

**Vorwort**

Der „USB-Komfort-Monitor“ ist ein "Arbeitsmonitor" für den AC1 und nicht als Test- oder DebugMonitor zu verstehen. Der Monitor dient vornehmlich zum Laden und Speichern von Files über verschiedene Medien und beinhaltet die wichtigsten Kommandos aus dem Monitor 10/88.

Es ist nur noch 1 Monitorversion für AC1-SCCH und AC1-2010 notwendig! Eine angepasste Version steht auch für den LLC2 zur Verfügung.

Fast alle wichtigen Einsprungadressen ab Monitor V 8.x sind unverändert ! RST-Befehle und Sprungverteiler sind kompatibel zum AC1 Monitor V3.1.

Die meisten bisher veröffentlichten AC1-Programme, die auf Monitor ab V8.x basieren, bleiben lauffähig. Ausnahme: TurboTape!

Da viele TurboTape-basierten Programme recht wild und undokumentiert in die Routinen im Monitor springen, ist im Monitor ab 10.3 ein Abbruch von derartigen Aufrufen aus TurboTapebasierten Programmen eingebaut. Das betrifft z.B. TurboLoadCopy, MultiCopy oder Clist-Turbo.

Im Vergleich zum Vorgänger Monitor 10.3 ist SAVE/LOAD per V.24 (ehemals Monitorbuchstaben K und L) aktuell nicht mehr enthalten, jedoch als **nachladbares Kommando** berücksichtigt, siehe dort!

Die normale Ein- und Ausgabe per V.24 (z.B. für den Druckeranschluss) wird jedoch weiterhin realisiert und sind direkt ansprechbar

Eine Besonderheit gilt beim **Grafik-Sound-Basic 3.2**:

Die CLOAD & CSAVE Routinen verwenden dank spezieller Programmierung ab Monitor 11.0 jetzt direkt HD oder USB! Auf die Unterstützung der Kassettenarbeit bzw. V.24 (Mon. 10.3) wurde verzichtet. Grafik-Sound-Basic 3.2 selbst ist unverändert.

**Inhalt**

[Bedienung Monitor 11.0 3](#_Toc49995)

[Einschalten 3](#_Toc49996)

[Tastatur 4](#_Toc49997)

[Integrierte Kommandos 5](#_Toc49998)

[Nachladbare Kommandos 9](#_Toc49999)

[Beschreibung von Systembestandteilen 10](#_Toc50000)

[Startroutinen des Monitors 10](#_Toc50001)

[Dateitypen 10](#_Toc50002)

[Dateinamen 10](#_Toc50003)

[Ein-/Ausgabe 10](#_Toc50004)

[V.24 - Schnittstelle 11](#_Toc50005)

[V.24 - Kontrollregister 11](#_Toc50006)

[Hardware 12](#_Toc50007)

[Ablauf Betriebssystem Monitor 11.0 12](#_Toc50008)

[Weitere Betriebssysteme 13](#_Toc50009)

[Erweiterung der Funktionalitäten von GS-BASIC 14](#_Toc50010)

[Wahl des Modus/Speichermediums 14](#_Toc50011)

[Inhaltsverzeichnis anzeigen 14](#_Toc50012)

[Datei Laden 14](#_Toc50013)

[Datei sichern 14](#_Toc50014)

[Felder 15](#_Toc50015)

[Verwendung von CLOAD/CSAVE in BASIC-Programmen 15](#_Toc50016)

[Externes SAVE/LOAD 16](#_Toc50017)

[Hinweise für HD-Nutzung 17](#_Toc50018)

[GIDE 17](#_Toc50019)

[Einstellung HD-Typ 17](#_Toc50020)

[HD-Puffer 17](#_Toc50021)

[Defragmentierung 17](#_Toc50022)

[Hinweise für USB-Nutzung 18](#_Toc50023)

[Anhang 19](#_Toc50024)

[Betriebssystem-Routinen und Voreinstellwerte 19](#_Toc50025)

[Speicherbelegung (Memory Map) 20](#_Toc50026)

[Peripherie 21](#_Toc50027)

[Weitere Hardwarebezüge 22](#_Toc50028)

[Belegung PIO 1/B 22](#_Toc50029)

[BREAK-Taste/NMI 22](#_Toc50030)

[Grafiktaste 22](#_Toc50031)

[Joystick 22](#_Toc50032)

# Bedienung Monitor 11.0

Der USB-Komfort-Monitor 11.0 belegt wie seine Vorgängerversionen 4 kB Speicher und läuft nur im Adressbereich 0 - 0FFFH.

Der Monitor umfasst das Betriebssystem und stellt die wichtigsten Routinen zum Laden und Ausführen der Software dar. Er ist nicht als Test- und Debug-Monitor konzipiert. Hierzu empfiehlt es sich den Monitor V10/88 zu verwenden, der nebenbei einige Fehler der V8.x Monitore beseitigt.

Das Betriebssystem (Software) dient der Verbindung der Computerhardware mit den Nutzerprogrammen (Anwendersoftware). Es realisiert die Ein/Ausgabe (Tastatur, Bildschirm, V24-Schnittstelle, USB, Festplatte), die Verwaltung des Speichers und ist erweiterungsfähig.

Der Monitor enhält nunmehr universelle Routinen für den Zugriff auf moderne Massenspeichermedien.

Die nutzbaren Betriebssystemroutinen und die Belegung des Arbeitsspeichers sind den Tabellen im Anhang zu entnehmen.

## Einschalten

Nach dem Einschalten startet sofort ab Adresse 0H der Monitor die Initialisierung und meldet sich mit der Überschriftszeile und 2 Zeilen weiter mit dem Promptsymbol (Doppelkreuz), dahinter blinkt der Kursor.

Beim 1.Kaltstart wird der gesamte RAM von 18D6h bis FFFFh mit FFh gelöscht. **RESET** oder **J 0** löschen jedoch keinen RAM !

Der AC1 befindet sich jetzt im Maschinensprache-Monitor und es lassen sich Kommandos aufrufen, Programme laden und starten oder in eine andere Betriebssystemebenen springen (Basicinterpreter, Programmpaket X, CP/M).

## Tastatur

Neben der Eingabe von Zeichen und Zahlen erfolgt in Verbindung mit der CTRL-/Strg-Taste die Eingabe von Steuerfunktionen. Diese sind notwendig, um z.B. fehlerhafte Eingaben zu korrigieren (Zeichen überschreiben, einfügen, löschen usw.).

Steuer-/Editier-/Kursortasten (Taste Strg und Taste Buchstabe gleichzeitig drücken !)

|  |  |  |  |
| --- | --- | --- | --- |
| **Code** | **Taste** | **Funktion** | |
| 01h | A | Home, Kursor oben links | |
| 02h | B | Bildschirm ab Kursorposition löschen | |
| 03h | C | Zeile ab Kursorposition löschen | |
| 04h | D | Delete; 1 Zeichen löschen, Zeile rückt nach links | |
| 05h | E | Insert; Space einfügen, Zeile rückt nach rechts | |
| 06h | F | Kursor an den Anfang der Zeile | |
| 07h | G | BEL, akustisches Signal | |
| 08h | H | Kursor nach links | |
| 09h | I | Kursor nach rechts | |
| 0Ah | J | Kursor nach unten | |
| 0Bh | K | Kursor nach oben | |
| 0Ch | L | Bildschirm löschen, Kursor Home | |
| 0Dh | M | CR, Kursor an Anfang nächster Zeile; Enter | |
| 0Eh | N | Kursor direkt positionieren (vgl. Hinweis) | |
| 0Fh | O | Tabulator 8 Spalten | |
| 10h | P | Folgende Zeichen normal darstellen | |
| 11h | Q | Folgende Zeichen invers darstellen (bis Zeilenende bzw. Ctl.+ P) | |
| 12h | R | F-Tasten in BASIC | BREAK |
| 13h | S | PAUSE |
| 14h | T | LIST |
| 15h | U | RUN |
| 16h | V | CLOAD“ “ |
| 17h | W | CSAVE |
| 18h | X | Drucker ein, V 24-Schnittstelle | |
| 19h | Y | Ein-/Ausgabe normal, Drucker aus | |
| 1Ah | Z | Zeichensatz umschalten | |
| 7Fh |  | Kursor nach links und Zeichen löschen (Del/Rubout) | |

Strg+ N : Direkte Kursorpositionierung

Nach Strg + N steht der Kursor Home.

