BIC A5105

zum Bildungscomputer BIC gibt es (fast) alles bei U. Zander http://www.sax.de/~zander/index2h.html

Hier gibt es deshalb nur den orig. System-Quellcode und ein paar weitere spezielle Sachen zum BIC A5105.

Ich hab die orig. Quellen (aus UDOS) ein wenig aufbereitet, so dass ein Rekompilieren unter CP/M möglich ist[1)]. Damit schneller kompiliert werden kann, ist ein CP/M-Emulator und eine make.cmd für Windows dabei 🙂

2021: reass. Quellcode für nachladbaren RAM-Disk-Treiber VDISK. Dieser Treiber erzeugt eine 123K-RAM-Disk im ungenutzten Bildspeicher des GDC U82720.

Systembeschreibung



Der Bildungscomputer robotron A 5105 sollte die KC-Reihe auf einem wesentlich höheren Niveau fortsetzen. Sein Haupteinsatzgebiet war, wie es der Name sagt, in der Informatik-Ausbildung der Polytechnischen Schulen und in der Berufsausbildung. Der BIC besteht aus drei Baugruppen (Grundgerät CGG, Diskettenspeichereinheit DSE, Monitor). Als Betriebssystem kommt das ROM-orientierte Basisbetriebssystem RBASIC oder SCPX 5105 zum Einsatz. SCPX 5105 ist vollständig zum SCP des PC 1715 kompatibel.

technische Daten

Merkmal	Beschreibung

Merkmal	Beschreibung
CPU	U880
ROM	dsemc 16 KByte (= k5651_4000.bin, DSE) scprom 8 KByte bicmc 32 KByte
RAM	64 KByte RAM Arbeitsspeicher 128 KByte RAM als Bildspeicher
Takt	3,75 MHz
Anzeige	Videocontroller U 82720 25 Zeilen x 40/80 Zeichen,16 Vordergrundfarben,8 Hintergrundfarben 320 x 200 Pixel mit 16 Farben 640 x 200 Pixel mit 4 aus 16 Farben
Tastatur	ASCII
Peripherie	Grundgerät mit Netzteil und 5,25„ Floppylaufwerk 12“-Monitor (grün)
Software	RBASIC SCPX 5105


Standard-Zeichensatz

Am verbreitetsten ist „BIC System Version 2.01 robotron MKD 1989“. Die Version 2.00 ist 2022 aufgetaucht. Sie unterscheidet sich nur in der Funktion BEEP von der Version 2.01 (Version 2.00 ueber Kanal C mit Low-Frequ.= 55h)

Zusammensetzung der ROMs

```
bicmc  = bic_N021.bin = BICSYS + BIC1       (0000-22FF,2300-7FFF)
scprom = bic_N120.bin = SCPROM1         (8000-9EFF)
dsemc  = dse32.bin    = DSE1 + DSE2 + SCPROM2
(4000-63FF,6420-77FF,7880-7FFF)
```

Der Hauptspeicher des BICs ist in 4 SLOTs eingeteilt, die jeweils 64 KBytes Speicher im Adressbereich von 0H bis FFFFH umfassen. Der Adressbereich von 64 KBytes wird in 4 PAGEs von je 16 KBytes eingeteilt. Zur Speicherumschaltung dient der E/A-Port mit der Adresse **A8H**.

	RBASIC Slot 0	Modul DSE Slot 1	SCPX Slot 2	Modul CCG Slot 3
FFFF	n.n.	frei f. Modul	RAM	frei f. Modul
C000	n.n. / scprom N120	frei f. Modul	RAM	frei f. Modul
8000	bicmc N021	dsemc D47#D46	RAM	frei f. Modul
4000	bicmc N021	frei f. Modul	RAM	frei f. Modul
0000				

Literatur

1. http://robotron.foerderverein-tsd.de/322.html (Entwicklung ausgewählter Produkt-Linien › Klein-, Prozeß- u. Mikrorechentechnik › Heimcomputer), dort
2. http://robotron.foerderverein-tsd.de/322/robotron322a.pdf (Heim-, Klein- und Bildungscomputer des VEB Kombinat Robotron, Autor K.-D. Weise; dort Abschnitt 4.6. ff)

Downloads

- Anleitung, .. s. U.Zander
- bic.zip orig. UDOS-Disketten mit Betriebssystem-Quellcode im UDOS-Format
- bic_src.zip Betriebssystem-Quellcode, angepasst auf CP/M und Z80ASM
- bic_vdisk.zip reass. Quellcode zum RAM-Floppy-Treiber VDISK LW M: 123 kByte, HUB/IV M. Krzikalla, U. Konzack, 14.04.89
- bicsys_v3.pdf Betriebssystem-Quellcode-Listings als PDF zusammengefasst (02/2022)

