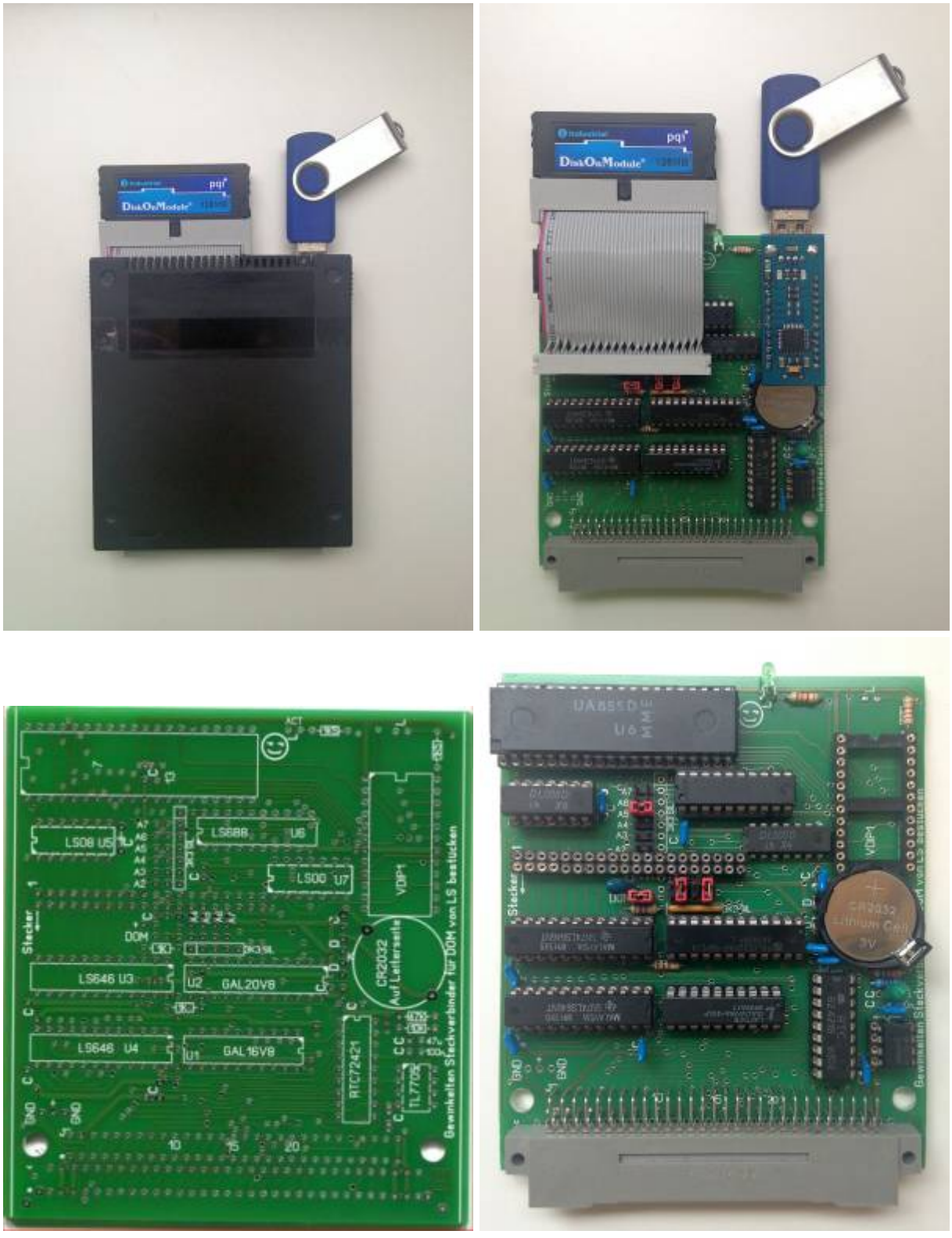
GIDE+USB-Modul

Wolfgang Harwardt hat Herbst 2013 eine kompakte Leiterplatte mit GIDE-Interface und USB (VDIP1 oder V2DIP) für den K1520-Bus entwickelt. Diese Karte lässt sich auch direkt am Z9001/KC87 nutzen!

http://buebchen.jimdo.com/selbst-gebaut-diy-homebrew-feito-por-mim/8-bit-selbstbau/gide-für-k1520/



v.l.n.r.voll bestücktes Modul, geöffnetes Modul, Leiterplatte, bestückte Leiterplatte

