

USB-Anschluss



Hier ist er endlich - mein 2.50-Euro-USB-Anschluss!

Und nein - es können keine USB-Geräte wie USB-Sticks am KC angeschlossen werden, der KC wird an den PC angeschlossen. Eigentlich ist es also ein USB-Interface für den PC.

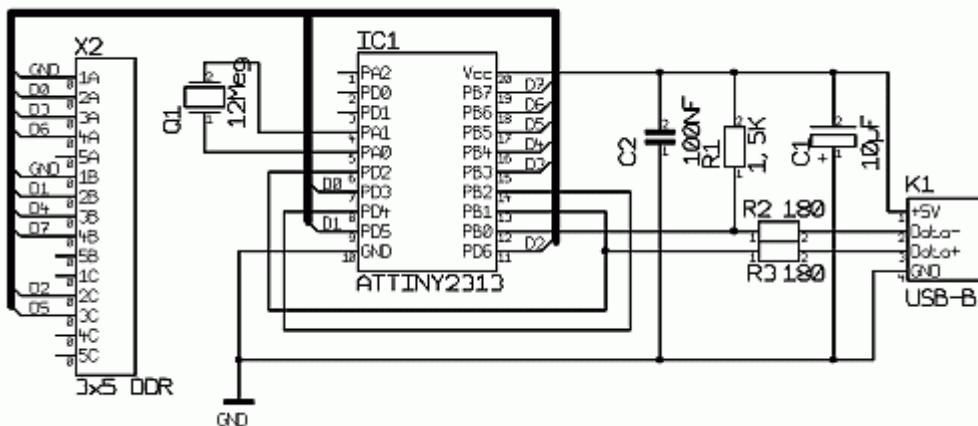
Ich begebe mich hier auf Messers Schneide: Ich bastel' was für den Z9001, aber verlasse die gewohnte und zeitangepasste Technik. Aber der Aufwand ist gering und es gibt immer mehr Rechner ohne seriellen und parallelen Anschluss, und eine serielle Lösung mit MAX232 ist auch nicht billiger und wesentlich einfacher) (Und mein nächstes Projekt ist noch frevelischer: ein 1-EUR-MMC-Anschluß mit 6 Widerständen)

Schaltung

Die Minimal-Schaltung und die gesamte USB-Treiber-Software stammt von Ing. Igor Cesko <http://members.chello.cz/cesko/index.php>, [IgorPlug-USB to RS232 converter](#), a simple and cheap universal USB device: RS232, 8-bit I/O port, EEPROM), komplett in Software realisiert mit einem Microcontroller Atmel AVR tiny2313.

Von Atmel gibt es diese Lösung als freie Version, die als „**AVR309**“ bekannt ist:
http://www.atmel.com/dyn/resources/prod_documents/doc2556.pdf,
http://www.atmel.com/dyn/resources/prod_documents/AVR309.zip

AVR309: Software Universal Serial Bus (USB) (23 pages, revision B, updated 02/06)
This application note describes the USB implementation in a low-cost microcontroller through emulation of the USB protocol in the firmware. Supports Low Speed USB (1.5 Mbit/s) in accordance with USB2.0.



Andere Schaltungs- und Softwarevariationen sind auch

<http://www.iknowu.net/Files/AVR/AVR309-1.6.zip> (AVR309-1.6, die originale Version von Ing. Igor Cesko kombiniert mit der AVR309.DLL) und hier <http://www.xs4all.nl/~dicks/avr/usbtiny/> zu finden.

Von allen Möglichkeiten des USB-Interface wird in meiner Variante nur der I/O-Port (8 Datenleitungen) genutzt, die an die User-PIO1B des Z9001 angeschlossen werden.

Alle Bauteile gibt's bei Pollin (zusammen unter 2,50 €):

- AVR tiny2313
- Quarz 12 MHz
- USB-Steckverbinder

Die Widerstände und Kondensatoren und eine KC-3x5-Buchse findet sich sicherlich in der Bastelkiste.

Der Aufbau erfolgte auf einer Lochrasterplatte 30x45 mm und mit Handverdrahtung (Verowire Fädelstift) und funktionierte fast auf Anhieb. Deshalb hier ein paar Hinweise, die ich im Netz gefunden habe:

- die USB-Datenleitungen arbeiten mit 3,3V. Deshalb haben die beiden D+ und D- -Leitungen des USB-Anschlusses Widerstände. Original sind es in der Schaltung 68 Ohm, damit funktioniert es aber nicht an allen Laptops. Mit 380 Ohm-Widerständen soll es besser gehen. Bei meinem Exemplar haben 180 Ohm gereicht, um an 4 verschiedenen Laptotypen zu funktionieren. Mit 68 Ohm war die Spannung auf den D-Leitungen mit 3,8V wohl doch zu hoch, obwohl laut USB-Spezifikation sogar 5V verkraftet werden müssen. So gibt es Applikationen, die ganz auf diese beiden Widerstände verzichten.

