



LambdaConcept Taka

Lecteur Biométrique ARCH

Aug 04, 2022

DOCUMENTATION

1 Fonctionnalités	3
1.1 Lecteur carte à puce	3
1.2 Capteur d'empreinte digitale	3
1.3 Authentification	4
1.4 Ecran LCD	4
1.5 Indicateurs	4
1.6 Journaux	4
1.7 Batterie	4
1.8 USB	5
1.9 Réseau (Ethernet)	5
2 Utilisation	7
2.1 Démarrage	7
2.2 1. Insertion de la carte d'identité	8
2.3 2. Lecture de la carte	8
2.4 3. Capture de l'empreinte digitale	10
2.5 4. Vérification des minutiae	11
2.6 5. Journalisation des événements	13
3 API	15
3.1 USB	15
3.2 Exemple WebUSB	16
3.3 Détails du protocole API USB	22
4 Journaux	25
4.1 Format	25
4.2 Accès	26
5 Mise à jour	27
5.1 Mode DFU	27
5.2 WebDFU	28
5.3 DFU-Util	31
6 Diagnostics	33
6.1 Erreurs courantes	33
6.2 Firmware de test	33
6.3 Procédure de test	34
7 Boîtier	41
7.1 Démontage	41
7.2 Batterie	42

7.3	Carte mère	43
7.4	Module caméra/touch	44
7.5	Module LED/antenne	45
7.6	Module carte à puce	45
7.7	Module écran	46
8	Développement	49
8.1	Console de debug	49
8.2	Telnet	51
8.3	Compilation	52
8.4	Architecture NuttX	52
8.5	Defines	54
8.6	Changement des APDU	54

LECTEUR BIOMÉTRIQUE ARCH



Taka est un lecteur biométrique qui permet de réaliser l'authentification forte d'une personne bénéficiaire ARCH (Assurance pour le Renforcement du Capital Humain),

A ce titre, le boîtier Taka est pourvu:

- d'un lecteur de carte à puce permettant de communiquer avec une carte nationale d'identité biométrique,
- d'un lecteur d'empreinte digitale permettant de vérifier la bonne correspondance entre l'empreinte stockée dans la carte et le doigt de la personne bénéficiaire,
- d'un écran permettant de connaître le résultat de l'authentification.

FONCTIONNALITÉS

1.1 Lecteur carte à puce

La lecture de la carte d'identité se fait par contact (insertion de la carte verticalement dans le boîtier Taka).

Spécifications du lecteur:

- **Type de lecture:** Contact
 - **Standard de la carte:** ISO/IEC 7816
 - **Authentification passive:** ICAO 9303 Part 11 PKCS#7 ECDSA
 - **Applications supportées:**
 - **ICAO:** (eTravel Doc 9303)
 - **ARCH:** (Assurance pour le Renforcement du Capital Humain)
 - **MoC (Match-on-card):** ISO 19794-2 NIST MINEX
-

1.2 Capteur d'empreinte digitale

La lecture de l'empreinte digitale du bénéficiaire se fait par un capteur optique situé à l'avant du boîtier Taka et sur lequel la personne place son doigt.

Spécifications du capteur:

- **Type de caméra:** Optique
 - **Zone de capture d'image:** 17x18 mm
 - **Résolution:** 640x480 pixels @1000 dpi
 - **Eclairage:** 1 LED Bleue + 2 LED Infrarouges
-

1.3 Authentification

Les minutiae sont extraits de l'image de l'empreinte digitale via une librairie satisfaisant aux critères de performance et de qualité spécifiés par MINEX, puis sont envoyés à la carte d'identité pour vérification.

- **Type d'authentification:** Match on Card
 - **Standard de l'empreinte:** ISO/IEC 19794-2:2005
-

1.4 Ecran LCD

Le boitier Taka est pourvu d'un écran LCD rétroéclairé permettant d'afficher des instructions pour l'utilisateur et de suivre les opérations en cours.

- **Affichage:** Ecran LCD 2x16 caractères
 - **Rétroéclairage**
-

1.5 Indicateurs

Le résultat de l'authentification (succès en cas de concordance des empreintes, ou échec en cas de non concordance) est indiqué sur le boitier Taka via un système lumineux et sonore.

- **Indicateur lumineux:** LED RGB (7 couleurs)
 - **Indicateur sonore:** Buzzer
-

1.6 Journaux

Chaque opération d'authentification et son résultat sont enregistrés dans un journal. Ce journal peut être lu via l'API USB (ou en retirant la carte SD du boitier)

- **Stockage:** Carte micro SD (1 GB)
 - **Système de fichier:** FAT32
-

1.7 Batterie

Le boitier Taka est autonome et peut fonctionner sur batterie.

- **Autonomie sur batterie:** 24h minimum
 - **Changement:** Via câble USB (5V)
 - **Type de batterie:** Li-ion 1 cellule 3.7V 10000mAh
-

1.8 USB

Le connecteur USB situé à l'arrière du boîtier Taka permet une connexion à un PC.

- **Connecteur:** USB2 type-B (Full-Speed)
- **API USB**

L'API USB permet d'accéder aux données suivantes:

- **Données ICAO** contenues sur la carte d'identité
 - **Données ARCH**
 - **Journaux:** logs des tentatives d'authentification et résultats
 - **Image et minutiae** de l'empreinte digitale en cours de traitement
-

1.9 Réseau (Ethernet)

- **Connecteur:** Ethernet RJ45 10/100 Mbits
- **API réseau:** Non implémentée
- **Serveur Telnet** (Debug ou développement)

UTILISATION

2.1 Démarrage

Tout d'abord, allumer le boitier Taka en actionnant l'interrupteur On/Off situé sur le côté droit.

Lors du démarrage, plusieurs vérifications sont effectuées pour garantir le bon fonctionnement du système. Notamment:

- Vérification de la date/heure
- Vérification du capteur caméra
- Vérification de la présence de la carte SD



En cas d'échec des vérifications système, le message d'erreur correspondant s'affiche sur l'écran. On se référera alors à la section *Diagnostics*.

Le scénario d'authentification d'un bénéficiaire se déroule comme suit:

2.2 1. Insertion de la carte d'identité

Le boitier Taka affiche le message INSEREZ LA CARTE et attend. Les LEDs sont éteintes.



Lorsque l'insertion d'une carte est détectée, la machine passe à l'étape 2. *Lecture de la carte*.

2.3 2. Lecture de la carte

Affichage du message LECTURE EN COURS. Les LEDs restent éteintes.



La communication s'établit avec la carte d'identité et les actions suivantes sont effectuées:

- Lecture des 4 conteneurs ARCH (Health, Formation, Credit, Retirement)
- Vérification des dates d'éligibilité à l'assurance maladie, en comparant les dates de validité spécifiées dans le conteneur ARCH "Health" avec la date actuelle.
- Lecture des fichiers ICAO (SoD, DG1-DG16), qui contiennent les informations sur l'identité de la personne.
- Authentification cryptographiques passive de la carte d'identité.

L'authentification passive (Standard ICAO 9303 Part 11) s'assure de l'authenticité et de l'intégrité des données ICAO en validant les signatures cryptographiques des fichiers et la chaîne de confiance depuis le certificat national PKCS#7.

Lorsque les données de la carte sont lues avec succès, passage à l'étape 3. *Capture de l'empreinte digitale*.

En cas d'échec de la lecture (carte illisible, carte retirée, carte mal provisionnée, carte non authentique, etc), l'erreur correspondante est affichée sur l'écran, la LED devient rouge et demande le retrait de la carte. Retour à l'étape 1. *Insertion de la carte d'identité*.

2.4 3. Capture de l'empreinte digitale

Affichage du message POSEZ LE DOIGT INDEX DROIT. La LED RGB devient bleue, et la LED caméra s'allume aussi en bleu pour inciter l'utilisateur à poser son doigt. Les LED infrarouges du capteur caméra sont activés pour assurer une meilleure qualité d'image.



L'application passe en attente jusqu'à ce que l'utilisateur pose son doigt sur le prisme de la caméra optique. La présence du doigt est détectée par un capteur tactile situé à l'avant du prisme.

Lorsque le doigt est stable, une image est capturée par la caméra et traitée par une librairie d'extraction de minutiae satisfaisant aux critères de performance et de qualité spécifiés par MINEX.

Si la qualité de l'image est insuffisante (trop peu de minutiae) ou que le placement du doigt est mauvais (pas assez appuyé, trop haut, trop bas, etc), un message est affiché sur l'écran LCD pour inviter l'utilisateur à repositionner son doigt correctement. Le processus recommence l'étape 3. *Capture de l'empreinte digitale*.

Si l'image est de bonne qualité, la machine passe à l'étape 4. *Vérification des minutiae*.

2.5 4. Vérification des minutiae

Les minutiae sont convertis au format Card-Compact (ISO/IEC 19794-2:2005) et soumis à la carte d'identité pour vérification, via l'application MoC (Match-on-card).

Si les minutiae soumis correspondent à ceux provisionnés, la carte accepte l'authentification de l'utilisateur, et le boîtier Taka affiche alors le message DOIGT ACCPTE ainsi que le statut d'éligibilité ou non à l'assurance maladie (lu lors de l'étape 2. *Lecture de la carte*). Le buzzer émet un son "succès".

La couleur de la LED change en fonction des cas:

- Authentifié et éligible: LED verte



- Authentifié mais non éligible: LED jaune



En cas de refus des minutiae par la carte, Taka affiche le message DOIGT REFUSE. La LED devient rouge et le buzzer émet un son “échec”. Le nombre d’essais restants avant le blocage de la carte est alors affiché sur l’écran.





Dans tous les cas (authentifié ou non, éligible ou non), la tentative est logguée. Passage à l'étape 5. *Journalisation des événements*.

2.6 5. Journalisation des événements

L'opération d'authentification et son résultat sont enregistrés, avec l'identité de la personne, la date et l'heure de passage, dans un journal stocké dans une mémoire de type carte micro SD.

Ces journaux sont notamment accessibles par l'intermédiaire de l'*API*.

- En cas de succès de l'authentification:

La procédure est terminée. Le boîtier Taka passe dans un état d'attente, jusqu'à ce que l'utilisateur retire sa carte d'identité. La procédure recommence alors à l'étape 1. *Insertion de la carte d'identité* pour l'utilisateur suivant.

- En cas d'échec de l'authentification:

La procédure retourne automatiquement l'étape 3. *Capture de l'empreinte digitale* après quelques secondes pour un nouvel essai du même utilisateur.

CHAPTER
THREE

API

3.1 USB

Le connecteur USB situé à l'arrière du boîtier Taka permet une connexion à un PC.



3.2 Exemple WebUSB

Un exemple d'utilisation de cette API est fournie dans une application WebUSB utilisable depuis un navigateur Web (Chrome). Cette application implémente l'ensemble des fonctionnalités fournies par l'API et peut donc servir de référence.

