

# Robots d'encodage

Concernant la fonctionnalité de "robot", elle est livrée de la même façon que les versions d'esup-sgc-client pour l'édition en 1 temps avec evolis ou zebra.

Aussi vous pouvez vous référer au README d'esup-sgc-client pour les détails techniques d'installation, notamment les parties spécifiques à la marque de l'imprimante souhaitée :

- [Imprimantes Evolis Primacy 2 et Primacy 1, ....](#)
- [Imprimantes Zebra.](#)

Cependant, à notre connaissance, seule l'imprimante Zebra ZXP3 a été mise en oeuvre dans plusieurs établissements actuellement pour faire office de robot d'encodage.

En effet celle-ci se prête bien au positionnement d'une webcam pour lecture du QRCode. De plus, cette imprimante se retrouve dans des établissements car elle était la seule imprimante supportée par un SGC du marché.

Plus commercialisée, reconvertir la Zebra ZXP3 en robot d'encodage permet de recycler du matériel encore fonctionnel mais dont les consommables pour l'impression ne sont plus disponibles.

Ci-dessous, on vous propose donc d'utiliser la Zebra ZXP3 comme robot d'encodage.

## Matériel

Il vous faut une Zebra ZXP3 avec un firmware à jour : V02.15.00 ; logiquement c'est le cas sur les ZXP3 détenus par les établissements.

Si vous avez comme encodeur intégré le Identiv SDI010, si vous souhaitez fonctionner sous windows, nous vous conseillons de ne pas mettre à jour le firmware de l'encodeur (qui a un firmware spécifique/propres à zebra qu'on ne retrouve pas au téléchargement).

Si vous souhaitez fonctionner sous linux, il faut par contre que le firmware de l'encodeur soit à jour avec la version d'identiv officielle pour le SDI010 (dont pcsd-lite avec cet encodeur s'avère malheureusement instable, cf ci-dessous).

Pour ce faire téléchargez le firmware (fichier FWUpdate\_SDI011\_FW7.45.zip) depuis <https://support.identiv.com/sdi010-011/> et lancez l'exécutable FwUpdate.exe depuis un pc windows connecté à l'imprimante zebra xzp3.

**Attention, la mise à jour du firmware est donc irréversible** (puisque le firmware identiv chargé avec les xzp3 n'est pas disponible au téléchargement) ; à nouveau, si vous souhaitez utiliser votre ZXP3 depuis windows, ne mettez pas à jour ce firmware.

## Positionnement de la webcam avec un support webcam imprimé en 3D

Si vous avez la possibilité d'utiliser une imprimante 3D et que vous avez une webcam logitech c922 pro, nous avons fait réaliser par [l'Atelier Autonome de Rouen](#) un support en imprimante 3D afin de positionner au mieux la webcam sur la zebra xzp3 au dessus de la carte lors de la phase d'encodage / enrôlement.

Ce support d'impression est [disponible librement sur printables.com](#), nous le proposons également ici :

- [Zebra\\_Cam\\_full.stl](#)
- [spacer 4mm.stl](#)

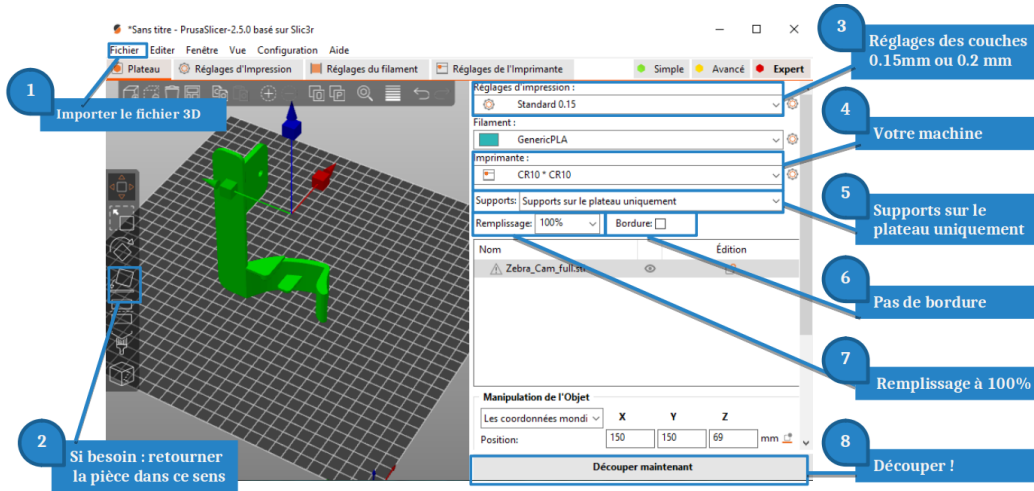
Une fois le support [Zebra\\_Cam\\_full.stl](#) téléchargé, téléchargez un slicer. Cet outil permet de transformer un fichier 3D en jeu d'instruction machine.

