

einfaches Modul,nur CH376-Aufsatz, Platine von „felge“, und das benötigte USB-CH376-Modul

Alternativ: in <https://www.robotrontechnik.de/html/forum/thwb/showtopic.php?threadid=19015> hat „Hobi“ in Eintrag 022 ein kleines Modul mit ROM und RAM vorgestellt mit minimaler Hardware, was nur noch um ein Port-Signal für den CH376 erweitert werden muss.

## Downloads

- techn. Unterlagen zum Modul (ROM-Inhalt, Schaltplan, Bestückungsplan)  
ch376-modul.zip
- bzw. aktuell unter [https://github.com/felge1966/Kleincomputer/tree/main/KC87/USB\\_CH376](https://github.com/felge1966/Kleincomputer/tree/main/KC87/USB_CH376)
- Software f. USB-Stick → s. [USB-OS](#)
  - Binär-Version  
usb-os-bin.zip  
, enthält auch aktuellen ROM-Inhalt ch376os.bin für den Modul-ROM
  - Handbuch  
usb-os.pdf

## technische Daten

Das Modul kann am Z1013 und am Z9001 gesteckt werden. Es ist eine variable Bestückung bzgl. ROM möglich. Kommt ein ROM 27128/27256 zum Einsatz, kann über Jumper ein 10K (16K)-Bereich ausgewählt werden, der aktiviert wird.

|      |                                  |
|------|----------------------------------|
| Port | 28h..2Fh CH376                   |
| RAM  | 4000-BFFF (32k), 4000-E7FF (48k) |
| ROM  | C000-E7FF (10k)                  |

Die Portadr. gilt für Z9001 und Z1013.

Jumper auf der Platine v.o.n.u.

|             |     |  |
|-------------|-----|--|
| JP4 ROM SEL | 3-4 | Adressleitung A14 des ROMs, gesteckt: L, offen H   |
| JP4 ROM SEL | 1-2 | Adressleitung A15 des ROMs, gesteckt: L, offen H   |
| JP2 RAM SEL | 3-4 | RAM ON, gesteckt: RAM des Moduls aktiv, offen RAM deaktiviert  |
| JP2 RAM SEL | 1-2 | RAM 48K, gesteckt: RAM im Bereich C000-E7FF, offen ROM im Bereich C000-E7FF  |
| JP3 ROMDI   | 1-2 | gesteckt: Ausblenden eingebauter BASIC-ROM, muss beim KC87 gesteckt sein, beim Z9001 muss der Jumper offen bleiben |

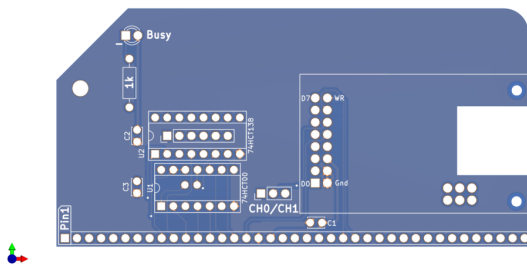
„RAM 48K“ ist momentan noch nicht im GAL umgesetzt!

## Aufbauhinweise

- Unterlagen zum Modul gibt es unter [https://github.com/felge1966/Kleincomputer/tree/main/KC87/USB\\_CH376](https://github.com/felge1966/Kleincomputer/tree/main/KC87/USB_CH376)
- Man braucht ein CH376-Module mit 6pol. Jumperfeld (s.u., erste Variante)
- Das CH376-Modul wird im Parallel-Modus betrieben (JP P\_S in Stellung P).
- Der RAM-Baustein ist ein 628128 (128k x 8Bit SRAM, z.B. AS6C1008-55SIN, BS62LV1027PCP70, WS628128LLPG). Es sollte eine extra flache Fassung genutzt werden, da er sonst mit dem Quarz des CH376-Moduls kollidiert. Oder man lötet den RAM direkt ohne Fassung ein.

- Als Jumper sollten abgewinkelte Exemplare genutzt werden.
- Das Widerstandsnetzwerk 4x10k sollte entweder eine niedrige Bauform haben oder aber flach auf der Platine montiert werden. Ich habe mangels passendem Widerstandsnetzwerk 4 einzelne Widerstände genommen.

Die Platine ist am CH376-Modul recht gedrängt. Zukünftige Platinen sollten hier mehr Platz für das CH376-Modul lassen; auch ist es sinnvoll, Aussparungen auf der Leiterplatte für den USB-Anschluss und das S/P-Jumperfeld vorzusehen, so dass das Modul direkt auf die Platine aufgelötet werden kann und damit die Gesamthöhe so klein bleibt, dass alles in ein originales Modulgehäuse passt. → <https://rc2014.co.uk/modules/ch375-usb-storage/>



Die Datei **CH376\_RR\_2.jed** wird in einen GAL 20V8 gebrannt.

Als ROM kommt ein beliebiger Typ 2764..27512 zum Einsatz (8K Byte .. 64KByte). Die Datei ch376os.rom wird in den ROM gebrannt; je nach gewünschtem 16K-Bereich muss man die Adresse im ROM passend wählen. Mit den beiden Jumpern J4 wird der 16K-Bereich ausgewählt, Standard ist beide Jumper gesteckt (A14 und A14 L), d.h. das USB-OS wird ab ROM-Adresse 0 gebrannt. Ein 128K-ROM hat bei mir nicht funktioniert, ein EEPROM Winbond W27E257 arbeitet tadellos.

Jeder Bereich des Moduls (RAM, ROM, CH376) kann einzeln in Betrieb genommen und getestet werden. Bei einem sauberen Aufbau sollte es keine Probleme geben. Das Modul muss sich beim Einschalten des Rechners mit „EOS>“ melden.

