die Tastatur, der Bildschirm, als auch die Laufwerke des angeschlossenen PC zur Verfügung gestellt. Die Laufwerke (Remote Drives) des PC/104 Moduls PC-386VGA sind wie folgt definiert:

Beschreibung:	Drives auf PC-386VGA:	Drives auf PC:
Extended Mode ROM Drive (DEV_A)	A:	
Extended Mode SRAM Drive (DEV_B)	B:	
Harddisk auf dem PC	C:	C:
Floppy auf dem PC	D:	A:

3.7 PC(CPU)-386CAN/VGA

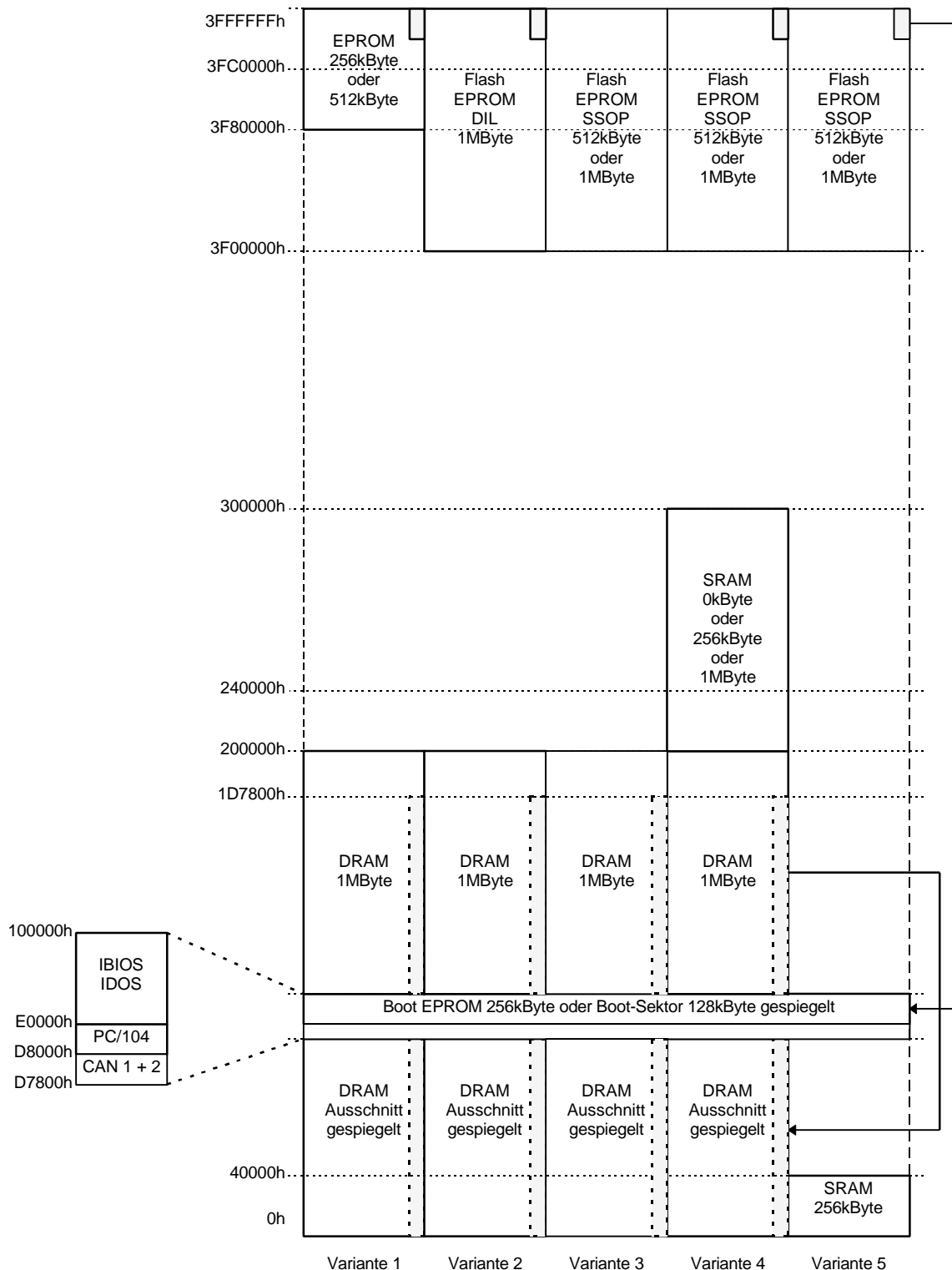
- 1) Serielle Schnittstelle COM1 (bzw. COM2) des eingeschalteten PC's über ein RS232 Null-Modem Schnittstellenkabel mit der Schnittstelle COM2 des ausgeschalteten PC/104 Modules CPU-386CAN/VGA verbinden.
- 2) System-Flash File's mit Hilfe der System-ROM Diskette erstellen.
Durch Aufrufen der Datei make.bat im Directory PROJEKT/CP386CV1/INI wird das Binary-File ccvp1.bin im Directory PROJEKT/CP386CV1/PROM erzeugt.
- 3) Flash mit Hilfe des Bootloaders Programmieren .
- 4) Das Terminalprogramm RDCON auf dem PC durch Eingabe von **rdcon /c1** (bzw. **rdcon /c2** bei Anschluss an COM2 des PC) starten.
- 5) Speisung des PC/104 Modules CPU-386CAN/VGA einschalten.

Damit werden dem PC/104 Modul CPU-386CAN/VGA über die serielle RS232 Schnittstelle sowohl die Tastatur, der Bildschirm, als auch die Laufwerke des angeschlossenen PC zur Verfügung gestellt. Die Laufwerke (Remote Drives) des PC/104 Moduls CPU-386CAN/VGA sind wie folgt definiert:

Beschreibung:	Drives auf CPU-386CAN/VGA:	Drive auf PC:
Extended Mode ROM Drive (DEV_A)	A:	
Extended Mode DRAM Drive (DEV_B)	B:	
Extended Mode FlashROM Drive	C:	
Extended Mode FlashROM Drive	D:	
Harddisk auf dem PC	E:	C:
Floppy auf dem PC	F:	A:

4. System Übersicht

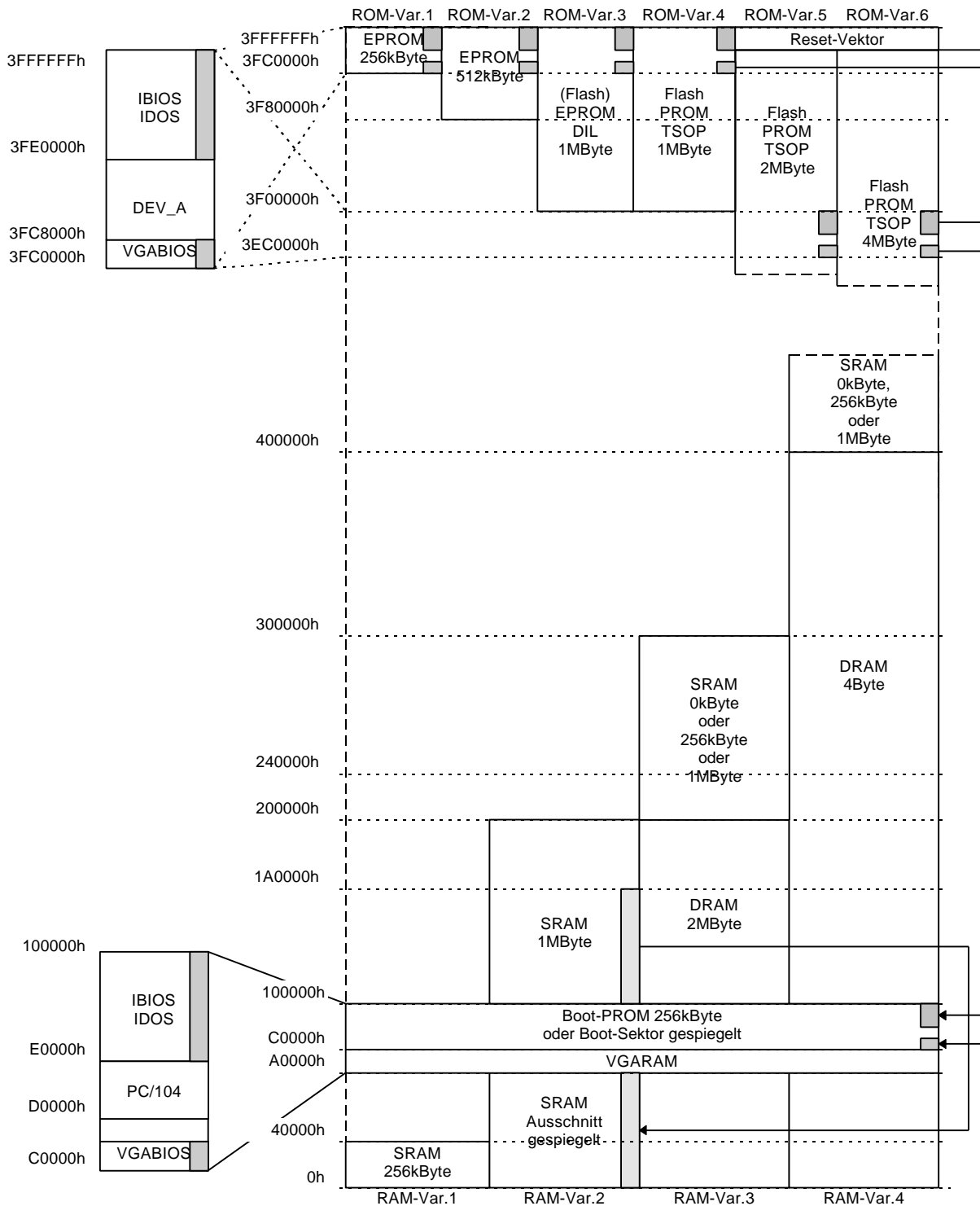
4.1 Memory-Mapping CPU-386CAN



Das Modul CPU-386CAN erlaubt fünf Bestückungsvarianten. Die Bestückungsvarianten unterscheiden sich im Wesentlichen durch die Nutzung der DIL-Sockel für RAM, EPROM oder Flash-EPROM Bausteine.

Bei der minimalen Bestückung von 256 kByte EPROM wird dem Anwender ein real-mode ROM-Drive (DEV_A) von 128 kByte zur Verfügung gestellt. Je grösser das (Flash-)EPROM ist, desto mehr Speicher steht dem Anwender in einem extended ROM Drive (DEV_B) zur Verfügung.

