



lecteur d'empreintes digitales

dispositif utile d'identification

Antonello Della Pia (Italie)

La reconnaissance d'empreintes digitales est une technologie couramment intégrée dans les appareils de pointe tels que les PC, les claviers avancés et le contrôle d'accès automatisé. Toutefois, il est assez rare de la trouver disponible en tant qu'appareil autonome. Ce projet pallie ce manque en offrant un circuit versatile et facile à assembler.

L'objectif de ce projet était de tester un capteur biométrique d'empreintes digitales et de développer un module autonome fonctionnant comme un interrupteur activé par reconnaissance d'empreintes digitales, sans nécessiter une connexion à un PC. La sortie du module peut être facilement connectée à divers circuits de contrôle de charge électrique courants, tels que des opto-isolateurs, des MOSFET, des relais, ainsi qu'à d'autres circuits numériques ou analogiques.

Comme alternative aux capteurs optiques plus répandus, j'ai utilisé un petit capteur capacitif FPM383C de Hi-Link, un modèle abordable et puissant. La partie sensible de ce capteur est représentée sur la **figure 1a**, tandis que la partie arrière est présentée à droite sur la **figure 1b**, où j'ai ajouté le schéma de brochage du connecteur pour faciliter son utilisation. Ce dispositif intègre un microcontrôleur, capable de gérer l'acquisition, la vérification, le stockage et la gestion de la base de données d'empreintes

digitales, en réponse à des commandes spécifiques reçues via une interface série. Ces commandes, envoyées sous forme de séquences d'octets, peuvent provenir d'un PC ou d'un microcontrôleur. La puissance et la capacité de mémoire d'un microcontrôleur ATtiny84A sont suffisantes pour réaliser les fonctions de base. L'interaction avec l'utilisateur est assurée par le capteur et la LED RGB intégrée. En outre, j'ai ajouté deux cavaliers, un bouton de réinitialisation, deux autres LED et un buzzer, tous montés sur une carte.

Circuit

Le schéma présenté dans la **figure 2** est simple, car la majorité du travail est faite par le module capteur et le micrologiciel ATtiny. L'alimentation du circuit est assurée par un régulateur LDO à 3,3 V. En conditions normales, le capteur et le microcontrôleur sont en mode *sleep/power-down* (veille/arrêt), avec une consommation de courant



Figure 1. La face active du capteur (a) et le brochage du connecteur (b).
(Source : Laboratoire de l'auteur)

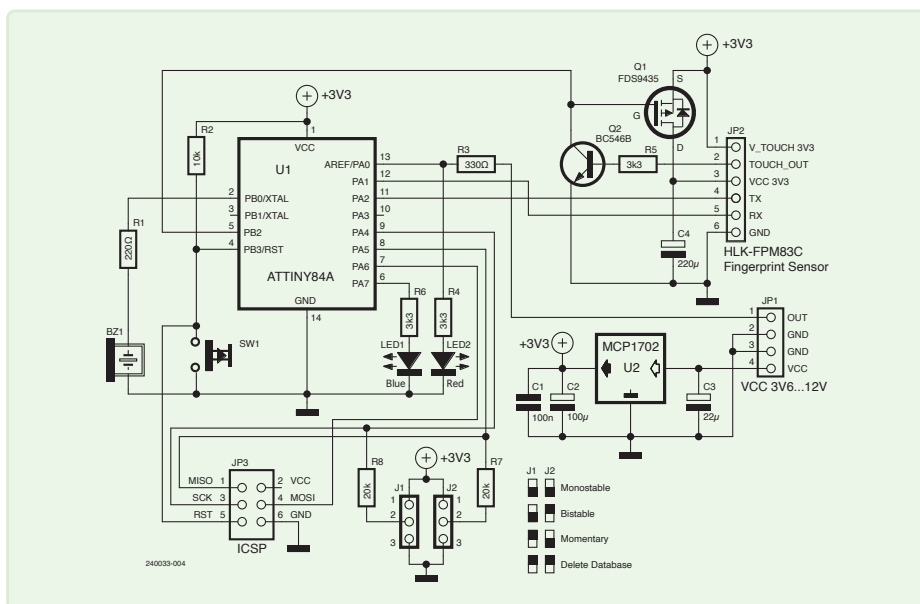


Figure 2. Schéma du lecteur d'empreintes digitales.

d'environ 40 μ A. Le timer watchdog réveille le microcontrôleur toutes les 4 s et fait clignoter brièvement la LED bleue pour indiquer que l'appareil est opérationnel. Lorsqu'un doigt est détecté, Q1 alimente le capteur et une interruption réveille le microcontrôleur pour qu'il exécute les fonctions requises. La LED bleue s'allume. La diode rouge s'allume lorsque la sortie est au niveau haut. La **figure 3** montre le prototype complet lors de son premier test. En mode *actif*, le circuit consomme environ 35 mA, ce qui le rend adapté également aux dispositifs alimentés par batterie. En configurant les cavaliers J1 et J2 selon le schéma de câblage, il est possible de choisir entre une activation *monostable (one-shot*, idéale pour actionner une serrure électrique), *bistable (latch)* ou *momentanée* (la sortie reste à l'état haut tant que le doigt est sur le capteur). Une fois les cavaliers réglés, il faut appuyer sur le bouton *reset* SW1 pour appliquer les modifications. Les tâches principales sont confirmées par un signal sonore émis par le buzzer. La résistance R3 de 330 Ω , protège la sortie PA0 contre surintensités, tandis que les condensateurs C1...C4 constituent les condensateurs de dérivation de l'alimentation. Le connecteur JP3 ICSP permet de connecter un programmeur USBasp pour flasher le microcontrôleur.

