

00liesmich.txt

```
-----  
; Z9001 MEGA-FLASH-Modul (2.5M-Modul)  
; (c) V. Pohlens 2011  
; letzte Änderung 15.03.2017  
-----
```

alternative Software fürs MegaROM, MegaFlash, Buebchen-Rx3
und auch f. Ulrichs ROM-Bank, 64K-SRAM Modul und das KOMBI-Modul

```
-----
```

Binaries:

packedroms.bin_kombi.txt enthält die Programmübersicht für kombimodul.bin
packedroms.bin_mega.txt enthält die Programmübersicht fürs Mega-Flash

kombimodul.bin der ROM fürs 64K-SRAM Modul und das KOMBI-Modul
rom_1.bin..rom_5.bin die ROMs fürs MEGA-Flash-Modul
megarom.bin der ROM für den JKCEmu (= rom_1.bin..rom_5.bin)

Beim alten Megamodul sind es nur 3 ROMs; rom_1.bin+rom_2.bin und
rom_3.bin+rom_4.bin sind jeweils zusammen in einem 1-MByte-ROM enthalten.
rom_5.bin kommt in den dritten ROM.

```
-----
```

Zum Selbstkompilieren benötigt man

- den Arnold-Assembler <http://john.ccac.rwth-aachen.de:8000/as/>
Version AS V1.42 Beta [Bld 82] oder aktueller
Achtung: ich nutze die Windows-Version asw.exe, umbenannt in as.exe
Die DOS-Version läuft unter Windows XP etc. nicht!
- eine Perl-Installation, z.B. <http://www.activestate.com/activeperl>
Hier sollte die Version egal sein; ich nutze noch v5.14.2

Im Makefile sind die Pfade zu den Programmen as, pbin und perl anzupassen.
In includes.asm wird die Konfiguration eingestellt (MEGA für Megarom und
Megaflash,
MEGA8 für die 8K-Segment-Variante, lastbank ist für kleinere FLash-Speicher zu
verändern)

assemblieren

```
make depend  
make  
make 2x make, damit die Prüfsummen von CHKROM stimmen
```

```
make mega für Megamodul (3 ROMs)  
make flash für Mega-Flash-Modul (5 ROMs)  
make kombi f. 64K-SRAM-Modul, KOMBI-Modul bzw. Buebchen-Rx3
```

(1 ROM)

testen

```
jkcemu (www.jens-mueller.org/jkcemu/)  
Konfiguration KC87 + 2x 16K-RAM-Module (oder ein 64K-RAM-Modul),
```

Megamodul

```
externe Datei "megarom.bin" f. Megamodul laden
```

ROMs:

Die Datei "megarom.bin" muss in 5 Teile a 512 KByte zerlegt werden (ROM 1..5).
Das erledigt "make flash". Beim Megamodul sind es nur 3 ROMs; 1+2 und 3+4 sind
jeweils zusammen in einem 1-MByte-ROM enthalten. Das erledigt "make roms".

Beim Kombimodul wird eine Datei "kombimodul.bin" erzeugt.

```

;-----
; Arbeitsweise
; Hinzufügen weiterer Programme
;-----

```

Das OS des Z9001 ist analog zum CP/M aufgebaut. Die oberste Schicht, die Kommandoeingabe CCP, kann durch ein eigenes Programm ersetzt werden. Dazu dient das Kommando "# ".

Die Mega-Flash-Software nutzt genau dies aus, um das CCP zu erweitern.

Zum einen wurde eine Bankrückschaltung integriert; so das Programme beim Beenden wieder die Bank mit der Systemerweiterung (kurz Systembank) aktivieren.

Dadurch kann jede Software, z.B. originale ROM-Modul-Software, unverändert bleiben.

Es muss keine spezielle Enderoutine o.a. gepatcht werden.

Zum anderen wurde ein neuer Kommandorahmen eingeführt.

Kurzer Exkurs: Was ist das?

OS-Programme werden anhand eines Kommandorahmens gefunden. Dieser OS-Rahmen liegt auf einer xx00h-Adresse und sieht so aus

```

org      xx00h
jp       start
db       "NAME      "      ; genau 8 Zeichen
db       0                ; Ende eines Kommandos
db       0                ; Ende der Liste

```

Details s. OS-Handbuch. Die hier stehenden Programmnamen können im CCP eingegeben werden. Das CCP suche den Programmnamen in allen solchen Kommandorahmen und starten bei gefundenem Programmnamen das Programm. Andernfalls erscheint die Ausschrift "start tape".