Bedienung

Sonstiges

http://waste.informatik.hu-berlin.de/diplom/robotron/studienarbeit/files/hardware/bic/bic.html

Ostern 2018: Bislang fehlte von DISK.006 der Quellcode von RBASIC. Ich habe es jetzt geschafft, die Diskettenimages direkt zu extrahieren! Das ganze war ein schwieriges Unterfangen. Irgendwie habe ich ein Diskettenbelegungsplan der einzelnen Sektoren und ihrer Zugehörigkeit zu den Dateien vermisst. Ich hatte keine Ahnung vom Aufbau des Filesystems unter UDOS. Es gibt nur wenige Tools

zum Bearbeiten von DMK-Dateien (erzeugt mit catweasel); das Einzige, was die großen DMK-Dateien lesbarer gemacht hat, war dmk2jv3.exe. Nur fehlte hier irgendwie die Verbindung zwischen den 128 Byte großen Häppchen der einzelnen Dateien. Daher habe ich zuerst versucht, anhand der vorhandenen extrahierten Daten die Belegung der Disk herauszufinden. Erste Baustelle: das Zeilenende ist unter UDOS nur 0Dh, nicht 0D0Ah. Zweites Ergebnis: irgendwie gibt es keinen erkennbare Struktur, die einzelnen Häppchen liegen of mit Sektorversatz 4, aber eben nicht immer, Zwischendurch gibt es immer wieder problematische Stellen, an denen es unvorhersehbar weiter geht. Ein neuer Versuch war daher, die DMK-Dateien selbst zu verarbeiten. Dabei bin ich über eine weitere Baustelle gestolpert: die Sektoren haben im DMK-Format unterschiedliche Länge, eine einfache Berechnung der Sektorposition ist nicht trivial. Aber ich habe gesehen, dass es nach den 128 Byte noch weitere Bytes gibt, die offensichtlich eine Art verkettete Liste bilden. So habe ich den Quellcode von dmk2jv3.exe analysiert und um zusätzliche 4 Byte pro Sektor ergänzt. Der Rest war fast trivial: Ein Perl-Programm zerlegt mir die von dmk2jv3_udos.exe erzeugten Diskettenabbilder in die einzelnen Dateien. Hilfreich dabei waren ein paar kurze Informationen im „Wissenspeicher Mikrorechentechnik“ zum Thema Diskettenbelegung unter UDOS. Wusste ich doch, schon mal irgendwo was zum Thema verkettete Dateien gesehen zu haben. Nun liegen also endlich die kompletten Quellen des BIC vor 🙂

Tim Mann's CatWeasel support tools: https://github.com/ezrec/cw2dmk, enthält dmk2jv3

Erweiterungen

Es gibt eine Neuentwicklung für den BIC A5105: - das KCNET-Modul https://www.robotrontechnik.de/index.htm?/html/eigenbau/k1520nw.htm, http://kc85.info/index.php/projekte-mainmenu/kcnet-mainmenu-130.html Das KCNET-Modul enthält einen BOOT-ROM, der wahlweise von EEPROM oder RBASIC/SCP bootet.

Alle ROMs des BIC orientieren sich an der MSX-Spezifikation für Erweiterungen. Auf einer xx00h-Adresse stehen Bytes:

```
DEFB "AB" ; expansion ROM header
DEFW initcode ; start of the init code, 0 if no initcode
DEFW callstat; pointer to CALL statement handler, 0 if no such handler
DEFW device; pointer to expansion device handler, 0 if no such handler
DEFW basic ; pointer to the start of a tokenized basicprogram, 0 if no
basicprogram
DEFS 6,0 ; room reserved for future extensions
```

(s. https://www.msx.org/forum/msx-talk/general-discussion/header-rom-cartridge und https://www.msx.org/wiki/CALL)

Das betrifft auch die internen ROMs DSE und SCPROM und ebenso das Prüfmodul TPROS.

[1)]
Allerdings fehlt ein spezieller Linker, der nur Symbole extrahiert. Unter Windows erfolgt das mit einem Perl-Programm

From:
https://hc-ddr.hucki.net/wiki/ - **Homecomputer DDR**

Permanent link:
https://hc-ddr.hucki.net/wiki/doku.php/homecomputer/bic?rev=1733404302

Last update: **2024/12/05 13:11**