3.2.1 Installation

Voir le README <https://github.com/lambdacconcept/taka/tree/master/utils/takaweb>

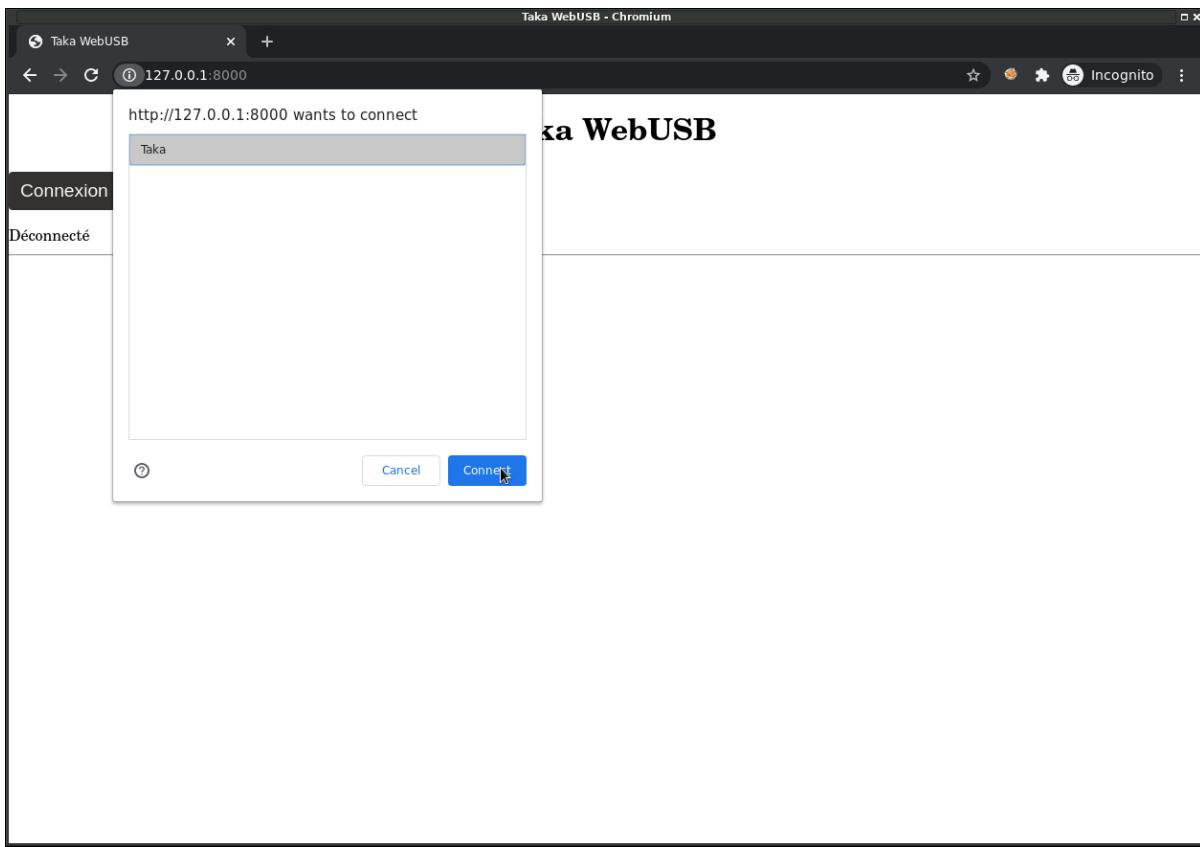
```
(Linux) Install Udev rules
=====
Write the rule in:
/etc/udev/rules.d/52-taka.rules
-----
ACTION=="add", SUBSYSTEM=="usb", ATTRS{idVendor}=="ffff", ATTRS{idProduct}=="1234", MODE=
↪"0664", GROUP="plugdev"
-----
Reload the rules:
# udevadm control --reload-rules && udevadm trigger

(Windows) Install WinUSB driver
=====
https://zadig.akeo.ie/
Assign "WinUSB" driver to the device.

Run the HTTP server for testing
=====
$ python3 -m http.server

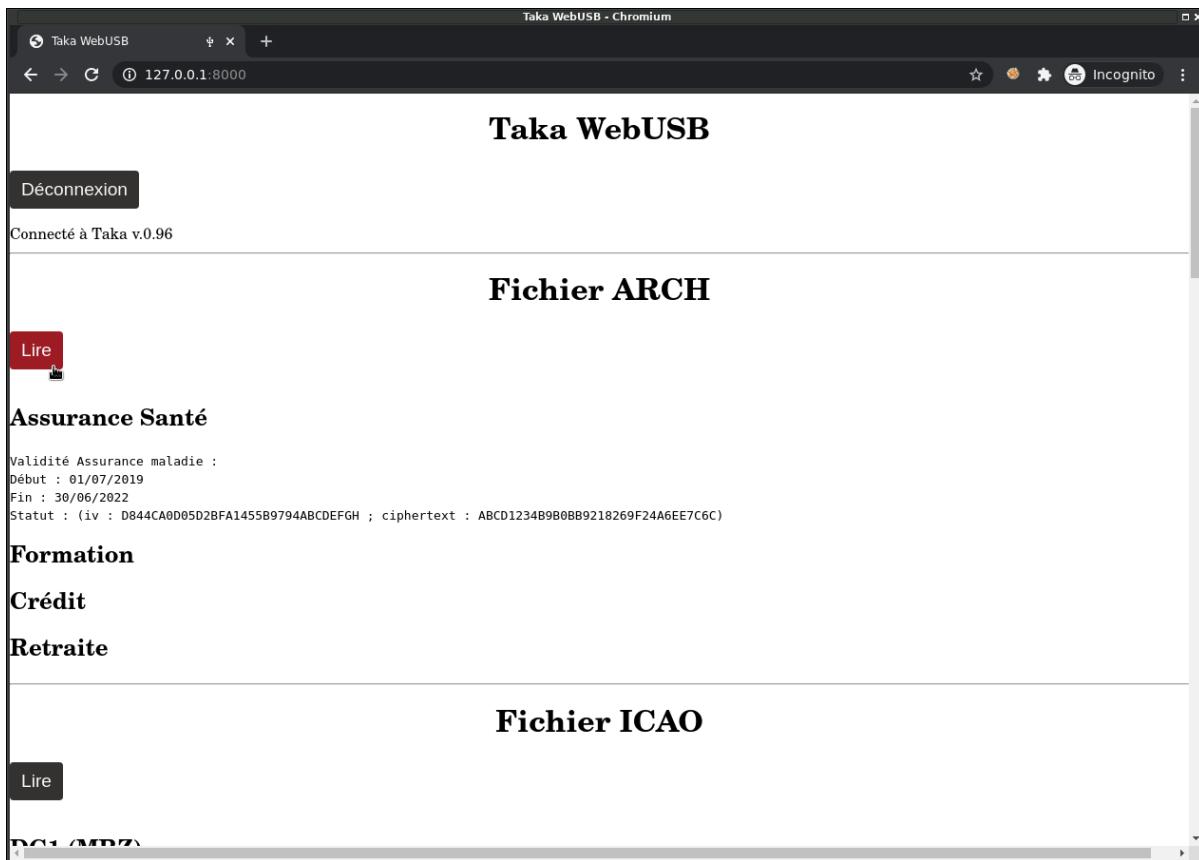
In a Web browser (Chrome), connect to:
http://127.0.0.1:8000/
```

3.2.2 Connection



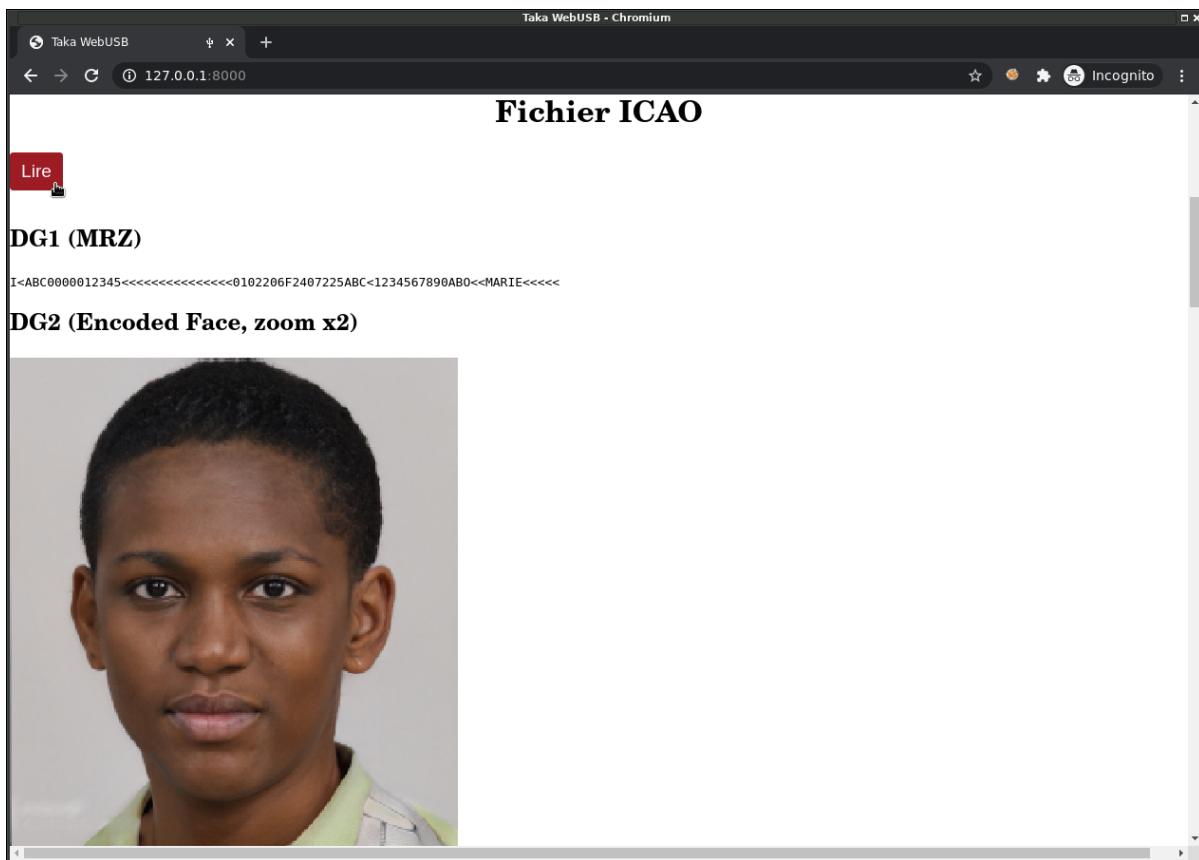
3.2.3 Lecture des fichiers ARCH

Lorsqu'une carte d'identité provisionnée avec l'application ARCH est insérée dans le boitier Taka, le contenu des fichiers ARCH peut être lu via l'API:



3.2.4 Lecture des fichiers ICAO

Lorsqu'une carte d'identité provisionnée avec l'application ICAO (eTravel) est insérée dans le boîtier Taka, le contenu des fichiers ICAO DG1 à DG16 peut être lu via l'API:



3.2.5 Lecture des journaux

A tout moment le contenu des logs stockés dans le boîtier Taka (sur carte SD) peut être lu via l'API. La requête doit préciser la date voulue (1 jour à la fois). Voir la section *Journaux* pour le descriptif du format journal.