Il en existe plusieurs, nous conseillons PrusaSlicer sous licence libre [www.prusa3d.com/prusaslicer](http://www.prusa3d.com/prusaslicer).

La configuration de votre machine se fait avec un assistant au premier lancement de PrusaSlicer.



Voici ensuite les paramètres conseillés pour l'impression du support :



Le logiciel de slicer va générer un fichier .gcode, à transférer sur votre imprimante 3D, en général à l'aide d'une carte SD ou via le wifi.

Une vidéo 'timelapse' d'une telle impression vous est proposée sur le Pod d'Esup-Portail.



*Tout comme la réalisation du modèle de la pièce et la documentation donnée ici à son propos, cette vidéo a été réalisée par l'Atelier Autonome de Rouen (un grand Merci à eux !) dans le cadre de l'Appel À Projets Services Numériques Aux Étudiants ESUP-SGC 2022/2023.*

Une fois imprimée, vous n'avez plus qu'à retirer les structures qui ont permis l'impression de la pièce en 1 seul tenant (cf vidéo ci-dessus à nouveau).

Puis vous pouvez fixer le support à l'aide d'une vis **M6** standard. Le support ainsi réalisé et positionné permet de faire également office de détrompeur ; l'imprimante peut fonctionner ainsi capot ouvert.

A nouveau 2 vidéos vous montrent ici la fixation du support et son utilisation avec esup-sgc-client :

- [ESUP-SGC - robot d'encodage ZXP3 - Positionnement de la caméra à l'aide du support réalisé via une imprimante 3D](#)



- [ESUP-SGC : robot xzp3 encodant en masse des cartes Mifare Desfire préalablement imprimées](#)



## Positionnement de la webcam sans le support

Si vous n'avez pas la possibilité d'utiliser une imprimante 3D pour imprimer le support, vous pouvez vous référer à la documentation [Tuto robot encodeur basé sur une Zebra ZXP3](#) pour cette mise en oeuvre.

## Logiciel

Au vu des problèmes de DLL sous windows avec les imprimantes Zebra - cf <https://developer.zebra.com/content/zebranativeusbadapter64dll-and-jdk-9> vous pouvez soit faire fonctionner esup-sgc-client avec Zebra depuis un Windows avec un JDK 8 et openjfx (distribution zulu voire Oracle si vous avez la licence pour) ... ou éventuellement le faire fonctionner depuis Linux si vous arrivez à fiabiliser l'usage du lecteur NFC Identiv SDI010 dans cet environnement.

## Sous windows (conseillé)

Vous pouvez utiliser un JDK (ou JRE) 8u242 zulu (Azul Zulu: 8.44.0.13) .

Il vous faudra également le SDK Zebra "Lien-OS Multiplatform SDK" depuis le site de zebra <https://www.zebra.com/fr/fr/support-downloads/printer-software/developer-tools/card-sdk.html>

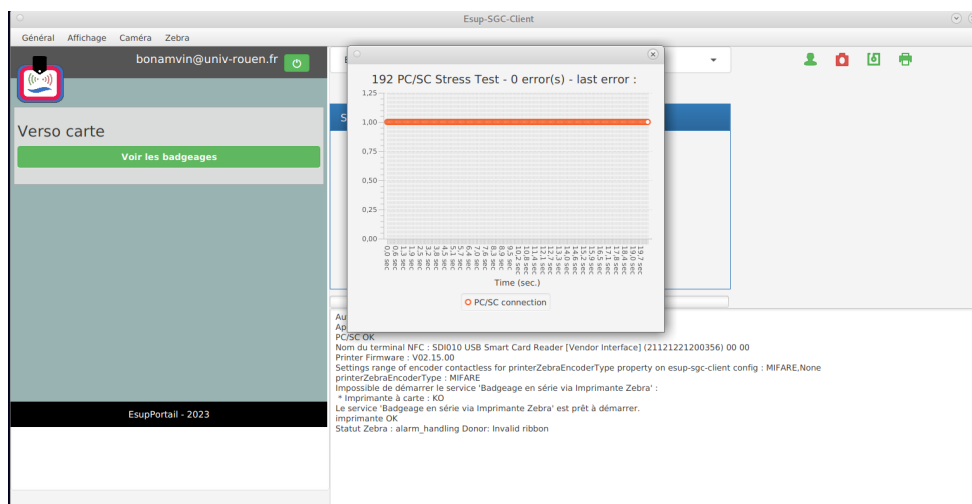
Dans variable d'environnement < variables systemes < Path, il faut alors ajouter le répertoire pointant sur la librairie du sdk ainsi installé : "C:\Program Files\Zebra Technologies\link\_os\_sdk\PC-Card\v2.14.5198\lib"

