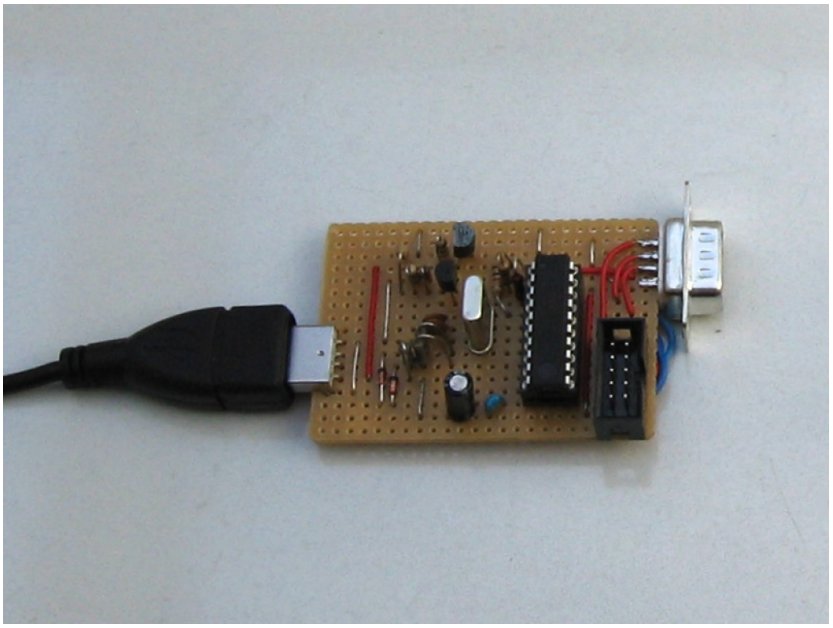
VUSB: Software-USB für AVRs

Ralf Beesner

1. August 2012



1 Überblick

Serielle Schnittstellen an PCs sterben so langsam aus. Man muss sich daher zunehmend mit gängigen USB- Seriell- Wandlern behelfen.

Es gibt aber auch eine reine Software- Lösung: Ein kleines Unternehmen aus Österreich (`www.obdev.at`) entwickelte eine USB- Library für AVR- Mikrocontroller, stellte sie unter die GPL und kümmert sich beständig um die Weiterentwicklung. In dem bereitgestellten Softwarepaket sind einige Beispiel- Implementierungen enthalten und auf der Website befinden sich Links zu weiteren Projekten, die mit dieser Library geschaffen wurden.

Leider ist der Code nicht in Bascom, sondern in C geschrieben. Einige VUSB-Projekte lassen sich jedoch gut in eigene Bascom- Projekte einbinden, wenn man die Aufgabe auf zwei Mikrocontroller verteilt.

2 AVR- Ressourcen

Da das USB- Protokoll recht aufwendig und zeitkritisch ist, bindet die Library einen großen Teil der AVR- Systemressourcen und benötigt einen relativ hohen Controller- Takt. Auch muss z.B. sichergestellt sein, dass die Interrupts des USB- Systems Priorität haben.

Der USB stellt zwar 5 V Betriebsspannung zur Verfügung, die beiden USB-Datenleitungen arbeiten aber mit 3,3 V Pegel. In den Schaltungsvorschlägen wird daher entweder die Spannung auf den Datenleitungen mit 3,6 V- Z- Dioden begrenzt oder die gesamte Schaltung mit 3,3 V betrieben, indem man die 5V durch zwei Si- Dioden oder eine rote LED um etwa 1,5 V herabsetzt (im letzteren Fall sollte laut Datenblatt der Controllertakt nicht wesentlich mehr als 10 MHz betragen).

Die älteren VUSB- Implementierungen erfordern den Betrieb mit einem Quarztakt von 12 MHz, in den neueren Versionen sind auch einige höhere Taktraten wie z.B. 18 oder 20 MHz zulässig, und besonders interessant ist eine Variante für AtTiny 45 und 85, die mit 16,5 MHz "peripheral clock" (PCK) arbeitet. Der Takt für die Timer wird dann mit der selten verwendeten PLL- Schaltung dieser AtTinys erzeugt, die aus 8,25 MHz des internen RC- Oszillators zunächst die achtfache Frequenz (also 66 MHz) gewinnt und diese auf den gewünschten Takt herunterteilt (in diesem Falle auf 16,5 MHz, die doppelte Frequenz des RC- Oszillators).