Mit den Zifferntasten dezimale Bildschirmposition angeben:

1. Reihenposition ( 00 bis 31 )
2. Zeilenposition ( 00 bis 63 )

Der Kursor kann an jede beliebige Bildschirmposition gebracht werden. Dargestellte Zeichen werden dabei nicht verändert (der Kursor blinkt abwechselnd zum Zeichen). Fehlerhafte Zeichen können nur durch Überschreiben mit anderen Zeichen, mit Delete oder mit DEL/Rubout korrigiert werden.

Um den Kursor um mehrere Bildschirmpositionen zu bewegen oder Zeichen mehrfach zu schreiben, braucht die entsprechende Taste nur gedrückt zu bleiben. Das Zeichen wird mit erhöhtem Tempo ständig wiederholt (Repetierfunktion). Wird der Bildbereich nach oben verlassen, erscheint der Kursor an gleicher Stelle am unteren Bildrand.

## Integrierte Kommandos

Kommandos können eingegeben werden, sobald sich der Monitor mit seinem Promptsymbol (Doppelkreuz) meldet. Dabei hat die Befehlszeile folgende Form:

# X aaaa bbbb cccc

**X** ist das entsprechende Zeichen, mit dem das Kommando abgekürzt wird.

**aaaa bbb cccc**

Hinter dem Kommandozeichen können bis zu 3 max.4-stellige Argumente folgen (hexadezimal). Hinter jedem Argument muss mindestens 1 Leerzeichen stehen, führende Nullen können entfallen. Werden weniger Argumente angegeben, als zu dem Kommando erforderlich, wird für die fehlenden der Wert 0 gebildet. Die Angabe eines Doppelpunktes hinter dem Kommandozeichen bewirkt, dass die Argumente des vorhergehenden Kommandos verwendet werden.

Die Befehlszeile wird mit Taste "Enter" (entspricht Taste "CR") vom Bildschirm übernommen, analysiert und ausgeführt. Die Übernahme der kompletten Befehlszeile vom Bildschirm in den Computer hat den Vorteil, dass Korrekturen jederzeit noch möglich sind oder ein Befehl mehrfach ausgeführt wird, indem der Kursor erneut in die Befehlszeile positioniert wird (beliebige Position hinter Doppelkreuz).

Debugger:

Der im Monitor V10/88 enthaltene Debugger ist zugunsten der Lademöglichkeit von GIDE und

USB entfallen. Einige Funktionen sind jedoch noch vorhanden und können verwendet werden. Die Kommandozeichen sind mit den geforderten Argumenten zu übergeben. In Klammern stehende Argumente sind nicht unbedingt erforderlich.

|  |  |  |
| --- | --- | --- |
| ! | BWS Farbe | BWS Farbe setzen (COLOR-BWS nötig) |
| 4 | 2/4Mhz | schaltet AC1 zwischen 2/4Mhz Takt um (BWSPort nötig) |
| b | Basic | Start GraficSoundBasic via Modul 1 |
| C | Compare | Vergleich von 2 Speicherbereichen |
| D | Dump | Hexdump und ASCII-Darstellung, sowie Modifikation |
| J | Jump | Sprung in Programm ab Adresse |
| l | HD Load | Lädt ein in DVHD gespeichertes File von HD |
| N | CRC | CRC Checksumme berechnen und ausgeben |
| P | Pattern | Füllt Speicherbereich mit Datenbyte |
| p | Clear RAM | Löscht RAM bis FF00H mit FFH |
| r | Basic Warmstart | Warmstart Grafik-Sound-Basic via Modul 1 |
| s | HD Save | Speichert ein File in DVHD |
| T | Transfer | Kopiert Speicherbereich in anderen Speicherbereich |
| U | USB Load | Lädt Files vom USB |
| u | USB SAVE | Speichert/Löscht Files auf USB |
| W | IO-Port beschreiben | Byte an IO-Port ausgeben |
| w | IO-Port lesen | liest den Portwert an IO-Port ein und gibt sie aus |
| X | Prog X | Sprung in Programmpaket X via Modul 1 |
| ? | Help | Liste der verfügbaren Kommandos |

Tabelle 1 Liste der Monitorkommandos

#### Befehlssatz

|  |  |
| --- | --- |
| **?** | **Help** |

Der Speicherbereich wird nach Startkommandos durchsucht und diese aufgelistet. Neben den Startkommandos des Debuggers werden Startbefehle geladener Anwenderprogramme angezeigt.

|  |  |
| --- | --- |
| **! aa** | **BWS Farbe setzen** |

Beschreibt den gesamten Farb-RAM mit dem Farbbyte aa (siehe Tabelle). Dazu wird die FarbResetRoutine verwendet, die das Vorhandensein des COLOR-BWS testet und ggf. ohne Reaktion abbricht.

Die neue Farbe aa gilt bis zum nächsten Kaltstart des Monitors, bzw. J 0. Einsprünge via GetCo und Modul X über Adresse 006EH ändern keine Farbe.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Bit 0 | Zeichen ROT | Bit 4 | Hintergrund ROT | |
| Bit 1 | Zeichen GRÜN | Bit 5 | Hintergrund GRÜN | |
| Bit 2 | Zeichen BLAU | Bit 6 | Hintergrund BLAU | |
| Bit 3 | Zeichen INTENSIV | Bit 7 | Hintergrund INTENSIV | |
| **4** |  |  |  | **Taktfrequenz umschalten** |

Schaltet zwischen 2/4Mhz die Taktfrequenz um und setzt den COLOR-BWS voraus.

|  |  |
| --- | --- |
| **b** | **BASIC starten** |

Kaltstart des Basic - Interpreters V.3.2 (via Modul 1), dabei werden auch die Funktionstasten geladen.

|  |  |
| --- | --- |
| **C aaaa bbbb cccc** | **Compare** |

Vergleicht die Speicherbereiche ab aaaa und bbbb mit der Länge cccc. Ungleichheiten werden mit Adress- und Datenbyteangabe durch Taste "CR" fortlaufend angezeigt. Jede andere Taste bewirkt vorzeitigen Abbruch.

|  |  |
| --- | --- |
| **D aaaa (bbbb)** | **Dump** |

Der Speicherbereich von Adresse aaaa bis bbbb wird angezeigt. Angezeigt wird: Adresse, 8 Datenbytes, ASCII-Interpretation.

Datenbytes können beliebig überschrieben werden, dazu wird der Kursor mit den Kursortasten in die entsprechende Zelle positioniert und die erforderlichen Datenbytes einfach überschrieben. Die geänderten Bytes werden sofort übernommen.