Schaltplan

- gide_usb.pdf von H. Poppe

Tipp: Für den 8-Bit-Comparator 74LS688 kann auch ein 74ALS520N zum Einsatz kommen. Dieser enthält bereits intern die benötigten Pull-Up-Widerstände, d.h. man braucht kein Widerstandsnetzwerk rechts neben den Jumpern. (Und ein 74ALS520N ist sogar billiger als ein 688)

Jumper:

- USB Jumper JP5 (Jumperleiste unterhalb der PIO) v.u.n.o. A2..A7. Es ist nur A5 gejumpert (Port DCh)
- GIDE Jumper JP2 (Jumperleiste Modulmitte) v.l.n.r. A4..A7, hi-Teil-PortAdr. Es ist A5 und A7 gejumpert (Port 50h)
- DOM Jumper J2 (Modulmitte, quer): gesteckt für Stromversorgung des DOM-Moduls

Forum: http://www.robotrontechnik.de/html/forum/thwb/showtopic.php?threadid=9958

Adressbereich	-	
I/O-Adressen	GIDE	54h-5Fh (empfohlener Port für Z9001)
	USB	DCh..DFh (empfohlener Port für Z9001)

GIDE

Die GIDE ist eine generische IDE-Schnittstelle für Z80-basierten Computer („Generic IDE“). Sie ermöglicht den Anschluss von bis zu zwei IDE-Geräten wie Festplatten oder CD-ROMs. Zusätzlich wird auch eine batteriegepufferte Echtzeituhr (RTC, Seiko-Epson-72421) unterstützt.
Die Schnittstelle wird über I/O-Befehle angesprochen.
Erdacht und realisiert wurde die GIDE von Tilmann Reh, 1995 und Herb Johnson (RTC und Software).



Start von GIDEC.COM, Menüpunkt 1 (Info)

Das Modul von Bübchen lief auf Anhieb, nachdem der GAL ST 20AS25HB1 gegen einen PALCE20V8H ausgetauscht wurde. Der Wellon-Brenner VP-280 macht leider Probleme beim ST-GAL. Obwohl der

Brennvorgang als erfolgreich gemeldet wird, ist nichts programmiert.

Zum Bild: Folgende Module sind gesteckt (von vorn nach hinten):

- Mega-Flash-ROM von Bübchen (mit Mega-Flash-Software, der 32K RAM ist deaktiviert)
- 64K dyn. RAM (Nachbau U. Zander, mit LEDs an der Seite)
- GIDE+USB von Bübchen mit 128MB DOM und VDIP1
- BIC-FDC-Platine (Nachbau U. Zander, mit Ports für Z9001)

Am GIDE-Bus ist ein DOM-Modul vom Pollin angeschlosssen. Der Flash-Speicher hat eine Kapazität von 128 MByte und kostet nur 1,50€ (2013).

Als Basis-Adresse ist bevorzugt **050H** zu nutzen (A5 und A7 jumpern). LLC2, AC1 und Z1013 nutzen den Adressbereich 84h..8Fh. Dieser ist leider am Z9001 nicht mehr frei.

Hardware

(nach http://www.gaby.de/gide/GIDE.txt)

Einführung

Die GIDE ist eine generische IDE-Schnittstelle für Z80-basierten Computer. Sie ermöglicht den Anschluss von bis zu zwei IDE-Geräten wie Festplatten oder CD-ROMs. Zusätzlich wird auch eine batteriegepufferte Echtzeituhr (RTC, Seiko-Epson-72421) unterstützt.
Die Schnittstelle wird über I/O-Befehle angesprochen.
Erdacht und realisiert wurde die GIDE von Tilmann Reh, 1995.

Schaltungsoptionen

Der RTC-Teil kann entfallen. Dies betrifft den RTC-Chip und die zugehörigen Bauteile zur Batteriegepufferung.

Schaltungsbeschreibung

Die Schaltung kann in mehrere Funktionsblöcke unterteilt werden. Der erste Block ist der Adressendecodierer, ein GAL 20V8 (IC2). Er vergleicht die Adressleitungen A4 bis A7 mit den Werten durch die Basisadresse via Jumper (J1), und dekodiert alle benötigten Auswahlsignale innerhalb des ausgewählten Adressbereich. Er puffert auch die Adressleitungen A0 bis A2 für den IDE-Port.