4.2 Memory-Mapping CPU-386VGA



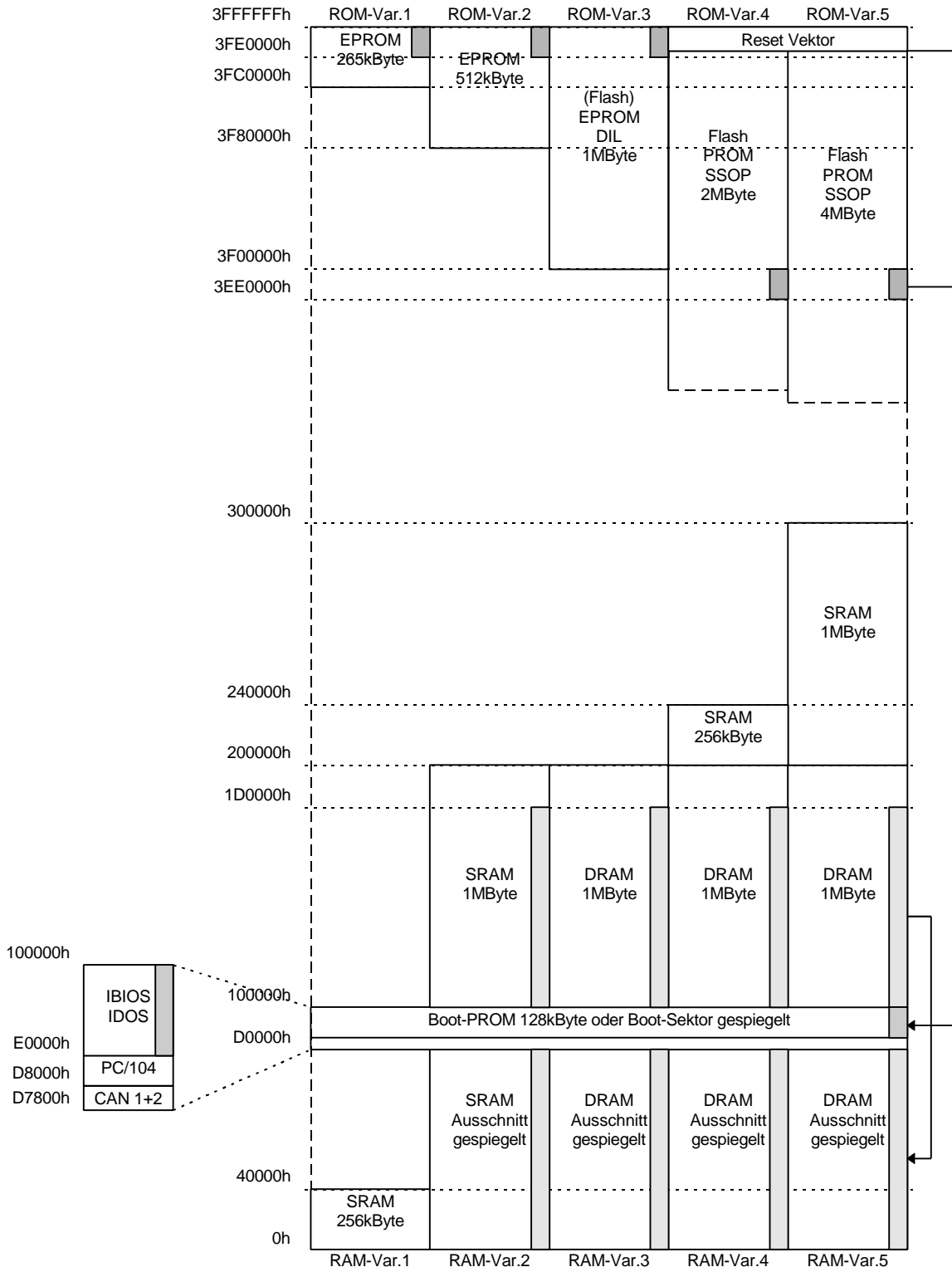
Das PC/104 Modul CPU-386VGA erlaubt die vier RAM Bestückungsvarianten mit 256 kByte bzw. 1 MByte SRAM, 2 MByte DRAM mit 256 kByte (1 MByte) SRAM sowie die sechs (Flash-)EPROM Bestückungsvarianten mit 256 kByte, 512 kByte bzw. 1 MByte DIL (Flash-)EPROM oder 1 MByte, 2 MByte bzw. 4 MByte SMD Flash-EPROM.

Bei der minimalen Bestückung von 256 kByte (Flash-)EPROM wird dem Anwender ein real-mode ROM-Drive (DEV_A) von 64 kByte zur Verfügung gestellt. Je grösser das (Flash-)EPROM ist, desto mehr Speicher steht dem Anwender in einem extended ROM Drive (DEV_B) zur Verfügung.

Achtung:

Werden EPROM's vom Typ 27C040 eingesetzt, so muss der 0Ω Widerstand R2 anstelle des 0Ω Widerstandes R1 bestückt werden.

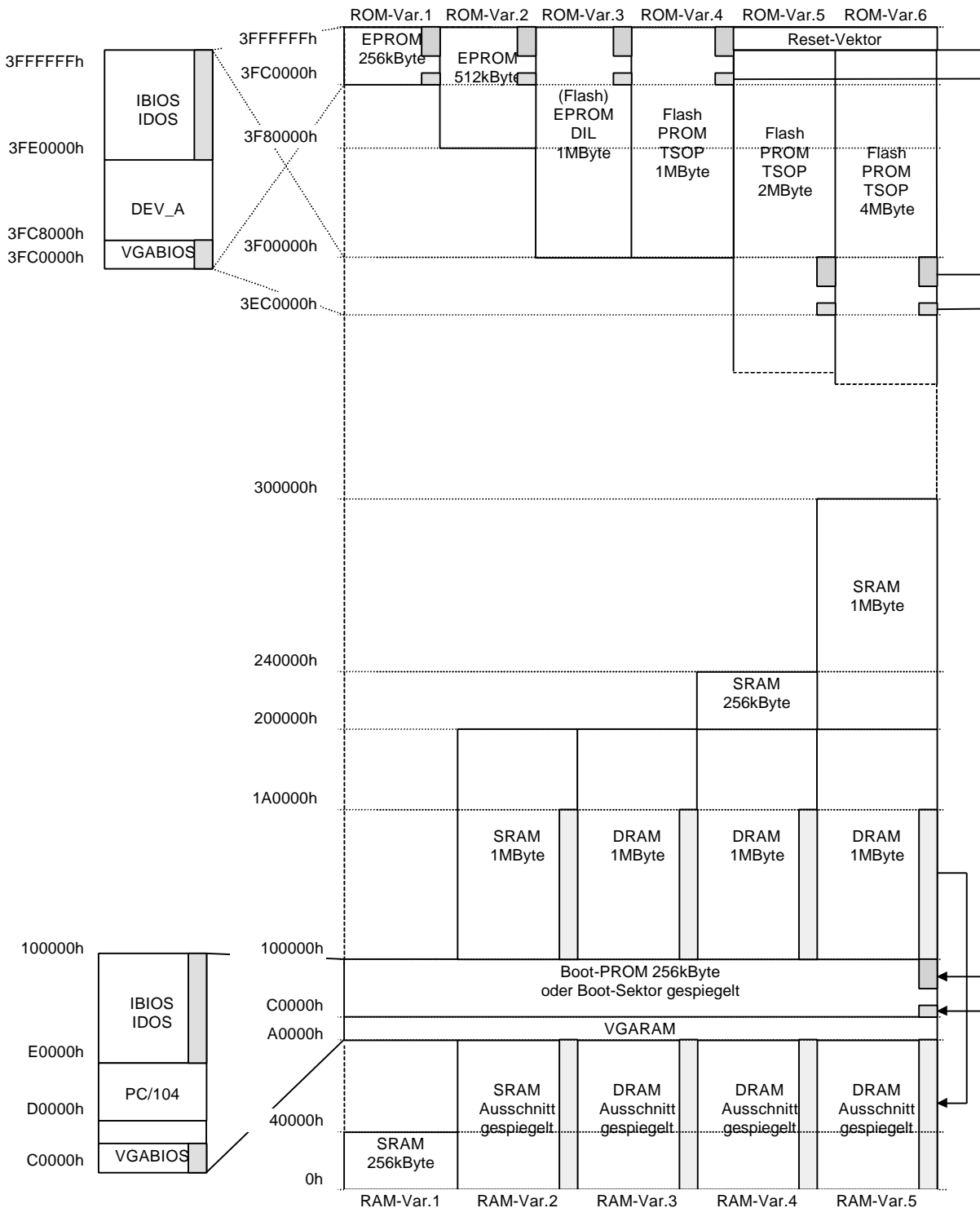
4.3 **Memory-Mapping PC-386CAN**



Das PC/104 Modul PC-386CAN erlaubt die fünf RAM Bestückungsvarianten mit 256 kByte bzw. 1 MByte SRAM, 1 MByte DRAM, 1 MByte DRAM mit 256 kByte bzw. 1MByte SRAM sowie die fünf (Flash-)EPROM Bestückungsvarianten mit 256 kByte, 512 kByte bzw. 1 MByte DIL-Flash-EPROM oder 2 MByte bzw. 4 MByte SMD Flash-EPROM.

Bei der minimalen Bestückung von 256 kByte EPROM wird dem Anwender ein real-mode ROM-Drive (DEV_A) von 64kByte zur Verfügung gestellt. Je grösser das (Flash-)EPROM ist, desto mehr Speicher steht dem Anwender in einem extended ROM Drive (DEV_B) zur Verfügung.

