USB mit VDIP Viculum

Der **Anschluss eines USB-Sticks** an einen alten Heimcomputer ermöglicht einen einfachen Datenaustausch mit dem PC. Seit 21006 ist dies dank fertiger Module wie dem **VDIP1** oder **V2DIP** von Viculum/FTDI DevelopmentModules.htm, DS_VDIP1.pdf recht einfach zu realisieren.

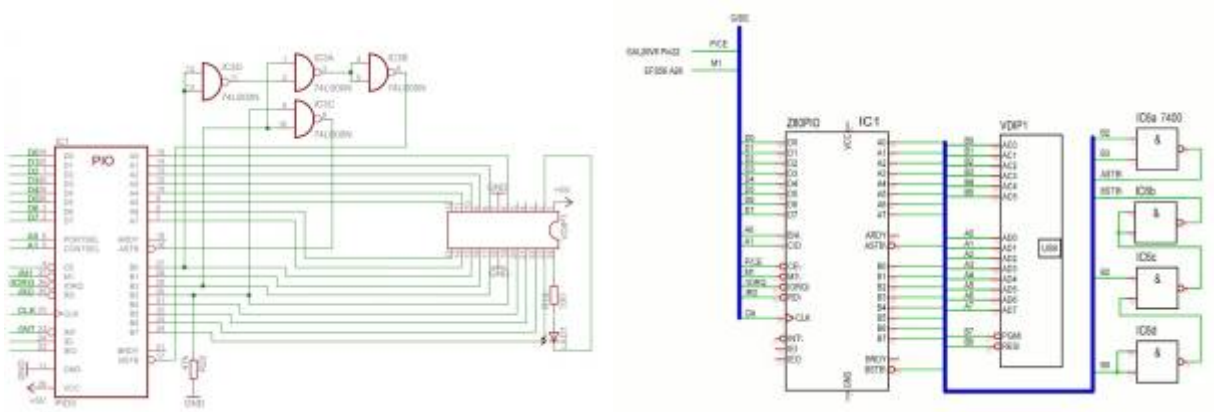


Basis dieser Entwicklung ist der USB-Anschluss aus „KC85 Labor susowa“. Mario Leubner hat die Hard- und Software entwickelt.

- http://kc85.info/index.php/component/finder/search.html?q=utools
- http://kc85.info/index.php/component/finder/search.html?q=usb-tools
- http://kc85.info/index.php/usb/72-usb-stick-am-kc.html (erster Artikel)
- http://kc85.info/index.php/download/download/19-ml-dos/420-utools18.html (aktuelle Version)

Hardware

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



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

Mario Leubner schreibt dazu im obigen artikel 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 umgeschaltet. 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).

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 Initialisieren installiert das VDIP1 dann automatisch seine neue Software.

Einrichtung

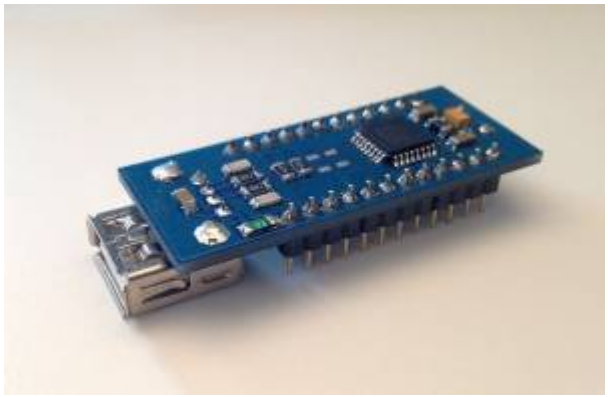
- auf dem VDIP1 muss JP3 1-2 und J4 3-2 gesteckt sein (Parallel FIFO)
- auf VDIP1 muss die passende Firmware aufgespielt sein (VDAP Version 3.68 oder neuer)

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.

2016: von W. Hardwardt gibt es ebenfalls einen V2DIP-Modul-Nachbau: https://buebchen.jimdo.com/8-bit-selbstbau-8-bit-diy/gide-f%C3%BCr-k1520/ sowie https://eb-harwardt.jimdo.com/8-bit-technik/v2dip-zum-gur-modul-mit-usb-lp-vorh/

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

Selbstbau

Leider muss man den VNC2-32 selbst flashen, ihn gibt es nicht vorprogrammiert. Kingstener hat in

seinem Download-Paket eine Dokumentation „usbmodul.pdf“, in der nicht nur der Selbstbau, sondern auch das Flashen selbst sehr gut beschrieben ist. **Tipp** Die PDF-Datei hat Anhänge mit dem BIN-File sowie Gerber-Dateien zum Nachbau!

