



# Manuel d'installation de PSC

( Prototype Sonnette Connectée )

## Sommaire :

<b>Partie Android</b>	<b>3</b>
Manuel d'Installation - Application Ouverture Porte (AOP)	3
Prérequis	3
Étapes d'Installation	3
<b>Partie C</b>	<b>5</b>
Manuel d'installation - SoftSonnette (partie A7)	5
Prérequis	5
Étapes d'installation sur la stm32mp157c-dk2	5
Implémentation de la librairie d'Intelligence artificielle	6
Manuel d'installation - SoftPorte (partie M4)	7
Prérequis	7
Compilation du projet	7
Modifier le projet existant :	7
Exécuter SoftPorte	8
<b>Partie Documentation</b>	<b>9</b>
Manuel d'installation pour générer la documentation Doxygen	9

## Partie Android

### Manuel d'Installation - Application Ouverture Porte (AOP)

Cette partie du manuel d'installation présente les étapes nécessaires pour installer l'application AOP sur votre appareil Android à partir du code source fourni.


#### Prérequis

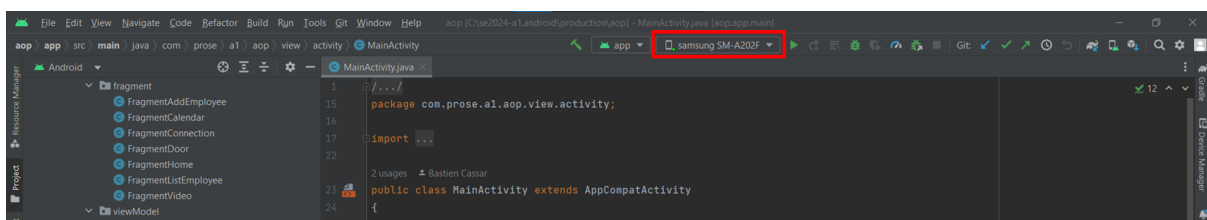
Avant de commencer, assurez-vous de disposer des éléments suivants :

- Un ordinateur avec Java JDK version 1.8 ou ultérieure installée.
- Android Studio installé sur votre ordinateur.
- Un câble USB pour connecter votre appareil Android à l'ordinateur.
- Un appareil Android compatible.

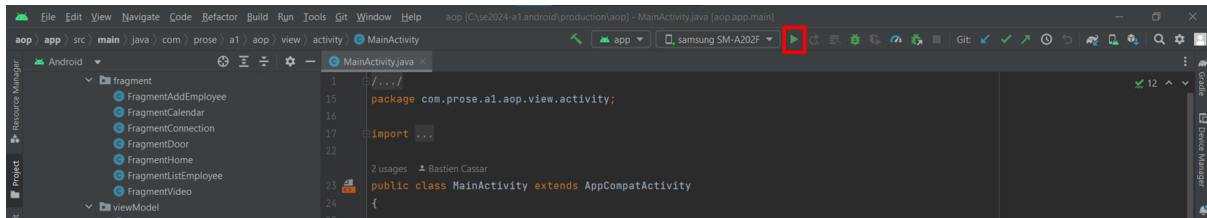
#### Étapes d'Installation

Suivez les étapes ci-dessous pour installer l'application AOP sur votre appareil Android :

1. Téléchargement du code source
  - Télécharger le fichier ZIP contenant le code source de l'application AOP fourni par l'équipe de développement.
  - Extraire les fichiers du code source de l'archive ZIP téléchargée.
2. Configuration du projet dans Android Studio
  - Lancer Android Studio sur son PC 
  - Dans la fenêtre de démarrage, sélectionner "Open an existing Android Studio project" (Ouvrir un projet Android Studio existant).
  - Naviguer vers le répertoire où sont extraits les fichiers du code source de l'application AOP, puis sélectionner le fichier "build.gradle" situé à la racine du projet.
  - Attendre que Android Studio importe et configure le projet.
3. Connexion de l'appareil Android :
  - Connecter un appareil Android à l'ordinateur à l'aide du câble USB.
  - S'assurer que le mode de débogage USB est activé sur l'appareil Android. Si ce n'est pas le cas, se référer au manuel d'utilisation du constructeur de l'appareil.
4. Configuration du périphérique cible :
  - Dans Android Studio, cliquer sur l'icône "Select Deployment Target" (Sélectionner la cible de déploiement) dans la barre d'outils supérieure.



