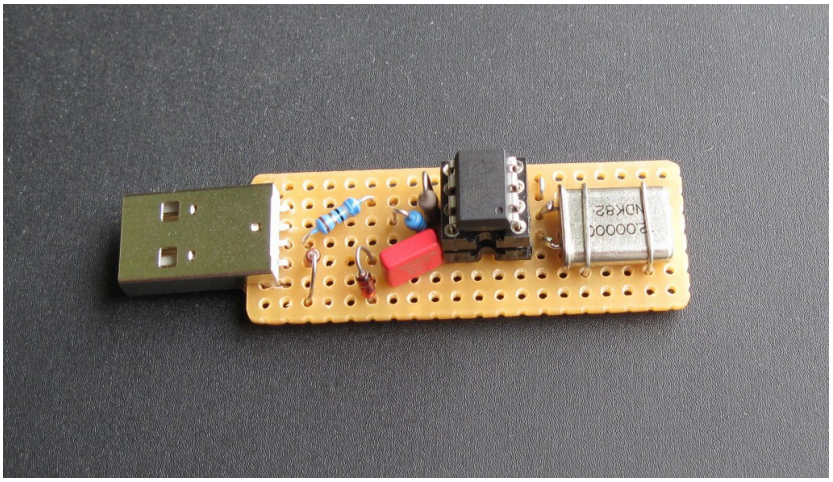
Windows- Rebooter

Ralf Beesner

1. August 2012



1 Übersicht

Unter den auf `http://www.obdev.at/products/vusb/projects-de.html` vorgestellten Keyboard- Emulatoren gibt es neben ernsthaften Anwendungen auch einige Spaß- Schaltungen, die man arglosen Zeitgenossen "unterjubeln" kann.

Steckt man die präparierten USB- Sticks unbemerkt in fremde PCs, treiben sie als heimliche Zweit-Keyboards ihr Unwesen, indem sie unregelmäßig Tastendrücke einfügen und den arglosen PC- Nutzer an seinem Verstand zweifeln lassen.

"Capslocker" drückt ab und zu die CapsLock- Taste und schaltet so dauerhaft auf Großschreibung um. "Cenillard" erzeugt ein wildes Geflacker der LED-Tastaturleuchten, und das (sogar kommerziell gefertigte) "Haunted USB Cable" betätigt neben der CapsLock- Taste noch einige weitere Tasten, die in die Keyboard- Eingaben der regulären Tastatur eingestreut werden. Die gesamte Schaltung ist in SMD ausgeführt und unsichtbar in den Stecker eines USB-Verbindungskabels eingegossen; daher der Name.

Als Programmierübung habe ich versucht, das Prinzip auf die Spitze zu treiben

und die Software so zu modifizieren, dass sie einen Windows- PC willkürlich herunterfährt bzw. rebootet.

Da ich niemanden kenne, den ich so derbe veralbern möchte, veröffentliche ich die Lösung stattdessen, damit die Arbeit nicht völlig vergeblich war.

2 Tastendrücke

Um Windows XP über Tastaturkommandos zu beenden, gibt es mindestens zwei Wege: wurde in der Windows- Benutzerverwaltung für den aktuellen Benutzer der "Willkommensbildschirm" aktiviert, muss man die linke Windows- Taste (oder alternativ Strg - Escape) drücken. Ist das Menü aufgeklappt, gelangt man mit "a" in den Beenden- Dialog und kann zwischen Herunterfahren (Taste "n") und Reboot (Taste "a") wählen.

Ist die "klassische Benutzeranmeldung" aktiviert, muss man nach Aufklappen des Menüs ein "r" drücken, um in den Beenden- Dialog zu gelangen. Dort präsentiert sich ein Pulldown- Menü, in dem man durch Drücken der Taste "n" den Neustart aktiviert. Schließlich muss man die Auswahl noch mit der Return-Taste bestätigen.

Eine weitere Hürde tut sich auf, wenn gerade Dateien bearbeitet wurden, dann werden die zugehörigen Anwendungen vor dem Herunterfahren nicht automatisch beendet, sondern es ploppt jeweils ein Dateidialog auf, der fragt, ob und unter welchem Namen die Datei gespeichert werden soll.