Si vous avez mis à jour le firmware du lecteur NFC Identiv SDI010 avec la version officielle, vous devez rajouter le paramètre java - **DprinterZebraHackZxpNfcPower=true** ; sans ce hack, le lecteur NFC n'est pas alimenté précisément quand la carte est positionnée sur la station d'encodage. Si vous n'avez pas modifié le firmware tel que livré par zebra, vous n'avez pas à ajouter ce paramètre.

En plus d'une JVM ancienne, le SDK de Zebra pour la ZXP3 peut également conduire à des crash java aléatoires lors d'appel de fonctions DLL du type "Exception Access Violation". Pour éviter cela, au niveau du raccourci permettant de lancer votre esup-sgc-client, demander (clic droit / propriétés sur le raccourci) à exécuter le programme en mode de compatibilité pour windows 8 (onglet compatibilité) si vous êtes sur windows 10 par exemple.

Vous pouvez ensuite lancer le esup-sgc-client propre à la zebra.

Pour valider le bon fonctionnement de votre installation, n'oubliez pas de lancer en premier lieu depuis esup-sgc-client le "Stress test pc/sc" depuis le menu "zebra" d'esup-sgc-client sur une carte Mifare DESFIRE ; aucune erreur ne doit pas apparaître sur les 30 secondes de tests de communication mifare desfire.



Vous pouvez enfin lancer (bouton start) l'encodage en série en sélectionnant "Robot encodage via webcam et imprimante Evolis" et le salle d'encodage.

## Sous linux/debian (déconseillé actuellement du fait de l'instabilité du lecteur NFC ZXP3 sous linux)

Sous linux, et depuis la [page officielle Zebra de la ZXP3](#), vous pouvez télécharger le driver linux (fichier Unified\_Zebra\_ZXP3\_Series\_Card\_Printer\_Driver-1.0.0.0-Ubuntu\_Linux-x86\_64-Install.tar.zip pour debian/ubuntu) et l'installer.

Installez le paquet pcscd et mettez le en route

```
apt install pcscd
systemctl enable pcscd
systemctl restart pcscd
```

Installez également le driver linux propriétaire du lecteur NFC Identiv SDI010 (n'oubliez pas de mettre son firmware à jour depuis un windows) : <https://support.identiv.com/sdi010-011/>

Récupérez via <http://esup-sgc-client-web-installer> ou [packagez le jar](#) d'esup-sgc-client pour **zebra**.

Installez openjfx et openjdk :

```
apt install openjfx openjdk-11-jdk
```

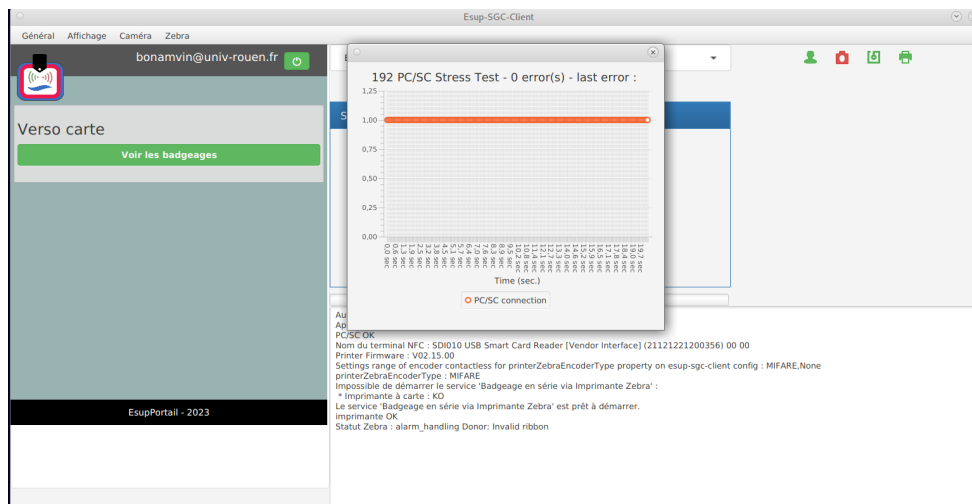
Vous pouvez enfin créer un script bash esup-sgc-client-zebra.sh permettant de lancer le client simplement en donnant les paramètres attendus :

```
#!/bin/bash

java --module-path /usr/share/openjfx/lib/ --add-modules javafx.controls,javafx.fxml,javafx.base,javafx.media,
javafx.graphics,javafx.swing,javafx.web -DprinterZebraEncoderType=MIFARE -DprinterZebraHackZxpNfcPower=true -
jar esup-sgc-client-zebra.jar
```