Der OS-Kommandorahmen ist im Modul für Programme nutzbar; die im Speicherbereich von C000-E7FF arbeiten (also z.B. Inhalte originaler ROM-Module), oder die eine eigene Umladeroutine besitzen, die das eigentliche Programm erst an die Zieladresse im RAM kopieren und dort starten.

Es gibt nun einen weiteren Kommandorahmen, den FA-Rahmen.

Dieser ist 32 Byte lang und liegt ebenfalls auf einer xx00h-Adresse.

Nach dem FA-Rahmen folgt das Programm. Der FA-Rahmen hat folgenden Aufbau:

```

org      xx00h                ; header
db       0FAh, 0FAh          ; +0 Kennbytes
db       Dateityp            ; +2 0-MC, 1-BASIC (s. includes.asm)
db       "NAME      "      ; +3 genau 8 Zeichen
dw       aadr                ; +11 Anfangsadresse im RAM
dw       eadr                ; +13 Endadresse im RAM
dw       sadr                ; +15 Startadresse im RAM (oder FFFFh -
nichtstartend)
dw       länge              ; +17 (Datei-)Länge des nachfolgenden Programms
db       Dateikategorie     ; +19 Standard 0 (s. includes.asm)
db       "Kommentar  "      ; +20 12 Zeichen, bel., z.B. Autor o.ä.
;
binclude      programm.bin

```

Dieser Kommandorahmen ist für beliebige RAM-MC-Programme, aber auch BASIC-Programme nutzbar (geplant ist die Unterstützung weiterer Datentypen z.B. Forth-Programme).

Außerdem können alle Programme mit bitbuster_extreme komprimiert sein, um Platz im Mega-Modul zu sparen. Bei komprimierten Programmen muss im Dateityp das Bit 7 gesetzt sein (also 80h zum originalen Dateityp addiert).

MC-Programme mit diesem Rahmen werden zuerst an die korrekte Adresse aadr im

00liesmich.txt

Speicher umgelagert und dann auf der Startadresse sadr gestartet. programm.bin ist einfach der binäre Speicherabzug des Programms von aadr bis eadr. Praktisch sind das *.KCC-Dateien OHNE den Kopfblock. (Die Informationen aus dem Kopfblock stehen schon alle im FA-Rahmen; damit spart man ein bisschen Speicher im Mega-Modul).

Basic-Programme werden nach 0401h kopiert. Dann wird die BASIC-Bank zugeschaltet, Basic initialisiert und das Programm gestartet. Für BASIC-Programme ist programm.bin einfach die *.KCC-Datei (ohne die ersten 13 Byte!).

Zur einfachen Konvertierung von *.tap-Dateien ins binäre Format kann das Perl-Programm tap2bin.pl genutzt werden. Mit tap2fa.pl wird auch gleich ein FA-Header erzeugt.

Das Mega-Modul kann dadurch einfach um eigene Software erweitert werden: Einfach in einer beliebigen Bank (außer der Systembank) in einem freien Bereich auf einer xx00h-Adresse ein Programm mit OS-Rahmen oder mit FA-Rahmen speichern. Fertig!

Ebenso kann man in der Datei packedroms.asm eigene Einträge hinzufügen:

```
addFile
"robotron/r+hanoi.bin",0401h,,,"R+HANOI",ft_BASIC,fk_spiele_basic,""
```

```
Durch Aufruf von
make depend
make
```

wird aus der TAP-Datei robotron/r+hanoi.tap automatisch eine FA-Datei erzeugt und in den ROM aufgenommen.

Fügt man an den Filenamen noch die Endung .pck an und an den Dateityp

```
"ft_packed",
wird auch eine gepackte FA-Datei erzeugt.
```

```
addFile
"robotron/r+hanoi.bin.pck",0401h,,,"R+HANOI",ft_BASIC+ft_packed,fk_spiele_basic,
""
```

```
;-----
; Speicherbelegung
;-----
```

Es werden Speicherzellen im Bereich 00-100h genutzt.