The screenshot shows a Chromium browser window titled "Taka WebUSB - Chromium" with the URL "127.0.0.1:8000". The page is titled "Logs". It features a search bar with the placeholder "Date: 02/02/2021" and a red "Afficher" button. Below is a table with the following data:

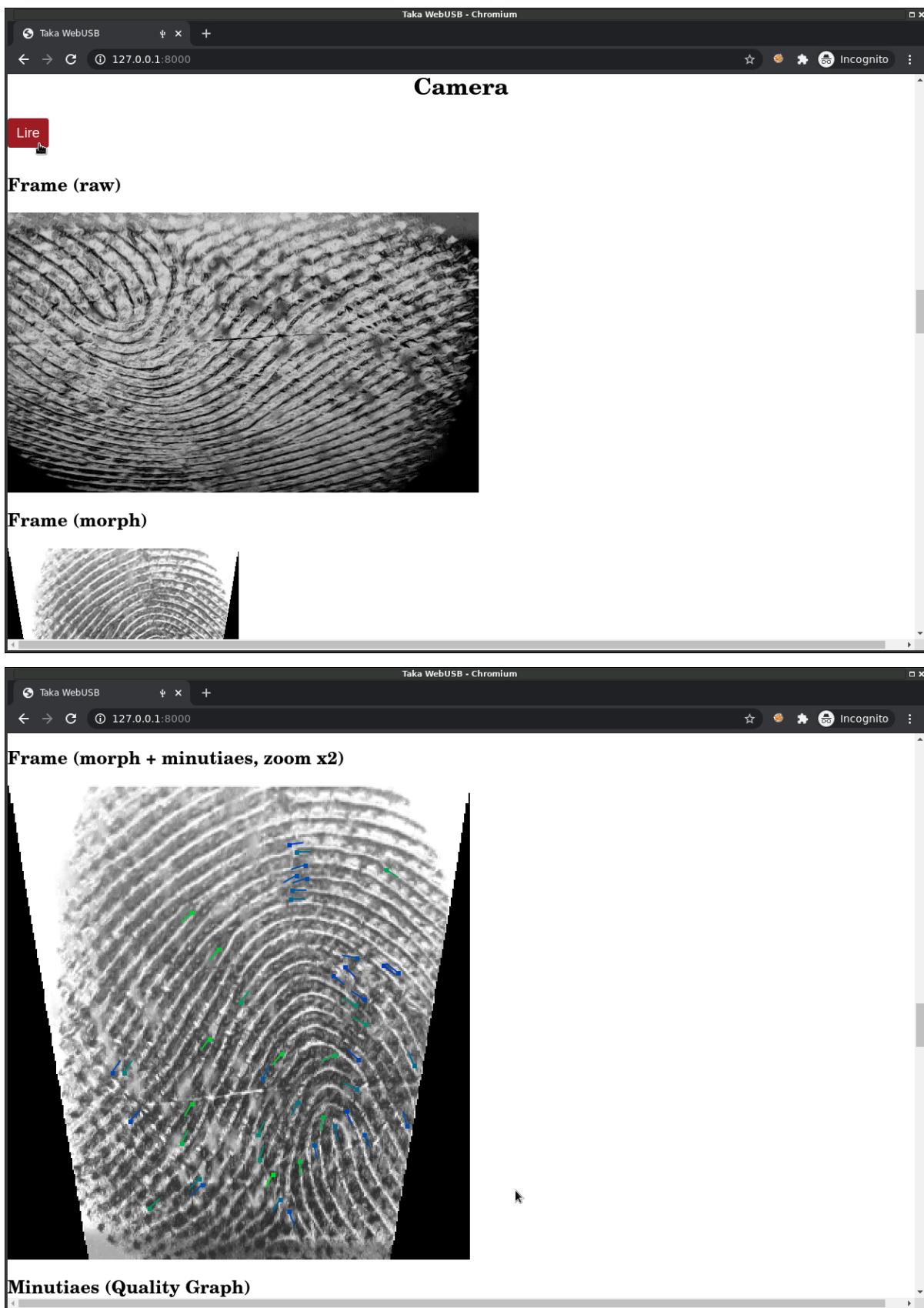
Nom Prénom	Numéro de carte	Date de naissance	Sexe	Authentification	Eligibilité	Date de passage	Status	Résidence
MARTIN ALFRED	000002345	90-07-10	H	R	V	2021-02-02T14:21:10.000		
DECHAMPS MARCELLE	000009999	89-02-15	F	V	R	2021-02-02T14:33:02.000		
DUBOIS PIERRE	000001234	01-02-20	H	R	R	2021-02-02T14:39:57.000		
BIO MARIE	000005678	10-11-20	F	V	R	2021-02-02T14:40:26.000		

3.2.6 Lecture d'empreinte

L'empreinte digitale capturée par la caméra optique est accessible par l'API. Seule la dernière image capturée est disponible, une nouvelle tentative écrasant l'image précédente.

3 images sont récupérables:

- **Frame Raw:** image reçue du capteur sans aucun traitement
- **Frame Morph:** image après traitements de correction de déformations et d'aberration
- **Minutiae:** empreinte + représentation des minutiae trouvés



3.3 Détails du protocole API USB

Le protocole est implémenté sous la forme de commandes/réponses. Les commandes sont toujours émises par le host (PC), auxquelles le device (Taka) répond. Taka n'émet pas de messages USB sans être sollicité au préalable par une commande reçue du host.

Implémentation de référence: (Javascript) voir <https://github.com/lambdacconcept/taka/blob/master/utils/takaweb/index.js>

Les commandes respectent le format suivant:

- **1er octet** (obligatoire): numéro de la commande
- **octet suivants** (optionnels): paramètres de la commande

Liste des commandes:

Nom	Numéro	Paramètres
Get Version	0x00	
Get ARCH	0x01	1 octet
Get ICAO	0x02	1 octet
Get Logs	0x03	8 octets
Get Frame Raw	0x04	
Get Frame Morph	0x05	
Get Minutiae	0x06	

3.3.1 Get Version

- **Description:** Lit la version actuelle du firmware programmé sur le boitier Taka.
- **Numéro:** 0x00
- **Paramètres:** Aucun
- **Réponse:** La version sous forme de chaîne de caractères (NULL terminated). Exemple: “v.0.96”

3.3.2 Get ARCH

- **Description:** Lit un fichier ARCH présent sur la carte d'identité. Le numéro du conteneur à lire est indiqué par un paramètre. Cette commande renvoie un paquet USB de taille vide (ZLP) lorsqu'aucune carte n'est insérée ou que le conteneur n'existe pas.
- **Numéro:** 0x01
- **Paramètres:** 1 octet, le numéro du fichier ARCH à lire (0x00-0x03)
- **Réponse:** Renvoie le contenu du fichier ARCH demandé

3.3.3 Get ICAO

- **Description:** Lit un fichier ICAO présent sur la carte d'identité. Le numéro du conteneur à lire est indiqué par un paramètre. Cette commande renvoie un paquet USB de taille vide (ZLP) lorsqu'aucune carte n'est insérée ou que le conteneur n'existe pas.
- **Numéro:** 0x02
- **Paramètres:** 1 octet, le numéro du fichier DG à lire (0x00-0x0f)
- **Réponse:** Renvoie le contenu du fichier ICAO demandé

3.3.4 Get Logs

- **Description:** Récupère les évènements ayant eu lieu le jour passé en paramètre.
- **Numéro:** 0x03
- **Paramètres:** 8 caractères ASCII au format “YYYYMMDD”, correspondant à la date du jour voulu
- **Réponse:** Le contenu du fichier journal pour le jour demandé. Voir la section *Journaux* pour le descriptif du format journal.

3.3.5 Get Frame Raw

- **Description:** Récupère la dernière image de l'empreinte capturée par la caméra. Il s'agit de l'image brute reçue du capteur sans aucun traitement.
- **Numéro:** 0x04
- **Paramètres:** Aucun
- **Réponse:** L'image au format binaire (framebuffer 8bit grayscale, 640x380 px)

3.3.6 Get Frame Morph

- **Description:** Récupère la dernière image de l'empreinte après les traitements de correction de déformations et d'aberration.
- **Numéro:** 0x05
- **Paramètres:** Aucun
- **Réponse:** L'image au format binaire (framebuffer 8bit grayscale, 314x322 px)

3.3.7 Get Minutiae

- **Description:** Récupère l'ensemble des minutiae extraits à partir de la dernière image capturée par la caméra. Cette commande renvoie un paquet USB de taille vide (ZLP) lorsque l'extraction a échoué (image de mauvaise qualité, doigt non présent, etc.)
- **Numéro:** 0x06
- **Paramètres:** Aucun
- **Réponse:** Renvoie les minutiae au format ISO/IEC 19794-2:2005 (Record Format, FMR)

CHAPTER
FOUR

JOURNAUX

Chaque opération d'authentification d'un bénéficiaire est enregistrée avec la date et l'heure de passage (UTC) dans un journal stocké dans une mémoire de type carte micro SD, au format CSV.

La carte SD contient un système de fichier de type FAT32, organisé de la manière suivante:

- Un dossier par année (2020, 2021, etc.)
- Pour chaque année, un fichier texte par jour (20210101.TXT, 20210102.TXT, etc.)

