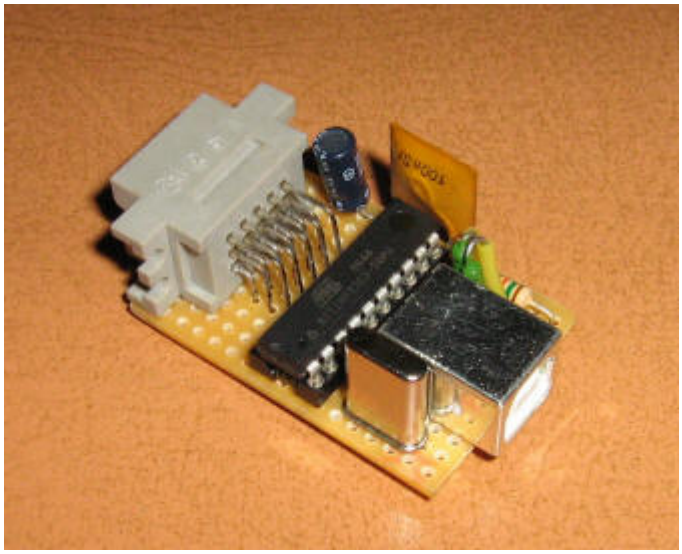
USB-Anschluss

2007 habe ich einen 8-Bit-Parallelport über USB gebastelt. Der Aufwand ist minimal, die Bauteile sind für knapp 2,50 Eur bei Pollin erhältlich.



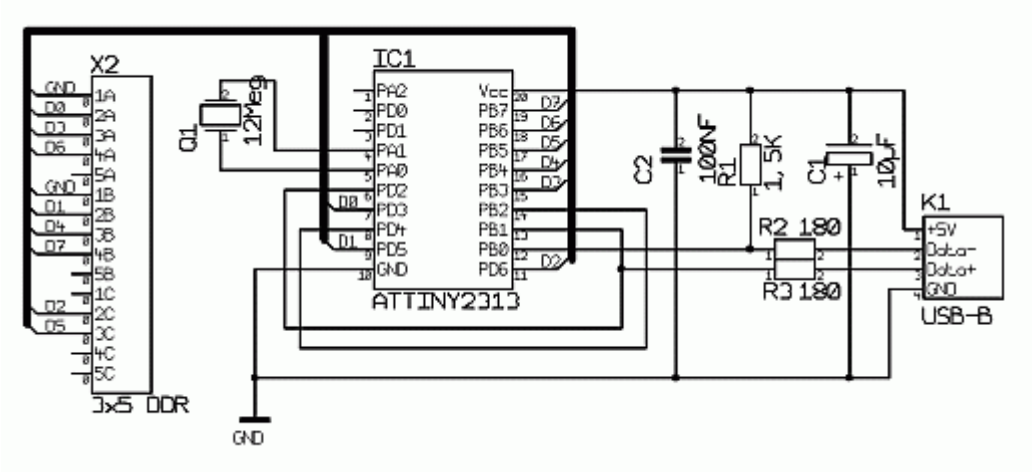
Schaltung und (alte) Software stammt von Ing. Igor Cesko und seinem Projekt IgorPlugUSB_RS232 (s.u.).

Sechs Jahre später (2013) habe ich begonnen, die Software auf die allgemein gebräuchlichere Variante V-USB (http://www.obdev.at/products/vusb/index-de.html) und die libusb-Treiber umzustellen. Damit ist auch eine Nutzung unter Linux möglich.

Zum Kopplen eines KC mit einem PC ist heutzutage sicherlich nach wie vor eine V24-Verbindung besser geeignet. Dafür steht i.allg. auf beiden Seiten der Kopplung Software zur Verfügung, während sie für diesen USB-Anschluss erst geschrieben werden muss. Steht am KC kein serieller Anschluss bereit, kann etwa mit einem kleinen USB To RS232 TTL-Anschluss mit PL2303HX (~ 1€ bei bei ebay) eine serielle Schnittstelle an freien PIO-Portanschlüssen realsiert werden

Schaltung

Die Minimal-Schaltung und die (alte) 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 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 sowie eine EGS-3×5-Buchse findet sich sicherlich in der Bastelkiste.

Der Aufbau erfolgte auf einer Lochrasterplatte 30×45 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 Laptoptypen 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 eine 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

V-USB

(Windows + Linux)

V-USB ist ein Software-USB-1.1-Treiber in C, ein Projekt von Objective Development http://www.obdev.at/vusb/. Vorteil dieses Treibers ist aufgrund der Implementation in C die leichte Anpassbarkeit an verschiedene Hardware und verschiedene Aufgaben. Es werden auch HID-Protokolle unterstützt. Damit muss man nicht einmal Treiber fürs USB-Interface installieren!