Wegen der eigentlich zu geringen Genauigkeit des RC- Oszillators wird bei Start des AtTiny- Programms zunächst eine Synchronisation mit dem USB- Takt vorgenommen und der RC- Oszillator über das OSCCAL- Register feinabgeglichen.

Man spart sich so nicht nur den Quarz, sondern gewinnt auch zwei der knappen IO- Pins.

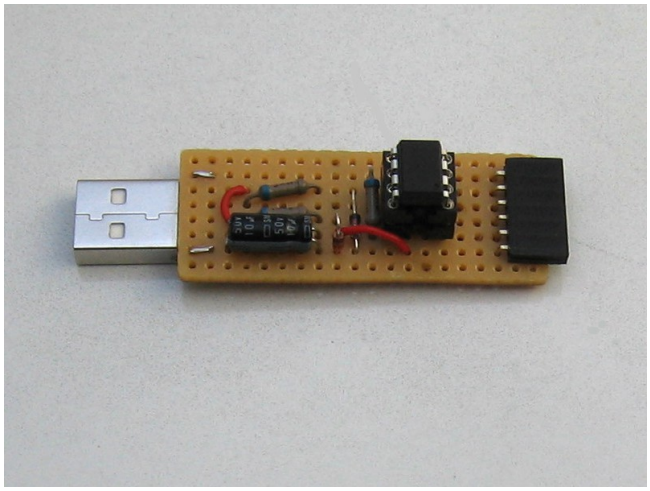


Abbildung 1: AtTiny45-Basisschaltung

3 virtuelle HID- Keyboards

Unter den auf `www.obdev.at/products/vusb/projects-de.html` vorgestellten Projekten fand ich besonders die Lösungen interessant, die ein HID- Keyboard emulieren.

Anders als gängige USB- Seriell- Wandler können sie direkt in PC- Anwendungsprogramme (z.B. eine Textverarbeitung oder eine Tabellenkalkulation) hineinschreiben.

Man benötigt also kein Terminalprogramm, das einlaufende Daten erst einmal in eine Datei schreibt, die man dann weiterverarbeiten kann, und man muss auch keine eigene PC- Software entwickeln, die sich mit einem virtuellen COM- Port "unterhält", sondern der Mikrocontroller erzeugt Tastendrücke, die beliebige PC- Programme bedienen können.

Obwohl ich keine Ahnung von C habe, konnte ich einige C- Programme minimal verändern und sie für das Zusammenspiel mit eigenen Bascom- Projekten "aufwerten".

Da es sich um GPL- Software handelt, sind nicht nur die geänderten Quelltexte, sondern das komplette jeweilige Originalpaket beigefügt. In jedem Paket gibt es ein HEX- File, so dass man die Software nutzen kann, wenn man keinen AVR-GCC istalliert hat.

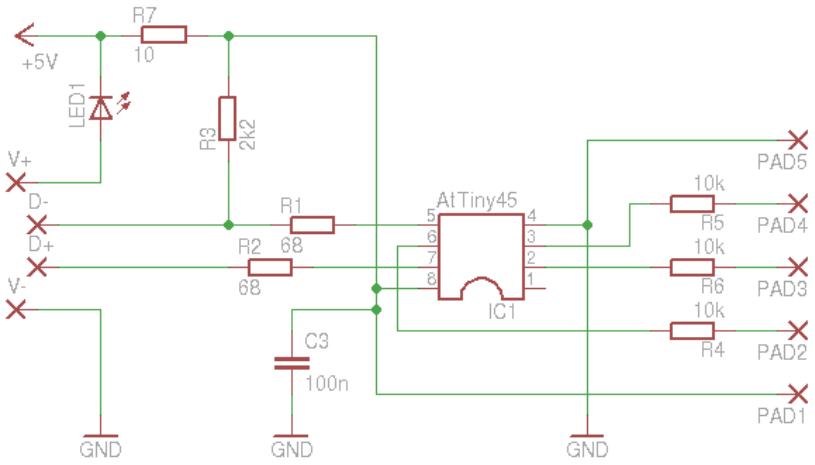


Abbildung 2: Schaltbild Basisschaltung 3,3V

4 Slideshow Presenter

Der Slideshow Presenter dient dazu, bei einem Vortrag eine Powerpoint- Präsentation oder ein PDF weiter- oder zurückzublättern.

