

émulateur de clavier USB

à la fois farce & attrape et clé USB de sécurité

Markus Hirsch
(Allemagne)

La familiarité actuelle avec l'interface USB n'a pas effacé le souvenir, voire la nostalgie, de l'interface RS232, bien plus commode à mettre en œuvre. Heureusement, il suffit désormais d'un modeste μC à 8 bits pour implanter le bus USB sur un montage. Avantages évidents : pas besoin de convertisseur USB/RS232 ; même sans recours à un pilote, émulation facile d'appareils USB, tels des claviers. Et pour cet accessoire de substitution à un clavier de PC, pas besoin de logiciel ni de matériel coûteux ! En plus, on peut s'en servir pour jouer des tours aux copains.

Apparences matérielles

L'idée est tordue, le schéma simple (**fig. 1**). Le cerveau est un AVR Attiny85 d'Atmel. Deux diodes 1N4148 réduisent les 5 V de l'alimentation à 3,6 V. La résistance R3 détermine la vitesse de la communication USB — au cas où l'appareil connecté est lent. Le quartz de 16 MHz n'est pas indispensable, si l'oscillateur intégré est configuré pour une fréquence de 16,5 MHz. La LED tient lieu de témoin.

Le circuit est facile à assembler sur une chute de carte d'essais munie d'une fiche USB-A (**fig. 2**).

Pour limiter l'encombrement, il n'y a pas de port SPI. Pour programmer l'ATtiny85, trois fils suffisent : *Debug Wire* (dW) et l'alimentation. Il peut les souder à même la carte ou utiliser des grip-fil.

Réalité logicielle

Le logiciel en C, basé sur le *USB Device Stack*, a été produit avec la suite Studio 6 d'Atmel [1]. Il est aussi disponible sous forme de bibliothèque autonome et gratuite. Le téléchargement de V-USB se fait directement sur le site de ses créateurs [2], où il est hébergé sur un micro-site [3] où se trouvent quelques exemples utiles, notamment pour lancer des projets USB sans avoir à se coltiner les affres du protocole.

Une fois la pile USB configurée, le μC AVR devrait être détectable, sans pilote spécifique, par la plupart des systèmes d'exploitation. Aussitôt l'état des LED des touches *Caps Lock*, *Num Lock* et *Scroll Lock* est envoyé à la clé USB par le PC qui la considère comme un clavier. Pour émuler la frappe sur ce clavier virtuel, les codes correspondant aux touches sont envoyés à l'ordinateur hôte, comme l'indiquent les **tableaux 1, 2 et 3**. Ces codes (différents des codes ASCII des caractères auxquels ils correspondent) peuvent comporter un

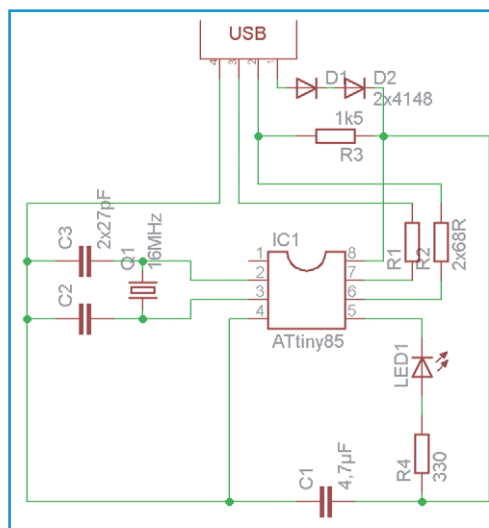


Figure 1.
Schéma de l'émulateur de
clavier USB.

modificateur, qui fait la différence entre p. ex. "A" (majuscule) et "a" (minuscule). Il reste à parler de l'utilisation de cette fonction d'émulation de clavier. En voici deux applications, l'une pour rigoler, l'autre pour ne pas pleurer.

Fonction 1 : farce & attrape pour la vie de bureau

L'application envisagée ici est destinée à animer plaisamment la vie de bureau (ou de famille) comme peut le faire une espièglerie du genre interversion (en son absence !) des cabochons des touches B et V du clavier d'un collègue.

Elle est inspirée par l'*Annoy-a-Tron*, un circuit conçu par *ThinkGeek* [4], et par plusieurs montages du même tabac proposés dans le passé par Elektor, comme p. ex. la **vermine sonore** [5]. En voici une nouvelle variante qui consiste à bloquer à l'insu de son utilisateur la fonction de verrouillage des majuscules (*Caps Lock*) sur un clavier.

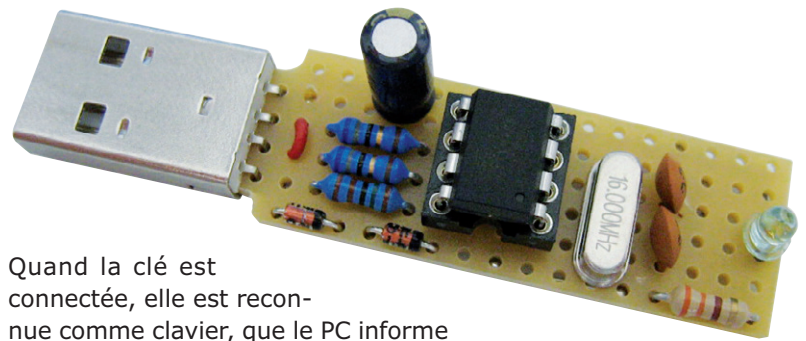


Figure 2. L'émulateur de clavier USB est facile à assembler.