Mit Taste HOME=Strg+A (01H) wird zwischen Hex- und ASCII-Mode umgeschaltet.

|  |  |
| --- | --- |
| **J aaaa** | **Jump** |

Sprung in Anwenderprogramm ab Adresse aaaa.

|  |  |
| --- | --- |
| **l CR**  **l "Filename" CR l AADR "Filename" CR** | **„kleines L“**  **HD-Lesen (für DVHD Files)** |

Liest DVHD Files nach Adresse AADR. Wird keine AADR angegeben, wird auf die originale Adresse geladen.

Der Dateiname muss 100% stimmen und in Hochkommas gesetzt werden, Groß/Kleinschreibung und auch Leerzeichen beachten !

Syntaxbeispiele:

"l zeigt alle DVHD Files (sortiert wie in DVHD) an.

"l "FILE" liest FILE auf originale Ladeadresse "l 2000 "FILE" liest FILE auf Adresse 2000H

Randbedingungen für die HD-Nutzung: siehe Hinweise für **HD-Nutzung**.

|  |  |
| --- | --- |
| **N aaaa bbbb** | **CRC - Prüfsumme** |

Prüfsummenberechnung über Speicher von Adresse aaaa bis bbbb entsprechend dem CRC (SDLC)-Standard.

|  |  |
| --- | --- |
| **P aaaa bbbb cc** | **Pattern** |

Speicherbereich von aaaa bis bbbb mit Datenbyte cc füllen.

|  |  |
| --- | --- |
| **p** | **Clear RAM** |

Der Speicher wird von 18D6h bis FEFFh mit FFh beschrieben = gelöscht. Die letzten 256Byte von FF00h…FFFFh werden NICHT gelöscht, um dort befindliche Hard- bzw. Softwareerweiterungen z.B. DVU/DVHD nicht zu überschreiben.

|  |  |
| --- | --- |
| **r** | **Basic Warmstart** |

Das Kommando löst einen Warmstart vom Grafik-Sound-Basic V3.2 über Modul 1 aus, dabei werden auch die Funktionstasten geladen.

|  |  |
| --- | --- |
| **s CR**  **s:"Filename" P CR**  **s AADR EADR SADR "Filename" P CR** | **HD-Schreiben (DVHD Files)** |

Schreibt ein DVHD File von AADR..EADR und trägt SADR und P=Dateityp ein.

Syntaxbeispiele:

s CR alle HD-Files anzeigen (passiert auch bei falscher Syntax)

s 0 FFF 0 "Monitor" P CR schreibt DVHD File "Monitor "

s 6000 7038 0000 "GRAFIKSHOW V10" B schreibt BasicFile "GRAFIKSHOW V10"

Die Angabe eines FILENAMENS in Hochkomma ist zwingend und wird ggf. mit Leerzeichen aufgefüllt (Filename in DVHD immer 16Bytes). Achtung Groß/Kleinschreibung beachten !

Files können bis FFFFH geladen werden. Bei SAVE ist das leider nicht möglich. Der HD-Puffer vom GIDE liegt normal auf FD00h..FEFFH und zerstört ggf. den zu sichernden bisherigen Inhalt!. Im Einzelfall kann das durch **Verlagerung des Pufferbereiches** geändert werden.

Randbedingungen für die HD-Nutzung: siehe **Hinweise für HD-Nutzung**

**Neu: Beim Sichern erfolgt ein Test auf Vorhandensein der Datei. Ist eine gleichnamige Datei bereits vorhanden, besteht die Auswahl zwischen Überschreiben oder Abbruch.**

|  |  |
| --- | --- |
| **T aaaa bbbb cccc** | **Transfer** |

Speicherbereich verschieben von Adresse aaaa nach bbbb. cccc ist die Länge. Überschneidung der Bereiche ist zulässig.

|  |  |
| --- | --- |
| **U aaaa "FILENAME.Z80"** | **USB File laden** |

Laden eines FILES von USB in den Speicher. Es wird NICHT unbedingt das HEADERSAVEDateiformat erwartet, sondern es können alle FILES geladen werden !!

Die Namensangabe MUSS mit " beginnen UND 100% übereinstimmen (!), sonst wird das File nicht gefunden und die Meldung "FILE ERROR" ausgegeben. Ohne Namensangabe wird kein FILE geladen, sondern das DIR vom USB angezeigt.

Bei Angabe der Ziel-Ladeadresse aaaa wird das FILE auf die Adresse aaaa geladen. Ohne Angabe der Zielladeadresse wird auf die ggf. im HEADERSAVE-Header enthaltene originale Adresse des Programms geladen, andernfalls auf Adresse 0H!

Hinweis: Bei echten HEADERSAVE-Files wird beim Laden der HeaderInfoText mit ausgegeben. Bei allen anderen Files wird das unterdrückt.

Ein Wechsel in Unterverzeichnisse des Sticks ist nicht möglich.

|  |  |
| --- | --- |
| **u AADR EADR SADR "FILENAME.Z80" P Fileinfo u-"FILENAME.Z80"** | **USB File speichern USB File löschen** |

Schreiben eines FILES im Headersave-Format auf USB. Der Filename muss zwingend im 8+3 DOS Dateiformat in Hochkomma stehen. Der Dateityp (P) ist frei wählbar und als FileInfo sind 16Bytes möglich.

Wird nur u oder eine ungültige Syntax angegeben, werden die Files von USB angezeigt.

Eine spezielle Form des Schreibens ist die neue Möglichkeit **löschen.** Wichtig ist exakte Schreibweise ohne Leerzeichen nach dem „u“. Stimmt der Dateiname nicht, passiert außer der Meldung "CF Error" nichts. Ansonsten wird ohne Rückfrage gelöscht!

**Neu: Beim Sichern erfolgt ein Test auf Vorhandensein der Datei. Ist eine gleichnamige Datei bereits vorhanden, besteht die Auswahl zwischen Überschreiben oder Abbruch.**

|  |  |
| --- | --- |
| **w aa** | **Port lesen** |

Der Portwert an IO-Adresse aa wird eingelesen und angezeigt.

|  |  |
| --- | --- |
| **W aa bb** | **Port schreiben** |

Der Portwert bb wird an IO-Adresse aa ausgegeben.

|  |  |
| --- | --- |
| **X** | **Programmpaket X starten** |

Sprung in weiteres Betriebssystem. Vorgesehen ist Programmpaket X ab E000h. Näheres siehe Bauanleitung Modul 1.

## Nachladbare Kommandos

Mit Monitorversion 10.x wurden einige Stimmen laut, die entfallene Kommandos vermissten. Dem wurde mit einer großen Neuerung Abhilfe geschaffen:

***Nicht im Monitor enthaltene („eingebaute“) Kommandos können bei Bedarf als "nachladbare Kommandos" realisiert werden!***

Das gilt sowohl für die „alten“ Kommandos als auch eventuelle neue. Für selten benötigte Kommandos ist dies eine echte Alternative, denn es wird kein wertvoller Speicherplatz im SystemEPROM benötigt.

Prinzip:

Ein nachladbares Kommando wird wie ein eingebautes aufgerufen, also mit gleicher Syntax und ggf. auch mit Argument. Die Routine des Kommandobuchstabens liegt aber nicht im EPROM sondern als Datei auf einem externen Speichermedium.