4.4 Memory-Mapping PC-386VGA



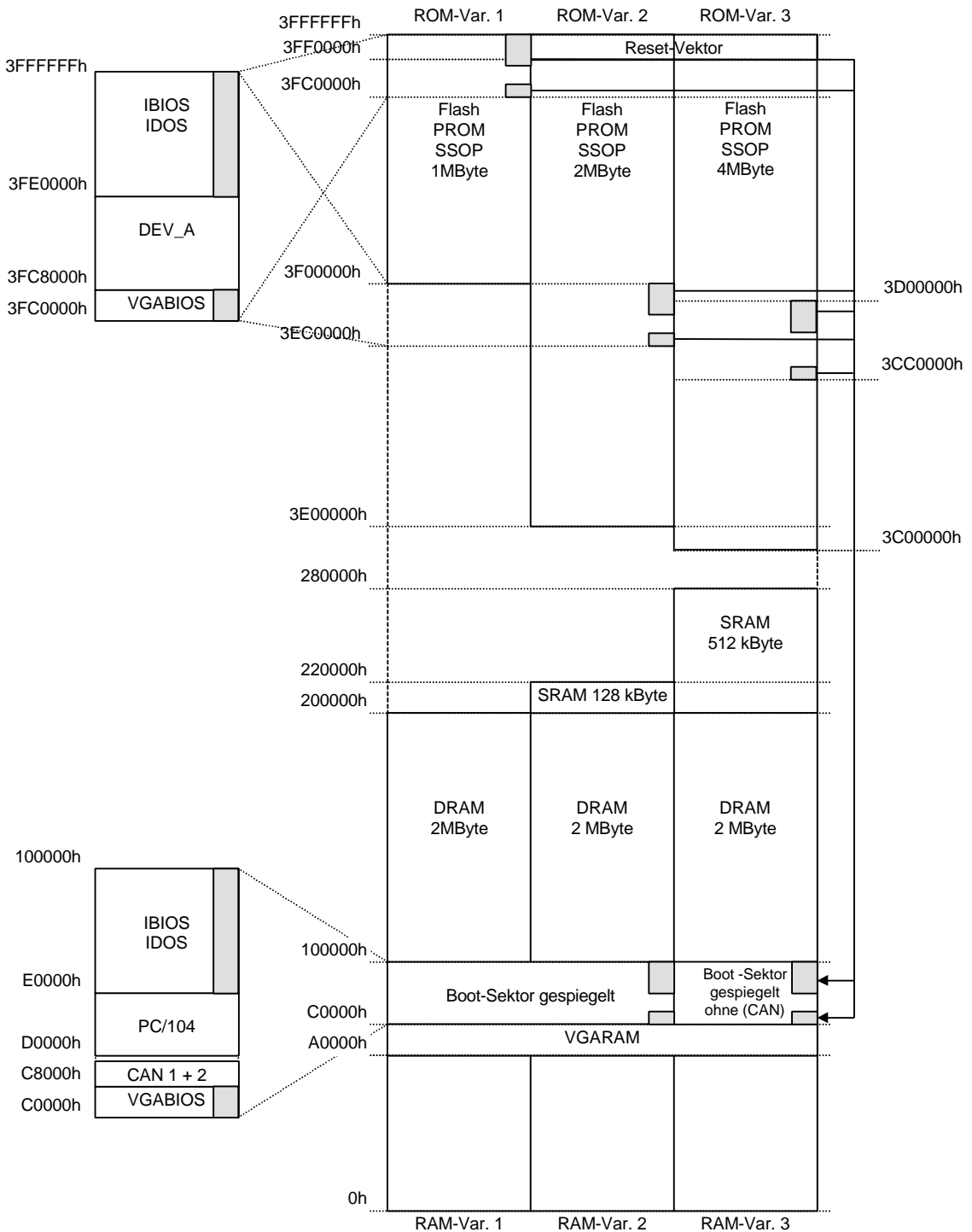
Das PC/104 Modul PC-386VGA erlaubt die vier RAM Bestückungsvarianten mit 256 kByte bzw. 1 MByte SRAM, 1 MByte DRAM, 1 MByte DRAM mit 256 kByte SRAM sowie die fünf (Flash-)EPROM Bestückungsvarianten mit 256 kByte, 512 kByte bzw. 1 MByte DIL (Flash-)EPROM oder 2 MByte bzw. 4 MByte SMD Flash-EPROM.

Bei der minimalen Bestückung von 256 kByte (Flash-)EPROM wird dem Anwender ein real-mode ROM-Drive (DEV_A) von 64 kByte zur Verfügung gestellt. Je grösser das (Flash-)EPROM ist, desto mehr Speicher steht dem Anwender in einem extended ROM Drive (DEV_B) zur Verfügung.

Achtung:

Werden EPROMs vom Typ 27C040 eingesetzt, so muss der 0Ω Widerstand R2 anstelle des 0Ω Widerstandes R1 bestückt werden.

4.5 **Memory-Mapping PC(CPU)-386CAN/VGA**



Das PC/104 Modul PC(CPU)-386CAN/VGA erlaubt die drei RAM Bestückungsvarianten mit 2 MByte DRAM, 2 MByte DRAM mit 128 kByte SRAM, 2 MByte DRAM mit 512 kByte SRAM sowie die drei Flash-ROM Bestückungsvarianten mit 1 MByte Flash-ROM, 2 MByte Flash-ROM oder 4 MByte Flash-ROM.

Dem Anwender stehen zwei extended ROM-Drives mit 768 kByte und 960 kByte zur Verfügung.

4.5 Interrupt Struktur

Der i386EX hat zwei Interrupt Controller 82C59A integriert, welche wie im PC/AT kaskadiert sind. Das BIOS initialisiert diese Interrupt Controller gemäss der üblichen DOS Konfiguration. Wie die Interrupts des i386EX auf den ModuNORM® PC/104 Modulen genutzt werden, ist den untenstehenden Tabellen zu entnehmen.

PC/104 Standard:

Priorität	Typ	Nr	Adresse	PC-Standard	i386 intern	extern
0 höchste	IRQ 0	08	020 - 023	Timer	Timer 0	
1	IRQ 1	09	024 - 027	Tastatur		INT 0
2	IRQ 8	70	1C0 - 1C3	Echtzeituhr		INT 4
3	IRQ 9	71	1C4 - 1C7	frei	SSIO oder	INT 5
4	IRQ 10	72	1C8 - 1CB	frei	Timer 1	
5	IRQ11	73	1CC - 1CF	frei	Timer 2	
6	IRQ12	74	1D0 - 1D3	frei	DMA	
7	IRQ13	75	1D4 - 1D7	math. Copr.		INT 6
8	IRQ14	76	1D8 - 1DB	Festplatte		INT 7
9	IRQ 15	77	1DC - 1DF	frei	Watchdog	
10	IRQ 2	0A	028 - 02B	(Echtzeituhr)	-	
11	IRQ 3	0B	02C - 02F	2. Serielle	SIO 1	
12	IRQ 4	0C	030 - 033	1. Serielle	SIO 0	
13	IRQ 5	0D	034 - 037	2. Parallele		INT 1
14	IRQ 6	0E	038 - 03B	Diskettenlw.		INT 2
15 tiefste	IRQ 7	0F	03C - 03F	1. Parallele		INT 3

PC/104 CPU's:

Priorität	CPU-386CAN	CPU-386VGA	PC-386CAN	PC-386VGA	PC(CPU)-386 CAN/VGA	
0 höchste						
1	INTCAN2	PC/104 IRQ1 auf IRQ12 1)	PC/104 IRQ1 auf IRQ12 1)	PC/104 IRQ1 auf IRQ12 1)	PC/104 IRQ1 auf IRQ12 1)	
2	Stecker J1/121	INTLCD	CAN 2, Stecker J9/25	INTLCD	INTLCD Stecker J9/25	
3	Stecker J1/23	PC/104 IRQ9	PC/104 IRQ9	PC/104 IRQ9	CAN2 PC/104 IRQ9	
4						
5						
6						
7	Stecker J1/55	Stecker J9	Stecker J9/22	Stecker J9/22	Stecker J9/22	
8	INTCAN1	PC/104 IRQ14	CAN 1 PC/104 IRQ14	PC/104 IRQ14	CAN 1 PC/104 IRQ14	
9						
10						
11						
12						
13	Stecker J1/89	PC/104 IRQ5	PC/104 IRQ5	PC/104 IRQ5	PC/104 IRQ5	
14	Stecker J1/122	PC/104 IRQ6	PC/104 IRQ6 Stecker J9/21	PC/104 IRQ6 Stecker J9/21	PC/104 IRQ6 Stecker J9/21	
15 tiefste	Stecker J1/67	PC/104 IRQ7	PC/104 IRQ7	PC/104 IRQ7	PC/104 IRQ7	

- 1) Da diese Konfiguration nicht der PC/104-Norm entspricht, kann der INT0 durch Weglassen des Widerstandes R97 vom PC/104 Bus getrennt werden.

PC/104 Adapter:

Priorität	DP-386CAN	DP-386COM	PC-Adapter Quad-COM	PC-Adapter Ethernet A104		
0 höchste						
1						
2						
3	9, konfig	9, konfig	9, konfig			
4	10, konfig	10, konfig	10, konfig	10, konfig		
5	11, konfig	11, konfig	11, konfig	11, konfig		
6	12, konfig	12, konfig	12, konfig	12, konfig		
7						
8	14, konfig	14, konfig	14, konfig			
9	15, konfig	15, konfig	15, konfig	15, konfig		
10						
11	3, konfig	3, konfig	3, konfig	3, konfig		
12	4, konfig	4, konfig	4, konfig	4, konfig		
13	5, konfig	5, konfig	5, konfig	5, konfig		
14	6, konfig	6, konfig	6, konfig			
15 tiefste	7, konfig	7, konfig	7, konfig			

PC/104 Basis:

Priorität	EVA-386 CPU-386VGA	Basis WINbloc PC-386CAN	Basis Terminal PC-386VGA	Basis Touch PC-386VGA	Basis Touch PC-386 CAN/VGA	
0 höchste						
1			Tastatur-Controller, fix			
2		Jumper 8				
3						
4						
5						
6						
7		13, fix				
8						
9						
10						
11						
12						
13		Jumper 5				
14	6, fix CAN	6, fix				
15 tiefste		7, fix				

5. Industrial-BIOS

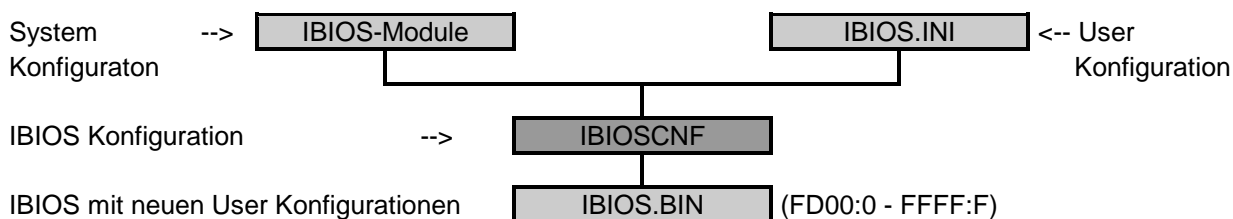
Das Industrial-BIOS IBIOS ist ein konfigurierbares BIOS für die ModuNORM® PC/104 Module. Das IBIOS ist aus einzelnen selbständigen Funktionsmodulen zusammengesetzt, welche unabhängig voneinander verwendet werden können. Je nach Anforderung der spezifischen Hardware-Plattform können einzelne Module hinzugefügt oder weggelassen werden. Das IBIOS läuft „location independent“, d.h. es muss im Speicher nicht an einer bestimmten Adresse liegen, sondern kann frei auf eine Segmentgrenze gelegt werden.

Während der Initialisierung der Hardware werden:

- die Port-Pins konfiguriert
- die Chip-selects richtig gesetzt
- das Memory getestet
- der Timer programmiert
- der Interrupt-Controller programmiert
- die Zusatz-BIOS-Module aufgerufen
- nach BIOS-Erweiterungen gesucht (z.B. VGABIOS)

Die Grundinitialisierung des IBIOS erfolgt vom System immer gleich. Hardware bezogene Initialisierungen (Chip-selects, COM1, COM2, D/A-Wandler Werte, etc.) können von Anwender im File IBIOS.INI gemäss der entsprechenden Hardware angegeben werden.

Mit dem Tool IBIOSCNF.EXE wird das neue IBIOS.BIN anhand der Konfiguration von IBIOS.INI erzeugt.