Quand la clé est connectée, elle est reconnue comme clavier, que le PC informe donc automatiquement de l'état des LED (*Caps*, *Num* et *Scroll Lock*). La clé lit le bit d'état de la touche *Caps Lock*. Aussitôt que celle-ci est actionnée sur le vrai clavier pour désactiver le verrouillage, la clé USB envoie au PC un code *Caps Lock* comme si cette touche était de nouveau actionnée. Autrement dit, les majuscules restent verrouillées (**fig. 3**), et l'utilisateur ne peut rien y faire. Pour qu'il ne s'en aperçoive pas trop vite, il aura fallu prendre la précaution de connecter discrètement la clé à une prise USB à l'arrière de son PC. Si vous ne reculez devant rien, vous pouvez même monter la clé dans le PC... Ça fera durer le plaisir.

Cela devrait aller sans dire, mais il n'est peut-être pas inutile de préciser que la mise en œuvre de cette application demande du

Tableau 1. Codage des touches ordinaires de clavier USB

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	Entrée	Espace	Éch.	Ver. Maj.	Ver. Num.	Arrêt Défil.
32	33	34	35	36	37	38	39	40	44	41	57	83	71

Tableau 2. Codage des touches de fonction de clavier USB

Ctrl	Maj	Alt	GUI	Ctrl. Dr.	Maj. Dr.	Alt. Dr.	GUI Dr.
0x01	0x02	0x04	0x08	0x10	0x20	0x40	0x80

Tableau 3. Codage des LED de clavier USB

Ver. Num.	Ver. Maj.	Arrêt Défil.
0x01	0x02	0x04

tact et de la bienveillance ; elle doit amuser tous ses protagonistes et ne doit en aucun cas contribuer à la dégradation des relations entre eux.

Fonction 2 : clé USB pour mots de passe

Assez rigolé. Cette fois il s'agit d'éviter les crises de désespoir en cas d'oubli de mot de passe, une mésaventure d'autant plus fréquente qu'on nous serine partout qu'il faut utiliser des mots de passe différents. Une des solutions consiste à s'en remettre à un utilitaire comme p. ex. KeePass. Mais là comme ailleurs et même plus qu'ailleurs, il faut un *bon* mot de passe, bon c'est-à-dire long et compliqué. Donc source d'oublis et ou d'erreurs. Avec notre clé USB pour mots de passe, votre vie va changer.

Pour que la clé donne le mot de passe, il faut valider la demande par une combinaison particulière des trois touches spéciales du clavier. L'utilisateur doit suivre les étapes prescrites dans le document *code.h*, qui fait partie du progiciel de la clé USB. Ces étapes sont : 1. *Num Lock*, 2. *Num Lock* + *Caps Lock*, 3. *Num Lock*, après quoi le mot de passe est envoyé par la clé comme s'il était saisi sur le clavier. Pour augmenter la sécurité, la première condition (ici *Num Lock* ou verrouillage du clavier numérique) doit être remplie AVANT même que la clé soit branchée. Ensuite, dans notre exemple, il faut activer le verrouillage des majuscules (*Caps Lock*) de façon à remplir la 2^e condition : *Num Lock* + *Caps Lock*, et enfin désactiver le verrouillage des majuscules pour obtenir la 3^e condition : *Num Lock*. Alors seulement la clé de sécurité « tapera » le mot de passe à votre place. Si vous vous trompez dans la séquence des touches, il faut déconnecter la clé USB et recommencer. Sur votre exemplaire de la clé, il faudra bien sûr modifier vous-même dans le fichier *code.h* la combinaison des étapes de validation en fonction de vos besoins.

C'est du gâteau

Tous les documents requis par la réalisation du projet sont sur Elektor.LABS [6], y compris les schémas et le circuit imprimé au format EAGLE.



Figure 3.
La LED de la touche de verrouillage des majuscules (*Caps Lock*) ne cesse de se rallumer quand on cherche à l'éteindre. Choisissez vos victimes de telle sorte que ça reste drôle (pour elles aussi).

Vous pourrez à partir de cette proposition imaginer d'autres applications sur le même principe, p. ex. mettre sur la même clé USB plusieurs mots de passe différents, et utiliser différentes combinaisons de touches pour les appeler. Si vous êtes plutôt du genre farceur et que vous avez des idées d'applications aussi drôles qu'intéressantes, ne manquez pas de nous en faire profiter.

(120583)

Liens

- [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.fr/090084
- [6] www.elektor-labs.com/120583

Liste de composants

Résistances (1 %) :

R1,R2 = 68 Ω, 0,25 W
R3 = 1,5 kΩ, 0,25 W
R4 = 330 Ω, 0,25 W

Condensateurs :

C1 = 4,7 μF 100 V électrolytique
C2,C3 = 27 pF 100 V cér.

Semiconducteurs :

D1,D2 = 1N4148
LED1 = LED, 5 mm, jaune

Divers :

IC1 = ATtiny85, programmé
X1 = quartz 16 MHz
K1 = fiche USB-A