

USB-Stick als Tastatur

Emuliertes Keyboard mit AVR-Controller

Von Markus Hirsch (D)

Für den Hobbyanwender scheint ein RS232-Interface bequemer als ein USB-Port zu sein. Doch heute ist es auch mit kleinen 8-bit-Mikrocontrollern möglich, den USB-Bus im Device-Mode zu unterstützen. Die Vorteile liegen auf der Hand: Man spart die Kosten für einen USB/RS232-Konverter. Darüber hinaus kann man USB-Devices wie eine Tastatur emulieren, ohne einen eigenen Treiber zu benötigen. Man kann also eine programmierbare USB-Tastatur mit minimalen Hardware-Kosten und kleinem Software-Aufwand realisieren. Und wie Sie sehen werden, ist solch ein „USB-Tastatur-Stick“ nicht nur vielseitig einsetzbar, sondern kann auch richtig Spaß machen!

Hardware: Die Basis

Wie schon erwähnt: Die Hardware ist einfach und preiswert (**Bild 1**). Wichtigstes Bauelement des USB-Tastatur-Sticks ist ein ATtiny85 von Atmel. Die 5-V-Versorgung des USB ist über die zwei Dioden D1 und D2 (1N4148) auf etwa 3,6 V reduziert; C1 dient der Siebung. Der 1,5-k Ω -Widerstand R3 setzt die USB-Geschwindigkeitsidentifikation auf „Lowspeed-Device“. Ein 16-MHz-Quarz erzeugt einen präzisen Takt. Man kann ihn auch weglassen, wenn man dem internen Oszillator vertraut, der dann auf 16,5 MHz eingestellt wird. Eine LED mit Vorwiderstand R4 dient als Statusanzeige.

Die Schaltung kann auf einer Lochrasterplatine mit einem USB-A-Stecker einfach und recht kompakt aufgebaut werden (siehe **Bild 2**). Eine SMD-Version mit geätzter Platine ist noch kompakter und kann z.B. in ein altes USB-Stick-Gehäuse eingebaut werden.

Aus Platzgründen wurde auf einen SPI-Header verzichtet. Der ATtiny85 kann mittels Debug Wire (dW) und einer Spannungsversorgung über nur drei Leitungen programmiert werden. Diese Leitungen können temporär mit Testclips an den Controller angeschlossen werden.

Software: Der Überbau

Die Software wurde mit dem bekannten und frei erhältlichen AVR Studio 6 [1] in C erstellt. Grundlage der Software ist der USB Device Stack. Dieser ist als Library ebenfalls frei erhältlich. V-USB kann direkt von der Webseite der Entwickler [2] beziehungsweise von einer eigenen Unterseite [3] heruntergeladen werden. Hier sind auch zahlreiche Beispiele verfügbar, mit denen man USB-Projekte erstellen kann, ohne tief in das heftige USB-Protokoll eintauchen zu müssen.

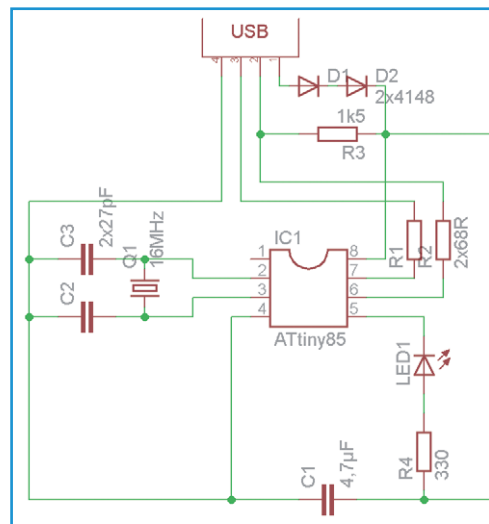


Bild 1. Schaltung des USB-Tastatur-Sticks.



Wird der USB-Stack passend konfiguriert, kann der Mikrocontroller von den meisten Betriebssystemen ohne weitere Treiber als Tastatur erkannt werden. In der Folge werden daher die LED-Infos für die Caps-, Num- und Scroll-Lock-Tasten zum Stick übertragen. Um Tastendrücke zu emulieren werden Codes entsprechend der **Tabellen 1, 2 und 3** zum PC geschickt. Diese Tasten können einen Modifier enthalten werden, um z.B. zwischen einem großen „A“ und einem kleinen „a“ zu unterscheiden. Doch Achtung: Diese Codes entsprechen nicht dem ASCII-Code des äquivalenten Zeichens.

Nachfolgend werden zwei Anwendungen des USB-Tastatur-Sticks beschrieben. Der Source-Code für AVR Studio ist zusammen mit den Hex-Dateien via Elektor.LABS [6] verfügbar.

Funktion 1: USB-Annoy-a-tron - unverzichtbarer Quälgeist

Beim Annoy-a-tron handelt es sich um ein populäres Versteck- und Such-Spiel von

ThinkGeek [4]. Vor drei Jahren wurde in Elektor eine vereinfachte und preiswertere Variante [5] veröffentlicht. Nun also stellen wir das USB-Annoy-a-tron vor, das auf einem anderen Konzept basiert und über *verbesserte Nervqualitäten* verfügt, die Ihre Kollegen, Verwandten und (noch) Freunde in den Wahnsinn treiben werden. Ein typischer und schon bekannter Ulk ist, die beiden Zeichen „v“ und „b“ auf einer normalen Tastatur physikalisch zu vertauschen, da sie nebeneinander liegen. Das USB-Annoy-a-tron geht da noch ein Stück weiter, wie Sie noch sehen werden...