- Sélectionner l'appareil Android connecté dans la liste des périphériques disponibles.
5. Installation de l'application :
- Dans Android Studio, cliquer sur le bouton "Run" (Exécuter) dans la barre d'outils supérieure.



- Choisir l'option "Run app" (Exécuter l'application).
- Attendre que Android Studio compile le projet et installe l'application sur l'appareil Android.
- Une fois l'installation terminée, l'icône de l'application AOP s'affiche sur l'écran d'accueil de l'appareil.

## Partie C

### Manuel d'installation - SoftSonnette (partie A7)

Cette partie du manuel d'installation présente les étapes nécessaires pour utiliser l'application SoftSonnette sur la carte stm32mp157c-dk2 à partir du code source fourni.

#### Prérequis

Pour pouvoir installer l'application sur la carte vous devez au préalable:

- Avoir installé tous les outils pour la cross-compilation présentés par ST : [ici](#).
- Être connecté à la carte via ethernet/usb/... afin de pouvoir envoyer des fichiers sur la carte.
- Configurer la connexion Wi-Fi afin de pouvoir se connecter sur la carte en Wi-Fi. Pour cela suivez les indications de ST présentées [ici](#). Le Wi-Fi doit être le même que sur le tuto de ST.
- Avoir l'OS OpenST-Linux fourni par ST. (starter package)

#### Étapes d'installation sur la stm32mp157c-dk2

##### Avec le code source :

1. Téléchargement du code source :
  - a. Télécharger le fichier ZIP contenant le code source de l'application SoftSonnette fourni par l'équipe de développement.
  - b. Extraire les fichiers du code source de l'archive ZIP téléchargée.
2. Cross-compilation :
  - a. Ouvrir un terminal et entrer la commande pour accéder à la cross-compilation
    - i. Il faut se mettre juste avant le dossier SDK
    - ii. Puis exécuter la commande suivante :  
  
*source SDK/environment-setup-cortexa7t2hf-neon-vfpv4-ostl-linux-gnueabi*
  - b. Cross-compiler le code source
  - c. Pour tout souci de cross compilation : se référer au [wiki](#) de ST.
3. Envoie sur la carte :
  - a. Envoyer l'exécutable cross-compiler sur la carte via scp/...
4. Tester l'exécutable :
  - a. Lancer l'exécutable depuis le terminal de commande, directement sur la carte.
  - b. L'application se lance

##### Avec le fichier binaire :

1. Télécharger le code source :
  - a. Récupérer le fichier binaire et le fichier csv contenant les informations des employés
2. Envoie sur la carte :