- Alternativ werden auch Zener-Dioden an den D-Leitungen empfohlen.
- Kondensatoren für den Quarz braucht man nicht unbedingt.
- Original ist die Schaltung für einen AT90S2313 ausgelegt. Diese Mikrocontroller werden nicht mehr produziert und wurden durch den TINY2313 ersetzt. Dieser ist code-kompatibel zum AT90S2313, d.h. das Hex-Binary kann ohne Änderung gebrannt werden. Zu beachten ist, dass Fuses gesetzt werden müssen, sonst funktioniert es nicht!

Unter http://www.mikrocontroller.net/articles/AVR_Fuses gibt es allgemeine Hinweise zu den Fuses. Wichtig sind keine Teilung des Taktes durch 8, externer Quarz, nützlich ist auch der brown out detector, aber keine Pflicht. Mit <http://palmavr.sourceforge.net/cgi-bin/fc.cgi> kann man die Fuses leicht zusammenstellen. Diese Zusammenstellung der Fuses sollte funktionieren.

- Hat man keinen geeigneten „Brenner“ zur Hand, kann man den eingebauten TINY2313 über 5

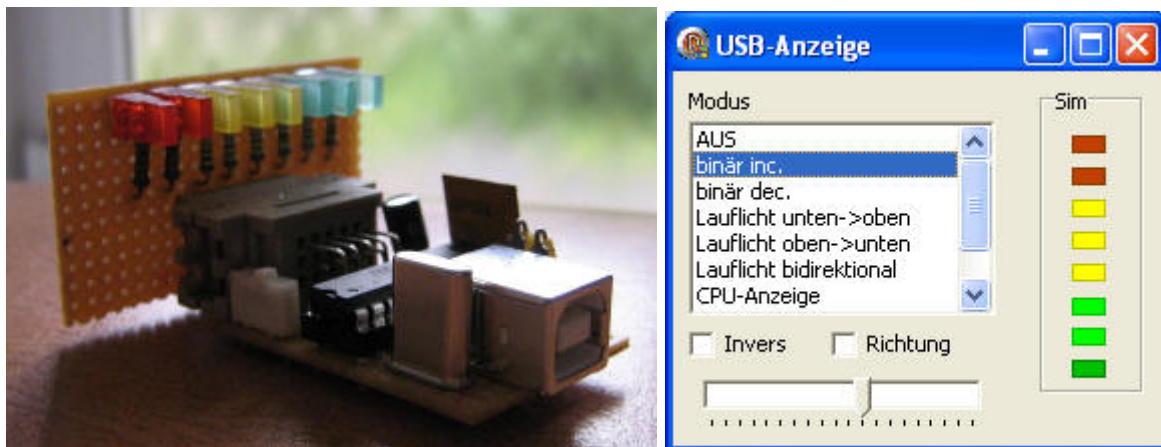
Leitungen und 2 Widerstände vom Parallelport eines PC aus programmieren. Bei <http://s-huehn.de/elektronik/avr-prog/avr-prog.htm> gibt es die Schaltung und Software. Andere einfache Brenner sind <http://rumil.de/hardware/avrisp.html> und <http://www.rowalt.de/mc/avr/progd.htm>.

Software

Am PC muss ein **USB-Treiber** installiert werden. Dieser liegt inkl. reich bebildeter Installationsanleitung im Paket von Atmel [AVR309.zip](#). Ebenfalls in diesem Paket ist auch ein Testprogramm **AVR309USBdemo.exe** zu finden, mit dem die Schaltung getestet werden kann. Ein kleines BASIC-Programm auf dem KC, was den PIO1B-Port ausliest und anzeigt, dient zur Kontrolle der Kommunikation mit dem KC, sowie ob alle Datenleitungen richtig angeschlossen wurden (Achtung: Im PC-Programm alle 'Data Port Direction' anklicken):

```
10 PRINT INP(137): GOTO 10
```

Zum **Testen der Hardware** habe ich nach einer Idee von [meierspage](#) 8 Leuchtdioden angeschlossen und ein kleines Testprogramm [usb_irpj.zip](#) dazu geschrieben:



Auf einer Lochrasterplatte habe ich 8 LEDs nebst Vorwiderständen und einer 3x5poligen Buchse untergebracht (Für Nachbauer: die LEDs sind sinnvoller auf der anderen Leiterseite bzw. abgewinkelt anzubringen, so zeigen sie in Richtung USB-Kabel).

Was kann man nun damit am Z9001 machen?

Als erste Lösung plane ich eine Rechnerkopplung, die sich in die Kassettenroutinen des Z9001 einklinkt und den PC als intelligenten und schnellen Kassettenrecorder nutzt. Als zweites gibt es dann ein angepasstes CP/M, die den USB-Anschluss als PUNCH/READER-Gerät einbindet.

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

Permanent link:
<https://hc-ddr.hucki.net/wiki/doku.php/elektronik/usbport?rev=1373026245>

Last update: **2013/07/05 12:10**