Findet die Suchroutine das Kommando nicht im EPROM, dann wird das eingestellte Speichermedium auf die zum Kommando passende Datei durchsucht und von dort nachgeladen. Die Dateinamen der „Kommandoprogramme“ müssen für diesen Zweck einer bestimmten Syntax genügen. Beispiel:

Find-Kommando für den AC1:

**MO11A\_46.Z80**

**| | ^^--------- ASCII-Code des Kommandobuchstabens**

**| +------------ Gerätekenner (A=AC1, L=LLC2)**

**+-------------- Monitorversion**

Damit wird garantiert, dass das nachladbare Kommando sowohl zum Gerät (AC1/LLC2) als auch zur Monitorversion passt. Durch die Angabe des ASCII-Codes können (trotz der Einschränkung bei USB) auch Kommandos in Kleinbuchstaben verwendet werden. Auch die normalerweise für Dateinamen „verbotenen Zeichen“ \ / | ä ö ü . Ö Ä Ü [ ] sind als Kommando“buchstaben“ möglich!

Das so nachgeladene Kommando befindet sich nun im RAM an geeigneter Stelle (abhängig von Art/Zweck des Kommandos) und kann nun auch ohne Nachladen verwendet werden, solange der Bereich nicht durch ein anderes Programm überschrieben/gelöscht wird.

Als Ablagemedium für die Dateien der Nachladekommandos ist sowohl USB (Voreinstellung) als auch HD möglich.

Hinweis:

Wer selbst nachladbare Programme entwickelt, sollte diese als verschiebliche Programme programmieren und genau wissen in welchem Programmbereich das Kommando läuft. Es ist sogar möglich, ganze Programme gleich als nachladbares Kommando zu speichern und damit automatisch bei Bedarf zu laden und zu starten.

Kleiner Nachteil:

Das Nachladen per USB ist jedoch etwas langsam. Wird ein gar nicht vorhandenes Kommando eingegeben, so dauert es bei gut gefülltem USB-Stick schon eine Weile, bis die „? #“-Meldung erscheint. Bei HD-Verwendung ist ein Geschwindigkeitsverlust kaum zu verzeichnen.

# Beschreibung von Systembestandteilen

## Startroutinen des Monitors

Diese werden anhand eines Codemusters im Adressbereich 0010h bis FFFFh gesucht. Jede Routine hat die Form : **00 09 xx 0D ... Programm ... C9 .**

xx ist der ASCII-Code des Start-Kennbuchstaben. Eigene Programme brauchen nur in diesen Rahmen gepackt zu werden und können über Kennbuchstaben (Zeichen/Ziffer) gestartet werden.

## Dateitypen

Resultierend aus dem historischem Turbo – Tape werden 4 Filetypen unterschieden :

P lauffähiges Maschinenprogramm B Basicprogramm D Maschinendatei F Datenfeld (Basic)

Die Dateitypen B und F des Basicinterpreter V.3.2 oder 4 können nunmehr auch vom Monitor aus geladen werden. Dabei ist das Laden eines F-Types nur im USB-Modus sinnvoll. Details zum Dateityp F siehe unter **Erweiterung der Funktionalitäten von GS-BASIC**.

## Dateinamen

Die maximale Länge der Dateinamen ist vom Modus abhängig: 8 bzw. 16 Zeichen. Längere Namen werden in beiden Modi beim SAVEn abgeschnitten!

* USB max. 8 Zeichen, keine Unterscheidung Klein-/Großschreibung Es wird immer mit Header als <dateiname>.Z80 gesichert.

Als Headerinfo wird der Dateiname geschrieben, der bis zu 16 Zeichen lang sein kann. Ein abgeschnittener Name bleibt in der Headerinfo erhalten!

* HD max. 16 Zeichen

Wird ein Dateiname abgeschnitten und ist der so entstandene Name schon vorhanden, so erfolgt beim SAVEn die Rückfrage "OverWrite?(J)".

Wird beim Laden ein längerer Name als 8 bzw. 16 Zeichen angegeben, so werden die überzähligen Zeichen des Namens ignoriert und die Datei geladen, deren Name bis zur zulässigen Maximallänge übereinstimmt (sofern vorhanden).

## Ein-/Ausgabe

Die Verbindung des Computers mit der Umwelt erfolgt über die Ein/Ausgabe. Die Ein/AusgabeKanäle werden über das I/O-Byte (Adresse 1821h) adressiert. Bit 0-3 sind für die Eingabe, bit 4-7 für Ausgabe zuständig. Es stehen somit jeweils 4 Möglichkeiten zur Verfügung:

|  |  |
| --- | --- |
| **Eingabe** | **Ausgabe** |
| bit 0 Tastatur bit 1 V 24 (Rs 232c) bit 2 Reserve bit 3 User | bit 4 Bildschirm bit 5 V 24 (RS 232 c) bit 6 Reserve bit 7 User |

Nach Einschalten des Computers wird das Ein/Ausgabebyte auf 11h gesetzt, somit Tastatureingabe und Bildschirmausgabe.

Setzen der Ein/Ausgabe

* Taste Control + Taste X schaltet zusätzlich Ausgabe über V 24 ein.
* Taste Control + Taste Y schaltet Ein/Ausgabe auf Standard (11h).
* Taste BREAK/NMI setzt Ein/Ausgabe auf Standard (11h).
* RESET-Taste verändert die Ein/Ausgabe nicht !

Die Ausgabe verarbeitet Datenbytes 00…7Fh, womit der ASCII-Code (0-7F) verarbeitet wird. Bit 7 wird dabei auf 0 gesetzt. Die Ausgabe lässt sich auf 00…FFh umstellen, indem bit 7 des V 24Kontrollregisters gesetzt wird.

|  |  |  |
| --- | --- | --- |
| Adressen: |  |  |
| Eingabekanal 08h | Ausgabekanal | 10h |
| V.24-Kontrollregister 1820h | I / O-Byte | 1821h |
| USER-Eingabe 18F0h | USER-Ausgabe | 18F3h |
| V 24 - Eingabe UP 04ECh | V 24 - Ausgabe UP | 0DF9h |

## V.24 - Schnittstelle

Die standardisierte Schnittstelle ermöglicht den Anschluss von Drucker, Modem u.a. Peripheriegeräten und die Kopplung mit weiteren Computern. Die Schnittstelle arbeitet mit TTLPegel, zum Anschluss von Peripherie wird in der Regel eine Pegelanpassung (-12V, +12V) benötigt, siehe Schaltbild.

V.24 (RS 232c) ist eine serielle Schnittstelle. Zur Kopplung können wahlweise 3 Leitungen (3 Draht Handshake) oder 5 Leitungen (X Draht-Handshake) eingesetzt werden. Letzteres ist zur Kopplung mit langsamer Peripherie (Drucker) notwendig. Zur Kopplung von 2 Computern genügt der schnelle 3-Draht Handshake.

Über das Kontrollregister 1820h werden die Standardwerte der Schnittstelle eingestellt, siehe Tabelle.

### V.24 - Kontrollregister

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Baudrate** | **bit 2** | | **bit 1** | | **bit 0** | | **dez.** | |
| 300 | 1 | | 1 | | 1 | | 7 | |
| 600 | 1 | | 1 | | 0 | | 6 | |
| 1200 | 1 | | 0 | | 1 | | 5 | |
| 2400 | 1 | | 0 | | 0 | | 4 | |
| 4800 | 0 | | 1 | | 1 | | 3 | |
| 9600 | 0 | | 1 | | 0 | | 2 | |
| 19200 | 0 | | 0 | | 1 | | 1 | |
| 14400 | 0 | | 0 | | 0 | | 0 | |
| **Stop-Bits (Ausgabe)** | | | | **bit3** | | | **dez** | |
| 1 | | | | 0 | | | 0 | |
| 2 | | | | 1 | | | 8 | |
| **Parität (Ausgabe) :** | | **bit 5** | | | | **bit 4** | | **dez.** |
| 8 Datenbit, ohne P. | | 0 | | | | 0 | | 0 |
| 7 Datenbit + P. ungerade | | 0 | | | | 1 | | 16 |
| 7 Datenbit + P. gerade | | 1 | | | | 0 | | 32 |
| **Handshake (Ausgabe) :** | | | | | | **bit 6** | | **dez.** |
| 3 - Draht | | | | | | 1 | | 64 |
| X - Draht | | | | | | 0 | | 0 |
| **Ausgabe :** | | | | | | **bit 7** | | **dez.** |
| 8 Datenbit | | | | | | 1 | | 128 |
| 7 Datenbit bit 0-6 + bit 7-0 | | | | | | 0 | | 0 |

Der Eingabekanal arbeitet ohne Paritätskennung, eine Einstellung der Stop Bits und Handshake ist nicht notwendig.