Exemple:

```
nsh> ls LOGS/  
2020/  
2021/  
  
nsh> ls LOGS/2021/  
20210128.TXT  
20210129.TXT  
20210202.TXT  
  
nsh> cat LOGS/2021/20210202.TXT  
MARTIN<<ALFRED,000002345,90-07-10,H,R,V,2021-02-02T14:21:10.000,,  
DESCHAMPS<<MARCELLE,000009999,89-02-15,F,V,R,2021-02-02T14:33:02.000,,  
DUBOIS<<PIERRE,000001234,01-02-20,H,R,R,2021-02-02T14:39:57.000,,  
BIO<<MARIE,000005678,10-11-20,F,V,R,2021-02-02T14:40:26.000,,
```

4.1 Format

Les logs sont stockés au format texte (CSV), et contiennent les colonnes suivantes:

- **Nom/Prénom:** Format NOM<<PRENOM (Comme sur le MRZ de la carte)
- **Numéro de carte:** Exemple: 000001234
- **Date de naissance:** Format yy-MM-dd. (L'année est stockée seulement sur 2 chiffres, le format long 4 chiffres n'étant pas provisionné dans les cartes fournies)
- **Sexe:** H = Homme, F = Femme
- **Authentification:** Résultat de la vérification d'empreinte (V = Valide, R = Rejet)

- **Eligibilité:** Résultat d'éligibilité à l'assurance maladie (V = Valide, R = Rejet)
- **Date de passage:** Format yyyy-MM-dd'T'HH:mm:ss.000 (Heure UTC)
- **Statut:** (Non implémenté, le statut de pauvreté étant chiffré)
- **Résidence:** (Non implémenté, l'adresse n'étant pas provisionnée dans les cartes fournies)

La fonction d'export des logs est visible ici:

<https://github.com/lambdacconcept/taka-apps/blob/master/examples/taka/logs.c>

4.2 Accès

Les logs sont accessibles par l'API, voir [Lecture des journaux](#)

Nom Prénom	Numéro de carte	Date de naissance	Sexe	Authentification	Eligibilité	Date de passage	Status	Résidence
MARTIN ALFRED	000002345	90-07-10	H	R	V	2021-02-02T14:21:10.000		
DESCHAMPS MARCELLE	000009999	89-02-15	F	V	R	2021-02-02T14:33:02.000		
DUBOIS PIERRE	000001234	01-02-20	H	R	R	2021-02-02T14:39:57.000		
BIO MARIE	000005678	10-11-20	F	V	R	2021-02-02T14:40:26.000		

Note: Bien sur les logs sont aussi accessibles en retirant la carte SD du boîtier Taka.

MISE À JOUR

5.1 Mode DFU

Le lecteur Taka peut être mis à jour via USB. Prérequis:

- **Etape 1:** Connecter Taka à un PC à l'aide d'un câble USB.
- **Etape 2:** Démarrer Taka en **mode DFU** (Device Firmware Update).

Pour cela, il faut maintenir le bouton DFU enfoncé à l'aide d'une aiguille ou autre objet pointu, tout en démarrant Taka en actionnant l'interrupteur On/Off.

Le bouton DFU est situé sur le côté gauche du boitier:



5.2 WebDFU

Une application web est fournie, utilisable depuis un navigateur Web (Chrome).

5.2.1 Installation

Voir le README <https://github.com/lambdacconcept/taka/tree/master/utils/takadfu>

```
(Linux) Install Udev rules
=====
Write the rule in:
/etc/udev/rules.d/45-stdfu-permissions.rules
-----
ACTION=="add", SUBSYSTEM=="usb", ATTRS{idVendor}=="0483", ATTRS{idProduct}=="df11", MODE=←"0664", GROUP="plugdev"
-----

Reload the rules:
# udevadm control --reload-rules && udevadm trigger

(Windows) Install WinUSB driver
=====

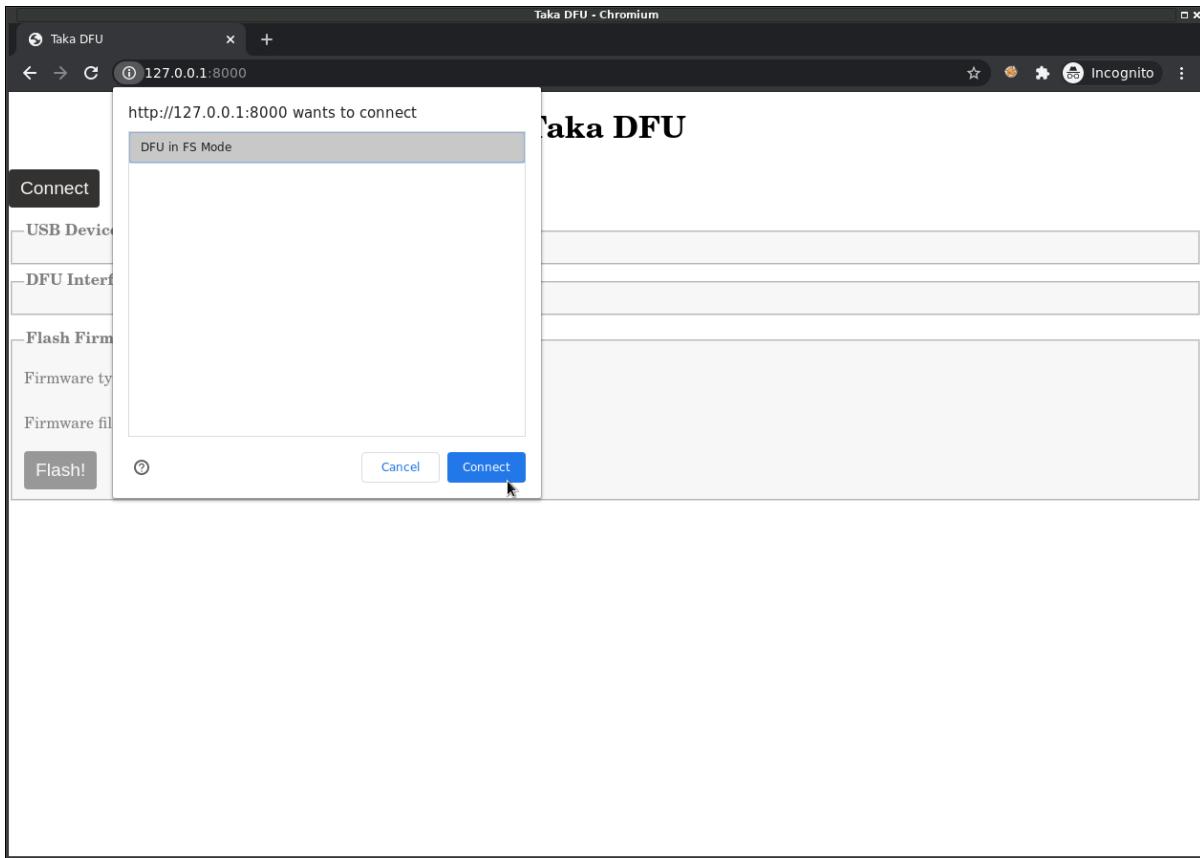
https://zadig.akeo.ie/
Assign "WinUSB" driver to the device.

Run the HTTP server for testing
=====

$ python3 -m http.server

In a Web browser (Chrome), connect to:
http://127.0.0.1:8000/
```

5.2.2 Connection

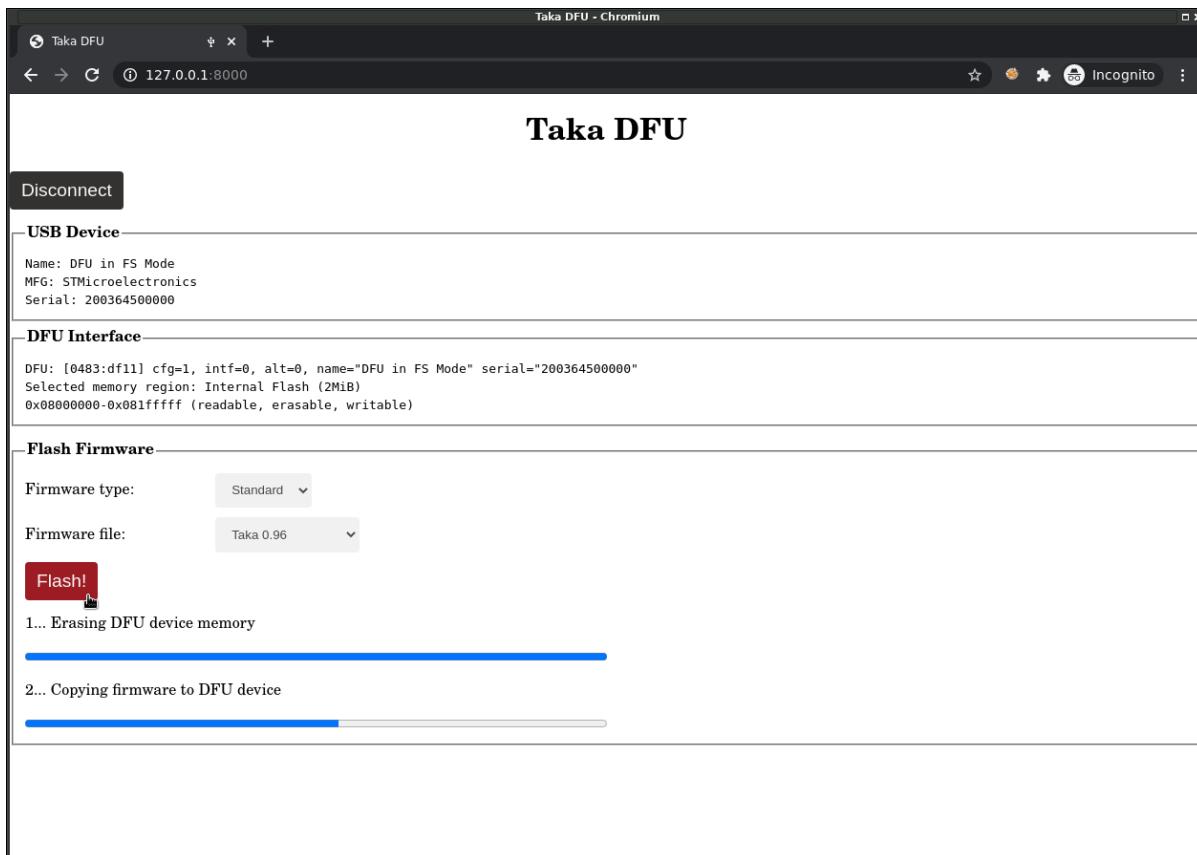
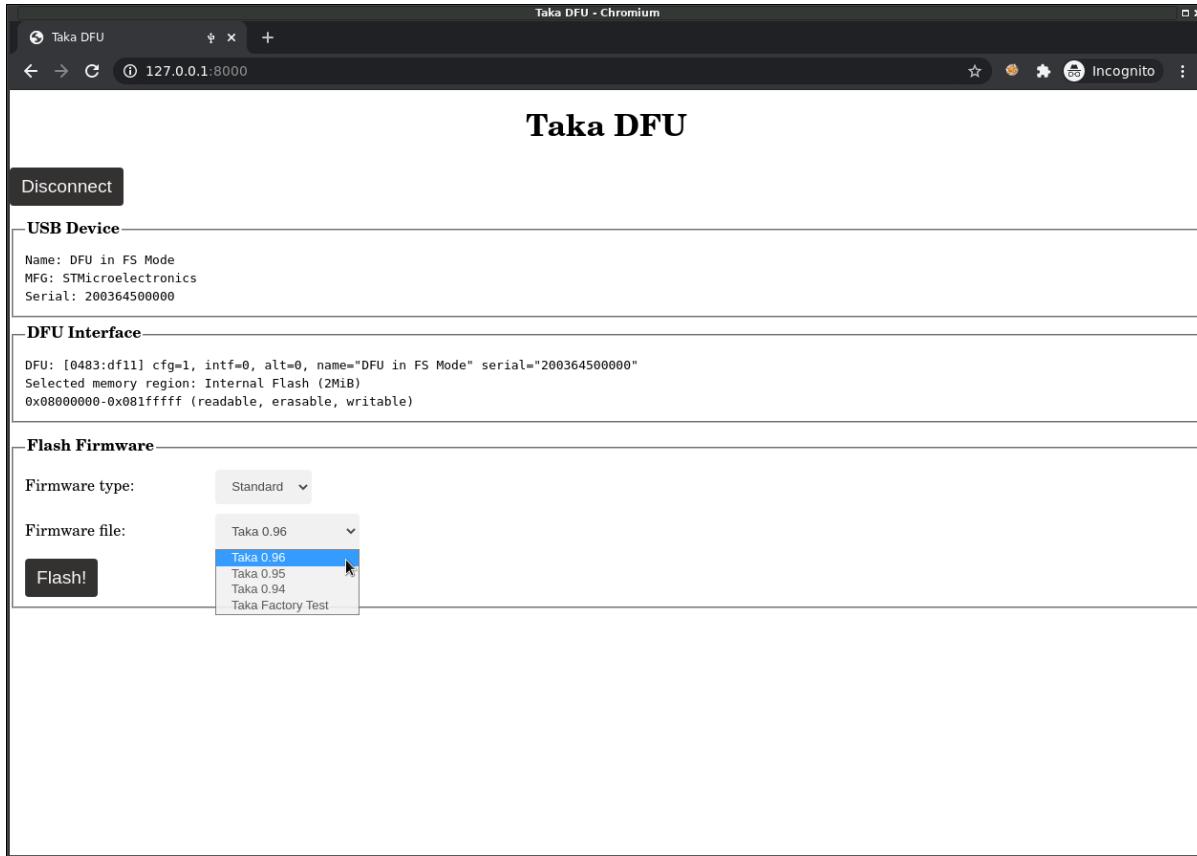