Der zweite Funktionsblock ist die IDE-Zugriffs-Zustandsmaschine. Er besteht aus einem GAL 16V8 (IC1) und zwei bidirektionalen 8-Bit-Registern (IC3, IC4), dieser Block wickelt den Datentransfer zwischen der Z80-CPU und dem IDE-Gerät ab. Hauptzweck dieses Funktionsblocks ist es, eine Schnittstelle zwischen den 16-Bit-Datenübertragungen vom IDE-Gerät und dem 8-Bit-Zugriff des Z80-Prozessors herzustellen.

Für Datenzugriffe wird eine Hälfte jedes 16-Bit-Datenwort in einem der Register gespeichert, bis der nächste 16-Bit-Zugriff gemacht werden kann. Da dies nur eine I/O-Adresse des Z80 belegt, kann die Datenübertragung mit Block I/O-Anweisungen (INIR/OTIR) durchgeführt werden. Die Zustandsmaschine GAL bietet auch Strobe-Signale für die IDE-Geräte und ein maskiertes /RD-Signal für den Zielcomputer.

Der dritte und letzte Funktionsblock ist der RTC-Block. Die Adressleitungen A8 bis A11 werden

verwendet, um die Register der RTC anzusprechen. Daher wird nur ein I/O-Port benötigt, der Zugriff muss aber mit OUT (C),r oder IN r,(C) erfolgen, und die RTC-Registeradresse muss dabei im B-Register stehen.

Programmdetails

Belegte I/O-Adressen (Alle Adressen sind hexadezimal):

x0..x3	werden nicht genutzt und sind frei für andere Erweiterungen
x4	reserviert für IDE expansion board
x5	RTC access
x6	IDE alternate status / digital output register
x7	IDE drive address register
x8	IDE data register
x9	IDE error/feature register
xA	IDE sector count register
xB	IDE sector number register
xC	IDE cylinder low register
xD	IDE cylinder high register
xE	IDE drive/head register
xF	IDE command/status register

„x“ steht für die Basisadresse (mit J1 ausgewählt).

Auf die sechzehn RTC Register wird mit 16-Bit-I/O-Anweisungen zugegriffen:

y0x5	seconds, units
y1x5	seconds, tens
y2x5	minutes, units
y3x5	minutes, tens
y4x5	hours, units
y5x5	hours, tens & AM/PM flag
y6x5	day, units
y7x5	day, tens
y8x5	month, units
y9x5	month, tens
yAx5	year, units
yBx5	year, tens
yCx5	day of week
yDx5	control register D (status/control)
yEx5	control register E (pulse output control)
yFx5	control register F (master control)

„x“ steht für die Jumper Basis-Adresse, und „y“ für einen beliebigen Wert, Jedes RTC-Register besteht aus nur 4 Bits, die oberen vier Daten-Bits werden einfach ignoriert.

Programmierbeispiele

Für eine einfache Implementierung werden Routinen sind sowohl für IDE-und RTC-Zugang benötigt: die Initialisierungsroutine, und Routinen zum Lesen resp. Schreiben von Daten.

Die Datei GIDEIDE.MAC enthält einen Beispiel-IDE-Treiber, GIDERTC.MAC einen Beispiel-Treiber für die RTC.

Für erste Tests mit einer frisch angeschlossenen IDE-Festplatte gibt es das Testprogramm GIDETEST, das einige grundlegende Test der Schnittstelle und der Festplatte ermöglicht. Das Programm ist in Turbo-Pascal 3.0 geschrieben und steht in Quell-und Objektcode zur Verfügung.

Hinweise

Kommt es beim Schreiben auf die Disk zu Problemen, hilft vielleicht der Tipp aus http://p112.sourceforge.net/index.php?gide. Ein Widerstand von 100 Ω .. 1 kΩ ist in die /IOWR-Leitung zwischen PAL und IDE-Anschluss einzubauen.

DOM-Modul

Bei Pollin gibt es für nur 1,50 € sogenannte DOM-Module. Das sind 128 MByte Flash-Speicher, die wie Festplatten mit IDE-Interface angesteuert werden (eine Art Vorläufer heutiger SSD-Festplatten). Original werden sie einfach auf die IDE-Wannenstecker des Motherboards gesteckt, daher haben die Module weibl. Buchsen anstelle der bei Festplatten üblichen Pfostenstecker. Pin 1 liegt links oben (von unten auf Modul gesehen).