Nach Einschalten des Computers sind folgende Werte voreingestellt : 1820h=42h

9600 Baud 1 Stop-Bit ohne Parität

3-Draht-Handshake

8 Datenbit (Ausgabe bit 7 auf 0 gesetzt)

Somit kann übliche Peripherie (Drucker) unmittelbar betrieben werden, z.B. K6304 und Schreibmaschine S 6006. Druckaufräge funktionieren damit auch im JKCEMU!

### Hardware

Die serielle Schnittstelle verwendet 4 Kanäle (A0-A3) der PIO 2 (LLC2 : D 47). Bei AC1 ist PIO 2 zu erweitern, Adressierung über IOSEL 2. Die PIO hat gegenüber einer SIO den Vorteil des geringeren Aufwandes. Die Baudraten werden softwaremäßig getaktet und sehr genau eingehalten(+-0%). Wie bereits ab Monitor 10.3 kommt eine Baudraten-Tabelle zum Einsatz, mit der bei Bedarf für ein konkretes System das Timing optimiert werden kann.

Die Leitungen werden auf eine 5-polige Diodenbuchse gelegt. 2 Computer (AC1/LLC2) können über ein Überspielkabel gekoppelt werden. Der Anschluss an PIO2 ist wie folgt definiert:

|  |  |  |  |
| --- | --- | --- | --- |
| **PIO2** | **Signal** | **Name** | **Buchse** |
| A0 | Eingang Daten | RxD | 1 |
| A1 | Ausgang Daten | TxD | 3 |
| A2 | Eingang empfangsbereit | CTS | 4 |
| A3 | Ausgang sendebereit | DTR | 5 |

Die Kopplung mit weiteren Computern (z.B. Commodore 64, +4, KC 85) erfolgt entsprechend der Angaben in den Handbüchern. Es können problemlos Daten ausgetauscht werden (z.B. Basicprogramme).

## Ablauf Betriebssystem Monitor 11.0

Nach dem Einschalten des Computers :

* Interrupt wird gesperrt,
* SP initialisiert,
* Initialisierung der Peripherie (PIO/CTC),
* Warmstartvektor abfragen,
* Sprungtabelle für RST, NMI in den RAM übertragen,
* Warmstartvektor, Ein/Ausgabe-Byte, V 24-Kontrollbyte in den RAM übertragen,
* Voreinstellungen für HD-Puffer und GS-BASIC-Methode übertragen,
* Anwender-RAM löschen, Interruptregister wird gesetzt,  Farb-RAM initialisieren, Bildschirm löschen und Überschrift,  Tastaturschleife/Warten auf Eingabe Monitor.

## Weitere Betriebssysteme

Die Anwendung weiterer Betriebssysteme ist im Monitor vorbereitet. Vorgesehen sind BasicInterpreter V.3.2/4 und Programmpaket X sowie CP/M. Die weiteren Betriebssysteme werden fest auf Eprom installiert; entsprechend Bauanleitung Modul 1.

Welchem Betriebssystem die höchste Priorität zusteht, kann der Anwender frei festlegen. Um in das jeweilige Betriebssystem zu gelangen, gibt es verschiedene Möglichkeiten :

Starten des Basic-Interpreters V.3.2

# b (CR)

Rückkehr in den Monitor mit BYE.

Starten des Programmpaket X

# X (CR) oder

Taste X drücken und RESET betätigen oder

Taste X während des Einschaltens des Computers drücken.

Starten von CP/M:

Je nach benutztem Rechnertyp, Speichermedium und CP/M-Version, z.B.:

AC1: Laden von HRDOS, HRCPM nach E600h und Start ab E600h

LLC2: Laden von CPL9 nach C800h und Start ab DE00h

Verlassen von CP/M durch RESET.

# Erweiterung der Funktionalitäten von GS-BASIC

GS-BASIC selbst ist unverändert. Der neue Monitor erweitert jedoch die Möglichkeiten erheblich: Die ursprünglich für die Kassettenarbeit geschaffenen Anweisungen CLOAD und CSAVE wirken nun direkt auf das entsprechende Speichermedium HD bzw. USB!

Es ist damit nicht mehr nötig, die Ende-Adresse eines BASIC-Programms aus dem Inhalt der BASIC-Arbeitszelle 60D2h zu bestimmen und das BASIC-Programm nach Verlassen von BASIC separat zu sichern!

Per USB wird immer mit ".Z80"-Files gearbeitet. Es gelten die unter **Dateinamen** beschriebenen Regeln. Einzig die Dateierweiterung z80 muss beim Laden/Sichern hier in GS-BASIC nicht mit angegeben werden!

## Wahl des Modus/Speichermediums

* In GS-Basic kann entweder mit HD oder USB als Speichermedium gearbeitet werden.  Die Umschaltung erfolgt aus BASIC heraus so:

POKE 6241,0 Einstellung auf USB

POKE 6241,32 Einstellung auf HD (Voreinstellung)

* Wird USB als Speichermedium bevorzugt, so ist die Voreinstellung in 0238H zu ändern.

## Inhaltsverzeichnis anzeigen

* Die Anweisung **CLOAD“ “** (Leerzeichen als Dateiname) zeigt das Inhaltsverzeichnis an (HD bzw. USB je nach aktuellem Modus).
* Tipp: Funktionstaste **Strg+V** => spart die Tipparbeit :-)

## Datei Laden

* Mit **CLOAD „Dateiname**“ wird vom aktuell eingestellten Medium eine z80-Datei geladen.
* Der Dateiname muss 100% stimmen, sonst Ladeabbruch mit "NAME FILE Error". Die ehemalige Funktionalität CLOAD“X (lädt das nächste mit X beginnende Programm) wird nicht mehr unterstützt.
* Sowohl im HD- als auch USB-Modus ist ein "MERGE" (Programme anhängen) realisiert: CLOAD "DATEI1" 1. Datei laden

CLOAD "DATEI2" 2. Datei laden (muss höhere Zeilennummern haben!) "0 REM" eingeben das ist wichtig!

Das bedeutet, dass vor einem "normalen" CLOAD der Benutzer immer erst (sicherheitshalber) ein "NEW" anzuwenden hat.

* CLOAD lädt auch ausführbare Programme (Typ "P") an ihre korrekte Adresse. Damit lassen sich z.B. MC-Routinen für Basic-Programme nachladen. Innerhalb eines Programms angewandt, wird das Basic-Programm nach erfolgreichem Laden und Ausschrift normal fortgesetzt. Schlug das Laden fehl, wird abgebrochen.
* Wird versucht, einen anderen Typ als "B" oder "P" zu laden, so erfolgt Abbruch mit "TYPE FILE Error"

## Datei sichern

* Die Anweisung **CSAVE“Dateiname“** sichert das im Speicher befindliche BASIC-Programm auf das aktuell eingestellte Speichermedium.
* Existiert der gewählte Dateiname schon, so kann entweder überschrieben (Overwrite = J) oder abgebrochen und ein CSAVE mit neuem Dateinamen begonnen werden.