Mit einfachen I/O-Abfragen kann die Funktion des CH376 geprüft werden

```
IN 29 --> 80 device status
OUT 29 1
IN 28 --> 43 chip version (>28)
OUT 29 6
OUT 28 55
IN 28 --> AA (bit reverse)
```

## CH375/CH376-Module

Der CH376 ist ein Controller für kleine Mikrocomputersysteme zum Lesen und Schreiben von Dateien auf USB-Disk oder SD-Karte. Der CH376 unterstützt den USB-Gerätmodus und den USB-Host-Modus, ist dabei kompatibel mit USB V2.0. Der CH376 unterstützt drei Kommunikationsschnittstellen: 8-Bit-parallel, SPI oder asynchron seriell. Mikrocomputersysteme können den CH376-Chip über eine der genannten Kommunikationsschnittstellen steuern und auf Dateien oder Dateien auf USB-Disk oder SD-Karte zugreifen. Der CH376 unterstützt FAT16 und FAT32, allg. das FAT12-Dateisystem mit





|     |     |    |
|-----|-----|----|
| GND | TX  | •  |
| RX  | GND | TX |

RS232  
interface

|     |     |    |
|-----|-----|----|
| GND | TX  | •  |
| RX  | GND | TX |

Parallel  
interface

**Achtung:** Es gibt ähnliche „CH376S U DISK READ WRITE MODULE“, die aber eine andere Steckverbinderbelegung und auch nur einen 3pol. Jumper haben. Diese Module können NICHT mit dieser Leiterplatte verwendet werden!

Der CH376 kann direkt an den Z80-Bus angeschlossen werden (nur Port-Selektion nötig, die restl. Anschlussleitungen D7..D0, RD, WR, A0 gehen direkt zum Z80-BUS).

Jumper P\_S kommt in Stellung „P“.

**Hinweis 5V:** Das Modul betreibt den CH376S-Chip mit 3,3V. Das ist laut Unterlagen nicht zulässig, wenn die Peripherie mit 5V arbeitet. **Man sollte daher den Spannungsregler AMS1117 auf dem Modul entfernen und die beiden oberen Pins 2 und 3 (in Richtung USB-Buchse) überbrücken.**

Andere Projekte (rookiedrive, RC2014) verwenden diese Module aber auch ohne Änderungen?

## Unterlagen

- Produktseite: <http://www.wch.cn/product/CH376.html>
- Anleitung CH376 (engl): ch376ds1.pdf [http://www.wch.cn/downloads/CH376DS1\\_PDF.html](http://www.wch.cn/downloads/CH376DS1_PDF.html)
- zusätzliche Unterlagen + C-Code f. Microcontroller (8051), leider in chinesisich: [http://www.wch.cn/downloads/CH376EVT\\_ZIP.html](http://www.wch.cn/downloads/CH376EVT_ZIP.html)

Es gibt auch den zweiten Teil der Anleitung CH376DS2.PDF auf obiger Produktseite, ebenfalls leider nur in chinesisich: Beschreibung der grundlegenden Übertragungsbefehle und Beschreibung der externen Firmware des Gerätemodus. Mit google translate kann man die PDF übersetzen.

## Arduino

- Arduino-Library: <https://github.com/djuseeq/Ch376msc>

## Z80-Anschluss

**Bruce Abbott** hat für seinen Mattel Aquarius ein Micro-Expander-Modul entwickelt. Sein Modul umfasst 32k RAM, 4x16k ROM und obiges CH376-USB-Modul. Auf einer zweiten Leiterplatte ist ein Soundchip AY-3-8910 und ein zweites CH376-USB-Modul. Auf der Webseite gibt es komplette Z80-Assemblerquellen.

- [http://www.bhabbot.net.nz/micro\\_expander.html](http://www.bhabbot.net.nz/micro_expander.html)

- archive:  
[https://web.archive.org/web/20220126104846/http://www.bhabbot.net.nz/micro\\_expander.html](https://web.archive.org/web/20220126104846/http://www.bhabbot.net.nz/micro_expander.html)

## RookieDrive für MSX

Ein virtuelles Disketten-Laufwerk für MSX-Computer. Disketten liegen als .DSK image files (720kByte) auf einem USB-Stick. Es kommt obiges CH376-Modul zum Einsatz.

<http://rookiedrive.com/en/>, unter <https://github.com/Konamiman/RookieDrive-FDD-ROM> liegen die Assemblerquellen (rom1)

## RC2014

- <https://rc2014.co.uk/modules/ch375-usb-storage/>

## CP/M

Mittlerweile ist das Modul bei den Z80-Fans bekannt und wird auch genutzt. Ich habe zwei interessante Ansätze zum Thema CP/M gefunden:

- <https://github.com/gotaproblem/Z80Playground>  
Ein orig. CP/M mit Diskettenimages A.DSK ... D.DSK auf dem USB-Stick
- <https://github.com/z80playground/cpm-fat>  
ein neues BDOS, das direkt auf das FAT-Filesystem zugreift. Die CP/M-Dateien liegen so alle in Verzeichnissen A\...P\ auf dem USB-Stick. Allerdings werden keine direkten BIOS-Aufrufe unterstützt?

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

Permanent link:  
[https://hc-ddr.hucki.net/wiki/doku.php/z9001/module\\_sonstige/usbmodul?rev=1756966308](https://hc-ddr.hucki.net/wiki/doku.php/z9001/module_sonstige/usbmodul?rev=1756966308)

Last update: **2025/09/04 06:11**