Mit jeweils einem "n" für "nicht speichern" lassen sie sich auf brutalstmögliche Weise beenden.

Die komplette Tastenfolge für einen Reboot lautet also:

<Strg-Escape>annnnnn bzw.

<Strg-Escape>rn<Return>nnnnn

Um beide Varianten abzudecken, lautet sie zweckmäßigerweise:

<Strg-Escape>annnnnnrn<Return>nnnnn

Einige Fenster lassen sich allerdings nicht automatisch oder per "n" schließen, z.B. Excel (die "nnnnn" werden in die Tabelle geschrieben).

Ob die Tastenkombinationen auch unter Windows 7 oder Vista funktionieren, weiß ich nicht - es wäre nicht ganz unwahrscheinlich, dass Microsoft sich neue Tastaturkommandos ausgedacht hat.

3 Software

Beim Herumspielen mit den oben genannten Lösungen stellte sich schnell heraus, dass (zumindest auf meinem Windows XP) die auf der AtTiny- PLL- Fre-

quenzerzeugung basierenden Lösungen einen Reboot bzw. einen Kaltstart des Rechners nicht "überlebten" (unter Linux trat das Problem übrigens nicht auf). Der Windows- Rebooter würde also nur genau einmal funktionieren.

Eine partielle Abhilfe besteht darin, den 1,5 kOhm- Pullup- Widerstand, der dem USB ein low-speed- Gerät signalisiert, nicht ständig an Plus zu legen, sondern über einen Ausgang des Mikrocontrollers erst nach einiger Zeit auf high zu ziehen, nachdem also Windows hochgefahren ist. Das funktioniert aber nur nach einem Kaltstart zuverlässig (weil der Mikrocontroller dadurch stromlos war und neu startet).

Die Kalibrierung des Mikrocontrollers erfolgt durch die Funktionen in osccal.h auch nur einmal nach dem Start des Mikrocontrollers; bei Rechnern, die nur selten heruntergefahren werden, könnte der RC- Oszillator daher so weit wegdriften, dass die USB- Kommunikation zusammenbricht.

Die obdev.at- Lösungen mit einem Quarz werden jedoch in jedem Fall unter Windows erkannt; daher habe ich das Herumgefummel an usbconfig.h und den osccal- Funktionen aufgegeben und 18 Cent für einen 12 MHz- Quarz investiert.

Ausgangspunkt war die Software für den "CapsLocker".

Die geänderte Version durchläuft zunächst 30 min. Wartezeit, in der sich der Rebooter passiv verhält. Dann legt die Software PB1 auf high und signalisiert dem PC ein low-speed- Gerät.

Danach wird die Variable TimerDelay auf 32000 + rand() gesetzt (was in der Hauptschleife zu einer weiteren Wartezeit zwischen 2 * 12 und 2 * 25 Minuten führt).

Anschließend geht das Programm in die Hauptschleife:

- watchdog resetten
- USB pollen
- falls der USB bereit ist, einen neuen Report senden
- Funktion TimerPoll aufrufen

Der Timer ist auf einen Überlauf nach 1/45 sec. programmiert.

In der Funkion TimerPoll werden in einer Doppelschleife die Variablen i und TimerCnt so lange inkrementiert, bis i = 2 und TimerCnt größer als der Wert TimerDelay ist. Dann werden die Tastaturkommandos "abgefeuert".

4 Hardware

Verwendet wird die 3V- Standardschaltung, die aus den 5V des USB mit zwei in Durchlassrichtung in Reihe geschalteten Dioden ca. 3,5V gewinnt. Der Widerstand R3, der dem USB ein Low-Speed- Gerät signalisiert, liegt nicht ständig an Plus, sondern wird über PB1 des AtTiny45 auf High gezogen. An PB3 und PB4