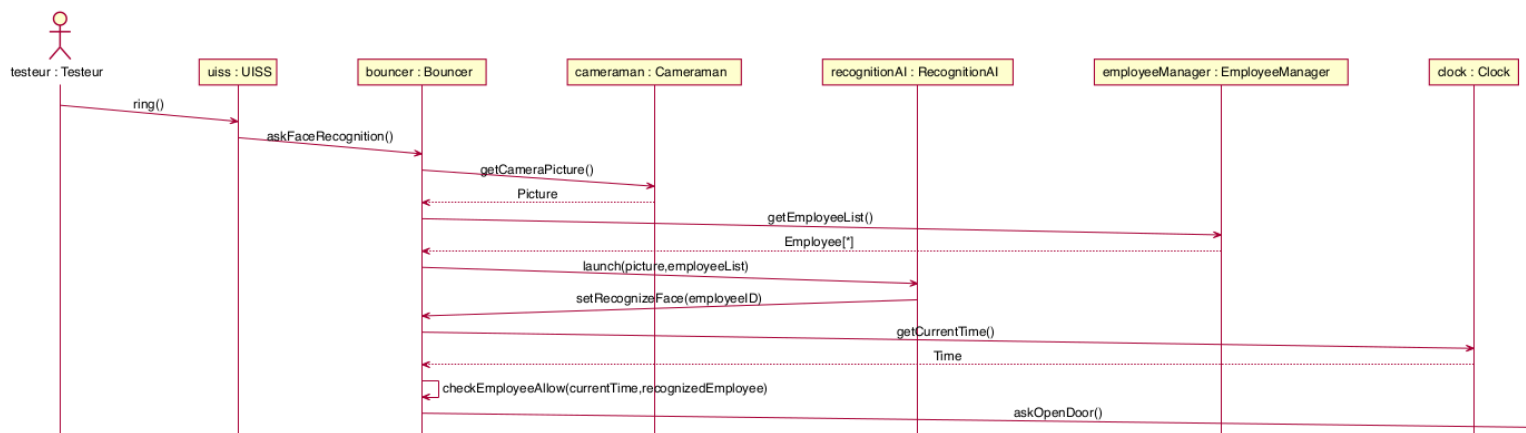
- a. Envoyer les fichier binaire et le fichier csv à la carte via la commande scp.
3. Tester l'exécutable :
  - a. Lancer l'exécutable depuis le terminal de commande, directement sur la carte.
  - b. L'application se lance

## Implémentation de la librairie d'Intelligence artificielle

Afin d'ajouter votre module de reconnaissance faciale, vous devez :

1. Dans le code, l'objet "recognitionAI.c" est conçu pour au choix :
  - a. Appeler un script permettant de lancer la reconnaissance faciale sur un processus externe à SoftSonnette à l'aide de la fonction recognitionAI\_lauch()
  - b. Être remplacé directement par des appels à une librairie de reconnaissance faciale en C
2. En fonction de ce que retourne votre IA vous devez adapter la méthode "setRecognizeFace()" de la classe Bouncer: vous trouverez dans les premières lignes de la classe un ensemble de trois lignes (printf/scanf) représentant la sortie de l'algorithme de reconnaissance faciale. L'objectif est que la variable "answer" ait comme valeur l'employeeID de la personne reconnue ou la chaîne "UNKNOWN".

Une fois ces deux modifications faites, le logiciel SoftSonnette sera prêt à reconnaître vos employés en fonction de la réponse du bloc d'intelligence artificielle, comme illustré ci-dessous:



## Manuel d'installation - SoftPorte (partie M4)

Cette partie du manuel d'installation présente les étapes nécessaires pour compiler l'application **SoftPorte** en vue de son exécution sur le Cortex M.

### Prérequis

- STM32CubeIDE version 1.12.1 (ou supérieur) est installé sur un PC tournant sous Linux (seule la version Linux permet de déboguer le Cortex M)

### Compilation du projet

1. Téléchargement du code source :
  - a. Télécharger le fichier ZIP contenant le code source de l'application SoftSonnette fourni par l'équipe de développement.
  - b. Extraire les fichiers du code source de l'archive ZIP téléchargée.
2. Ouvrir STM32CubeIDE et créer un nouvel espace de travail (workspace)
3. Importer le projet :
  - a. "Import->Existing Projects into Workspace"
  - b. Renseigner le chemin où se situent les fichiers source et choisir le dossier "softPorte2-BareMetal"
  - c. Sélectionner les 3 sous-projets (en cliquant sur "Select All")
  - d. Cliquer sur "Finish"
4. Compiler le projet
  - a. Clic droit sur softPorte2-BareMetal\_CM4 -> "Build target"

### Modifier le projet existant :

Les fichiers sources du projet se situent dans le projet "softPorte2-BareMetal\_CM4->Core->App"

Le projet est fourni avec le fichier de configuration CubeMX "softPorte2-BareMetal.ioc" qui permet de configurer les différents périphériques du M4.