Das USB-Annoy-a-tron ist im Grunde eine permanent aktive Feststelltaste. Sobald der Stick in einem PC steckt, wird er als Tastatur erkannt. Anschließend schickt der PC die Status Infos der drei LEDs (Caps-, Num- und Scroll-Lock). Der Stick wertet das Bit aus, das der Feststelltaste Caps-Lock

Bild 2. Der USB-Keyboard-Stick passt gut auf ein Stück Lochrasterplatine.

Tabelle 1. Code der Standard-Tasten einer USB-Tastatur

A	B	C	D	E	F	G	H	I	J	K	L	M	N
4	5	6	7	8	9	10	11	12	13	14	15	16	17
O	P	Q	R	S	T	U	V	W	X	Y	Z	1	2
18	19	20	21	22	23	24	25	26	27	28	29	30	31
3	4	5	6	7	8	9	0	ENTER	SPACE	ESC	CAPS LOCK	NUM LOCK	SCROLL LOCK
32	33	34	35	36	37	38	39	40	44	41	57	83	71

Tabelle 2. Code der Spezial-Tasten einer USB-Tastatur

CTRL	SHIFT	ALT	GUI	R. CTRL	R. SHIFT	R. ALT	R. GUI
0x01	0x02	0x04	0x08	0x10	0x20	0x40	0x80

Tabelle 3. Bit-Code der LEDs einer USB-Tastatur

NUM LOCK	CAPS LOCK	SCROLL LOCK
0x01	0x02	0x04

entspricht. Wenn man nun die Feststelltaste betätigt um sie zu deaktivieren, wird unser USB-Annoy-a-tron kurz darauf einen virtuellen Tastendruck simulieren, mit dem sich Caps-Lock einfach wieder einschaltet und somit immer aktiv ist (**Bild 3**). Der Mensch vor dem PC kann tippen was er will - es hilft nichts.

Nun muss also der Stick nur noch an einem gut versteckten Punkt wie z.B. auf der Rückseite des PCs eingesteckt werden. Wenn man richtig böse ist, installiert man den Stick im Inneren des PCs über ein kleines Adapter-Kabel.

Funktion 2: Masterkey – Safe für Passwörter

Ein USB-Annoy-a-tron ist nur ein Scherzartikel und eigentlich zu nichts nütze, wenn man die Möglichkeiten eines solchen Geräts bedenkt. Richtig nützlich aber ist die zweite Funktion als Passwort-Speicher. Heute sind Passwörter absolut notwendig, was ein Problem mit sich bringt: Man kann sich sichere Passwörter auf Grund ihrer kryptischen Natur nur schwer merken. Für diesen Zweck gibt es Passwortmanager wie Kee-Pass. Aber auch hier wird ein starkes Masterpasswort benötigt. Und genau da kommt der USB-Masterkey zum Zuge. Das Konzept ist nicht neu. Allerdings hat die hier vorgestellte Version eine entscheidende Funktion mehr als andere USB-Passwort-Speicher. Um das Passwort auszugeben reicht es nicht, den Stick in den PC einzustecken. Dies ist sehr unsicher, falls der Stick verloren gehen sollte. Um das Passwort zu aktivieren muss hier zunächst eine bestimmte Kombination der drei Tastatur-LEDs erzeugt werden.

In unserer Beispiel-Firmware wäre dies 1. Num Lock, 2. Num Lock + Caps Lock und 3. Num Lock. Der erste Schritt muss durch Aktivieren der Num-Lock-LED schon getan worden sein, bevor der Stick eingesteckt wird. Dann tippt man Caps Lock und ist schon bei Stufe 2. Eine weiteres Caps Lock schaltet die LED wieder aus und das Passwort wird ausgegeben. So eine kurze „Leuchtfolge“ kann man sich leicht merken. Wird eine falsche Kombination eingegeben, muss der Stick erst wieder vom PC getrennt werden.



Bild 3. Und die Feststelltaste leuchtet immer wieder. Spaßig für uns – nicht unbedingt für das Opfer.

Zugabe

Alle Dateien wie der Schaltplan und die Layout-Dateien der Platine im Eagle-Format sind wie immer auf der Elektor.LABS-Webseite für dieses Projekt [6] verfügbar.

Ein Vorschlag zum Selbermachen wäre eine Erweiterung der Funktion 2: Man könnte unterschiedliche Passwörter speichern und sie über verschiedene Kombinationen der LEDs freigeben. Das nur für den Fall, dass Sie etwas Ernsthaftes machen wollten. Falls Sie aber den Schalk im Nacken sitzen haben, können Sie sich auch andere Späße ausdenken. Ideen sind willkommen, jedenfalls solange nicht wir die Opfer davon sind ;-).

(120583)

Weblinks

- [1] www.atmel.com/microsite/atmel_studio6
- [2] www.obdev.at
- [3] www.obdev.at/products/vusb/index.html
- [4] www.thinkgeek.com/product/8c52/
- [5] www.elektor.de/090084
- [6] www.elektor-labs.com/120583

Stückliste

Widerstände:

(alle 0,25 W, 1%)
 R1, R2 = 68 Ω
 R3 = 1k5
 R4 = 330 Ω

Kondensatoren:

C1 = 4µ7/100 V, Elko
 C2, C3 = 27 p/100 V
 keramisch

Halbleiter:

IC1 = ATtiny85
 D1, D2 = 1N4148
 LED1 = LED, 5 mm,
 gelb

Außerdem:

X1 = Quarz, 16 MHz
 K1 = USB-Stecker
 Typ A