Benötigt wird von FTDI das Flash-Programm FT_Prog_v3.1.72.360 Installer.exe (http://www.ftdichip.com/Support/Utilities.htm#FT_PROG) und die zu flashende Hex-Datei. Kingstener hat eine Datei vnc2_32v2dap_incl_reflash.rom erstellt, die neben der VDAP-Software auch einen Reflasher enthält, mit dem über USB zukünftige Softwareupdates eingespielt werden können. Die originale V2DAP Firmware unterstützt eigentlich keinen FIFO-Mode, aber Kingstener hat das umgeschrieben und den „245 FIFO-Mode“ für den Vnc2-32 implementiert, also sollte man nicht die originale Firmware verwenden.

Wer die PDF-Datei mit Anhang nicht hat, oder wer selbst entwickeln will, kann sich mit der Entwicklungsumgebung Vinculum II Installer V2.0.2-SP2.exe (http://www.ftdichip.com/Firmware/VNC2tools.htm#VNC2Toolchain) eine solche Datei ggf. selbst erstellen oder die in diesem Paket enthaltene Datei VNC1L\V2DAP\$0\V2DAP.bin nutzen (ohne Reflasher, ReflashFATFile.rom muss noch hinzugefügt werden, und ohne FIFO Mode).

Zum Thema Flashen und Reflashen siehe AN_159 Vinculum-II Firmware Flash Programming.pdf (http://www.ftdichip.com/Support/Documents/AppNotes.htm).

USB-Stick

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.

Es wird nur das FAT-Filesystem unterstützt. Die VDIP-Firmware nutzt nur die kurzen 8.3-Dateinamen!

Der USB-Stick sollte im Problemfall mit **rufus** https://rufus.ie/de/ formatiert werden. Es ist MBR, FAT, Clustergröße 2048 oder größer einzustellen. Bei einer Clustergröße von 1024 werden nicht alle Dateien im Root-Verzeichnis angezeigt.

Am V2DIP wurde mein alter USB-1.1-64MB-Stick mit FAT-12 nicht erkannt, am VDIP1 lief er. Aber wer nutzt noch solche alten Sticks?

Software

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.

- UTOOLS vom Mario Leubner. Programme für CP/M usb-tools
- utools14.zip UTools V1.4 Mario Leubner, incl. CP/M-SLR-Assembler-Umgebung zum Kompilieren unter Windows.
- utools15.zip UTools V1.5 Mario Leubner, incl. CP/M-SLR-Assembler-Umgebung zum Kompilieren unter Windows.
- Die Programme aus dem Paket zum USB-Modul für den Z1013 können ohne Änderungen auch am Z9001 genutzt werden!!

VDIP_LIB von mir. Kleine Bibliothek mit Funktionen zum Datei lesen/schreiben/Verzeichnisse etc.

```
; VDIP_INIT     Initialsierung (zuerst aufrufen!)
; VDIP_CDROOT       Wechsel ins Root-Verzeichnis
; VDIP_CD       Verzeichniswechsel
; VDIP_CLOSE        Datei schließen nach Lesen/schreiben
; VDIP_DELETE       Datei löschen
; VDIP_OPENDIR      DIR beginnen
; VDIP_GETENTRY     nächsten DIR-Eintrag holen (immer alle holen bis Ende!)
; VDIP_GETFILESIZE  Dateigröße ermitteln, auch zum Test, ob Datei ex.
; VDIP_OPENR        Datei zum Lesen öffnen
; VDIP_OPENRA       Datei zum vollst. Lesen öffnen (immer alle Bytge lesen
bis Ende!)
; VDIP_READB        Ein Byte aus Datei lesen
; VDIP_OPENW        Datei zum Schreiben öffnen (immer Datei am Ende
schließen!)
; VDIP_WRITEB       1 Byte schreiben
; VDIP_WRITE        n Bytes schreiben
```

Nutzung von VDIP am Rechner:

- Z1013 USB-Modul
- Z9001 Hardware GIDE+USB+RTC-Modul, Software OS-Erweiterung USB+SD
- LC80 (USB-Modul von Rolf Weidlich, Unterlagen in http://lc80ex.de/downloads/software/LCTOOLS_V2.zip)
- AC1 #pio2pio3-usb oder auch als Zusatzmodul für den picoAC1. Software: Komfort-Montitor V11, VDU 2.1, MONX
- TINY VDIP/USB am TINY (andere Hardware!)

Z9001/Z1013: 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.

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

Permanent link:
https://hc-ddr.hucki.net/wiki/doku.php/elektronik/vdip

Last update: **2026/09/13 10:44**