0000	: jp FWBOOT	Sprung zu Megamodul-Init
0028	: jp SBOS	RST 28H für den Sprungverteiler
0042	: currbank	aktuelle Bank
0043	: firstent	temp. Zelle f. Menu
0044	: DATA	Konvertierungsbuffer
0046	: ARG1	1. Argument
0048	: ARG2	2. Argument
004A	: ARG3	3. Argument
004C	: ARG4	4. Argument
004E	: bkswcode	Bankumschalt-Code 004E-0059
0110	: tmpcmd	temporärer Programmcode 0110h-01CFh

Auf dem Stack liegt zusätzlich die Bankrückschaltung zur Systembank. Damit wird beim Beenden eines Anwenderprogramms wieder ins EOS des Mega-Moduls zurückgeschaltet, auch wenn während der Ausführung des Anwenderprogramms eine andere Bank aktiv war.

Mit Eintritt in ein Anwenderprogramm ist SP = 1FAH. Auf Adresse 1FAH steht die Adresse der Fehleranzeige (ERDIS), auf Adresse 1FCH die Adresse der Bankrückschaltung setbk0, und auf Adresse 1FEH die Adresse des EOS (GOCPM).

Der Bankumschalt-Code 004E-0059 sollte nicht überschrieben werden. Alle anderen Speicherzellen können im Anwenderprogramm überschrieben werden!

00liesmich.txt

Der Bereich 0042-005B ist vom OS als frei für systemrelevante Programme wie IDAS, DEBUGGER, usw. gekennzeichnet. Sie sollten nicht im Anwenderprogramm benutzt werden. ZM, IDAS, EDAS nutzen diesen Bereich nicht. Wenn diese Speicherzellen doch im Anwenderprogramm benutzt werden, muss dieses Anwenderprogramm mit JP F000 oder RESET-Taste beendet werden!

Der Bankumschalt-Code:

```

004E : 21 03 F0      FWBOOT:      ld      hl,WBOOT      ; OS-Routine
0051 : E5                push    hl          ; Return-Adresse auf
Stack
0052 : 3E 00          setbk0:      ld      a,systembank
0054 : D3 06                out     6, a        ; Hi-RAM eines
64K-RAM-Moduls wieder ausschalten
0056 : D3 FF          setbk1:      out     bankport, a    ; Systembank einschalten
0058 : C9                ret              ; WBOOT starten

```

0110-01CF wird intern genutzt, wenn das EOS des Mega-Moduls aktiv ist, also bei Kommandosuche, DIR, Programme laden etc.

0042 kann vom Anwenderprogramm ausgelesen werden, um die aktive Bank zu ermitteln.

Die Speicherzelle kann in eigenen Programmen überschrieben werden.

0043-004C wird für die Kommandoparameter-Aufbereitung im Sprungverteiler genutzt und kann in eigenen Programmen überschrieben werden.

```

;-----
; Sprungverteiler
;-----

```

Für Systemkommandos wurde ein Sprungverteiler programmiert, der häufig benötigte Routinen bereitstellt. (analog Monitor-Sprungverteiler des Z1013)

Details zu den Routinen sind derzeit dem System-Quelltext zu entnehmen

Aufruf:

```

rst 28h
db nr

```

Nr	Routine	
0	OUTHX	Ausgabe (A) hexa
1	OUTHHL	Ausgabe (HL) hexa
2	WAIT	Unterbrechung Programm, wenn <PAUSE> gedrueckt; weiter mit beliebiger Taste
3	COLOR	Vordergrundfarbe setzen (E)
4	CPROM	Suchen Namen
5	FMOV	FA-Programm in Speicher kopieren
6	FRUN	FA-Programm starten
7	KDOPAR	Kommandoparameter aufbereiten: bis zu vier Hex-Parameter werden in ARG1..ARG4 abgelegt. Wenn erster Parameter ein ':' ist, gelten die alten Werte erneut. Fehlende Parameter werden 0 gesetzt
8	INHEX	Konvertierung ASCII-Hex ab (DE) --> (HL)
9	PRST7	Ausgabe String bis Bit7=1
10	GOCPM	Warmstart
11	JMPHL	Program starten (nach CPROM)
12	CP_NXBK	Bankumschalt-Code umlagern nach tmpcmd
13	STOPKEY	Test, ob <STOP> gedrueckt -> Cy=1
14	CLOAD	Datei laden. in: (fcb), hl, a A=0 => Dateiname+Typ ist bereits im FCB eingetragen A=1 => Dateiname "Name[.Typ]" muss in CONBU abgelegt sein A=2 => zuerst Abfrage "Filename:" HL = 0 => orig. aadr wird genommen