Si le périphérique USB DFU n'apparaît pas dans la liste, vérifiez que Taka est bien démarré en *Mode DFU*.

5.2.3 Selection du firmware

Les firmwares précompilés sont déjà inclus dans l'application web, il suffit de sélectionner la version voulue dans la liste déroulante et de cliquer sur le bouton “Flash!”.

LambdaConcept Taka



La mise à jour est terminée.

5.3 DFU-Util

Note: Sous linux, il est aussi possible d'utiliser dfu-util pour faire la mise à jour en ligne de commande plutôt que par l'application web

```
apt-get install dfu-util  
dfu-util -d 0483:df11 -a 0 -D taka.bin --dfuse-address 0x08000000
```

DIAGNOSTICS

6.1 Erreurs courantes

6.1.1 Taka ne s'allume pas

Taka ne s'allume pas ou s'éteint aussitôt.

- La batterie est déchargée, charger la batterie.
- Vérifier que le firmware est correctement programmé.
- Vérifier ou remplacer la *Carte mère*

6.1.2 Le doigt n'est pas détecté

La présence du doigt n'est pas détectée.

- Nettoyer la vitre de la caméra optique.
- Vérifier ou remplacer le *Module caméra/touch*

6.1.3 La date est invalide

Lorsque la batterie est débranchée, la date et l'heure doivent être resynchronisés via un serveur de temps (NTP). La synchronisation est automatique lorsque le boîtier Taka est connecté au réseau (internet). (Adresse du serveur NTP: pool.ntp.org)

6.2 Firmware de test

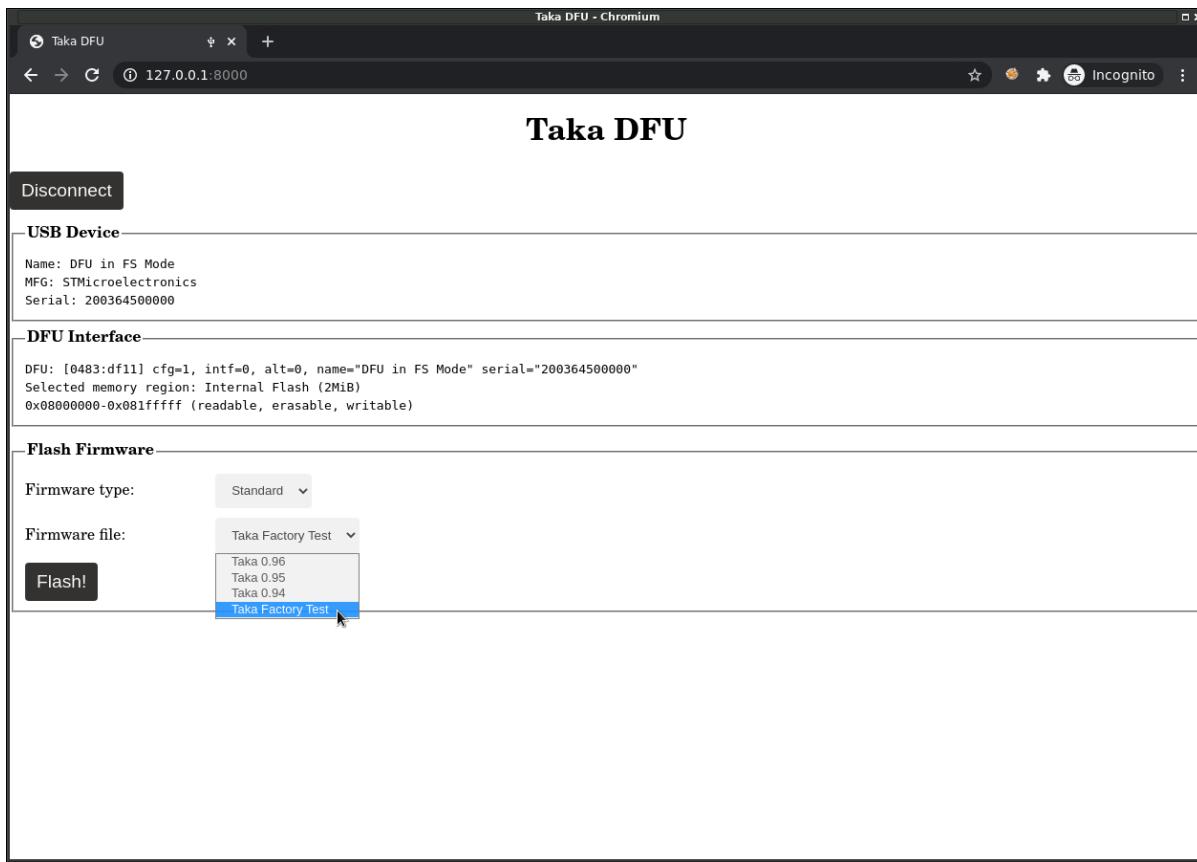
En cas d'autres dysfonctionnements de l'appareil Taka, il est possible de passer en mode diagnostic afin de déterminer l'origine du problème et de remplacer la pièce défectueuse le cas échéant.

Pour cela, il faut:

- 1. Connecter Taka à un PC via un câble USB.
- 2. Connecter Taka au réseau via un câble Ethernet.
- 3. Programmer le firmware de test usine et redémarrer Taka.

- 4. Suivre la procédure affichée sur l'écran et utiliser l'application de test depuis un navigateur web.

On commence par programmer Taka avec le firmware de test usine nommé “Taka Factory Test”, via le mode DFU et l'application WebDFU comme expliqué dans la section *Mise à jour*.



Ensuite, redémarrer le boitier Taka.

En cas d'erreur:

- Vérifier ou remplacer la *Carte mère*
-

6.3 Procédure de test

La procédure de test se déroule en étapes, chaque étape vérifiant un élément du système. En cas de succès, le test passe à l'étape suivante. En cas d'échec, la même étape recommence en boucle. Les instructions à suivre s'affichent sur l'écran LCD. Certaines étapes sont automatisées, d'autres requièrent une intervention humaine (par exemple pour vérifier que les LEDs sont bien allumées).

6.3.1 1. Ecran LCD

Dès le démarrage, l'écran affiche le message suivant:



Action humaine:

- Vérifier le bon affichage du message sur l'écran LCD
- Vérifier la présence du rétroéclairage

En cas d'erreur:

- Vérifier que Taka est allumé (Bouton On/Off)
 - Vérifier que Taka est alimenté (Par USB)
 - Vérifier ou remplacer l'*Module écran*
-

6.3.2 2. LEDs



La LED caméra s'allume en bleu fixe et la LED située sur le dessus du boîtier change successivement de couleur (Rouge, Vert, Bleu, Jaune, Magenta, Cyan).

Action humaine:

- Vérifier l'allumage de la LED caméra en bleu
- Vérifier l'allumage de la LED boîtier en multicolore
- Passer à l'étape suivante en touchant le capteur caméra.

En cas d'erreur:

- Vérifier ou remplacer le *Module LED/antenne*
 - Vérifier ou remplacer le *Module caméra/touch*
-

6.3.3 3. Buzzer



Le buzzer émet des sons à intervalles réguliers.

Action humaine:

- Vérifier que le buzzer émet bien un son.
- Passer à l'étape suivante en touchant le capteur caméra.

En cas d'erreur:

- Vérifier ou remplacer la *Carte mère*
-

6.3.4 4. Mémoire SDRAM

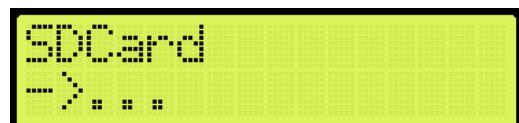


Test automatique: validation du bon fonctionnement du composant mémoire SDRAM présent sur la carte mère.

En cas d'erreur:

- Vérifier ou remplacer la *Carte mère*
-

6.3.5 5. Carte SD

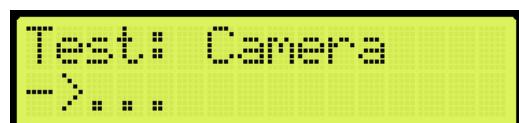


Test automatique: vérification de la présence et du bon fonctionnement de la carte micro SD. Si nécessaire, la carte sera automatiquement formatée avec un système de fichiers de type FAT32.

En cas d'erreur:

- Vérifier ou remplacer la Carte SD sur la *Carte mère*
-

6.3.6 6. Caméra



Test automatique: vérification de la présence et du bon fonctionnement du capteur caméra optique.