Bestellnr. 94-701 790 DOM, IDE, 128MB, PQI

Für die Neuauflage der Leiterplatte müssen die beiden Anschlussreihen A und B nicht mehr vertauscht werden! Nachfolgendes gilt nur für die Leiterplattenvariante 1.

Auf der Leiterplatte ist Pin 1 ebenfalls links oben (!) (auf Bestückungsseite gesehen!) Die Leiterplatte ist eigentlich dafür gedacht, dass Modul rückseitig aufzustecken, s. Originalbilder bei Bübchen, aber dann kann man im Z9001 kein weiteres Modul dahinter stecken. Um hier Platz zu sparen, erfolgt ein bestückungsseitiger Anschluss des Moduls.

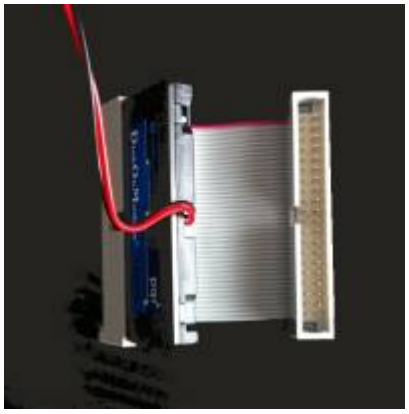
Das Modul kann aufgrund der „vertauschten“ Anschlussreihen nicht einfach auf die Leiterplatte aufgesteckt werden, sondern es wird ein spezielles Kabel benötigt, bei dem die beiden Anschlussreihen A und B vertauscht sind. Es wird ein Wannenstecker und ein Steckverbinder genutzt. Zum Vertauschen der Reihen A und B habe ich am Steckverbinder jeweils 2 nebeneinander liegende

Leitungen verdreht.






Zum Anschluss der DOM-Module am PC nutze ich einen **externen USB-Adapter**. Um hier das DOM-Modul zu verbinden, reicht ein Flachbandkabel mit 2 Wannensteckern auf derselben Seite. Die Reihen A und B des IDE-Anschlusses müssen nicht vertauscht werden.



Variante: Von **Rolf Weidlich** gibt es ein Programm DOM-Manager zum Bearbeiten der Module am PC. In der Doku zu seinem Programm wird der Anschluss des DOM-Moduls an einen USB-Adapter bebildert gezeigt.

http://www.ac1-info.de/galerie/weidlich_rolf/dom_manager.zip

http://buebchen.jimdo.com/app/download/8357963695/DOM-ManagerV1.zip

CP/M

Ein guter Startpunkt zum Thema GIDE ist

http://www.gaby.de/gide/ sowie
http://www.retrotechnology.com/herbs_stuff/gide.html

Empfehlenswert ist es, zuerst mit einem der Testprogramme GIDE*.COM zu beginnen, die direkt und ohne zusätzliche Treiber im normalen CP/M laufen. Hier ist GIDEC.COM oder das ältere Pascal-Programm GIDEtest09.zip zu empfehlen. Man sollte ein bisschen mit den Möglichkeiten herumzuspielen (Bilder s. http://www.mpm-kc85.de/html/GIDE.htm).

Zur Arbeit mit Festplatten unter CP/M muss i.W. ein passendes CP/M-BIOS erstellt werden. Quellcodebasis dafür ist GIDEprog.zip. Der GIDE-Treiber umfasst nur eine einfach zu übernehmende Schreib- und eine Leseroutine für das BIOS.

Von Kingstener kommt ein universell nachladbarer Treiber:
http://www.kingsteners.homepage.t-online.de/ (download →HP, Erweiterungen,GIDE).