00liesmich.txt

HL <> 0 => aadr

```
15      CSAVE   Datei speichern. in: (fcb), a (wie CLOAD)
16      COOUT   Ausgabe ab (HL) (B) Zeichen, nur Buchstaben
```

Beispiel: Abfrage Dateiname + Datei laden

```
ld      hl,0           ; orig. aadr nehmen
ld      a,2           ; Abfrage Dateiname
rst     28h
db      14            ; CLOAD
```

Beispiel: Abfrage Dateiname + Datei speichern 1000h-10FFh

```
ld      hl,1000h
ld      (fcb+17), hl  ; aadr
ld      (fcb+21), hl  ; sadr
ld      hl,10FFh
ld      (fcb+19), hl  ; eadr
ld      a,2           ; Abfrage Dateiname
rst     28h
db      15            ; CSAVE
```

Beispiel: Anzeige FCB

```
org     300h
OSPAC:  EQU     0F310H
OCRLF:  EQU     0F2FEH
fcb:    equ     5ch
jp      start
db      'FCB      ',0
db      0
start:  ld      hl,fcb
        ;Dateiname
        ld      b,8
        rst     28h
        db      16            ; COOUT
        call    ospac
        ;Dateityp
        ld      b,3
        rst     28h
        db      16            ; COOUT
        call    ospac
        ;aadr
        ld      hl,(FCB+17)
        rst     28h
        db      1            ; OUTHL
        call    ospac
        ;eadr
        ld      hl,(FCB+19)
        rst     28h
        db      1            ; OUTHL
        call    ospac
        ;sadr
        ld      hl,(FCB+21)
        rst     28h
        db      1            ; OUTHL
        call    ospac
        ;ende
        call    OCRLF
        rst     28h
        db      10           ; GOCPM
```

```
-----
; kurze Historie
-----
```

08.01.2012 18:58:20

s.a. <http://hc-ddr.hucki.net/wiki/doku.php/z9001:module:megaflash>
Es fehlt noch eine Suchfunktion, die Startfunktion für Basic, das Entpacken gepackter Programme und natürlich reichlich Software.

00liesmich.txt

Evtl. wird es später auch eine graphische Oberfläche ala A.S. geben...

09.01.2012 15:54

ok: Startfunktion für Basic, das Entpacken gepackter Programme
ok: robotron-software
todo: Suchfunktion in DIR
Fehler: R+MOND, R+DEMO2, ...
R+DEMO1 geht!

10.01.2012 09:47

ok: unpack-routine
todo: Suchfunktion in DIR
todo: crt40p, 80p-Treiber: Farbattribut wird zu spät beachtet (bei Zeilenumbruch ist erstes Zeichen noch in alter Farbe)?? wird erst das Farbattribut gesetzt und dann gescrollt?
NEIN -> Farbe + CR LF ist falsch. Besser FARBE + LF + CR ! Oder CR+LF + Farbe !
Oder kein Farbattribut setzen/schreiben bei Cursorsteuerung -> Quellcode anschauen!
todo: 64K-Modul zurücksetzen
run aadr bank

12.01.2012

Code umstrukturiert: Systembank hat eigenes Unterverzeichnis
ok: 64K-Modul zurücksetzen
run aadr bank
neu: bank 1..15 kann individuell zugeschaltet werden

9.2.2012: Der Suchalgorithmus für DIR wurde komplett neu programmiert. Als Wildcards zur Suche gibt es '*' für beliebige und beliebig viele Zeichen und '?' für genau ein beliebiges Zeichen. Gesucht wird nun nach Namensfragmenten, d.h. eine Suche 'DIR V' listet alle Programme, die ein 'V' im Namen haben.