Le paramètre **printerZebraHackZxpNfcPower=true** permet de faire fonctionner le lecteur NFC avec le **firmware officiel d'Identiv** ; sans ce hack, le lecteur NFC n'est pas alimenté précisément quand la carte est positionnée sur la station d'encodage.

Pour valider le bon fonctionnement de votre installation, n'oubliez pas de lancer en premier lieu depuis esup-sgc-client le "Stress test pc/sc" depuis le menu "zebra" d'esup-sgc-client sur une carte Mifare DESFIRE ; aucune erreur ne doit pas apparaître sur les 30 secondes de tests de communication mifare desfire.



Pour s'assurer de la stabilité du lecteur NFC, driver et pcscd-lite, on conseille d'utiliser en plus le "Stress test PC/SC DES Blank Desfire". Pour cela et par défaut, il vous faut des cartes Desfire vierges (avec clef master-key DES à 0000000000000000 donc).

Ce test opère des authentifications DES complètes sur la master-key pendant 20 secondes.

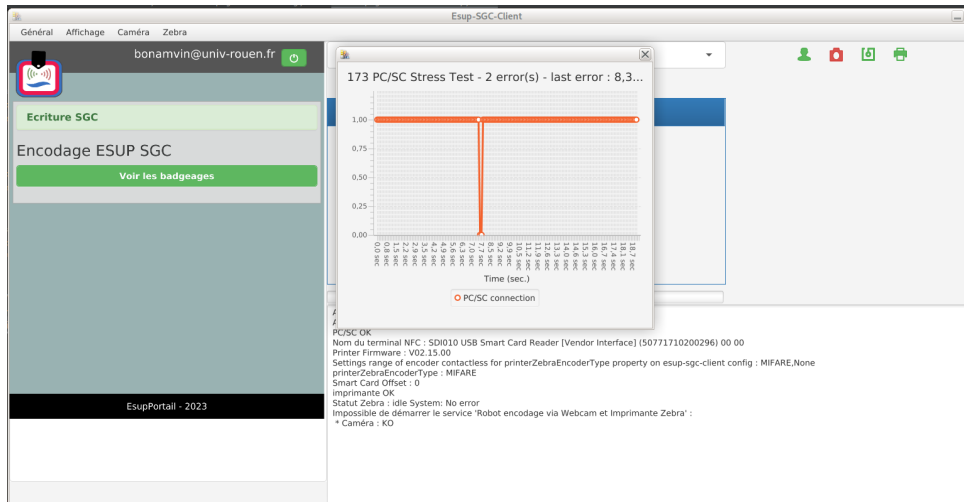
Sur le graphe ci-dessous, on voit que 2 erreurs ont eu lieu.

Dans les logs Java, on retrouve par exemple :

```
jnasmartcardio.Smartcardio$JnaPCSCException: SCardTransmit got response 0x80100016 (SCARD_E_NOT_TRANSACTED: An attempt was made to end a non-existent transaction.)
```

Alors que dans /var/log/syslog, on a

```
2023-08-31T11:43:59.849758+02:00 dgs-13-5912-1752 pcsd[11742]: 09458199 ifdwrapper.c:543:IFDTransmit() Card not transacted: 613
2023-08-31T11:43:59.850062+02:00 dgs-13-5912-1752 pcsd[11742]: 00000025 winscard.c:1616:SCardTransmit() Card not transacted: rv=SCARD_E_NOT_TRANSACTED
```



Avec un test dysfonctionnel de la sorte, des erreurs d'encodage auront lieu de manière aléatoire, une telle installation ne doit pas être considérée comme fiable et ne doit pas être utilisée en production sous peine d'avoir un taux de perte de cartes de l'ordre de 20% par exemple (suivant ce que vous encodez). Nous n'avons pour l'instant pas trouvé le moyen de fiabiliser l'usage du lecteur NFC Le robot d'encodage

## Paramétrage du offset

Si ce n'est déjà fait, il vous faudra (re)paramétrer le offset de l'imprimante à 0 (via la commande `+OS 0`) pour que la carte présente au mieux le QR-Code lorsque celle-ci est positionnée sur la station d'encodage, comme illustré sur la capture d'écran suivante :