Test

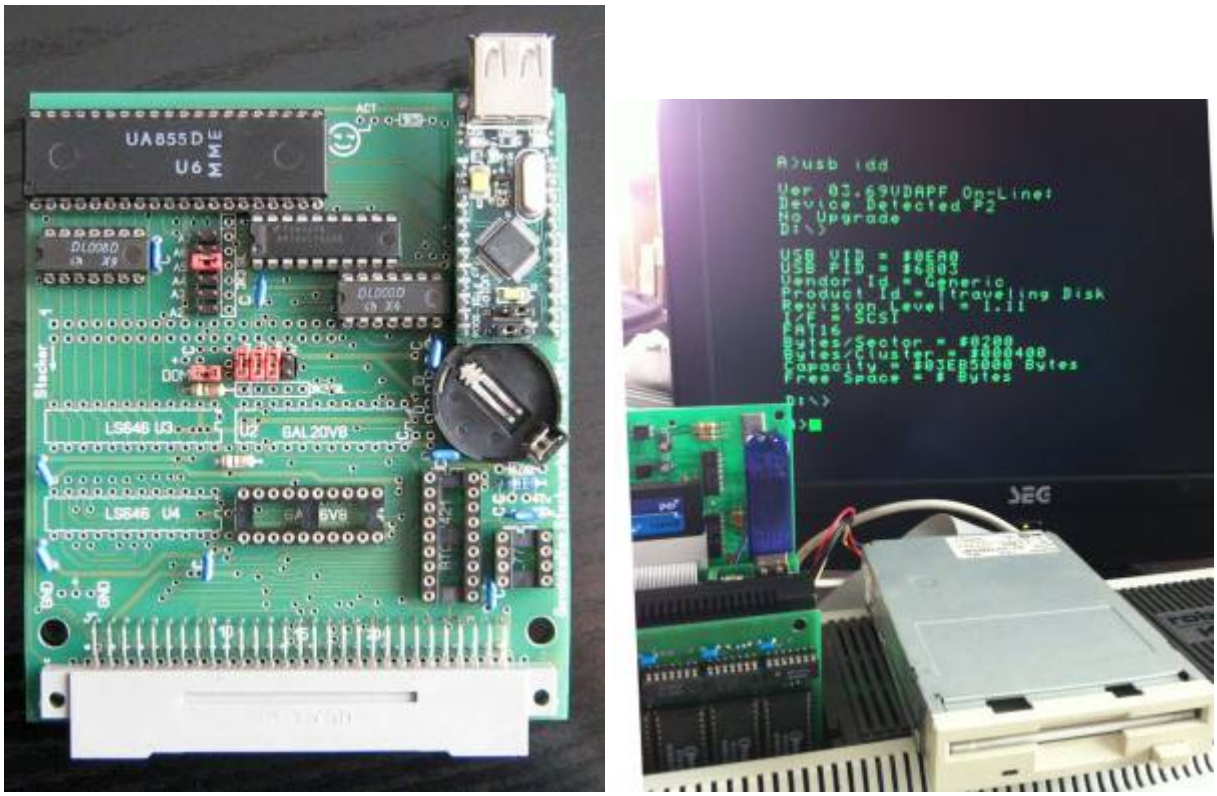
```
MINICPM     RAM-Disk formatieren J
B:
GIDEC
```

Unter CP/M:

```
GIDEC oder GIDE starten
p auswählen
Port angeben (hier 50), ECB-Bus-IDE Interface N
1 (read drives  ID data) - Anzeige der Disk-Daten
weitere Punkte nach Belieben
```

USB

Der **Anschluss eines USB-Sticks** an einen alten Heimcomputer ermöglicht einen einfachen Datenaustausch mit dem PC. Dank fertiger Module wie dem **VDIP1** oder **V2DIP** von Viculum/FTDI DevelopmentModules.htm, DS_VDIP1.pdf ist dieser Wunsch recht einfach zu realisieren.



li. Teilbestückung für USB, re. Start von USB.COM unter CP/M, angesteckt ist ein 64 MByte-USB 1.1-Stick

Basis dieser Entwicklung ist der USB-Anschluss von KC85 Labor susowa. Mario Leubner hat die Software entwickelt.

Hardware

Das VDIP-Modul wir an einer PIO angeschlossen. Das erlaubt die Nutzung des parallelen Datenmodus.

Mario Leubner schreibt dazu (http://susowa.homeftp.net/index.php/projekte-mainmenu/usb-mainmenu-131/72-usb-stick-am-kc.html):

Zur Nutzung der Parallelschnittstelle war zunächst klar, dass der Anschluss am M001 erfolgen wird. Hier stehen zwei PIO-Ports zur Verfügung, und Kanal A kann auch bidirektional arbeiten – also Daten empfangen und senden. Kanal B muss dazu im Bitbetrieb arbeiten und kann so für die Bedienung der Statussignale herangezogen werden.