En cas d'erreur:

- Vérifier ou remplacer le *Module caméra/touch*
-

6.3.7 7. USB

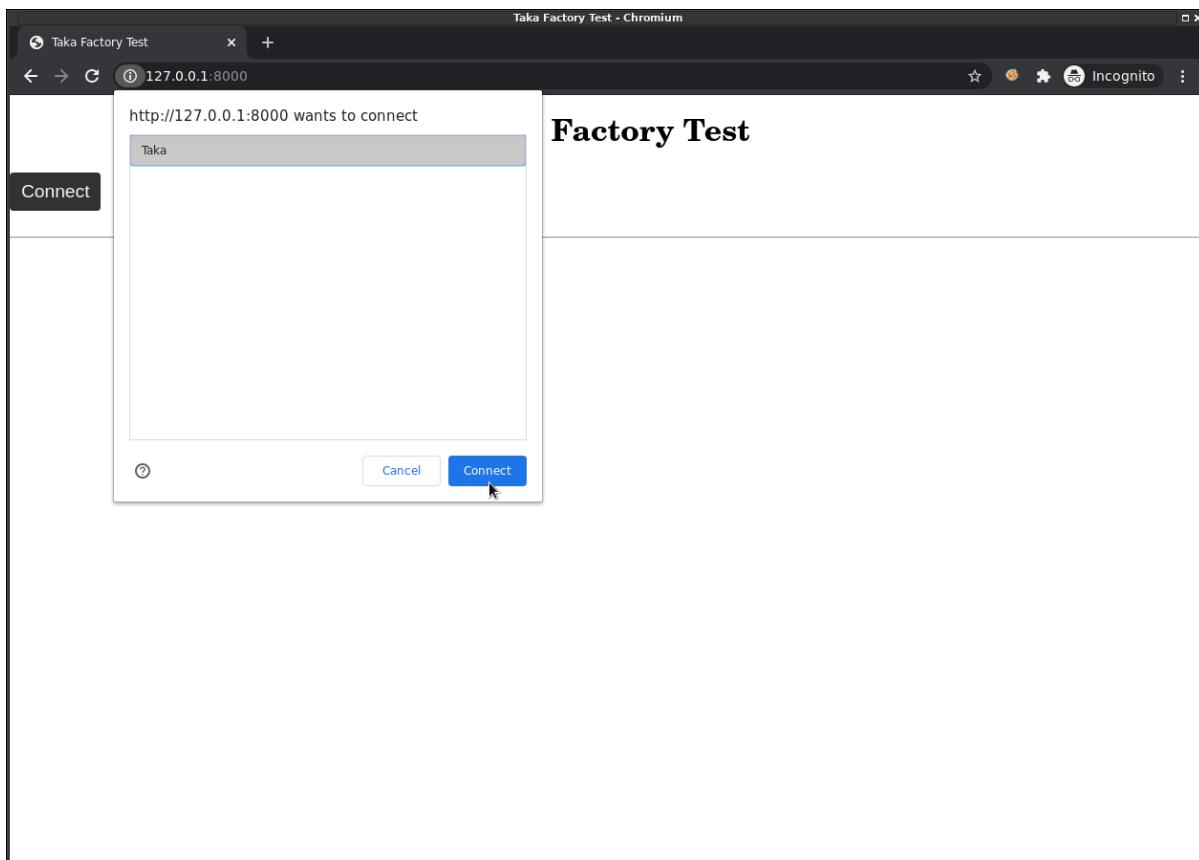


Taka est en attente d'une connexion USB.

Action humaine:

- Connecter Taka à un PC via un câble USB.
- Utiliser l'application web dédiée aux tests, nommée "takatest", depuis un navigateur Web (Chrome)
- Cliquer sur "Connect".

Voir le README <https://github.com/lambdacconcept/taka/tree/master/utils/takatest>



En cas d'erreur:

- Vérifier ou remplacer le câble USB.
- (Sous linux): vérifier les bonnes permissions udev pour webusb.
- Vérifier ou remplacer la *Carte mère*

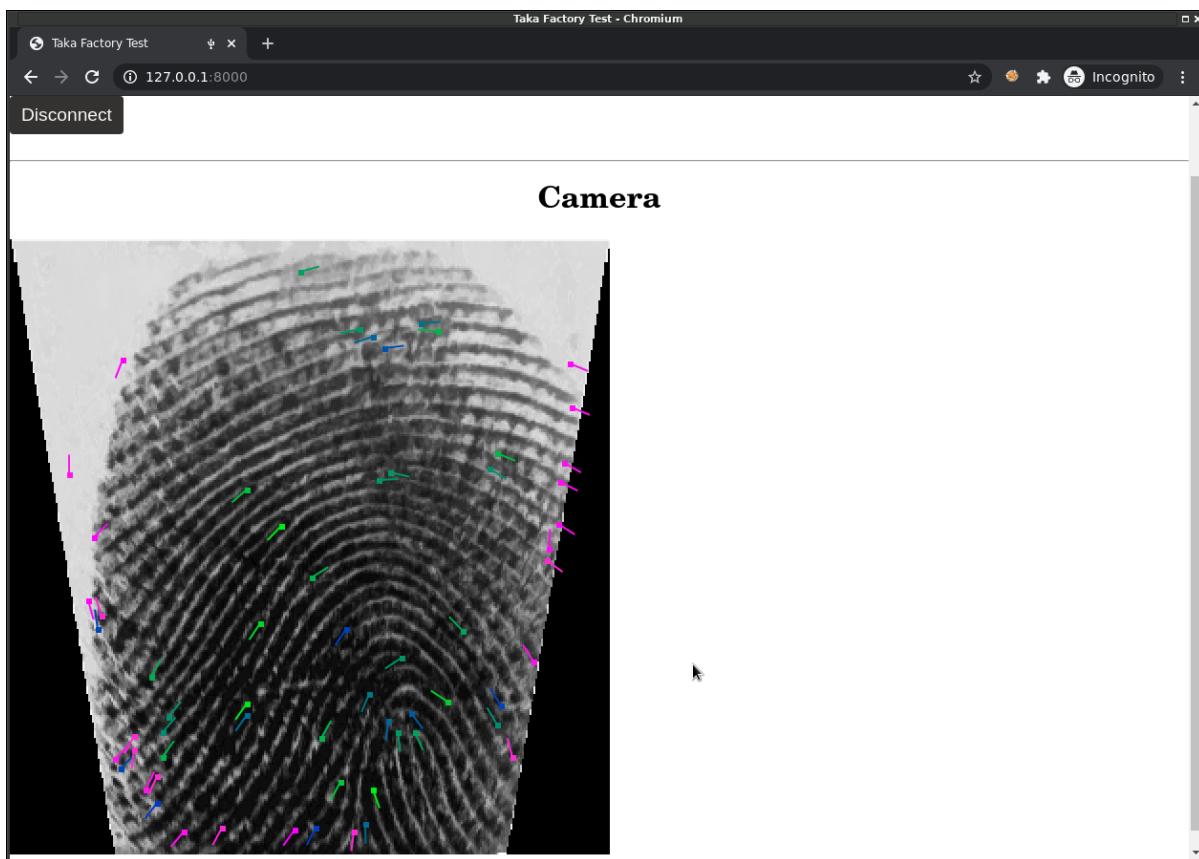
6.3.8 8. Empreinte digitale



La caméra capture des images en boucle et les envoie au PC par l'intermédiaire de l'API USB.

Action humaine:

- Placer le doigt sur le capteur.
- Vérifier dans l'interface web la bonne qualité de l'image.
- Le test passe automatiquement à l'étape suivante lorsque l'image est de qualité suffisante.



En cas d'erreur:

- Nettoyer la vitre de la caméra optique.
- Vérifier ou remplacer le [Module caméra/touch](#)

6.3.9 9. Carte à puce (Contact)



Taka est en attente de l'insertion d'une carte à puce dans son connecteur pour initier une communication (ATR).

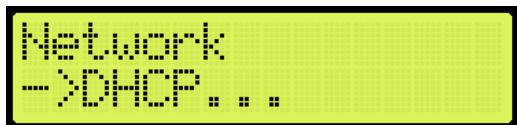
Action humaine:

- Insérer une carte d'identité.

En cas d'erreur:

- Vérifier le sens d'insertion (puce vers soi)
 - Vérifier ou remplacer le *Module carte à puce*
-

6.3.10 10. Réseau



Taka se connecte au réseau Ethernet et récupère une adresse IP par DHCP.

Action humaine:

- Connecter Taka au réseau via un câble Ethernet.

En cas d'erreur:

- Vérifier ou remplacer le câble Ethernet.
 - Vérifier la présence d'un serveur DHCP.
 - Vérifier ou remplacer la *Carte mère*
-

6.3.11 11. Date/Heure



Test automatique: configuration de la date et heure.

La synchronisation est automatique lorsque le boîtier Taka est connecté au réseau (internet). (Adresse du serveur NTP: pool.ntp.org)

En cas d'erreur:

- Vérifier la connexion internet.
- Vérifier ou remplacer la *Carte mère*

6.3.12 12. Batterie



Test automatique: test de charge de la batterie (par USB).

En cas d'erreur:

- Connecter Taka au courant via un câble USB.
 - Vérifier la présence de la batterie.
-

6.3.13 Succès

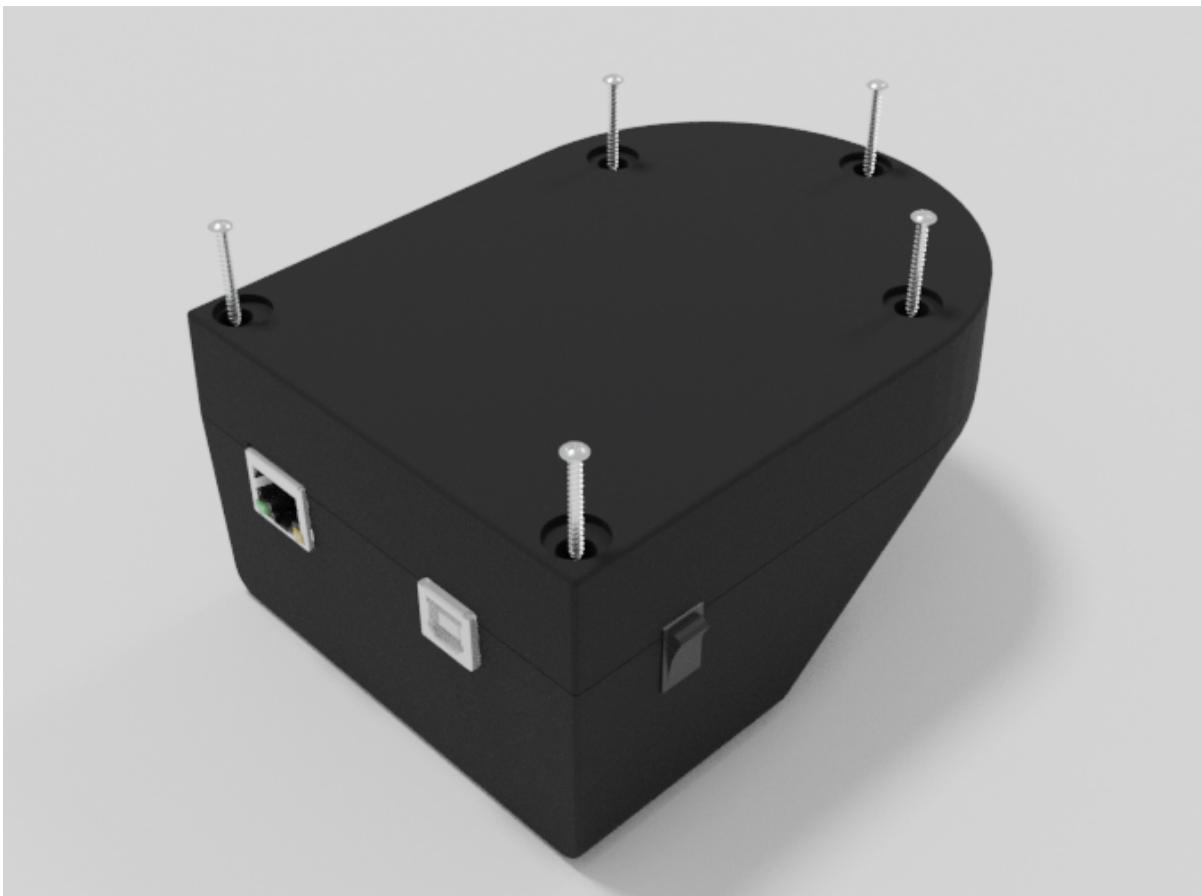


Le test est terminé, le boitier Taka est complètement fonctionnel. Il faut maintenant le reprogrammer avec le firmware de production, comme expliqué dans la section *Mise à jour*.