10.02.2012: DIR kennt nun den Parameter 'C' (comment)

12.02.2012: neue Kategorie fk_hidden: Datei wird bei DIR nicht aufgelistet
Das ist für Loader gedacht, wenn das Programm nicht direkt gestartet werden kann,
sondern ein Loader spezielle Anpassungen/Ausgaben vornehmen und das Programm selbständig laden soll

13.02.2012: neue BASICs: basic_16d.rom: 16 Farben, 80 Zeichen/Zeile, korr. CSAVE
basic_16dp.rom: 16 Farben, 80 Zeichen/Zeile, korr. CSAVE, neues PRINT-AT
letzteres Basic wird automatisch genutzt, wenn ein alternativer CONS-Treiber geladen ist. CSAVE wurde so geändert, dass ein brauchbarer Block 0 geschrieben wird
(in Vorbereitung auf DiskOS)

xx.02.2012: robotron2: robotron-Entwicklungen, die nicht auf Kassette veröffentlicht wurden

24.02.2012: DiskOS ist als DOS4 bzw. DOSX mit drin.

10.03.2012: Es gibt 3 CP/Ms und als viertes das MiniCPM. In der include-Datei includes.asm kann festgelegt werden, ob 1 oder 2 ROM-Floppies genutzt werden sollen. Außerdem sind diverse Testprogramme wie CHKROM, BANKTEST, LPRO etc. hinzugefügt.

21.03.2012: Die Programme von Peter Weigoldt sind hinzugekommen. s. Webseite <http://home.tiscali.de/petwe/kc.html>

02.04.2012: Es gibt einen neuen Filetyp ft_HELP. Dieser ist für das neue Systemkommando Help gedacht (Ausgabe kurzer Hilfetexte). Dateien vom Typ ft_HELP sind einfache Textdateien mit Farbbefehlen u.a. Dateiende ist 00h. Die Dateien

00liesmich.txt

können auch komprimiert werden.

Der Text wird in den Shadow-RAM ab 7800h entpackt; ist kein 64K-Modul gesteckt, wird der normale RAM überschrieben.

19.09.2012: Zum Erstellen und leichten Bearbeiten der Hilfe-Dateien gibt es den einfachen Editor hlpedit.pl.

Neu ist ebenfalls das (derzeit noch externe) Kommando MENU, bekannt vom Megamodul.

Allerdings werden hier die Datei-Dialoge dynamisch erzeugt :-)

****14.10.2012****: Korrekturen in der Systemsoftware bzgl. Startverhalten.

Bei gepackten Dateien wurde AF' nicht gesichert (wichtig f. GVAL), bei gepackten Dateien steht die Ladebank in B (f. CHKROM)

****25.10.2012****: megarom_info.asm ist eine kleine Info zur Übersicht über den gepackten ROM (make megarom_info.bin). Auf die Ausgabe beim Kompilieren achten! Es wird kein Programm erzeugt :)

Anzahl FA-Programme: 277, Gesamt-gepackt: 1214718, Gesamt-Speicher: 1744575.

****12.12.2012****: Für das neue MEGA-8KSEG-ROM-RAM-Modul von Bübchen wurde der Code

überarbeitet. In der includes.asm wird voreingestellt, für welches MEGA-Modul die Software erzeugt werden soll. Beim neuen MEGA-8KSEG-ROM wird nur die Datei packedroms.bin benötigt und linear auf die EPROMs aufgeteilt; eine Umsortierung auf 2K-Häppchen entfällt.

****27.12.2012****: 100 neue Programme, größtenteils aus dem orig. MegaROM rekonstruiert (Verzeichnis Soft5)

****01.04.2013****: neues DiskOS, jetzt auch für Floppies; Installation überprüft

****25.04.2013****: Anpassung für OS 1.1 (Z9001.84)

****19.08.2013****: Fehler in MEM behoben; Eingabe eines einzelnen 00-Werts war nicht möglich.

Neue Systemkommandos SAVE und FCB. Das OS-SAVE heißt nun OSSAVE.

Kommando SDX in System-ROM aufgenommen. Ebenso V24X.

****25.01.2015****: nach diversen Miniänderungen und Anpassungen an andere Modulooptionen ist nun die Bankumschaltung modifiziert. MENU läuft wieder!

****30.04.2015****: Fehlerkorrektur DIR-Wildcard-Routine, MENU Anzeige OS-Namen Version für UZ-64K-SRAM-Modul mit abwechselnd 10K und 6K-Bänken.