Er hat nur zwei (oder drei) Tasten, die über ein längeres Kabel oder eine Infrarot- bzw. Funkstrecke angebunden sind, so dass man sich bei einem Vor-

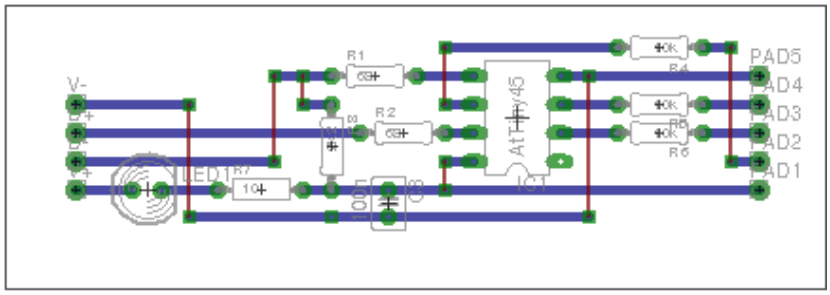


Abbildung 3: Layout Basisschaltung 3,3V

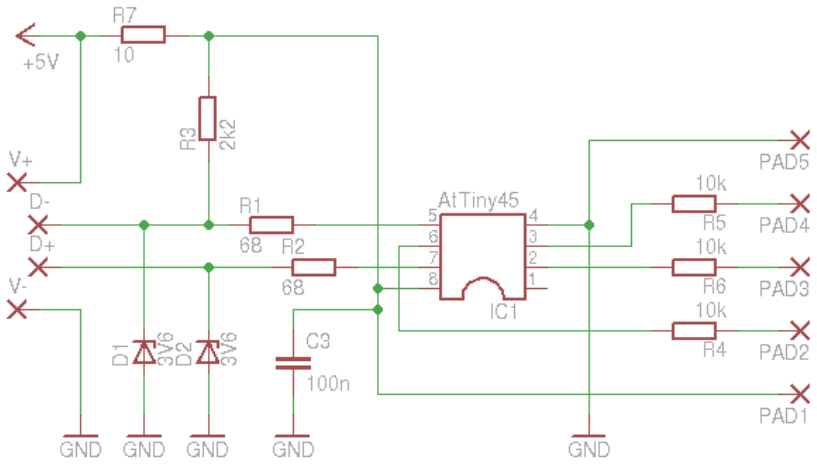


Abbildung 4: Schaltbild Basisschaltung 5V

trag vom Notebook entfernen kann (und auch nicht versehentlich auf eine falsche Taste der Notebook- Tastatur drückt).

Eine RC5- Infrarot- Strecke lässt sich recht einfach mit zwei Attiny 13 und Bascom realisieren - siehe `http://www.elektroniklabor.de/AVR/Infrarot.html`

Auf der Website `http://blog.flipwork.nl/?x=entry:entry100224-003937` ist die Originalschaltung "4-Key- Keyboard" beschrieben. Mit vier Tastern an einem Attiny 45 lassen sich die Tastendrücke "1" bis "4" an den PC senden.

Ich habe einen Taster weggelassen (er hängt am Reset und man müßte den Reset "wegfusen", um den Pin zu nutzen) und den Sourcecode minimal geändert. Auf den drei verbleibenden Eingängen liegen nun die Signale für "Bild auf" (an PB4), "Bild ab" (an PB3) und "F9" (an PB1). Die Taster (bzw. der Bascom-Mikrocontroller) müssen die Pins jeweils gegen Masse ziehen.