BOÎTIER

7.1 Démontage

Ouvrir le boîtier en retirant les patins autocollants en caoutchouc puis en dévissant les 5 vis longues.



L'appareil est constitué d'une carte mère et de 4 cartes modules reliées par des fils, et d'une batterie.

- *Carte mère*
- *Module caméra/touch*
- *Module LED/antenne*
- *Module carte à puce*

- *Module écran*
-

7.2 Batterie

L'emplacement batterie est situé sous la carte mère. Les fils de la batterie doivent remonter en passant dans l'encoche proche de l'interrupteur, puis être connectés à CN9 : BATTERY de la carte mère.

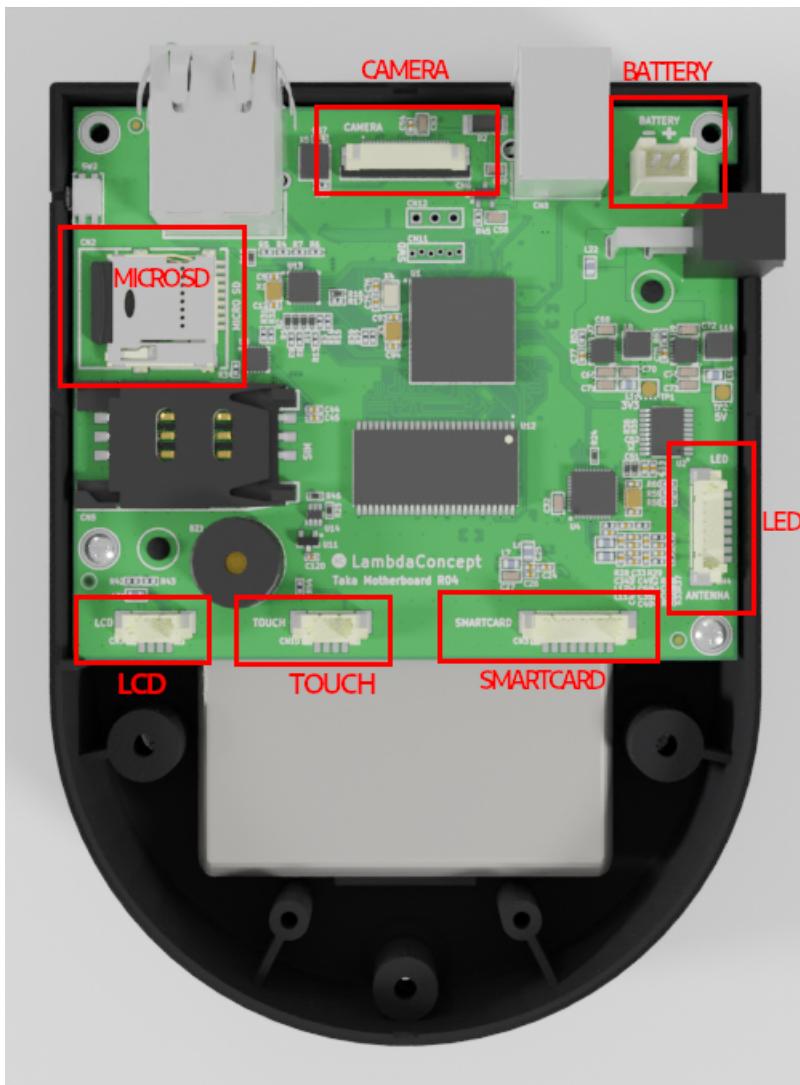


7.3 Carte mère

La carte mère est vissée dans la partie inférieure du boîtier par 2 vis (en bas).

La carte mère doit être reliée aux autres modules par les groupes de fils (4 ou 8 pins) et par la nappe flex.

- **Connecteur batterie:** CN9: BATTERY
- **Connecteur nappe flexible:** CN6: CAMERA (Flex)
- **Connecteur touch:** CN10: TOUCH (4 pins)
- **Connecteur LED:** CN4: LED ANTENNA (8 pins)
- **Connecteur carte à puce:** CN3: SMARTCARD (8 pins)
- **Connecteur écran:** CN7: LCD (4 pins)
- **Connecteur carte SD:** CN2: MICRO SD (1 GB)

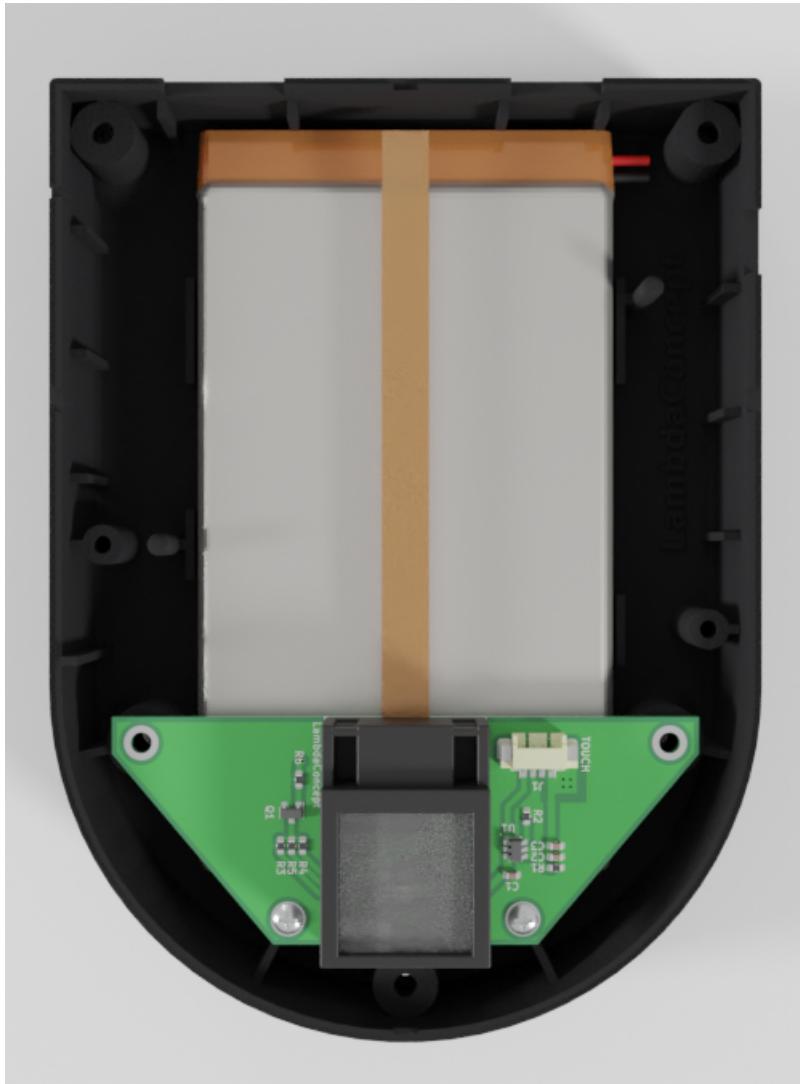


7.4 Module caméra/touch

Ce module contient 2 capteurs:

- Caméra optique pour capturer l'empreinte digitale.
- Capteur tactile (touch) pour détecter la présence du doigt.
- **Connecteur nappe flexible:** CN6: CAMERA (Flex)
- **Connecteur carte mère:** CN10: TOUCH (4 pins)

Il est vissé dans la partie inférieure du boîtier par 2 vis (en bas).



7.5 Module LED/antenne

Ce module se visse sur la partie supérieure du boîtier (2 vis).

- **Connecteur carte mère:** CN4: LED ANTENNA (8 pins)



7.6 Module carte à puce

Ce module se glisse et se visse sur la partie supérieure du boîtier (2 ou 3 vis + rondelles).

- **Connecteur carte mère:** CN3: SMARTCARD (8 pins)



7.7 Module écran

Ce module doit être vissé dans la partie supérieure du boîtier (4 petites vis).

- **Connecteur carte mère:** CN7: LCD (4 pins)



DÉVELOPPEMENT

Le firmware de Taka est basé sur le système d'exploitation temps réel NuttX.