5.1 System Konfiguration

Viele Port-Pin's des i386EX haben eine Doppelfunktion. So kann z.B. der Port 2.0 als -CS0 oder als I/O Port definiert werden. Die Initialisierung der Port-Pins ist im IBIOS fest konfiguriert. Dem Anwender bleibt es jedoch freigestellt, nach dem Aufstarten des Modules (IBIOS und IDOS) die Port-Pin's nach seinen Wünschen neu zu programmieren. Es ist jedoch darauf zu achten, dass die gewählte Konfiguration mit der Hardware kompatibel ist!

Port 1:

Port.Pin	1.7	1.6	1.5	1.4	1.3	1.2	1.1	1.0
P1CFG= 0x02	P1.7	P1.6	P1.5	P1.4	P1.3	P1.2	-RTS0	P1.0
P1LTC = 0xFF	Input	Input	Input	Input	Input	Input	--	Input
P1DIR = 0xFF	High Imp.	High Imp.	High Imp.	High Imp.	High Imp.	High Imp.	--	High Imp.

Port 2:

Port.Pin	2.7	2.6	2.5	2.4	2.3	2.2	2.1	2.0
P2CFG= 0xF7	-CTS0	TXD0	RXD0	-CS4	-CS3	-CS2	-CS1	-CS0
P2LTC = 0xFF	--	--	--	--	--	--	--	--
P2DIR = 0xFF	--	--	--	--	--	--	--	--

Port 3:

Port.Pin	3.7	3.6	3.5	3.4	3.3	3.2	3.1	3.0
P3CFG= 0xBC	COMCLK	P3.6	INT3	INT2	INT1	INT0	P3.1	P3.0
P3LTC = 0xFF	--	Output	--	--	--	--	Input	Input
P3DIR = 0xBF	--	Com-out	--	--	--	--	High Imp.	High Imp.

Pin Configuration:

Pin	--	2	89/91/92	128	113	112	77	79
PINCFG = 0x0D	--	-CS6	coprocess or signals	DACK0	-CTS1	TXD1	SRXCLK	-RTS1

5.2 User Konfiguration

Das IBIOS ist aus einzelnen separaten Modulen zusammengesetzt, die unabhängig voneinander funktionieren. Je nach Anforderung einzelner Hardware-Varianten können einzelne Module dazugefügt oder weggelassen werden. Einzelne IBIOS-Module kann der Anwender entsprechend seiner Hardware konfigurieren.

IBIOS-Module:

IBIOS-Modul	Beschreibung	Eingesetzt in:
i386ex.bin	Initialisierung des Prozessors mit HW	allen Modulen
ib_com.bin	Defaultwerte der seriellen Schnittstelle	allen Modulen
dc_dc.bin	Einstellung DC/DC-Wandler für LCD	allen Modulen mit VGA-Controller
ib_key.bin	INT-16h Keyboard Interface	allen Modulen
ib_rtx.bin	Delay Handler und RTX Emulator	allen Modulen
ib_ext.bin	Sucht nach BIOS-Erweiterungen	allen Modulen
ds1302.bin	RTC-Initialisierung	allen Modulen mit bestücktem RTC
kbmatrix.bin	Tastaturmatrix-Initialisierung	PC-Terminals
kb8151.bin	Keyboardcontroller-Initialisierung	PC/104 Modulen mit Keyboardcontroller

i386EX:

Definition i386EX-Modul im File IBIOS.INI.

```
[COMMANDS]
MAIN = i386ex.bin
```

Folgende Parameter sind vom Anwender im File IBIOS.INI in der Sektion MAIN einzustellen: 'x' steht in den folgenden Parametern für die Nummer der Chip-selects (0-6).

```
[MAIN]
RAM_SERCH_TOP = 0x4000

REF_UNIT_EN = 1

CSx_BASISADDRH = 0x000D
CSx_BASISADDRL = 0x78
CSx_MASKADDR = 0x0000
CSx_WAITSTATES = 6
CSx_BS16 = 0
CSx_MEM = 1
CSx_RDY = 0
CSx_ENABLE = 1
```

RAM_SERCH_TOP: Obere Grenze, bis zu welcher das DOS nach RAM Speicher sucht. (Die Grenze muss eine Segmentadresse einer 64KB-Adresse sein, d.h. Bit11..0 der Segmentadresse müssen Null sein)

Bei bestückten 256 kByte SRAM = 0x4000 (256kByte DOS RAM)
 Bei bestückten 1 MByte SRAM = 0xA000 (640kByte DOS RAM)
 Bei bestückten 1 MByte DRAM = 0xA000 (640kByte DOS RAM)

REF_UNIT_EN: Enable der Refresh UNIT des i386EX.

0 = Disable 1 = Enable

CSx_BASISADDRH: Basis Adresse HIGH des entsprechenden Chip-select.

z.B.: CS3_BASISADDRH = 0x000D

CSx_BASISADDRL: Basis Adresse LOW des entsprechenden Chip-select.

z.B.: CS3_BASISADDRL = 0x78

Chip-select als MEM definiert:

Die Basis Adresse HIGH und LOW setzt sich folgendermassen zusammen:
Basis Adresse = Basis Adresse High x 10000h & Basis Adresse LOW x 100h

z.B.: 000Dh x 10000h & 78h x 100h = D7800 = D780:0

Chip-select als I/O definiert:

Die Basis Adresse HIGH und LOW setzt sich folgendermassen zusammen.
Basis Adresse = Basis Adresse HIGH x 40h & Basis Adress LOW / 4h

z.B.: 000Dh x 40h & 78 / 4h = 340h & 1Eh = 35Eh

CSx_MASKADDR: Maske des entsprechenden Chip-select.

Mit diesem Wert kann die Länge der einzelnen Chip-selects definiert werden.
Zu beachten ist, dass die Start-Adresse eines Chip-selects einem ganzzahliges Vielfachen der Blockgrösse dieses Chip-selects entsprechen muss.

Chip select als MEM definiert:

0x0000 = 2 k	0x003F = 128 k	0x0FFF = 8 M
0x0001 = 4 k	0x007F = 256 k	0x1FFF = 16 M
0x0003 = 8 k	0x00FF = 512 k	0x3FFF = 32 M
0x0007 = 16 k	0x01FF = 1 M	0x7FFF = 64 M
0x000F = 32 k	0x03FF = 2 M	
0x001F = 64 k	0x07FF = 4 M	

Chip select als I/O definiert:

0x0000 = 2	0x003F = 128	0x0FFF = 8 k
0x0001 = 4	0x007F = 256	0x1FFF = 16 k
0x0003 = 8	0x00FF = 512	0x3FFF = 32 k
0x0007 = 16	0x01FF = 1 k	0x7FFF = 64 k
0x000F = 32	0x03FF = 2 k	
0x001F = 64	0x07FF = 4 k	

CSx_WAITSTATES: Wait-state des entsprechenden Chip-selects.

CSx_BS16: Bus-size des entsprechenden Chip-selects.

0 = 8-Bit 1 = 16-Bit

CSx_MEM: Bus-type der entsprechender Chip-selects.

0 = I/O 1 = Memory

CSx_RDY: Ready Definition des entsprechenden Chip-selects.

0 = Ready intern 1 = Ready extern

CSx_ENABLE: Enable des entsprechenden Chip-selects.

0 = Disable 1 = Enable

DC_DC.BIN:

Definition DC_DC-Modul im File IBIOS.INI

```
[COMMANDS]
MODULEn = dc_dc.bin
```

 n = Eindeutige Modulnummer

Folgende Parameter sind vom Anwender im File IBIOS.INI in der Sektion MIKRAP einzustellen:
 'x' steht in den folgenden Parametern für die Nummer des D/A-Wandlers (1, 2).

```
[MIKRAP]
DAX_ADDR    = 0x0300
DAX_VALUE   = 0x00A0
DAX_UPPER   = 0x00B0
DAX_LOWER   = 0x0090
```

DAX_ADDR: Adresse des D/A-Wandlers im I/O-Bereich (Chip-Select 2).
 Standardmässig ist die Adresse des D/A-Wandlers 1 auf 300h und die Adresse des D/A-Wandlers 2 auf 302h eingestellt.

DAX_VALUE: Parameter der entsprechenden D/A-Wanler. Anhand dieser Parameter werden die Spannungsausgänge VPOS, VNEG und AOUT_B eingestellt.

Spannung VPOS und VNEG:

Diese Spannung wird mit Hilfe des D/A-Wandlers 1 eingestellt.

VPOS [V]	Wert D/A-1	VPOS [V]	Wert D/A-1	VPOS [V]	Wert D/A-1
12.0	F4	22.0	80	32.0	59
12.5	EA	22.5	7E	32.5	57
13.0	DF	23.0	7B	33.0	56
13.5	D8	23.5	79	33.5	55
14.0	CF	24.0	76	34.0	54
14.5	C8	24.5	74	34.5	53
15.0	BF	25.0	72	35.0	51
15.5	BA	25.5	6F	35.5	50
16.0	B4	26.0	6D	36.0	4F
16.5	AF	26.5	6B	36.5	4E
17.0	A9	27.0	69	37.0	4D
17.5	A4	27.5	67	37.5	4C
18.0	9F	28.0	66	38.0	4B
18.5	9B	28.5	64	38.5	4A
19.0	97	29.0	62	39.0	49
19.5	93	29.5	60	39.5	48
20.0	8F	30.0	5F	40.0	47
20.5	8B	30.5	5D	40.5	46
21.0	87	31.0	5C		
21.5	84	31.5	5B		

$$VNEG[V] = -(VPOS - 0.5) [V]$$

(Gemessen mit je 1.5 kOhm Belastung an VPOS und VNEG)

Die Toleranz der Spannungen VPOS und VNEG beträgt $\pm 10\%$.**Spannung AOUT_B:**

$$AOUT_B = - \left(\frac{Wert_h}{FF} \right) * 5 [V] = - \left(\frac{Wert_d}{255} \right) * 5 [V]$$

- DAx_UPPER: Im IBIOS abgelegter oberer Spannungswert, bis zu welchem der aktuelle D/A-Wandler initialisiert werden darf.
 Lesen der DAx_UPPER-Werte: Mittels INT 15h Funktion EEh.
- DAx_LOWER: Im IBIOS abgelegter unterer Spannungswert, bis zu welchem der aktuelle D/A-Wandler initialisiert werden darf.
 Lesen der DAx_LOWER-Werte: Mittels INT 15h Funktion EEh.

Beim Initialisieren des IBIOS werden die beiden D/A-Wandler auf I/O-Adresse 300h und 302h mit den Spannungswerten von DA1_VALUE und DA2_VALUE initialisiert. Da die aktuellen Werte nicht vom D/A-Wandler zurückgelesen werden können, werden diese Spannungswerte DA1_VALUE und DA2_VALUE im BIOS Variablen-Segment abgespeichert. Mit Hilfe dieser abgespeicherten Spannungswerte, sowie den zugehörigen oberen und unteren Grenzwerten, kann die Kontrastspannung eines angeschlossenen LCD-Displays im Betrieb mit einem einfachen Anwendungsprogramm verstellt werden.