Logiciel

J'ai développé et compilé le code source en utilisant l'EDI Arduino 1.8.19, avec ATtiny core 1.5.2 de Spence Konde. Le croquis, téléchargeable sur la page web Elektor Labs de l'auteur [1], se compose de deux fichiers : le fichier principal, *Fingerprint_Sensor_Switch.ino*, et *FPM383C.h*, qui contient les tableaux d'octets de commande déduits de la documentation du capteur et les fonctions nécessaires pour contrôler le capteur. Aucune bibliothèque tierce n'est utilisée. Les fonctions les plus cruciales pour la gestion des empreintes digitales sont `fpm383cGetImage()`, `fpm383cConvertImage()`, `fpm383cFingerSearch()` et `fpm383cAutoEnroll(uint16_t pageID)`.

Les commandes définies dans les tableaux d'octets offrent également un contrôle total sur la LED RGB intégrée au capteur, créant des effets visuels très dynamiques. Le fichier *Fingerprint_Sensor_Switch.ino* file suit la structure typique d'un croquis Arduino.

Après la définition des broches et des variables, la boucle `loop()` est introduite. Cette routine inclut essentiellement des expressions conditionnelles nécessaires pour surveiller l'état du capteur, appeler d'autres fonctions, gérer l'état et les interruptions du microcontrôleur, ainsi que pour la lecture et l'écriture de données dans l'EEPROM. La fonction `enrollFingerPrint(uint16_t id)` est dédiée à l'enregistrement des empreintes digitales, tandis que `activateOutput()` et `setOutputMode()` contrôlent les modes de sortie. La fonction `wdtSleepEnable()` active le mode `SLEEP_MODE_PWR_DOWN` pour le microcontrôleur. Je vous recommande de consulter ce croquis, qui est enrichi de commentaires détaillés, d'informations approfondies, et de liens pertinents. Les noms des fonctions et des variables ont été choisis pour être le plus claires possible

Fonctionnement

Une fois le micrologiciel chargé, la LED du capteur clignote en bleu lors du premier démarrage, pour indiquer que l'appareil est prêt à enregistrer l'empreinte digitale de l'administrateur. L'administrateur peut alors accéder à toutes les fonctions en plaçant le bout du doigt sur le capteur et en comptant les clignotements bleus. Retirer le doigt après un clignotement active la sortie. Retirer le doigt après trois clignotements permet d'enregistrer d'autres utilisateurs (jusqu'à neuf empreintes digitales). Après cinq clignotements, l'empreinte digitale de l'utilisateur est effacée. Après sept clignotements, la base de données d'empreintes digitales, y compris celle de l'administrateur, est complètement effacée. Les opérations réussies sont confirmées par le clignotement vert, tandis qu'une erreur est signalée par un clignotement rouge. Un clignotement continu en rouge indique que le nombre maximal de dix empreintes a été atteint. Les utilisateurs enregistrés après l'administrateur peuvent uniquement activer la sortie. Personnellement, bien que la technologie de reconnaissance des empreintes digitales soit fascinante, elle présente une vulnérabilité :

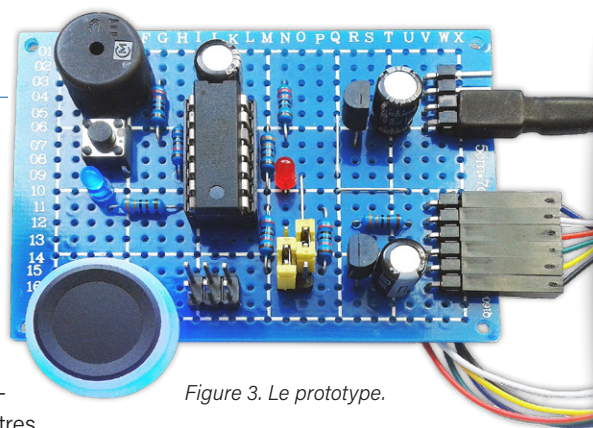



Figure 3. Le prototype.

une petite blessure sur le doigt peut suffire à invalider la reconnaissance. C'est pourquoi je pense qu'il est judicieux d'avoir une alternative. Je recommande de stocker plusieurs empreintes digitales de différents doigts pour pallier ce type de situation ! 

240033-04

Questions ou commentaires ?

Envoyez un courriel à l'auteur (anto62mail@gmail.com), ou contactez Elektor (redaction@elektor.fr).



À propos de l'auteur

Dès son plus jeune âge, Antonello Della Pia a été captivé par l'électricité et les appareils électroniques. Il est titulaire d'un diplôme de technicien en génie électrique. Antonello a toujours nourri et enrichi sa passion pour l'électronique analogique et numérique. Actuellement, il s'amuse avec les microcontrôleurs et la programmation, s'efforçant d'améliorer ses compétences. Antonello prend plaisir à concevoir et proposer des projets aussi innovants que possible.



Produits

➤ W. A. Smith, *Explore ATtiny Microcontrollers using C and Assembly Language (E-book, Elektor 2021)*
www.elektor.fr/20008

LIEN

[1] Ce projet sur Elektor Labs :
<https://elektormagazine.fr/labs/fingerprint-sensor-switch>