## Felder

* Mit **CSAVE\*“dateiname“** können wie im kassettenbasierten Original BASIC-Felder gesichert und mit **CLOAD\*“Dateiname“** geladen werden.
* Das Sichern und Laden von Feldern beinhaltet (ebenfalls wie im Original!) alle Felder des Basicprogramms. Es funktioniert nur dann, wenn mindestens ein Stringfeld vorhanden ist. Rein numerische Felder können nicht gesichert werden.
* Wird ein Feld mit geladen, so wird das aktuelle Programm gelöscht und das zum Zeitpunkt der Feldsicherung vorhandene Programm (nur dazu passt ja das gesicherte Feld) restauriert; es erfolgt also kein "MERGE"!
* **Einschränkung: NUR IM USB-MODUS VERWENDBAR!**

## Verwendung von CLOAD/CSAVE in BASIC-Programmen

Im Normalfall wird aus der Kommandoebene von BASIC geladen und gesichert. Es ist auch mit den neuen Funktionalitäten möglich, beide Kommandos in einem BASIC-Programm zu verwenden. Eine sinnvolle Anwendung von CLOAD besteht z.B. im automatischen Nachladen von MC-Routinen. Beispiel:

10 REM-CLOAD-MC 15 CLS 20 PRINT"LADE MC..."

25 **CLOAD"SETCLOCK":REM-MC!(AADR=1900)**

#### 30 PRINT"STARTE MC..." 35 CALL\*1900 40 PRINT:PRINT"FERTIG!"

Wird der im Programm angegebene Dateiname auf dem aktuell gewählten Speichermedium nicht gefunden, so erfolgt ein Programmabbruch mit entsprechender Fehlermeldung.

# Externes SAVE/LOAD

Eine der neuen Möglichkeiten des Monitor 11 ist die Verwendung der SAVE-LOAD-Routinen als Bausteine für eigene Programme. Ähnlich der festen Adressen des Sprungverteilers gibt es nunmehr fertige Routinen mit **festen Bezügen** für das Laden und Sichern von Dateien.

Aus eigenen Programmen können die SAVE-LOAD-Routinen unter folgenden Kriterien universell genutzt werden:

|  |  |
| --- | --- |
| 1. Monitorversion testen: | (0800h) => "MON11", nur dann ist Anwender-Save-Load möglich! |
| 2. Adressen angeben | beim Laden: Speicher-Anfang: => ARG1 185Bh beim Saven + Speicher-Ende: => ARG2 185Dh  + ggf. Startadresse: => ARG3 185Fh |
| 3. Methode wählen | => 18B3h setzen:  00h = USB laden 01h = USB sichern  20h = HD laden 21h = HD sichern |
| 4. Puffer befüllen | => ab 18B4h, siehe **Systemzellen** |
| 5. Aufruf der Methode | => CALL\*806 (separates INIT ist nicht nötig!) |

Hinweis:

Wird beim Sichern der Header-Info-Text nicht belegt, so erscheint in der Sicherung der aktuell vorhandene Pufferinhalt, also ggf. Müll! Gleiches gilt auch für eine nicht benutzte Startadresse.

Beispiel in BASIC:

|  |
| --- |
| 10 REM-EXTERNES.SICHERN.AUF.USB.BEISPIEL.MONITOR.SICHERN  15 CLS:PRINT"TESTE MONITOR-VERSION...";  20 A=DEEK(2048):B=DEEK(2050):IFA=28493ANDB=12593THEN30  25 PRINT"MONITOR UNGEEIGNET!":END  30 PRINT"OK: MO11."+CHR$(PEEK(2053))  35 PRINT"SICHERN MONITOR PER USB..."  40 DOKE6235,0: REM.ANFANGADRESSE.IN.185B(arg1)  45 DOKE6237,4095: REM.BIS.ENDEADR.IN.185D(arg2)  50 POKE6323,1: REM.KOMMANDO=SAVE-USB  55 D$="MONITOR!.Z80": REM.DATEINAME=8.3  60 POKE6324,34: REM.''  65 FORI=1TO12:POKE6324+I,ASC(MID$(D$,I,1)):NEXT  70 POKE6337,34: REM.''  75 POKE6338,32: REM.SPACE  80 POKE6339,80: REM.P=TYP  85 POKE6340,32: REM.SPACE  90 I$="TESTVERSION 11.0": REM.INFO=MAX.16ZEICHEN  95 FORI=1TO16:POKE6340+I,ASC(MID$(I$,I,1)):NEXT  100 CALL\*806: REM.EX\_SA\_LO  105 PRINT:PRINT"KONTROLL-LADEN..."  110 POKE6241,0: REM.CLOAD=USB 115 CLOADD$ |

# Hinweise für HD-Nutzung

## GIDE

* Das GIDE muss auf IO-Adresse 80h eingestellt sein.
* Auf ein als Slave angeschlossenes 2. Medium kann nur per DVHD3 oder mit speziellen Programmen zugegriffen werden, nicht mit dem Monitor!

## Einstellung HD-Typ

Die Geometrie-Daten der HD sind fest eingestellt (Systemadressen siehe Tabelle), können aber durch Patchen auch ganz individuell angepasst werden.

AC1-Version: 256MB SSD: 984\*16\*32

|  |  |  |  |
| --- | --- | --- | --- |
| 0036 | Geometriedaten der verwendeten HD | Zylinder -1 | D7 03 |
| 0064 | Köpfe -1 | 0F |
| 0065 | Sektoren | 20 |

LLC2-Version: 128MB DOM: 500\*16\*32

|  |  |  |  |
| --- | --- | --- | --- |
| 0036 | Geometriedaten der verwendeten HD | Zylinder -1 | F3 01 |
| 0064 | Köpfe -1 | 0F |
| 0065 | Sektoren | 20 |

Entsprechend der Nutzung im Zusammenhang mit DVHD3 kann ggf. auch der Start-Zylinder (Beginn des DVHD-Bereichs) modifiziert werden: Adresse **0B90h** (16Bit), Standardwert: 0002.

## HD-Puffer

Der HD-Modus benötigt für den Datenaustausch einen Puffer von 200h Bytes. Bisher war dieser Bereich fest auf FE00h gelegt. Das passte in Einzelfällen nicht. Um eine Kollision mit anderen gleichzeitig genutzten Programmen zu vermeiden, kann die Lage des Puffers verändert werden.

Die neue Pufferadresse kann je nach Anforderungen völlig frei gewählt werden. Dafür sind im Monitor zwei Arbeitszellen zuständig. Die HD Routinen holen den Puffer immer aus 1862/63h. Im BasicMode werden stattdessen die Inhalte von 1864/65 verwendet.

|  |  |  |
| --- | --- | --- |
| **Adresse** | **Bedeutung** | **Voreinstellung (nach Reset)** |
| 1862/63H | Pufferanfang für MonitorMode | FD00h (…FEFFh) |
| 1864/65H | Pufferanfang für BasicMode | 3E00h (…3FFFh) |

In Basic kann der Puffer wie folgt gesetzt werden: POKE 6245,25 :REM setzt Puffer auf 1900

## Defragmentierung

Die Benutzung des HD-Modus erzeugt beim Überschreiben auf dem Datenträger „leere“ Stellen, da die alte Datei gelöscht und eine gleichen Namens neu angehängt wird. Bedingt durch das gewählte Datenformat und die Organisation auf dem Datenträger lassen sich (wie mit DVHD3) insgesamt maximal 112 Einträge verwalten.