Spannungswert	BIOS-Variablen-Segment	Type
DA1_VALUE	0040:007A	Byte
DA2_VALUE	0040:007B	Byte

INT 15h Funktion EEh:

Mit der INT 15h Funktion können die Spannungs-Grenzwerte (welche im BIOS definiert wurden) ausgelesen werden.

- Input: AH = EEh
 AL = 00h für DA1_UPPER
 01h für DA1_LOWER
 02h für DA2_UPPER
 03h für DA2_LOWER

Output: AL = Wert des entsprechenden Spannungswert.

Achtung:

Es dürfen keine Peripheriekarten eingesetzt werden, welche die I/O-Adressen 300h bis 303h benutzen!

DS1302:

Definition ds1302-Modul in File IBIOS.INI.

```
[COMMANDS]
MODULEn = ds1302.bin
```

n = Eindeutige Modulnummer

Da der verwendete RTC-Baustein nicht DOS-kompatibel ist, wurden bisher nur die folgenden INT 1Ah Funktionen implementiert:

- INT 1Ah Funktion 00h: Read System Timer Counter
- INT 1Ah Funktion 01h: Set System Timer Counter
- INT 1Ah Funktion 02h: Read Real-Time Clock Time
- INT 1Ah Funktion 03h: Set Real-Time Clock Time
- INT 1Ah Funktion 04h: Read Real-Time Clock Date
- INT 1Ah Funktion 05h: Set Real-Time Clock Date

IB_COM:

Definition ib_com-Modul im File IBIOS.INI

```
[COMMANDS]
MODULEn = ib_com.bin
```

n = Eindeutige Modulnummer

Folgende Parameter sind vom Anwender im File IBIOS.INI in der Sektion (COM) einzustellen: 'x' steht in den folgenden Parametern für die Nummer der seriellen Schnittstellen (1-4).

```
[COM]
UART_x_ADR          = 0x3F8
UART_x_BAUD         = 12
UART_x_MODE         = 3
UART_x_INT          = 4
COMMAXSEARCH       = 2
```

UART_x_ADR: Adresse des UART-Bausteins im I/O-Bereich. Im i386EX sind diese Adressen fest auf COM1 = 3F8h und COM2 = 2F8h gesetzt.

UART_x_BAUD: Faktor für die Baudrate. Der Faktor ist gleich dem Teiler für die maximale Frequenz von 115 Kilobaud:

$$\text{Baudrate} = \frac{115'200}{\text{Faktor}}$$

Beispiel: Faktor 12 entspricht 9'600 Baud

UART_x_MODE: Bestimmt die physikalische Übertragungsart.

Bit 0,1 : Anzahl Data Bits:

0 = 5 Bits

1 = 6 Bits

2 = 7 Bits

3 = 8 Bits

Bit 2 : Anzahl Stop Bits:

0 = 1 Stop Bits

1 = 2 Stop Bits

Bit 3 : Parity Enable:

Bit 4 : Even Parity Select

0 = ODD

1 = EVEN

Bit 5 : Stick Parity

Bit 6 : Set Break

1 = Break Zustand

UART_x_INT: Definiert die Interruptleitung, an welcher der UART seine eventuellen Interruptrequests anlegt. Die Initialisierung des IBIOS sieht allerdings keinen Interruptbetrieb der seriellen Schnittstellen vor. Im i386EX sind diese Interruptleitungen für COM1 und COM2 fest definiert (INT_COM1 = IRQ4 und INT_COM2 IRQ3).

IB_EXT:

Nach dem Aufstarten der CPU sucht das IBIOS nach sogenannten BIOS-Erweiterungen. Dabei wird der angegebene Speicherbereich nach der Kennung (AA55h) abgesucht, welche den Beginn einer BIOS-Erweiterung markiert. Sobald eine solche Kennung gefunden wurde, rechnet das Modul EXT die Checksumme aus, und vergleicht diese mit der Checksumme der BIOS-Erweiterung. Stimmen die beiden überein, wird die Erweiterung angesprungen.

Definition `ib_ext`-Modul im File IBIOS.INI.

```
[COMMANDS]
MODULEn = ib_ext.bin
```

n = Eindeutige Modulnummer

Folgende Parameter sind vom Anwender im File IBIOS.INI in der Sektion (EXT) einzustellen:

```
[EXT]
BIOSEXTRANGE1_LO = 0xC000
BIOSEXTRANGE1_HI = 0xF800
```

BIOSEXTRANGE1_LO: Untere Grenze des Bereiches.

BIOSEXTRANGE1_HI: Obere Grenze des Bereiches.

Sind die untere und obere Grenze gleich Null, so wird nicht nach BIOS-Erweiterungen gesucht.

IB_KEY:

Definition `ib_key`-Modul im File IBIOS.INI.

```
[COMMANDS]
MODULEn = ib_key.bin
```

n = Eindeutige Modulnummer

IB_RTX:

Definition `ib_rtx`-Modul im File IBIOS.INI.

```
[COMMANDS]
MODULEn = ib_rtx.bin
```

n = Eindeutige Modulnummer

Das File IBIOS.INI muss nun so abgeändert werden, dass die IBIOS Konfiguration mit der Hardware übereinstimmt. Auf der System-Diskette befinden sich unter dem Directory <PROJEKT> Beispiele für die Konfiguration des IBIOS entsprechend der gewählten Hardware-Variante.

5.3 IBIOS.BIN Generierung

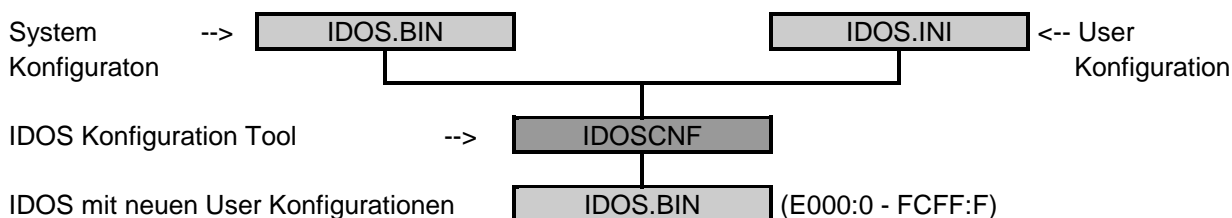
Ist die Konfigurationsdatei IBIOS.INI der entsprechenden Hardware angepasst worden, so kann eine neue Datei IBIOS.BIN mit folgender Eingabe erzeugt werden:

ibioscnf ibios

6. Industrial-DOS

Das Industrial-DOS IDOS ist ein Element des Real Time Disk Operating System RTXDOS-16 und ist kompatibel zum Industrie-Standard. Alle IDOS-Teile (inkl. Kommando Interpreter) laufen im ROM, d.h. es wird für den DOS-Teil kein zusätzliches RAM verwendet. Um die ModuNORM® PC/104 Module in kurzer Zeit in Betrieb nehmen zu können, wurde das Programm „Remote-Console“ im IDOS implementiert. Mit Hilfe dieses Programmes können über eine serielle RS232 Schnittstelle die Ein- und Ausgabeeinheiten eines PC an das Modul angeschlossen werden.

Mit dem Tool IDOSCNF kann eine anwendungsspezifische Variante von IDOS.BIN aufgrund der in IDOS.INI enthaltenen Konfigurationsangaben erzeugt werden.



Wird das PC/104 Modul über die Remote-Console betrieben, so muss darauf geachtet werden, dass nicht alle MS-DOS Funktionen auf die Remote Drives anwendbar sind.

Unterstützte Funktionen:

- 36h Get Disk Free Space
- 3Bh Change the current Directory
- 3Ch Create File
- 3Dh Open File
- 3Eh Close File
- 3Fh Read data
- 40h Write data
- 41h Delete File
- 42h Move File Ptr
- 4Eh Find 1st File
- 4Fh Find next File
- 57h Get/Set File Time

Damit können Daten von lokalen Disks auf eine Remote Disk kopiert werden, oder es können Programme direkt von einer Remote Disk aus gestartet werden. Beim Starten von Programmen über die Remote-Console muss beachtet werden, dass die Programme zuerst über die RS232 Schnittstelle in das RAM des PC/104 Moduls geladen werden. Dies bewirkt, dass solche Programme wesentlich mehr Zeit beanspruchen, als wenn sie direkt von einem DEVICE der CPU aufgerufen werden.

Achtung:

Programme, welche über die Remote-Console funktionieren sollen, dürfen nicht direkt ins Video-RAM schreiben. Alle Ausgaben müssen via DOS oder BIOS aufgerufen werden.

6.1 User Konfiguration

Im IDOS können zwei Sektionen vom Anwender entsprechend seinen Anforderungen eingestellt werden. In der Sektion (REM) kann die Remote-Console definiert werden. In der Sektion (DEVICES) können die lokalen Drive-Typen konfiguriert werden.

Sektion	Beschreibung
REM	Definition der Remote-Console
DEVICES	Konfiguration von lokalen Drive-Typen

REM:

Folgende Parameter sind vom Anwender im File IDOS.INI in der Sektion (REM) einzustellen: 'x' steht in den folgenden Parametern für die Nummer der seriellen Schnittstellen (1-4).

[REM]	
UART_x_ADR	= 0x3F8
UART_x_BAUD	= 1
UART_x_MODE	= 0x0F
UART_x_INT	= 4
VIDEOWAITMOD	= 0
REMOTECON	= 2
VIDEO_OUTPUT	= 2

UART_x_ADR: Adresse des UART-Bausteins im I/O-Bereich. Im i386EX sind diese Adressen fest auf COM1 = 3F8h und COM2 = 2F8h gesetzt.

UART_x_BAUD: Faktor für die Baudrate. Der Faktor ist gleich dem Teiler für die maximale Frequenz von 115 Kilobaud:

$$\text{Baudrate} = \frac{115'200}{\text{Faktor}}$$

Beispiel: Faktor 12 entspricht 9'600 Baud

UART_x_MODE: Bestimmt die physikalische Übertragungsart.

Bit 0 .. 1 = Anzahl Data Bits:

- 0 = 5 Bits
- 1 = 6 Bits
- 2 = 7 Bits
- 3 = 8 Bits

Bit 2 = Anzahl Stop Bits:

- 0 = 1 Stop Bits
- 1 = 2 Stop Bits

Bit 3 = Parity Enable

Bit 4 = Even Parity Select

- 0 = ODD
- 1 = EVEN

Bit 5 = Stick Parity

Bit 6 = Set Break:

- 1 = Break Zustand

UART_x_INT: Definiert die Interruptleitung, an welcher der UART seine eventuellen Interruptrequests anlegt. Die Initialisierung des IBIOS sieht allerdings keinen Interruptbetrieb der seriellen Schnittstellen vor. Im i386EX sind diese Interruptleitungen für COM1 und COM2 fest definiert (INT_COM1 = IRQ4 und INT_COM2 = IRQ3).