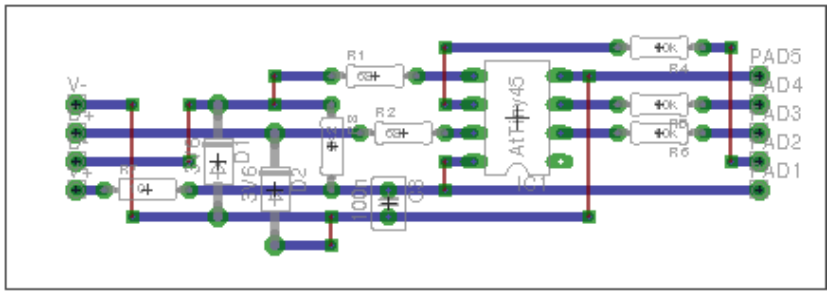


Abbildung 5: Layout Basisschaltung 5V

5 EasyLogger

Der Easylogger ist eines von Obdevs Beispielprojekten; es ist ebenfalls mit einem AtTiny 45 realisiert.

Ein ADC- Eingang (PB3) wird etwa im Sekundentakt abgefragt. Der ADC- Wert wird mit 2,5 multipliziert, mit einem Return abgeschlossen und als Tastendruck an den PC gesendet. Die Werte können z.B. in eine Spalte eines Tabellenkalkulationsblatts direkt untereinander geschrieben werden; in den Nachbarspalten kann man dann mit diesen Werten allerlei Berechnungen anstellen.

Als Spannungsreferenz wird die interne 2,56V- Referenz des AtTiny verwendet. Aufgrund der Multiplikation mit 2,5 erfolgt die Messwert- Ausgabe in mV-Schritten von 0mV .... 2560mV.

Ein Eingangspin (PB1) erzeugt den Start- Stop- Impuls, wenn er gegen Masse gezogen wird. Ein Ausgangspin (PB4) zeigt an, ob der Logger aktiv ist.

Will man das Messintervall auf einen anderen Zeittakt ändern, muss man den C- Quellcode ändern und den Controller neu flashen.

Durch meine kleine Änderung des Quellcodes wird pro Startimpuls nur ein einziger Messwert erzeugt. Den Startimpuls kann man mit einem Timer-IC oder einem weiteren Mikrocontroller und einem kleinen Bascom- Programm erzeugen und so das Messintervall in weiten Bereichen einstellbar machen. Es ist auch möglich, das Messintervall in Abhängigkeit von externen Ereignissen zu verändern.

Einige Schaltbeispiele erläutern, wie man den Messbereich des Easyloggers auf eine höhere Spannung (25V) erweitert, den Stromfluss durch einen Widerstand in eine Messpannung umsetzt (der Operationsverstärker verstärkt die Spannungsdifferenz um den Faktor 10, so dass an dem Messwiderstand nur max. 256 mV abzufallen brauchen) oder eine Temperaturmessung mit dem LM335 durchführt (da er 10 mV Ausgangsspannung pro Kelvin liefert, also bei Zimmertemperatur ca 3V, wird die Spannung auf die Hälfte geteilt, damit sie in den Messbereich des ADC fällt).

6 EasyLogger mit zwei Kanälen

Sparkfun `http://www.sparkfun.com` vertreibt einen leicht veränderten Easy-Logger unter dem Namen AVR-Stick. Die dort erhältliche Software fragt zwei ADC- Kanäle ab und gibt nach dem ersten Wert einen TAB aus; nach dem zweiten Wert ein RETURN. Auf diese Weise kann man z.B. in einem Tabellenkalkulationsblatt zwei Zahlenkolonnen "herunterschreiben" lassen.

Ich habe auch diese Version so abgeändert, dass über einen Startimpuls an PB2 ein einzelner Messzyklus gestartet wird, der PB3 und PB4 abfragt.