PowerSwitch

PowerSwitch http://www.obdev.at/products/vusb/powerswitch.html ist ein einfaches Beispiel für die Nuzung des V-USb-Treibers. Es werden 8 Ausgänge einzeln angesteuert. Die beschriebene Hardware entspricht obiger Schaltung, lediglich andere Port-Anschlüsse werden verwendet:

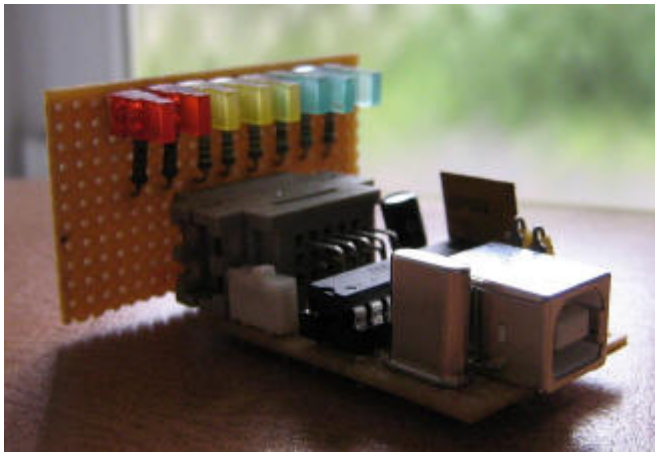
```
        USB D+    USB D-    P0    P1    P2    P3    P4    P5    P6    P7
        ------------------------------------------------------------------
---------
PowerSwitch PB1+PD2    PB0    PD4    PD5    PB2    PB3    PB4    PB5    PB6
PB7
I. Cesko    PB1+PD2    PB0    PD3    PD5    PD6    PB3    PB4    PB5    PB6
PB7
        zusätzlich sind PB2 und PD4 miteinander verbunden
```

Nur die Ausgänge P0 und P2 liegen an anderen Controlleranschlüssen, der Rest stimmt überein. die Änderungen in der Beispielsoftware beschränken sich damit auf die Ausgabefunktion in main.c:

```
static void outputByte(uchar b)
{
    PORTB = (PORTB & ~0xf8) | (b & 0xf8);   // Bit 7..3
    PORTD = (PORTD & ~0x68) | ((b << 3) & 0x08) | ((b << 4) & 0x60); //Bit
2..0 = PD6, PD5, PD3
}
```

Nach Compilieren und Flashen des TINY2313 mit main.hex und Fuses hfuse=db, lfuse=ef (s. Makefile) kann das Kommandozeilen-Programm powerSwitch.exe mit dem USB-Anschluss und der LED-Platine getestet werden:

```
powerSwitch off 0
powerSwitch on 3
powerSwitch status
```



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).

AVR309

(alt, nur Windows)

Von Atmel gibt es die Schaltung und Software nach Cesko 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.

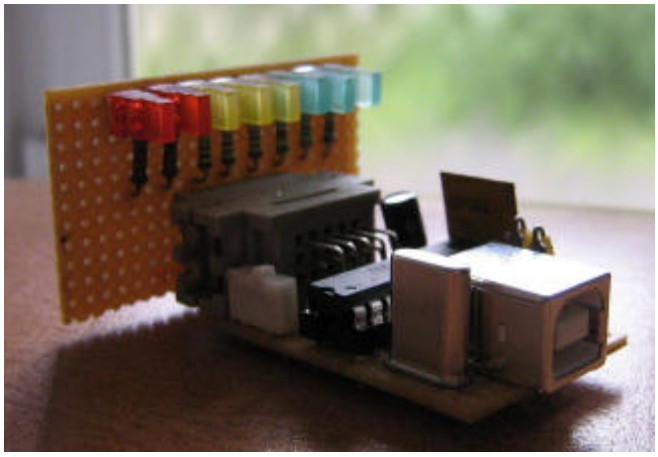
Andere Schaltungs- und Softwarevariationen sind auch http://www.iknowu.net/Files/AVR/AVR309-1.6.zip (AVR309-1.6, die originale Version von Ing. Igor Cesko kombiniert mit der AVR309.DLL) und hier http://www.xs4all.nl/~dicks/avr/usbtiny/ zu finden.

Am PC muss ein **USB-Treiber** installiert werden. Dieser liegt incl. reich bebilderter 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
```

Lauflicht

Zum **Testen der Hardware** habe ich nach einer Idee von meierspage 8 Leuchtdioden angeschlossen und ein kleines Testprogramm usb_irprj.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).

Verbindung mit KC

Was kann man nun damit am Z9001 machen?

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.

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 einen CP/M-Treiber, die den USB-Anschluss als PUNCH/READER-Gerät einbindet.

das Terminal-Programm SDX funktioniert schon.