- VIDEOWAITMODE:** Systemstart nur mit, oder auch ohne angeschlossene Remote-Console.
 0 = System startet, auch ohne angeschlossene Remote-Console
 1 = System startet nur auf, wenn die Remote-Console angeschlossen ist.
- REMOTECON:** Spezifiziert die COM-Schnittstelle, über welche die Remote-Console laufen soll.
 Wird dieser Wert auf Null gesetzt, so wird der Treiber für die Remote-Console nicht gestartet.
- VIDEO_OUTPUT:** Die Ausgaben können sowohl auf die LCD-Anzeige als auch auf die Remote-Console (PC-Bildschirm) geleitet werden:
 Bit 0 = Remote-Console aktiv
 Bit 1 = LCD-Anzeige aktiv
- Wenn die Remote-Console aktiv ist (Bit 0 = 1), wird unabhängig von der Einstellung von VIDEOWAITMODE beim Systemstart auf die Remote-Console gewartet

DEVICES:

Folgende Parameter sind vom Anwender im File IDOS.INI in der Sektion (DEVICES) einzustellen:
 'x' steht in den folgenden Parametern für die Nummer der Devices (1-8).

[DEVICES]	
Dx_FILESYSTEM	= RTF
Dx_AUTOINIT	= YES
Dx_ISBOOTDEVICE	= NO
Dx_DRIVERTYP	= RM_MEM
Dx_DRIVERSUBTYP	= ERAM
Dx_BASESEG	= 0x0200000
Dx_INITSECTORLEN	= 1
Dx_INITCLUSTERSIZE	= 32
Dx_INITTOTALSIZE	= 8192

- Dx_FILESYSTEM:** File-System:
 INACTIV File System inaktiv
 FAT MS-DOS kompatibles File System über BIOS,
 File System für Memory Devices (SRAM)
 RTF File System für Memory Devices (DRAM, ROM, FLASH, etc.)
 Ab IDOS V6.41b (ab System-ROM Packet 001210J) ist für ein
 DRAM-Device nur das FAT-Filesystem zu verwenden.
 DUMMY Laufwerksbuchstabe wird übersprungen
- Dx_AUTOINIT:** Autoinitialisierung des Devices:
 NO Keine Initialisierung des Devices
 YES Initialisierung des Devices, wenn kein gültiges File-System
 dedektiert wird.
- Dx_ISBOOTDEVICE:** Ein Device muss als Boot-Device definiert werden. Dieses Device muss die
 Dateien autexec.bat und config.sys beinhalten.
 NO kein Boot-Device
 YES Boot-Device
- Dx_DRIVERTYP:** File System Drive Type.
 BIOS_13: File System über BIOS (FAT File System)
 RM_MEM: File System für Memory Devices (RAM, ROM, FLASH, etc.)

Dx_DRIVERSUBTYP: Angabe des Driver-Sub-Typs:
 RAM File System in RAM (Adressbereich ≤ 1 MByte)
 ROM File System in ROM (Adressbereich ≤ 1 MByte)
 FLASH File System in Flash (Adressbereich ≤ 1MByte)
 ERAM File System in extended RAM (Adressbereich > 1 MByte)
 EROM File System in extended ROM (Adressbereich > 1 MByte)
 EFLASH File System in extended Flash (Adressbereich > 1MByte)

Dx_BASESEG: Basis Adresse des Devices:
 Dx_DRIVERSUBTYP = RAM,ROM oder FLASH:
 Es muss die Device-Segment-Anfangsadresse angegeben werden.

z.B.: Für die Basisadresse von 512 kByte ⇒ 0x8000

Dx_DRIVERSUBTYP = ERAM,EROM oder EFLASH:
 Es muss die Device-Anfangsadresse angegeben werden.

z.B.: Für die Basisadresse von 63 MByte ⇒ 0x3F00000

Dx_INITSECTORLEN: Sektor Grösse in Byte (bei FAT-File System = 512, RTF-File System = 32)

Dx_INITCLUSTERSIZE: Sektor Grösse in Cluster (bei FAT-File System = 1, RTF-File System = 1)

Dx_INITTOTALSIZE: Anzahl Cluster des Devices (max. 65535)

Device Grösse in Byte = SECTORLEN x CLUSTERSIZE x TOTALSIZE

6.2 Konfigurations Beispiele

RAM-Device (SRAM):

- Basis Adresse = 512 kByte (0x8000)
 - Device Grösse = 256 kByte

[DEVICES]	
Dx_FILESYSTEM	= FAT
Dx_AUTOINIT	= YES
Dx_ISBOOTDEVICE	= NO
Dx_DRIVERTYP	= RM_MEM
Dx_DRIVERSUBTYP	= RAM
Dx_BASESEG	= 0x8000
Dx_INITSECTORLEN	= 512
Dx_INITCLUSTERSIZE	= 1
Dx_INITTOTALSIZE	= 512

ROM-Device als Boot-Device:

- Basis Adresse = 512 kByte (0xC800)
 - Device Grösse = 64 kByte

[DEVICES]	
Dx_FILESYSTEM	= RTF
Dx_AUTOINIT	= NO
Dx_ISBOOTDEVICE	= YES
Dx_DRIVERTYP	= RM_MEM
Dx_DRIVERSUBTYP	= ROM
Dx_BASESEG	= 0xC800

ERAM-Device (SRAM):

- Basis Adresse = 2 MByte (0x0200000)
- Device Grösse = 256 kByte

[DEVICES]	
Dx_FILESYSTEM	= FAT
Dx_AUTOINIT	= YES
Dx_ISBOOTDEVICE	= NO
Dx_DRIVERTYP	= RM_MEM
Dx_DRIVERSUBTYP	= ERAM
Dx_BASESEG	= 0x0200000
Dx_INITSECTORLEN	= 512
Dx_INITCLUSTER SIZE	= 1
Dx_INITTOTALSIZE	= 512

EROM-Device:

- Basis Adresse = 63 MByte (0x3F00000)
- Device Grösse = 768 kByte

[DEVICES]	
Dx_FILESYSTEM	= RTF
Dx_AUTOINIT	= NO
Dx_ISBOOTDEVICE	= NO
Dx_DRIVERTYP	= RM_MEM
Dx_DRIVERSUBTYP	= EROM
Dx_BASESEG	= 0x3F00000

6.3 IDOS.BIN Generierung

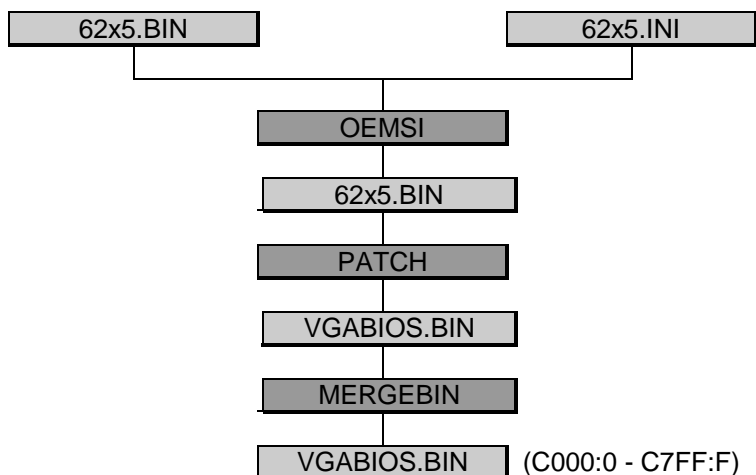
Sind alle Parameter in der Konfigurationsdatei IDOS.INI richtig eingestellt, so kann eine neue Datei IDOS.BIN mit folgender Eingabe erzeugt werden:

ibioscnf idos

7. VGA-BIOS

Das VGA-BIOS des auf den ModuNORM® PC/104 Modulen verwendeten VGA-Controllers von Cirrus-Logic, kann mit Hilfe des Tools OEMSI konfiguriert werden. Die Datei oemdata.txt gibt Auskunft über die Anpassungen, welche im File 62x5.ini vorgenommen werden können.

Im konfigurierten File 62x5.BIN müssen kleinere Anpassungen vorgenommen werden, damit das VGABIOS als DOS-BIOS lauffähig wird. Diese Anpassungen werden durch die Tools „PATCH“ sowie „MERGEBIN“ ausgeführt.



Die Konfiguration des VGABIOS erfolgte mittels folgender Eingabe:

```

oemsi -i 62x5.ini 62x5.bin
patch 62x5.bin -ovgabios.bin 9788=144 9789=144 9790=144 -v
mergeben vgabios.bin 0 vgabios.bin /bios=32
  
```

Soll an ein ModuNORM® PC/104 Modul ein Display angeschlossen werden, welches im Anhang des betreffenden Handbuches aufgeführt ist, so kann das entsprechende VGA-BIOS direkt übernommen werden.

Wurde ein neues System-ROM mit VGABIOS konfiguriert, so empfiehlt es sich, zuerst die Signale am VGA-Stecker zu messen, bevor ein Display angeschlossen wird.

8. Locale-Drives

Auf den ModuNORM® PC/104 Modulen können zwei verschiedene lokale ROM-Drive Typen verwendet werden:

- ROM-Drive (normales Real-Mode ROM-Drive)
- EROM-Drive (Extended ROM-Drive)

Die Files, welche den ROM-Drives übergeben werden sollen, werden mit dem Tool rtfmkdev bestimmt. Angaben zum Tool rtfmkdev sind dem Kapitel 12 zu entnehmen.

9. System-ROM

Ist die ROM Konfiguration abgeschlossen, so müssen die einzelnen Binary-Files zu einem System-ROM zusammengebunden werden. Das Erstellen eines System-ROM's (1 MByte) für ein ModuNORM[®] PC/104 Modul CPU-386VGA oder PC-386VGA erfolgt gemäss folgender Eingabe:

```
copy DEV_B.BIN PROM.BIN  
mergebin PROM.BIN 0xC0000 VGABIOS.BIN  
mergebin PROM.BIN 0xC8000 DEV_A  
mergebin PROM.BIN 0xE0000 IDOS.BIN  
mergebin PROM.BIN 0xFD000 IBIOS:BIN
```

Das nun erhaltene File PROM.BIN hat eine Grösse von 1 MByte. Durch das Teilen dieses Files in gerade und ungerade Adressen (splitten) werden die zwei Binary-Files PROM.EVE und PROM.ODD mit einem dafür geeigneten Tool erzeugt. Das Splitten des Files PROM.BIN kann z.B. mit dem Tool split2 gemäss folgender Eingabe erfolgen:

```
split2 PROM.bin PROM.evn PROM.odd
```

Das Splitten kann auch mit einem entsprechenden EPROM-Programmer erfolgen, welcher ein EVN und ein ODD EPROM direkt aus dem File 001200A.BIN programmieren kann.