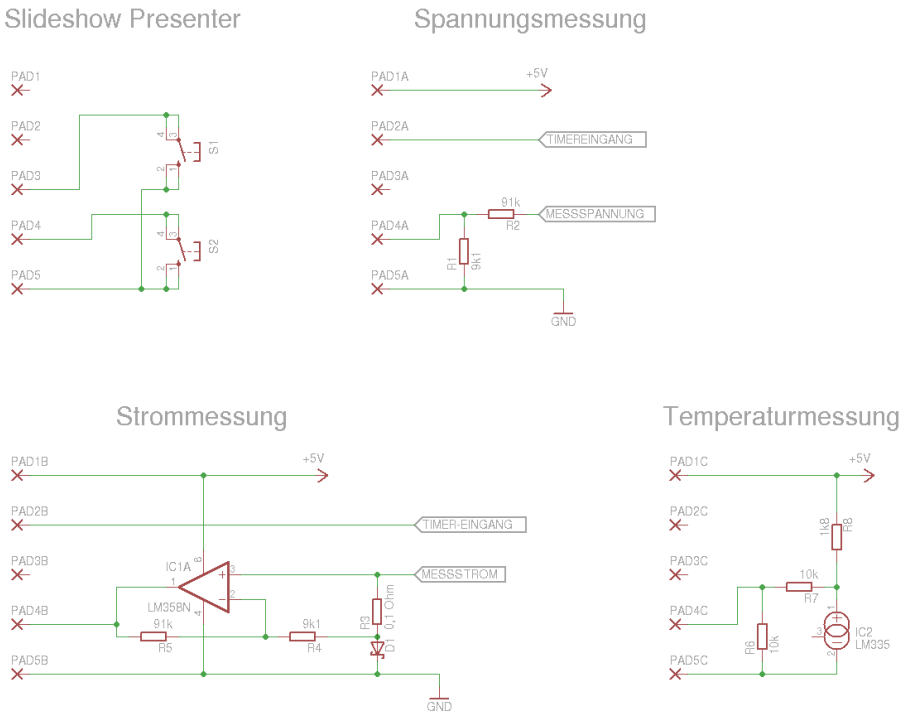


Abbildung 6: Peripherie für die Basisschaltung

7 Terminal- Keyboard

Die universelle Lösung für Bascom- User ist das Terminal Keyboard von Neil Stockbridge:

`http://www.obdev.at/products/vusb/prjdetail.php?pid=82` bzw. `http://hobby-electrons.sourceforge.net/projects/terminal-keyboard/`

Mit ihm kann man (fast) beliebige Signale als Tastatureingaben an ein PC-Programm senden.

Einziger Schönheitsfehler: Das Terminal Keyboard ist für amerikanische Tastaturen programmiert.

Es setzt ASCII- Zeichen in Tastaturcodes um. Die PC- Tastaturcodes sind jedoch an die Lage der Taste auf der Tastatur gebunden, nicht an die Bedeutung der Taste. Diese wird erst durch den (deutschen oder amerikanischen) Tastaturtreiber festgelegt.

Daher sind bei Nutzung an einem deutschsprachigen PC Y und Z vertauscht, statt Semikolon und Doppelpunkt erscheinen Umlaute, statt Minus ein Slash usw.

Ich habe daher versucht, die Tabelle, in der die ASCII- Werte in Tastaturcodes umgesetzt werden, für deutsche Tastaturen abzuwandeln. Bei den meisten Zeichen hat das geklappt; auf der Strecke geblieben sind folgende exotische Zeichen, die auf der deutschen Tastatur in Verbindung mit der Alt-Gr- Taste eingegeben werden: `'@‘{|}~`

Die beiden Zeichen `[` und `]` funktionieren jedoch. Das Zeichen `[` ist wichtig, weil es für Escape- Befehle verwendet wird, mit denen man den Cursor manipulieren kann.

Damit kann man mikrocontrollergesteuert in Kalkulationsblättern herumnavigieren, also z.B. mehrere Spalten in einer Zeile nacheinander ausfüllen und dann an den Anfang der nächsten Zeile springen, statt nur simpel eine vertikale Zahlenkolonne herunterzuschreiben.

Escape ist das Zeichen <0x27>. Mit den Zeichenfolgen `<Escape>[A` bis `<Escape>[D` wird der Cursor um ein Zeichen nach oben, unten, rechts und links bewegt.

Die Zeichenfolgen `<Escape>[5` bis `<Escape>[8` erzeugen die Signale PgUp, PgDwn, Home, End.

Falls der RS232- Eingang nicht benötigt wird, kann ein TTL- Signal über einen Schutzwiderstand von 1 kOhm bis 10 kOhm direkt auf Pin 2 des AtTiny2313 gegeben werden.