Attention ! Si le code est régénéré par CubeMX, il est nécessaire de réactiver manuellement de VirtualUART afin de compiler à nouveau le projet.

Pour cela:

- Ouvrir le fichier "openamp\_conf.h" (accessible depuis le dossier OPENAMP)
- Se rendre à la ligne 60 et décommenter le define présent:

```
55  /* ##### Virtual Diver Module Selection ##### */
56  /**
57   * @brief This is the list of modules to be used in the OpenAMP Virtual driver module
58   *       Please note that virtual driver are not supported on all stm32 families
59   */
60  #define VIRTUAL_UART_MODULE_ENABLED
61  //#define VIRTUAL_I2C_MODULE_ENABLED
```

## Exécuter SoftPorte

Il est possible de déboguer SoftPorte en utilisant les outils de débogage de STM32CubeIDE.

A noter que le lancement de l'application SoftSonnette permet automatiquement de lancer le programme du Cortex M.

Pour mettre à jour le fichier binaire utiliser par le Cortex M:

- S'assurer que le fichier SoftPorte-BareMetal\_CM4.elf est présent sur la Board dans le répertoire : /lib/firmware
- Ajouter le firmware au chemin de lancement du Cortex M: (x étant 0)

```
Board $> echo -n <firmware_name.elf> > /sys/class/remoteproc/remoteprocX/firmware
```

Pour démarrer manuellement le Cortex M:

```
Board $> echo start >/sys/class/remoteproc/remoteprocX/state
```

Des information supplémentaires sur l'exécution d'un programme sur le Cortex M peuvent être trouver sur le Wiki de ST:

[https://wiki.st.com/stm32mpu/wiki/Linux\\_remoteproc\\_framework\\_overview](https://wiki.st.com/stm32mpu/wiki/Linux_remoteproc_framework_overview)



## Partie Documentation

### Manuel d'installation pour générer la documentation Doxygen

#### Disclaimer :

- 1) La version du Doxygen utilisée pour générer la documentation est la 1.9.6. Si vous utilisez une version différente pour générer la documentation, l'équipe se dédouane de tous problèmes liés.
- 2) L'équipe recommande 2 fichiers différents pour générer le Doxygen en C et en Java.

En plus de vous livrer la documentation du Projet, l'équipe Projet Prose A1 vous donne la possibilité de la générer de vous même.

Pour la générer, il faut d'abord installer Doxygen sous votre OS :

[lien pour le télécharger](#)

Une fois que votre Doxygen est fonctionnel, il faut ouvrir un terminal dans l'emplacement de la génération de votre documentation (nous recommandons dans un dossier projet). Dans ce terminal, il faut entrer la commande "doxygen -g doxyfile", cela permet de créer le fichier de génération.

La prochaine étape est d'ouvrir le fichier créer. En fonction du langage que vous souhaitez générer, vous allez modifier les différents arguments :

(/>\ Attention : en fonction de la version du doxygen que vous utilisez, les options qui suivent peuvent déjà être dans leur état final !/)

- Pour le langage C :
  - OPTIMIZE\_OUTPUT\_FOR\_C = YES
  - INPUT = ./<chemin>/<vers>/softSonnette  
./<chemin>/<vers>/softPorte2-BareMetal/Cm4/Core/App
  - RECURSIVE = YES
  - GENERATE\_LATEX = NO
  - HAVE\_DOT = YES (permet la génération des diagrammes d'inclusion)
- Pour le langage Java :
  - OPTIMIZE\_OUTPUT\_FOR\_JAVA = YES
  - INPUT = ./<chemin>/<vers>/aop
  - RECURSIVE = YES
  - GENERATE\_LATEX = NO

Maintenant vous pouvez générer la documentation, avec "doxygen doxyfile". Cela créer un dossier html. Vous pouvez accéder à la documentation en lançant le "index.html" avec un navigateur internet.