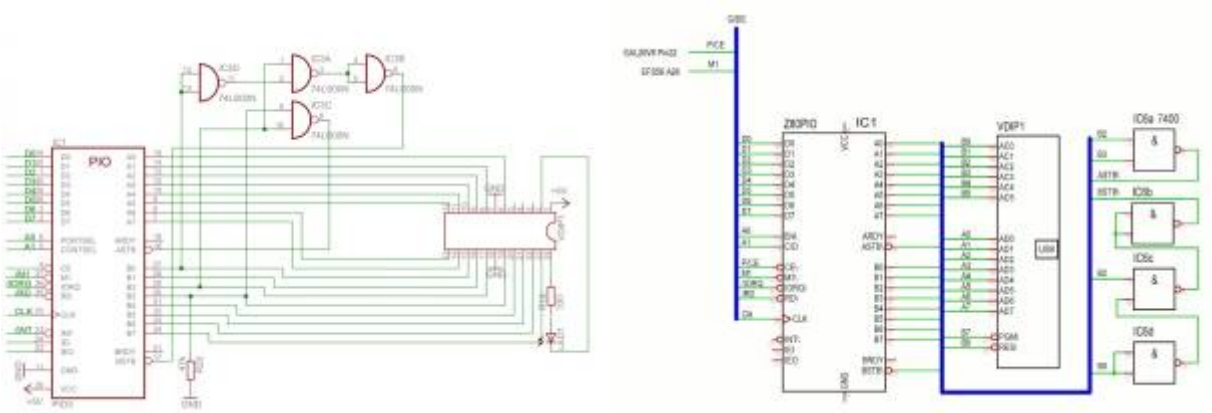
Zunächst galt es, die Signalspiele beim Lesen und Schreiben von Daten zu analysieren, also das Handshake zu begreifen. Der Vinculum-Chip zeigt an zwei Pins seinen Status an, zwei weitere Pins dienen als Steuersignale:

RXF# geht auf Low sobald Daten abgeholt werden können
TXE# zeigt mit Low an, dass Daten geschrieben werden können
RD# ein Low-Impuls liest ein Datenbyte
WR ein High-Impuls schreibt Datenbyte ein

Dieses Signalspiel eignet sich leider nicht, um die 4 Signale direkt mit den Strobe- und Ready-

Leitungen einer Z80-PIO zu verbinden und diese damit direkt bidirektional zu betreiben. So musste ich die Handshake-Signale an PIO-Port B legen und programmiertechnisch abfragen. Um den bidirektionalen Kanal A von Eingabe auf Ausgabe umzusteuern, ist es erforderlich, die PIO-Anschlüsse A-Strobe und B-Strobe zu beschalten. Dies übernimmt ein zusätzlicher DL000 (74LS00), der von den Leitungen RD# und WR mitgesteuert wird und so automatisch die passende Datenrichtung freigibt.

Dazu ist es ausreichend, ASTB zu beschalten - dann wird bereits das PIO- Ein/Ausgaberegister umgeschalten. So könnte BSTB zur Interrupt-Auslösung genutzt werden, wenn Daten verfügbar sind. Eine direkte Anschaltung von RXF# auf BSTB funktioniert aber nicht, auch nicht mit Negation. Denn BSTB muss zwingend Low sein, damit die Daten bei der Eingabe auch im Eingaberegister übernommen werden. BSTB kann jedoch auch Interrupts auslösen, und zwar mit der L/H-Flanke (!), mit dem Hintergrund, dass die Eingabedaten damit im Eingaberegister verfügbar sind. So habe ich die ursprüngliche Schaltung dahingehend modifiziert, dass sowohl bei der Eingabe BSTB Low ist, als auch das Signal „Daten verfügbar“ einen L/H-Wechsel verursacht, wenn neue Daten verfügbar sind, RXF# also auf Low geht (und nicht gerade eine Eingabe läuft).



li: Schaltung von Mario Leubner, rechts: DL. Hier sieht man die Logik besser!

Tipp: Auf der GIDE+USB-Platine müssen für den USB-Teil nur PIO, 74LS08, 74LS00 und der 8-Bit-Comparator 74LS688 sowie das VDIP1-Modul bestückt sein.

Als Adresse ist bevorzugt **0DCH** zu nutzen. Das ist kompatibel zum Z1013 und auch so im JKCEMU-Emulator umgesetzt: d.h., bei den Adress-Jumpern JP5 ist nur A5 zu stecken (das ist Adresse 0DCh-0DFh)

VDIP1



USB-Sticks

Der VDIP1 unterstützt USB 1.1 und USB 2.0-Sticks. Ein 8GB-Stick wurde erfolgreich getestet. Der Stick muss mit FAT12, FAT16 oder FAT32 formatiert sein.