Um die Konfiguration des System-ROM's zu erleichtern, sind auf der System-Diskette unter dem Directory Projekte mehrere Batchdateien abgelegt, mittels welchen System-ROM's für die einzelnen ModuNORM[®] PC/104 Module generiert werden können.

10. Systemerweiterungen

10.1 PC-Terminal

Die PC-Terminals mit s/w oder Farb ¼ VGA bzw. Farb TFT VGA LCD-Display und einer industrietauglichen Tastatur mit 30 Kurzhubtasten enthalten ein PC/104i Modul PC-386VGA mit zwei seriellen RS232 Schnittstellen. Damit die Tasten vom System-BIOS (IBIOS) eingelesen werden können, muss die standard IBIOS Konfiguration mit einem zusätzlichen IBIOS-Modul ergänzt werden.

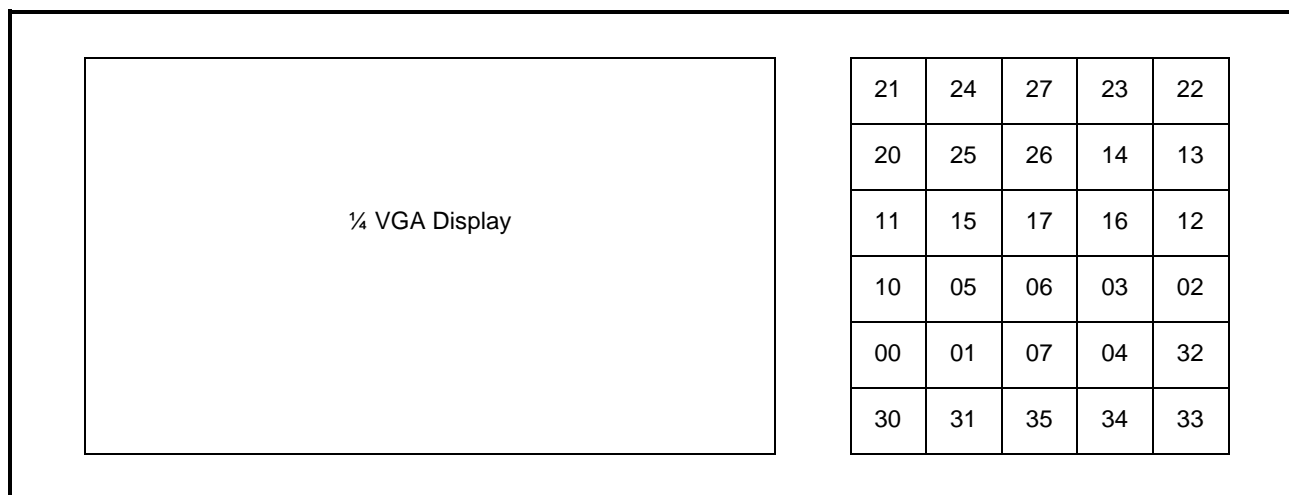
IBIOS-Modul KBMATRIX:

Dieses Modul initialisiert die Tastaturmatrix, welche maximal 32 Tasten unterstützt. Die aktive Taste wird mit Hilfe von vier Ausgangsleitungen (Port 1.4 bis 1.7), und acht Eingangsleitungen (D0 bis D7), sowie dem Interrupt IRQ1 bestimmt. Wird dieses Modul verwendet, so stehen dem Anwender die Portleitungen P1.4 bis P1.7 sowie die Interruptleitung IRQ1 nicht mehr zur Verfügung.

Die Numerierung der Tasten erfolgt nach folgender Matrixanordnung:

	IN7(D7)	IN6(D6)	IN5(D5)	IN4(D4)	IN3(D3)	IN2(D2)	IN1(D1)	IN0(D0)
OUT0 (Port 1.4)	07	06	05	04	03	02	01	00
OUT1 (Port 1.5)	17	16	15	14	13	12	11	10
OUT2 (Port 1.6)	27	26	25	24	23	22	21	20
OUT3 (Port 1.7)	37	36	35	34	33	32	31	30

Tastenanordnung PC-Terminal:



Initialisierung IBIOS-Modul KBMATRIX:

Definition KBMATRIX-Modul im File IBIOS.INI

```
[COMMANDS]
MODULEn = kbmatrix.bin
```

n = Eindeutige Modulnummer

Folgende Parameter sind vom Anwender im File IBIOS.INI für das Modul kbmatrix einzustellen:

```
[KBMATRIX]
KEYm = codes
```

m = Tastenindex im Bereich von 1 bis 88

codes = Sacn-Code wie PC/AT-Tastatur

Damit den einzelnen Tasten nicht nur ASCII-Codes, sondern auch die Codes von Funktionstasten etc. zugeordnet werden können, wird für die Belegung der Tasten der gleiche Scan-Code wie bei einer PC/AT-Tastatur verwendet.

Beispiel für die standard Tastenbelegung:

[KBMATRIX]	
KEY00=0x3F00 ; F5	KEY20=0x3C00; F2
KEY01=0x4B00; <<	KEY21=0x3B00; F1
KEY02=0x0044 ; 'D'	KEY22=0x0041 ; 'A'
KEY03=0x002E; ','	KEY23=0x0039 ; '9'
KEY04=0x4D00; >>	KEY24=0x0037 ; '7'
KEY05=0x0008 ; DEL	KEY25=0x0034 ; '4'
KEY06=0x0030 ; '0'	KEY26=0x0035 ; '5'
KEY07=0x4800 ; ^	KEY27=0x0038 ; '8'
KEY10=0x3E00; F4	KEY30=0x4000 ; 'F6
KEY11=0x3D00; F3	KEY31=0x001B; ESC
KEY12=0x0043 ; 'C'	KEY32=0x0045 ; 'E'
KEY13=0x0042 ; 'B'	KEY33=0x0046 ; 'F'
KEY14=0x0036 ; '6'	KEY34=0x000D; ENTER
KEY15=0x0031 ; '1'	KEY35=0x5000 ; √
KEY16=0x0033 ; '3'	
KEY17=0x0032 ; '2'	

Achtung:

Das VGABIOS muss gemäss dem PC/104i PC-386VGA Handbuch entsprechend dem Display-Type angepasst werden.

10.2 Tastaturkontroller 81C51SL

KB8151:

Dieses Modul initialisiert den über den PC/104-Bus angeschlossenen Tastaturkontroller 81C51. Wird dieses Modul verwendet, so steht dem Anwender der Interrupt IRQ1 nicht mehr zur Verfügung. Der empfangene Tastencode vom 81C51 Controller wird im IBIOS gemäss der geladenen Tabelle umgesetzt. Die Tastencodetabelle kann im IBIOS.INI gemäss der folgenden Form verändert werden.

Definition KBMATRIX-Modul im File IBIOS.INI

```
[COMMANDS]
MODULE $n$  = kb8151.bin
```

n = Eindeutige Modulnummer

Folgende Parameter sind vom Anwender im File IBIOS.INI für das kbmatrix-Modul einzustellen:

```
[KB8151]
KEY $m$  = codes
```

m = Tastenindex im Bereich von 1 bis 88

$codes$ = 32 bit Wert für die verschiedenen Codes einer Taste.

- bit 0..7 Basis-Tastencode
- bit 8..15 Code in Kombination mit <SHIFT> Taste
- bit 16..23 Code in Kombination mit <CTRL> Taste
- bit 23..31 Code in Kombination mit <ALT> Taste

Sind alle Bits einer Code-Kombination gesetzt, so wird diese nicht unterstützt.

Werden keine KEY m neu definiert, so ist die Standard IBM-Tastenbelegung implementiert

10.3 Flash-File

Auf den beiden DIL-Stecksockeln eines PC/104 Moduls können anstelle der beiden EPROM's auch zwei Flash-EPROM's eingesetzt werden. Diese können vom IDOS im Real-Time Filesystem (RTF) unterstützt werden. In einem Flash-Device können Dateien erzeugt und gelöscht werden.

RTF FLASH Device-Driver:

Die Flash-File System Unterstützung wird durch einen Device-Driver realisiert, welcher in der Datei config.sys eingebunden wird.

Beispiel: DEVICE = A:\RTFFLASH.EXE

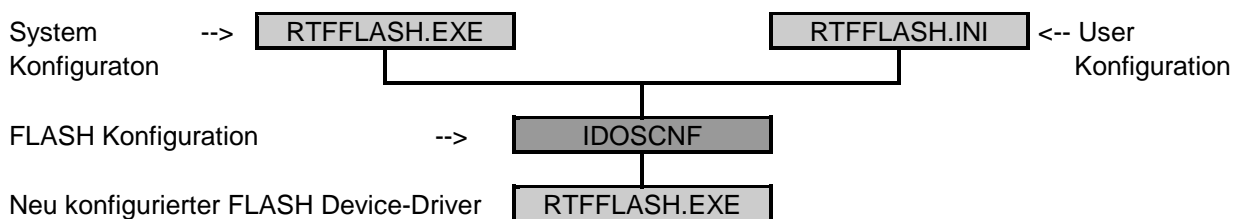
Dieser Device-Driver enthält die Hardware abhängigen Funktionen zum Schreiben und Löschen des jeweiligen Flash Typs. Zusätzlich unterstützt dieser Device-Driver die Möglichkeit, das Flash Device neu zu initialisieren. Bei der Initialisierung wird das Flash Device zuerst gelöscht und anschliessend das Dateisystem neu initialisiert. Diese Funktionalität wird auch beim „Autolnit“ verwendet.

Der Device-Driver unterstützt bis zu vier Devices. Die Konfiguration erfolgt mit dem Utility IDOSCONF.EXE. Der Device-Driver unterstützt Flash-EPROMs vom Typ AMD29F010, AMD29F040, Intel 28F016SV und Intel 28F400CVT.

Konfiguration:

Die Konfiguration des Device-Drivers RTFFLASH.EXE erfolgt mit dem Utility IDOSCONF.EXE. Der Device-Driver enthält interne Konfigurations-Tabellen für bis zu vier Devices.

Mit dem Tool IDOSCNF kann aufgrund der in RTFFLASH.INI angegebenen Parameter eine anwendungsspezifische Variante des Flash Device-Drivers RTFFLASH.EXE erzeugt werden.