8.1 Console de debug

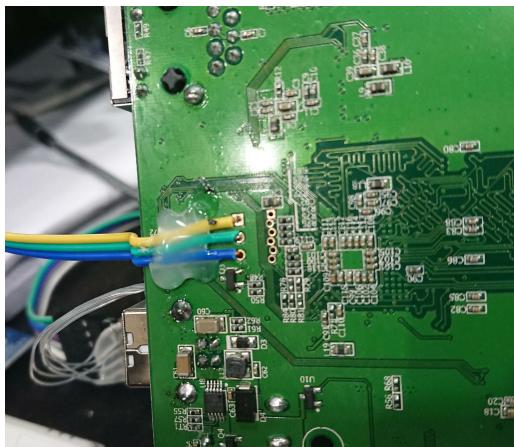
La console de debug de Taka est accessible par le connecteur 3 pins CN12 situé sur la carte mère. Il faut y connecter un câble de type debug UART. (Par exemple le câble LambdaConcept Jtag+Serial)

- Pin 1 (Carré): Taka TX
- Pin 2: Taka RX
- Pin 3: GND

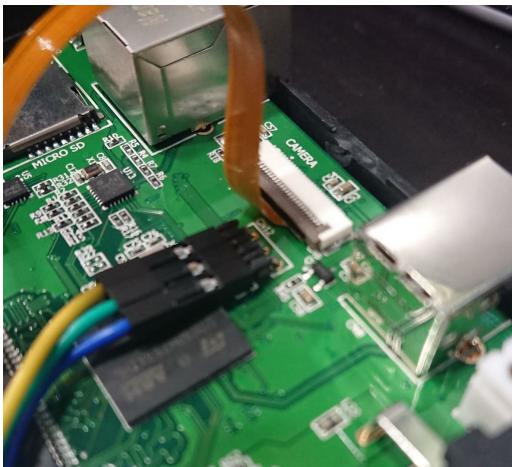


Exemples de branchement

- En soudant les fils au dos de la carte mère:

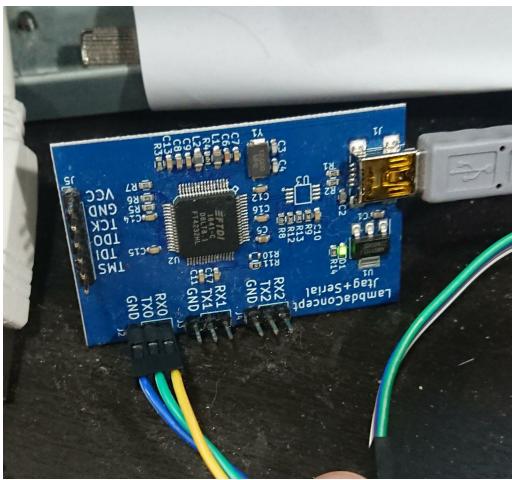


- Ou en faisant contact avec des pins coudés:



Câble USB UART

- RX0 <— Taka TX
- TX0 —> Taka RX
- GND <—> Taka GND



8.1.1 Minicom

Côté PC, le câble de debug UART est vu en tant que périphérique TTY USB. Si les fils ont été branchés sur le canal TX0/RX0, le périphérique côté PC sera en principe /dev/ttyUSB1.

On peut vérifier en laissant tourner dmesg tout en branchant le câble de debug:

```
# dmesg -w
[47587.720861] usb 1-2.4.2: new high-speed USB device number 29 using xhci_hcd
[47587.861696] usb 1-2.4.2: New USB device found, idVendor=0403, idProduct=6011, ↵
↳ bcdDevice= 8.00
[47587.861703] usb 1-2.4.2: New USB device strings: Mfr=1, Product=2, SerialNumber=0
[47587.861706] usb 1-2.4.2: Product: Quad RS232-HS
[47587.861710] usb 1-2.4.2: Manufacturer: FTDI
[47587.871543] ftdi_sio 1-2.4.2:1.0: FTDI USB Serial Device converter detected
[47587.871600] usb 1-2.4.2: Detected FT4232H
```

(continues on next page)

(continued from previous page)

```
[47587.871790] usb 1-2.4.2: FTDI USB Serial Device converter now attached to ttyUSB0
[47587.872008] ftdi_sio 1-2.4.2:1.1: FTDI USB Serial Device converter detected
[47587.872056] usb 1-2.4.2: Detected FT4232H
[47587.872249] usb 1-2.4.2: FTDI USB Serial Device converter now attached to ttyUSB1
[47587.872452] ftdi_sio 1-2.4.2:1.2: FTDI USB Serial Device converter detected
[47587.872495] usb 1-2.4.2: Detected FT4232H
[47587.872663] usb 1-2.4.2: FTDI USB Serial Device converter now attached to ttyUSB2
[47587.872908] ftdi_sio 1-2.4.2:1.3: FTDI USB Serial Device converter detected
[47587.872968] usb 1-2.4.2: Detected FT4232H
[47587.873122] usb 1-2.4.2: FTDI USB Serial Device converter now attached to ttyUSB3
```

Ici, on a:

- **ttyUSB0:** JTAG (Inutilisé ici)
- **ttyUSB1:** RX0/TX0 <—
- **ttyUSB2:** RX1/TX1
- **ttyUSB3:** RX2/TX2

On s'y connecte avec minicom:

```
$ minicom -s -D /dev/ttyUSB1

-> Serial port setup
-> Hardware Flow Control : No
```

Console nuttx:

```
Welcome to minicom 2.8

OPTIONS: I18n
Compiled on Jan  9 2021, 12:42:45.
Port /dev/ttyUSB1, 19:17:31

Press CTRL-A Z for help on special keys

ADE
taka [5:100]

NuttShell (NSH) NuttX-9.0.0
nsh>
```

8.2 Telnet

Le prompt NuttX est aussi accessible en Telnet (Pas besoin de câble de debug mais plus complexe).

1. Brancher un câble Ethernet à Taka
2. Redémarrer Taka
3. Taka prend une adresse IP par DHCP

Pour trouver l'IP de Taka:

- Soit par l'interface du routeur DHCP
- Soit en scannant le réseau (TCP/23): sudo nmap -sS -T5 -n -p 23 192.168.0.0/24

```
$ telnet 192.168.0.2

Trying 192.168.0.57...
Connected to 192.168.0.57.
Escape character is '^]'.

Welcome to NuttShell(NSH) Telnet Server...
login: taka
password: takatelnet

User Logged-in!

NuttShell (NSH) NuttX-10.2.0
nsh> ps
  PID PRI POLICY   TYPE   NPX STATE     EVENT      SIGMASK    STACK COMMAND
    0  0  FIFO     Kthread N-- Ready      00000000 0000000 Idle Task
    1 224 FIFO     Kthread --- Waiting  Signal      00000000 002028 hpwork
    2 100 FIFO     Kthread --- Ready      00000000 002028 lpwork
    3 100 FIFO     Task    --- Waiting  Semaphore  00000000 002028 init
    4 100 FIFO     Task    --- Waiting  Semaphore  00000000 002004 Telnet daemon
→ 0x38003790
    5 100 RR       Task    --- Waiting  Semaphore  00000000 131052 taka
    7 100 FIFO     Task    --- Waiting  Signal      00000000 002020 NTP daemon
    8 100 FIFO     Kthread --- Waiting  Semaphore  00000000 000996 telnet_io
    9 100 FIFO     Task    --- Running      00000000 002020 Telnet session
```

L'application Taka tourne en tâche de fond. Pour accéder au log de debug il faut la terminer et la relancer manuellement:

```
nsh> kill -9 5
nsh> taka
```

8.3 Compilation

Voir les explications ici: <https://github.com/lambdacconcept/taka>

Une fois le firmware compilé, programmer Taka avec le nouveau binaire comme expliqué dans la section *Mise à jour*.

8.4 Architecture NuttX

NuttX est contenu dans 2 dossiers:

- **nuttx**: contient le système d'exploitation, les drivers et la plateforme.
- **apps**: contient les applications

Plateforme Taka:

<https://github.com/lambdacconcept/taka-nuttx/tree/master/boards/arm/stm32h7/taka>

Application Taka:

<https://github.com/lambdacconcept/taka-apps/tree/master/examples/taka>

Description des fichiers:

Ce dossier contient le point d'entrée de l'application Taka ainsi que les librairies et sous fonctions.

Boucle principale de l'application:

taka_main.c	<-- Application de production
selftest_main.c	<-- Application test usine

Fichiers relatifs à la carte à puce:

apdu.c	
smartcard.c	<--- Couches de communication avec la carte à puce
arch.c	<--- Lecture des données ARCH
icao.c	<--- Lecture des données ICAO
icao_auth.c	<--- Authentification passive ICAO
icao_decode.c	<--- Décodage TLV des containers ICAO
libecc/	<--- Librairie cryptographique pour authentification passive
libpkcs7	<--- Librairie de lecture de certificats PKCS#7

Fichiers relatifs à la gestion des empreintes digitales:

cam.c	<--- Capture d'une image caméra
frame.c	<--- Traitements de correction de déformations
fingerjet/	<--- Librairie d'extraction des minutiae
minutiae.c	<--- Validation de la qualité des minutiae et conversion de format

Suivi des opérations:

led.c	<--- Couleur des LEDs
slcd.c	<--- Affichage sur l'écran LCD
tone.c	<--- Son du buzzer

Export des journaux:

logs.c

API USB:

usb.c

Fichiers système:

battery.c	<--- Niveau de charge de la batterie
buttons.c	
datetime.c	<--- Date et heure UTC
gpio.c	
network.c	
sdram.c	
taka.h	<--- Gestion des erreurs système

Lecture sans contact NFC (Non implémenté):

```
nfc.c  
libnfc/
```

8.5 Defines

Version de Taka:

- <https://github.com/lambdacconcept/taka-apps/blob/master/examples/taka/taka.h#L6>

Activer les prints de debug:

La plupart des fichiers ont un flags de debug qui permet d'avoir plus d'informations sur la console en cas dysfonctionnement, par exemple:

- https://github.com/lambdacconcept/taka-apps/blob/master/examples/taka/taka_main.c#L44
- <https://github.com/lambdacconcept/taka-apps/blob/master/examples/taka/arch.c#L14>
- <https://github.com/lambdacconcept/taka-apps/blob/master/examples/taka/icao.c#L15>

Activer/Désactiver l'authentification passive:

- <https://github.com/lambdacconcept/taka-apps/blob/master/examples/taka/icao.h#L9>

8.6 Changement des APDU

Adaptation des APDU ARCH pour supporter différentes cartes:

- <https://github.com/lambdacconcept/taka-apps/blob/master/examples/taka/apdu.c>

Construction des APDU:

1. APDU_ARCH_SELECT_APP

- <https://github.com/lambdacconcept/taka-apps/blob/master/examples/taka/apdu.c#L136>

Il modifier APDU_ARCH_SELECT_APP pour changer l'application ID et ajuster la longueur:

```
0x00, 0xa4, 0x04, 0x04,      (L'entête reste inchangée)  
0x0f,                      (Longueur de l'App ID = 15 octets)  
0xE8, 0x28, 0xBD, 0x08, 0x0F, 0xA0, 0x00, 0x00, 0x03, 0x63, 0x4D, 0x52, 0x4F, 0x41, 0x44,  
  ↪ (App ID, 15 octets)  
0x00                         (Octet final)
```

2. APDU_ARCH_SELECT_FILE

- <https://github.com/lambdacconcept/taka-apps/blob/master/examples/taka/apdu.c#L142>

Il modifier APDU_ARCH_SELECT_FILE pour changer le file ID et ajuster la longueur:

```
0x00, 0xa4, 0x02, 0x04,      (L'entête reste inchangée)  
0x02,                      (Longueur du File ID = 2 octets)  
0xC0, 0x01,                  (File ID, 2 octets)  
0x00                         (Octet final)
```

3. APDU_ARCH_READ_FILE

- <https://github.com/lambdacconcept/taka-apps/blob/master/examples/taka/apdu.c#L146>

Pas de changement c'est une commande standard “READ BINARY”.