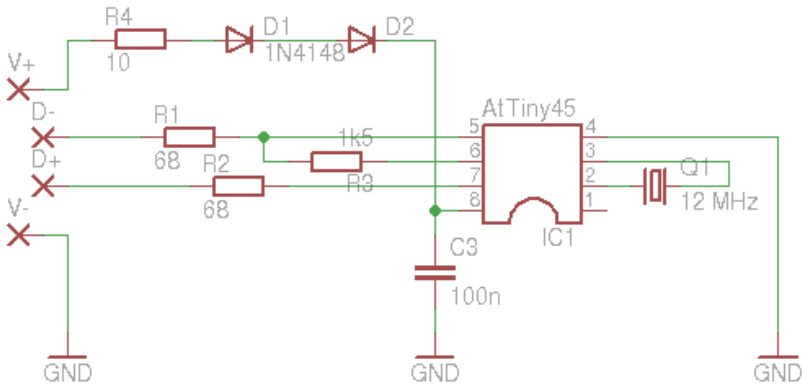


Abbildung 1: Schaltbild

liegt ein 12 MHz- Quarz. Die Bürdekapazitäten sind weggelassen; der Quarz schwingt trotzdem an. Dass seine Schwingfrequenz etwas zu hoch und weniger temperaturstabil ist (weil die entfallenen Bürdekondensatoren nicht die Wirkung der temperaturabhängigen Parasitärkapazitäten des AtTiny abmildern), stört in dieser Anwendung nicht.

Die USB- Pinbelegung entspricht einem USB-A- Stecker für Platinenmontage.

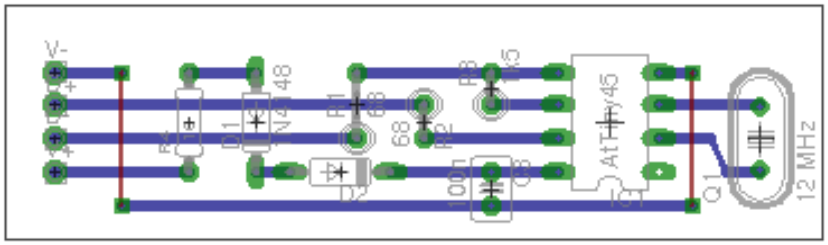


Abbildung 2: Streifenraster-Layout

5 Flashen

6 Flashen der AtTinys

Für den, der die C- Software nicht selbst kompilieren kann (das Makefile ist für Linux und müsste für Windows etwas angepasst werden), sind im gepackten Quellcode- Paket zwei HEX- Files enthalten, die direkt in den Mikrocontroller gebrannt werden können. Die "Test"- Variante arbeitet mit kürzeren Timern (ein bis zwei Minuten).

Unter `http://www.elektroniklabor.de/AVR/AVRdude.html` ist beschrieben, wie man die Hardware des LP Mikrocontroller zum Flashen und Umfusen mit mit dem Kommandozeilentool

`avrdude.exe` verwenden kann. Wurde der Attiny bereits auf Quarzbetrieb umgefust, kann man die Platine trotzdem benutzen, wenn man einen Quarz so in die Experimentierfassung einsteckt, dass er mit PB3 und PB4 verbunden ist.

Die erforderliche Fusebytes:

`lfuse=0xef` aktiviert den Quarzoszillator

`hfuse=0xdd` aktiviert den Brownout- Detektor mit einer Schwelle von 2,7V

Der komplette Aufruf lautet z.B:

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

7 Anwendung

Die Schaltung verhält sich etwa 30 min. lang passiv (damit man nach Platzierung des Rebooters erst mal "das Weite suchen kann"). War noch keine HID- Tastatur in dem USB- Port des Rechners eingesteckt, ploppt danach der Treiberdialog auf (der passende Treiber ist zwar Teil der Windows- Standard-Installation, er ist aber noch nicht an den benutzten USB- Port gebunden). Das Opfer klickt dann hoffentlich auf "automatisch installieren" - die meisten Windows- Nutzer sind ja konditioniert, kryptische Windows- Dialoge mit "ja" oder "weiter" wegzuklicken.

Danach läuft eine weitere Zufallszeit von 24-50 min. ab (um den Zusammenhang zu verschleiern), bis die Schaltung "zuschlägt".

Anschließend rebootet der Rechner im Abstand von 24-50 Minuten (nach einem Kaltstart kommen einmalig 30 min. dazu).

Die Chance ist also hoch, dass das Opfer auf einen Hardware- Defekt schließt oder Bill Gates mal wieder ergeben den Kopf für die vermeintliche Macke des Betriebssystems hinhalten muss ....