TODO: Hier funktionieren gepackte Programme nicht, wenn sie über das Bankende hinaus in die nächste Bank gehen

```
make megarom_info.bin
Anzahl FA-Programme: 409
Gesamt-gepackt: 1642496
Gesamt-Speicher: 2600448
```

wg. Fehler bei MEM Überarbeitung des MEM-Kommandos. Ursache für das Problem ist der Z1013-Monitor. Der schreibt direkt in den BWS alle nachfolgenden Zeichen sind Leerzeichen oder aber der Monitor beginnt (mit Nicht-Leerzeichen und Nicht-Hex-ASCII)

****17.06.2015****: Version für UZ-64K-SRAM-Modul: make ROM

In includes.asm ist die EPROM-Größe auszuwählen (für einen 256K-EPROM (27C020) werden z.B. nur 32 Bänke durchsucht), s.u.

Zusätzlich zum gewohnten Umfang gibt es Kommandos für die

Uhr (RTC, DAT, ...), die Entpackroutine ist anders, und ein paar kleine

Ergänzungen. Die Suche ist künstlich verlangsamt, damit man die Banknummern durchlaufen sieht (Wunsch von Ulrich, weil er das gut findet, wenn man was sieht).

Die Banknummer wird rechts oben mit

angezeigt. Uhrzeit und Datum sollten nach Reset zu sehen sein. Der Zusatzmonitor ZM wurde modifiziert, damit er besser mit der KRT zusammenarbeitet.

neue Entpackroutine unpack: Nun wird nach jedem Byte geprüft, ob

Bankweitzerschaltung

nötig ist. Das ist etwas langsamer als die optimierte Version für MegaROM, aber

00liesmich.txt

das fällt nicht weiter auf.

systemerweiterung: hier können Erweiterungen programmiert werden. In Bank 0 ist noch Platz. Beispiel bank0.asm und rtcdat_uz.asm

```
;syserv_reset equ 0C025h
;syserv_cold equ 0C028h
;syserv_gocpm equ 0C02Bh
```

auf C02E steht die Nr. der letzten Bank. Patchen auf tatsächliche ROM-Größe:

```
;27010 128K EPROM => 0fh
;27020 256K EPROM => 1fh
;27040 512K EPROM => 3fh
;27080 1M EPROM => 7fh
```

****22.05.2016****: neue Version für 64K-SRAM- und Kombi-Modul umfangreiche Überarbeitung der Bankrückschaltung, damit die Lade-Bank in der Anzeige stehen bleibt (bei Kombi-Modul aktiv). Unterstützung von 64K-SRAM und Kombi-Modul mit gemeinsamen Code. Unterstützung des USB-Moduls im OS (Kommando USB). Unpacker depack_extreme.exe für gepackte bin-Dateien (ohne Fa-Header). Als Folge der überarbeiteten Bankrückschaltung müssen Programme, die die Systembank nutzen (z.B. Sprungsverteiler), nun unbedingt ft_systembank im FA-Dateityp-Byte enthalten.

Jetzt wird eine Textdatei packedroms.bin.txt erstellt, die den Inhalt des ROMs auflistet (ähnlich dem DIR-Kommando im OS).

Für 64K-SRAM-Modul und KOMBI-Modul wird nun dieselbe ROM-Datei "kombimodul.bin" genutzt. Die Software erkennt das korrekte Modul und gibt eine entsprechende Meldung aus.

****07.07.2016****: MiniCPM für Kombi-Modul, Korrekturen f. Kombi-Modul, GIDE-Treiber, GIDE-RTC im Megaflash, CPM-48K um ROM-Floppy erweitert, auch für Kombi-Modul extra Version, INITKC zum Diskettenformatieren im OS. Damit kann man eine Bootdiskette nun selbst erstellen:
initkc, cpm-48k, pip a:=c:@cpmz9.com

****07.12.2016****: neu LOAD, USB heißt jetzt USBX, kleine Korrekturen im Code. Dank an Rolf W. fürs Testen und Fehlerfinden!

****15.03.2017****: MiniCPM wieder lauffähig (Dank an M.Bagola fürs Fehlerfinden). Zu ROM-Disk die KCNET-Software TFTP1287.COM und CPMN1587.COM hinzugefügt.

****01.08.2017****: neue Version vor allem fürs Kombi-Modul (vom April) Neues MINICPM speziell fürs KOMBI-Modul, mit RAM-Disk-Einbindung. Banknummer wird jetzt heruntergezählt (BWS r.o.). Wenn der ZM als Geräte-Treiber oder Debugginghilfe genutzt wird (rst 38), geht das nur mit der RAM-Version.