Die Geschwindigkeit des seriellen Eingangs beträgt beim Original- Code 57600 bit/s; ich habe sie in der deutschen Version auf 9600 bit/s herabgesetzt, da 9600 bit/s gebräuchlicher sind und für die meisten Zwecke ausreichen dürften.

8 Flashen der AtTinys

Da die Software unter der GPL steht, habe ich die Pakete komplett beigefügt. Für den, der sie nicht selbst kompilieren kann, ist in den 3 Paketen jeweils im Unterordner "firmware" ein HEX- File enthalten, das direkt in den Mikrocontroller gebrannt werden kann.

Unter `http://www.elektroniklabor.de/AVR/AVRdude.html` ist beschrieben, wie man die Hardware des LP Mikrocontroller mit dem Kommandozeilentool `avrdude.exe` programmieren und umfusen kann.

Das erforderliche Fusebyte lautet für den AtTiny45: `lfuse=0xe1` , es aktiviert die PLL.

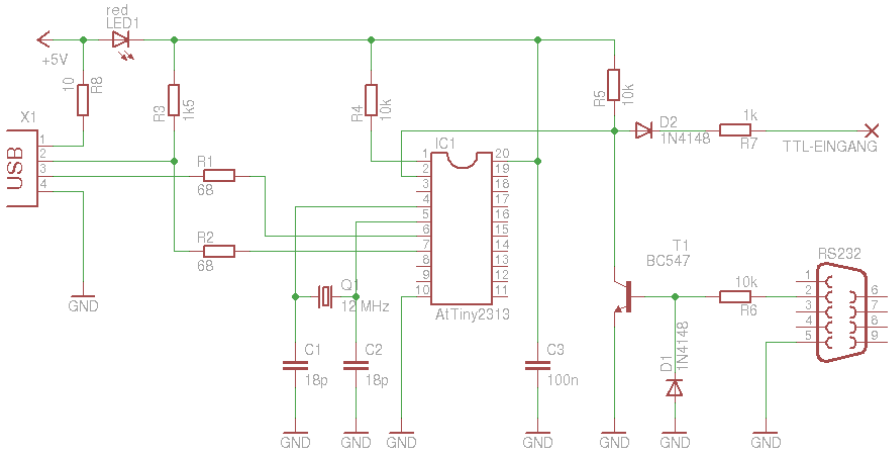


Abbildung 7: Schaltplan Serial Keyboard

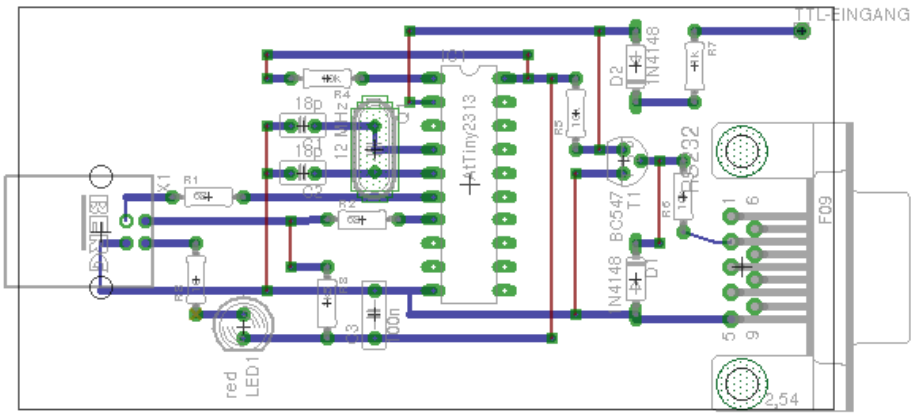


Abbildung 8: Layout Serial Keyboard

Der komplette Aufruf lautet:

```
avrdude.exe -p t45 -c burkhard -P com1 -U lfuse:w:0xe1:m
```

Das erforderliche Fusebyte lautet für den AtTiny2313: `lfuse=0xff`, es aktiviert den Quarzoszillator.

Der komplette Aufruf lautet:

```
avrdude.exe -p t2313 -c burkhard -P com1 -U lfuse:w:0xff:m
```