RTFFLASH.INI

```

[COMMANDS]
OPTION1 = -v
TARGET = RTFFLASH.EXE
[FLASHDEV]
D1_FILESYSTEM = RTF
D1_AUTOINIT = YES
D1_ISBOOTDEVICE = NO
D1_DRIVERTYP = RM_MEM
D1_DRIVERSUBTYP = EFLASH
D1_BASESEGFLASH = 0x3F00000 ; Base address of FLASH
D1_BASEOFS = 0x0 ; Base address of file system
D1_INITSECTORLEN = 1 ; sector size in bytes (fix for RTF)
D1_INITCLUSTER SIZE = 32 ; sectors in cluster (fix for RTF)
D1_INITTOTALSIZE = 16384 ; Device Size in clusters (512 kB)
D2_FILESYSTEM = RTF
D2_AUTOINIT = YES
D2_ISBOOTDEVICE = NO
D2_DRIVERTYP = RM_MEM
D2_DRIVERSUBTYP = EFLASH
D2_BASESEGFLASH = 0x3F00000 ; Base address of FLASH
D2_BASEOFS = 0x80000 ; Base address of file system
D2_INITSECTORLEN = 1 ; sector size in bytes (fix for RTF)
D2_INITCLUSTER SIZE = 32 ; sectors in cluster (fix for RTF)
D2_INITTOTALSIZE = 8192 ; Device Size in clusters (256 kB)
D3_FILESYSTEM = INACTIV
D4_FILESYSTEM = INACTIV
  
```

Folgende Parameter sind von Anwender im File RTFFLASH.INI einzustellen:

Dx_FILESYSTEM:	File System:	
	INACTIV	File-System ausgeschaltet
	RTF	Real-Time File System
	DUMMY	Laufwerkbuchstabe wird übersprungen
Dx_AUTOINIT:	Wird standardmässig auf YES gesetzt.	
Dx_ISBOOTDEVICE:	Wird standardmässig auf NO (kein Boot Drive) gesetzt.	
Dx_DRIVERTYP:	Wird standardmässig auf RM_MEM gesetzt.	
Dx_DRIVERSUBTYP:	Wird standardmässig auf EFLASH gesetzt.	
Dx_BASESEGFLASH:	Basis Adresse des FLASH Devices.	
Dx_BASEOFS:	Basis Adresse des File-Systems (relativ zum BASESEGFLASH).	
Dx_INITSECTORLEN:	Sector Grösse in Bytes (RTF-File System = 1)	
Dx_INITCLUSTERSIZE:	Sector Grösse in Cluster (RTF-File System = 32)	
Dx_INITTOTALSIZE:	Anzahl Cluster des Devices (max. 65535)	
	Device Grösse in Byte = SECTORLEN x CLUSTERSIZE x TOTALSIZE	

Um die Konfiguration des System-ROM's mit Flash-Device Treiber zu erleichtern, sind auf der System-Diskette unter dem Directory Projekte mehrere Batchdateien abgelegt, mittels welchen System-ROM's für die ModuNORM® PC/104 Module generiert werden können.

Starten des System-ROM's mit FLASH-Device:

Beim Starten des Device-Driver in der CONFIG.SYS Datei durch den Befehl `DEVICE = RTFFLASH.EXE` erstellt der Device-Driver ab dem ersten verfügbaren Laufwerk die konfigurierten FLASH Devices.

Fehlermeldungen des Device-Driver's:

- Not RTF file system

Im Konfigurations-File wurde für _FILESYSTEM nicht INACTIV oder RTF angegeben.

- Driver sub type not supported

In der Konfiguration _DRIVERSUBTYPE wurde ein nicht unterstützter Sub-Typ gewählt

- RTF file system not initialized (use Auto initialization)

Die automatische Initialisierung _AUTOINIT wurde auf NO gesetzt und das FLASH Device enthält kein gültiges RTF File System.

- Auto initialization failed

Das RTF File System konnte nicht initialisiert werden.

- FLASH type not supported

Der FLASH-Typ wird nicht vom Device-Driver unterstützt.

Erzeugen einer Datei im FLASH-File:

Die maximale Grösse einer Datei muss beim Erzeugen bekannt sein. Daher wird die eigentliche OPEN/CLOSE Funktion zwischengespeichert. Dazu merkt sich der Driver den Filenamem, die Operation, etc. in einer internen Datenstruktur im RAM. Bei der ersten Funktion, die auf das neue File angewandt wird, legt IDOS das File auch im eigentlichen RTF FLASH Device an. Der Dateiname wird in die Directory-Struktur aufgenommen, der notwendige Platz eingetragen etc.

Die erste Funktion auf das File sollte deshalb ein FSEEK mit der maximalen Grösse der Datei sein.

```
Beispiel:      fp = fopen ("c:\test.txt", "w");
               fseek (fp, 5000, SEEK_SET);
               fseek (fp, 0, SEEK_SET);
               fwrite (.....);
```

Damit wird eine neue Datei mit dem Namen TEST.TXT und der Grösse 5000 angelegt.

Formatieren des FLASH-Device:

Mit dem Tool RTFUTI.EXE kann ein FLASH Device formatiert werden.

Eingabe: rtfuti /F <Drive>

11. Konfigurations Beispiele

Um die Konfiguration des System-ROM zu erleichtern, sind auf der System-Diskette unter dem Directory Projekte mehrere Batchdateien abgelegt, mittels welchen System-ROM's für die einzelnen ModuNORM® PC/104 Module generiert werden können. Die Binay Files werden durch das Starten der Datei make.bat generiert.

Mit welcher Konfiguration Sie ein bestimmtes Modul in Betrieb nehmen können, entnehmen Sie der folgenden Tabelle.

ab Artikel-Nr:	Benennung:	Projekt Software:
102482A	CPU-386VGA	CPU386V1, CPU386V2 oder CPU386V3
102542A	CPU-386VGA	CPU386V1, CPU386V2 oder CPU386V3
102951A	PC-386VGA	CPU386V1, CPU386V2 oder CPU386V3
103101A	PC-386VGA	CPU386V1, CPU386V2 oder CPU386V3
103161A	PC-386VGA	CPU386V1, CPU386V2 oder CPU386V3
103251A	PC-386VGA	CPU386V1, CPU386V2 oder CPU386V3
103280A	PC-386VGA	PC386V1, PC386V2 oder PC386V3
102502A	CLCD-Controller	CPU386V1, CPU386V2 oder CPU386V3
103151A	CLCD-Controller	CPU386V1, CPU386V2 oder CPU386V3
103170A	PC-386CAN	PC386C1, PC386C2 oder PC386C3
103920A	CPU-386CAN/VGA	CP386CV1

12. Tools

Im Folgenden sind die auf der System-Diskette enthaltenen Tools kurz beschrieben:

RTFMKDEV.EXE

Den Devices (DEV_A, DEV_B, usw.) können die anwendungsspezifischen Programme wie folgt übergeben werden:

rtfmkdev device-file file1 file2 .. file n [optionen]

„file1“ und „file2“ bis „file n“ werden unter einem Device „DEVICE-FILE“ zusammengefasst.

Optionen:

/S=nnnnnn Device Grösse in Bytes

MERGBIN.EXE

Mehrere Binary Files können wie folgt zu einem einzigen Binary File zusammengefasst werden:

mergebin [/bios=len] [/code=xxxx] file1.bin offset file2.bin

In „file1“ wird das „file2“ geschrieben, ab offset Adresse „offset“.

Optionen:

/bios=len Generiert ein File mit einer BIOS Check-Summe.
Das File hat eine Grösse von „len“ kByte.

/code=0xhhhh Benutzer BIOS Signatur Code in Hex.

RTFDIR.EXE

Die Files im erstellten Device (DEV_A oder DEV_B) können wie folgt angezeigt werden:

rtfdir dev_a.bin

Ausgabe:	SIDTGPC	EXE	20192	(20192)
	CONFIG	SYS	31	(34)
	AUTOEXEC	BAT	34	(64)
	C_REMOTE	EXE	3192	(3200)
	DEVICES	EXE	12396	(12416)
	MEM	EXE	15747	(15776)
	6 File(s)		52128 Bytes used	
			13376 Bytes available	

RDCON.EXE

Das Programm RDCON.EXE hat zwei Funktionen:

- Remote-Console Programm
- Server für Remote Drives

Das Programm kommuniziert mit dem ModuNORM® PC/104 Modul über eine serielle Schnittstelle. Die Schnittstelle des PC's, über welche mit dem PC/104 Modul kommuniziert werden soll, kann mittels Parameter wie folgt eingestellt werden:

rdcon /c1 COM1 des PC

rdcon /c2 COM2 des PC

Um das Konsolenprogramm zu verlassen, müssen die folgenden Tasten gleichzeitig gedrückt werden:

[Left-Shift] [Right-Shift] [ESC]

In der Konfigurationsdatei IDOS.INI kann die Schnittstelle des ModuNORM® PC/104 Moduls definiert werden, über welche die Remote-Console mit dem PC kommunizieren soll.

OUTMODE.EXE

Die Anzeige des PC-Bildschirms sowie diejenige der am ModuNORM® PC/104 Modul angeschlossenen LCD-Anzeige können Ein- bzw. Ausgeschaltet werden.

outmode 0	Remote-Console aus	LCD-Anzeige aus
outmode 1	Remote-Console ein	LCD-Anzeige aus
outmode 2	Remote-Console aus	LCD-Anzeige ein
outmode 3	Remote-Console ein	LCD-Anzeige ein

SIDTGPC.EXE

Der Single Task Debugger auf dem ModuNORM® PC/104 Modul kommuniziert mit dem Programm SID.EXE auf dem PC über die serielle Schnittstelle. Das Starten des Single Task Debuggers erfolgt mit **sidtgp /c1** (bzw. mit **sidtgp /c2** bei Verwendung von COM2 des ModuNORM® PC/104 Moduls).

SID.EXE

Das Remote Debugger Modul auf dem PC kommuniziert mit dem SID.Debugger (SIDTGPC.EXE) auf dem ModuNORM® PC/104 Modul über die serielle Schnittstelle. Das Starten des Remote Moduls erfolgt mit **sid /c1** (bzw. **sid /c2** bei Verwendung von COM2 des PC). Die eingebundene Help Funktion beschreibt den Single Task Debugger.

Beispiel:

auf CPU-386EX: **sidtgps /c2**
auf PC: **sid /c1**

Der Single Task Debugger kommuniziert über die serielle Schnittstelle COM2 des ModuNORM® PC/104 Moduls mit der Schnittstelle COM1 des PC.

13. Versionstabelle

Die System-Diskette 001210K beinhaltet folgende Software Versionen:

Version Mikrap	File Name	Version Technosoftware
001550E	i386ex.bin	V1.09
001560A	kb8151.bin	V1.00
001570A	kbmatrix.bin	V1.00
001580C	dc_dc.bin	V1.11
001590D	ds1302.bin	V1.03
001600B	ib_com.bin	V1.0
001610B	ib_ext.bin	V1.0
001620A	ib_key.bin	V1.0
001630B	ib_rtx.bin	V1.01
001640D	ibioscnf.*	V1.05
001650F	idos.ini / idos.bin idosconf.*	V6.50 V1.13
001660C	PROM Util 386EX mergebin.exe rdcon.exe mem.exe devices.exe outmode.exe c_remote.exe edit.exe rtfdir.exe rtfmdkdev.exe	V1.2 V1.0 V6.30 V6.30 V1.0 V1.0 V1.01 V6.30 V6.30
001670A	SID-Debugger sid.exe sidtgp.exe	V3.11 V3.11
001790G	Flash-Device Treiber rtflash.exe rtfuti.exe	V1.05 V1.0