Achtung: Lange Dateinamen werden nicht unterstützt! Am günstigsten ist es, wenn man nur mit kurzen 8.3-Dateinamen arbeitet.

Flashen einer neuen Firmware

Aktuell ist Version 3.69; die Version 3.68 reicht aber auch. Unter http://www.ftdichip.com/Firmware/Precompiled.htm, Latest Vinculum (VNC1L) Firmware Releases, findet man ggf. eine neue Version. Es wird die **VDAP** Disk And Peripheral Firmware benötigt. Die Reflash (FTD)-Datei wird als FTRFB.FTD ins Root-Verzeichnis des USB-Sticks abgelegt. Beim Starten des Rechners bzw. auch beim Start von USB.COM installiert das VDIP1 automatisch seine neue Software.

Einrichtung

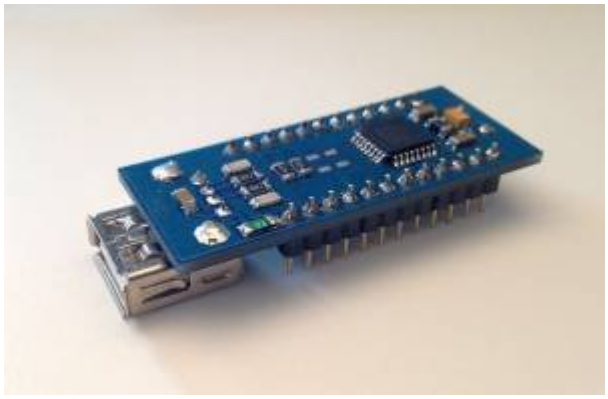
- auf dem VDIP1 muss JP3 1-2 und J4 3-2 gesteckt sein (Parallel FIFO)
- auf VDIP1 muss die passende Firmware aufgespielt sein (VDAP Version 3.68 oder neuer)

LEDs

Die beiden LEDs auf dem VDIP1 signalisieren den aktuellen Zustand:

LED1 (links)	LED2 (rechts)	Bedeutung
blinkt	blinkt	2 Sek. abwechselndes Blinken. Power On
an	aus	USB Stick init.
aus	an	USB Stick ready
aus	aus	kein USB Stick gesteckt
aus	blinkt	Ausführen eines Kommandos

V2DIP



Der V2DIP-Nachbau von Kingstener (http://www.kingsteners.homepage.t-online.de/Erweiterungen, USB Modul mit FTDI VNC2-32) hat die USB-Buchse unter der Leiterplatte. Damit passt dieses Modul komplett ins KC-Modulgehäuse.

Das V2DIP-Modul ist softwarekompatibel zum VDIP1 und kann ohne Änderung genutzt werden.

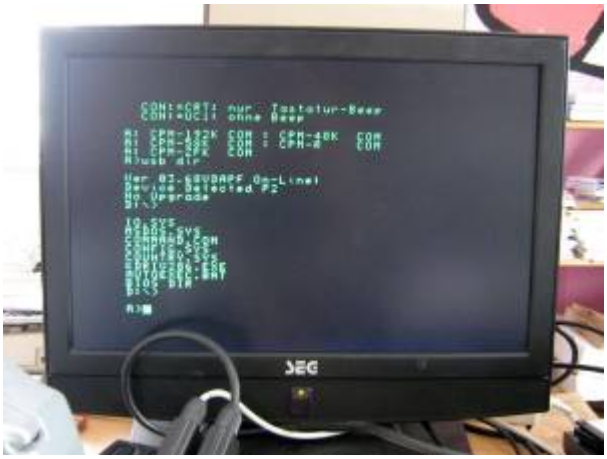
Mein alter USB-1.1-64MB-Stick wird nicht erkannt, am VDIP1 lief er. Aber wer nutzt noch solche alten Sticks?

CP/M

Unter CP/M stehen die **UTools von Mario Leubner** zur Verfügung. Die UTools sind die Programme USB.COM, UPUT.COM, UGET.COM, UDIR.COM.

Die UTools sind unter USB (VDIP) beschrieben.





teilbestückte Leiterplatte (nur USB); am KC mit Mega-Modul und BIC/KC-Floppymodul; Start von USB unter CP/M.

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

Permanent link:
https://hc-ddr.hucki.net/wiki/doku.php/z9001/module_sonstige/gide_usb?rev=1465822381

Last update: **2016/06/13 12:53**