Wird viel überschrieben, so empfiehlt sich deshalb die gelegentliche Anwendung der **„Kompaktieren“- Funktion von DVHD3**. Damit werden die leeren Stellen entfernt und es steht wieder mehr Speicherplatz zur Verfügung. Hinweis aus der Anleitung von DVHD3:

*„Achtung ! Kompaktieren funktioniert nur fehlerfrei, wenn alle Files fehlerfrei und direkt in den RAM ladbar sind !*

*Bei Lesefehlern kann unter Umständen die logische Diskstruktur zerstört werden !! Es wird dafür demnächst ein Update geben. Diese Routine erfordert etwas Geduld, grade zum Ende scheint es sich aufzuhängen, arbeitet aber noch. Das ist am Laufwerk am Flackern zu erkennen.“*

# Hinweise für USB-Nutzung

Der Monitor 11.0 benötigt (wie das USB-Verwaltungsprogramm DVU) für die USB-Funktionen ein VDIP1 oder VDIP2 auf der Basisadresse FCh (AC1) bzw. DCh (LLC2).

Es kann ein USB-Stick oder auch eine externe USB-Festplatte verwendet werden, welche am „BOMS“-Port („Bulk Only Mass Storage“) des VDIP anzustecken ist (beim VDIP2 der rechte Port).

Im Gegensatz zur HD müssen USB-Massenspeicher in der Regel nicht erst formatiert werden, FAT12, FAT16 oder FAT32 sind möglich.

Die Dateien sind „PC-kompatibel“. Somit ist ein bequemer Datenaustausch mit dem PC möglich.

Im Gegensatz zur HD (max. 112 Dateien möglich), gibt es für den USB-Zugriff theoretisch keine Beschränkung der Anzahl der gespeicherten Dateien.

Ist kein USB-Medium angeschlossen, so wird beim versuchten Zugriff „ND-ERROR“ gemeldet.

# Anhang

## Betriebssystem-Routinen und Voreinstellwerte

Beachte, dass es wie bereits im Monitor 10.3 einige Abweichungen in der Nutzung von Systemadressen (z.B. RST 20, RST 28,…) zur Ursprungsversion (Mon.8) gibt!

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Adr (hex)** | **Bedeutung** | | **Voreinstellungswerte (hex)** | | | |
| 0008 | Eingabekanal, normal Tastatur | | | | | |
| 0010 | Ausgabekanal, normal Bildschirm | | | | | |
| 0018 | Zeichenkettenausgabe bis einschließlich dem Byte, wo bit 7 gesetzt ist | | | | | |
| 0020 | USB (UP PUT Daten zum VDIP senden) | | | | | |
| 0028 | USB (UP GET Daten vom VDIP holen) | | | | | |
| 0030 | UP OUTHL | | | | | |
| 0033 | Ausgabe über Bildschirm | | | | | |
| 0036 | Geometriedaten der verwendeten HD, Beispiel:  128MB DOM: 500\*16\*32 | | Zylinder -1 | | | F3 01 |
| 0064 | Köpfe -1 | | | 0F |
| 0065 | Sektoren | | | 20 |
| 0066 | NMI | | | | | |
| 006E | EINSPRUNGPUNKT Rückkehr aus Programmpaket X | | | | | |
| 0071 | EINSPRUNGPUNKT Rückkehr aus Grafik-Sound-Basic 3.2 | | | | | |
| 009B | UP Beep | | | | | |
| 00A5 | UP "Eingabekanal" | | | | | |
| 011D | UP "MS 30" Zeitschleife 30 ms | | | | | |
| 0128 | UP "Inline", liest Zeile von Bildschirm ein (Adresse in 181Ah) | | | | | |
| 0146 | UP - Routine zu InHex | | | | | |
| 0172 | UP "InHex", wandelt eine 4-stellige Zahl im ASCII-Code in Hexzahl um. DE = Zeiger auf Zahl (Bildschirm), Ergebnis in HL (DE,HL) | | | | | |
| 0183 | UP "Outhex", gibt A als 2-stellige Hexzahl auf Schirm aus (A) | | | | | |
| 019A | UP "Out HL", gibt HL-Register als 4-stellige Hexzahl auf Schirm aus (HL) | | | | | |
| 01A5 | V 24 - Eingabe | | | | | |
| 01D3 | Standardmodus Nachlade-Funktion | | CALL 4DBH = Nachladen via USB CALL 9C4H = Nachladen via HD | | | |
| 01DE | Warteschleife Kommandozeicheneingabe | | | | | |
| 0220 | Tabelle RST, NMI, Hilfsregister, V 24, I/O | | | | | |
| 0238 | Voreinstellung Methode GSB SAVE/Load | | USB HD | | 00h  20h | |
| 0239 | Voreinstellung HighByte HD-Puffer | | FD (AC1) FE (LLC2) | | | |
| 0240 | Einsprung bei Programmfehler, Register werden gerettet in die RSA | | | | | |
| 0243 | Einsprung NMI, Register werden in RSA gerettet, I/O und PIO1/B werden auf Standard gesetzt | | | | | |
| 025D | PIO1-Init-Wert | | DAh für oberen Zeichensatz | | | |
| 0268 | UP "ERROR", schreibt Zeichenkette ERROR | | | | | |
| 0272 | UP "akustisches Signal" | | | | | |
| 0287 | UP "Ton", Reg. B=Tonlänge, C=Tonhöhe | | | | | |
| 0297 | UP "Taste", testet Tastaturstatus, kehrt bei gedrückter Taste nach 18 ms mit dem Kode zum Akku zurück, keine Taste-Rückkehr mit gesetztem Z-Flag und A=0, Code steht auch in 1822h | | | | | |
| 02F8 | Interrupt Tabelle PIO/CTC | | | | | |
| 04CA | Register mit Argumenten laden, aaaa=HL, bbbb=DE, cccc=BC | | | | | |
| 07E8 | V24OUT | | Sprungverteiler | | | |
| 07EB | MS 30 | |
| 07EE | OUTHEX | |  | | | |
| 07F1 | OUT HL | |
| 07F4 | INLINE | |
| 07F7 | INHEX | |
| 07FA | TASTE | |
| 07FD | GETCO | |
| 0800 | Monitorkennung | | "MO11.0" | | | |
| 0806 | EX\_SA\_LO Externer Aufruf Save/Load | | | | | |
| 0809 | USBINIT | | | | | |
| 080C | HDINIT | | | | | |
| 0A31 | UP TB aus (Ton und Interrupt wieder ein) | | | | | |
| 0A37 | UP TB ein (Ton und Interrupt sperren) | | | | | |
| 0A7B | UP Suche Autostartsequenz in DE !! AADR (DE) | | | | | |
| 0EB4 | UP „Joy“, Abfrage Joystick 1  keine Taste Z-Flag gesetzt und A=0, oben bit 0, unten bit 1, links bit 2, rechts bit 3, Feuerknopf bit 4 von Akku gesetzt | | | | | |
| 0AD8 | Tabelle Peripherie, CTC,PIO1,PIO2 | | | | | |
| 0B62 | FarbByte für Reset | | | | | |
| 0B63 | UP FarbInit in A Farbbyte | | | | | |
| 0DF9 | V 24 – Ausgabe | | | | | |
| 0B00 | Tabelle mit  Baudraten-Timings | 14400 | | 03 | | |
| 0B01 | 19200 | | 02 | | |
| 0B02 | 9600 | | 05 | | |
| 0B03 | 4800 | | 0B | | |
| 0B04 | 2400 | | 17 | | |
| 0B05 | 1200 | | 30 | | |
| 0B06 | 600 | | 61 | | |
| 0B07 | 300 | | C2 | | |
| 0B90 | Startspur für HD (Standard=2) | | | 02 00 | | |
| 0FC9 | Nur für AC1-Farb-BWS: Farbbyte | | | 0B | | |

grau markierte Zellen vom Anwender modifizierbar!

## Speicherbelegung (Memory Map)

|  |  |  |
| --- | --- | --- |
| 0000 | Betriebssystem | |
| 1000 | AC1: BWS LLC2: freier RAM | |
| 1800 | Zwischenspeicher Kursor | |
| 1802 | Sprungtabelle RST 8 | |
| 1805 | „ RST 10 H | |
| 1808 | „ RST 18 H | |
| 180B | „ RST 20 H | |
| 180E | „ RST 28 H | |
| 1811 | „ RST 30 H | |
| 1814 | „ RST 38 H | |
| 1817 | „ NMI | |
| 181A | Zwischenspeicher „Inline“ | |
| 181C | Warmstartcode | |
| 181F | Hilfsregister für FarbByte | |
| 1820 | Kommandobyte V 24 | |
| 1821 | Ein/Ausgabebyte | |
| 1822 | Tastencode der zuletzt gedrückten Taste | |
| 1823 | Hilfsregister Routine Q | |
| 1824 | Hilfsregister Repetierfunktion Tastatur | |
| 1825 | Hilfsregister Kursorpositionierung (Strg +N) | |
| 1856 | Stack | |
| 1857 | Hilfsregister UP „Inline“ | |
| 185B | Zwischenspeicher Argument 1 | |
| 185D | Zwischenspeicher Argument 2 | |
| 185F | Zwischenspeicher Argument 3 | |
| 1861 | Methode GSB SAVE/Load | 00h = USB  20h = HD |
| 1862 | HD-Puffer normal | 00 FD (AC1)  00 FE (LLC2) |
| 1864 | HD-Puffer GS-Basic | 00 3E |
| 1871 | RSA : IX, IY, PC, SP | |
| 187E | Speicher Save/Load Turbo-Tape/V24 | |
| 1885-  1894 | HeaderSave Header Teil 1 16Bytes | |
| 189E-  18AD | HeaderSave Header Teil 2 FILEINFO 16Bytes | |
| 18B3 | Methode für XX\_SAVE/LOAD | 00h = USB laden  01h = USB sichern  20h = HD laden  21h = HD sichern |
| 18B4 | Puffer für SAVE/LOAD-Arbeit | 1 Byte Hochkomma |
| 18B5 | Filename (8 . 3) |
| … | 1 Byte Hochkomma |
| … | 1 Byte Leerzeichen |
| … | 1 Byte Dateityp (P) |
| … | 1 Byte Leerzeichen |
| … | 16 Bytes Header-Info |
| 18D6 | freier RAM | |
| 18F0 | User Eingaberoutine | |
| 18F3 | User Ausgaberoutine | |
| 1900 | Anwender-Speicher bis FFFFh | |
| 2000 | Modul 1 : Basicinterpreter V.4 | |
| 6000 | Arbeitsspeicher Basicinterpreter | |
| 60F7 | Anfang Basic-RAM | |
| C000 | AC1: freier RAM LLC2: Bildspeicher 2k (Text), 16 k (Pixelgrafik) | |
| C800 | Modul 2 : 2 k Farb-RAM | |
| C000 | Modul 1 : ROM-Bank 1 und 2 | |
| E000 | Betriebssystem Programmpaket X | |
| FFFF | RAM – Ende | |

## Peripherie

|  |  |  |
| --- | --- | --- |
|  | **AC 1** | **LLC2** |
| CTC-Kanal 0-3 | 00h…03h | F8h…FBh |
| PIO 1 | 04h…07h | E8h…EBh |
| PIO 2 | 08h…0Bh | E4h…E7h |
| Modul 1 : Konfigurationsregister | 14h | ECh |
| Modul 3 : RAM-Disk/Bank | 15h | EDh |
| HiRes-Logik | - | EEh |
| Modul 2 : Mehrfarbenregister | 16h | EFh |
| USB | FCh…FFh | DCh…DFh |
| GIDE-Basisadresse | 80h | 80h |
| RCT-Basisadresse | 85h | 85h |
| ROM abschalten | - | E0h |

## Weitere Hardwarebezüge

### Belegung PIO 1/B

AC1: D17, LLC2: D44 bei Originaltastatur:

|  |  |  |
| --- | --- | --- |
| **Kanal B** | **AC 1** | **LLC2** |
| B0 | Ton-Ausgang | TB-Interface-Ausgang (Aufnahme) |
| B1 | Joystick 1 gem. A | TB-Interface-Eingang (Wiedergabe) |
| B2 | Grafiktaste | Grafiktaste |
| B3 | Bildschirm-Mode | Bildschirm-Mode (Doppel-ZG) \*) |
| B4 | frei | Joystick 1 gem. A. |
| B5 | Start/Stopp-Schaltung | frei \*\*) |
| B6 | TB-Interface-Ausgang | Ton-Ausgang |
| B7 | TB-Interface-Eingang | Start/Stop-Schaltung |

\*) Monitor aktiviert bei PIO-Init den oberen Zeichensatz (bei Bedarf: Patch Zelle 25D)

\*\*) CPL8/CPL9: Umschaltung invers/Pseudografik, nur mit Hardwarerweiterung

### BREAK-Taste/NMI

Die entprellte NMI-Taste ist mit dem NMI-Eingang der CPU zu verbinden. Im Monitor dient diese Taste zum Programmabbruch mit Rücksprung in die Maschinensprache-Monitor-Ebene.

### Grafiktaste

Diese Taste wird an PIO1 Port B2 angeschlossen (siehe Anhang). Mit gedrückter Grafiktaste schreibt die Tastatur die Grafikzeichen ab dem Code 128.

Hinweis : Im Maschinensprache-Monitor werden Zeichen ab Code 128 nicht dargestellt. Diese Taste wird daher erst in Verbindung mit GRAFIK-BASIC-Interpreter V.3.2 sinnvoll. Auch wird ein Zeichengenerator-Eprom 2K erforderlich.

Einbau der Grafiktaste (nur AC 1/LLC2 mit Originaltastatur): Der 1-polige rastende Schalter kommt an PIO 1/B2 und Masse. Von PIO 1/B2 ein R=330 Ohm und LED VQA 13 nach 5P anschließen. Die LED leuchtet bei eingeschalteter Grafiktastatur.

Änderung im Monitor notwendig: (Monitor ist bereits so vorgesehen)

AC 1 : Adresse 02A9h Datenbyte FFh in 21h ändern

LLC2 : Adresse 02C7h Datenbyte FFh in 21h ändern

### Joystick

Es können 1 oder 2 Stück angeschlossen werden, siehe Schaltbild. Die Abfrage der Joysticks kann in Anwenderprogrammen über UP „Joy“ erfolgen, siehe UP-Verzeichnis.

|  |
| --- |
| Copyright : SCCH-Monitor 10/88 sind © 1987 by E. Ludwig. die Erweiterungen © 2015 R.Hänsel ralphhaensel@gmx.de  Alle Rechte vorbehalten. Programme und Anleitungen können zur persönlichen Verwendung kostenlos weitergegeben und vervielfältigt werden unter Urheberangaben. |

\*\*\*

Viel Spaß mit dem neuen Monitor 11 am AC 1 & LLC2 wünschen euch

Ralph Hänsel (Programmierung)

&

Rolf Weidlich (Test & Dokumentation)

Stand: 26.02.15