



X-SCAN

**INTEGRATION DANS
DES APPLICATIONS
TIERCES**



PÉRIMÈTRE

Ce document présente les 3 options possibles pour intégrer le support du X-SCAN dans une application tierce

- Option 1 – Injection dans champ texte
- Option 2 – Intent Crosscall
- Option 3 – Intégration SDK

Les avantages et les limites de chaque méthode sont précisés ci-après



ETAPES D'UTILISATION DU X-SCAN

L'utilisation de X-SCAN avec une application tierce nécessite les 3 étapes suivantes :

**CONFIGURATION DU
X-SCAN**

**DÉCLENCHEMENT DU
X-SCAN**

**TRAITEMENT DU
CONTENU SCANNÉ**

Ces étapes sont gérées différemment selon le mode d'intégration choisi



OPTION 1 : INJECTION DANS CHAMP TEXTE

Principe

Aucune intégration n'est requise dans l'application tierce

Nécessite l'application X-TRACK pour la configuration et le contrôle de X-SCAN.

Le scan peut être déclenché via :

- Un bouton programmable
- Un bouton flottant

Le contenu du code scanné est inséré dans le champ texte sélectionné sur le smartphone.

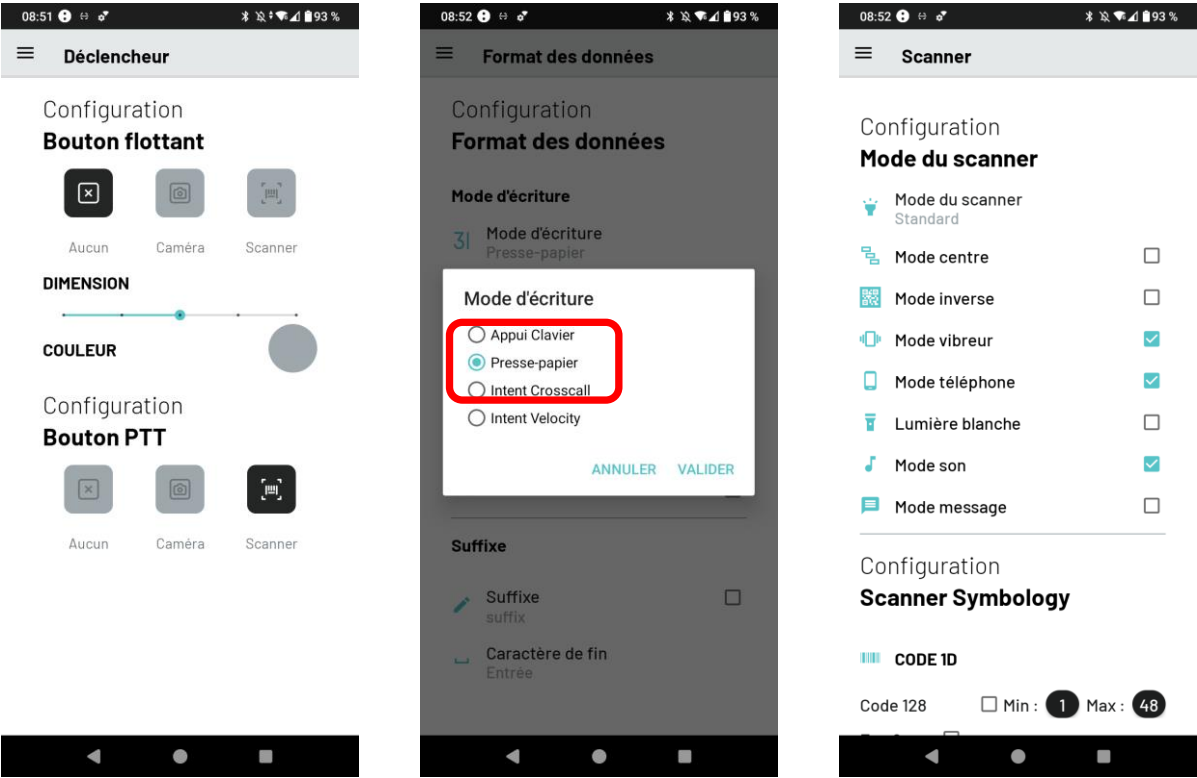
Deux modes d'injection sont supportés :

- Presse-papier (Plus rapide mais incompatible avec certaines applications comme Excel)
- Appui clavier (Fiable mais plus lent)



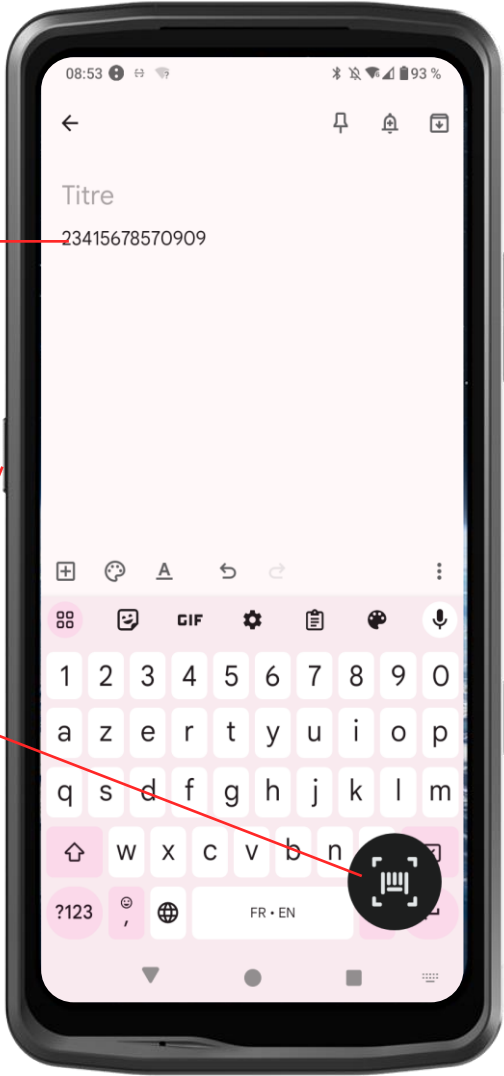
OPTION 1 : INJECTION DANS CHAMP TEXTE

Expérience Utilisateur



Le contenu du code
scanné est injecté
dans le champ texte
sélectionné

Appui sur le bouton
programmable ou le
bouton flottant



Configuration du X-SCAN dans l'application X-TRACK



OPTION 2 : INTENT CROSSCALL

Principe

Effort d'intégration dans l'application tierce limité

Besoin d'utiliser l'application X-TRACK pour la configuration et le contrôle du X-SCAN

Le scan peut être déclenché via le bouton programmable ou le bouton flottant

À chaque scan réussi, un message Intent est diffusé avec les caractéristiques suivantes:

ACTION : `com.crosscall.xtrack.action.BARCODE`

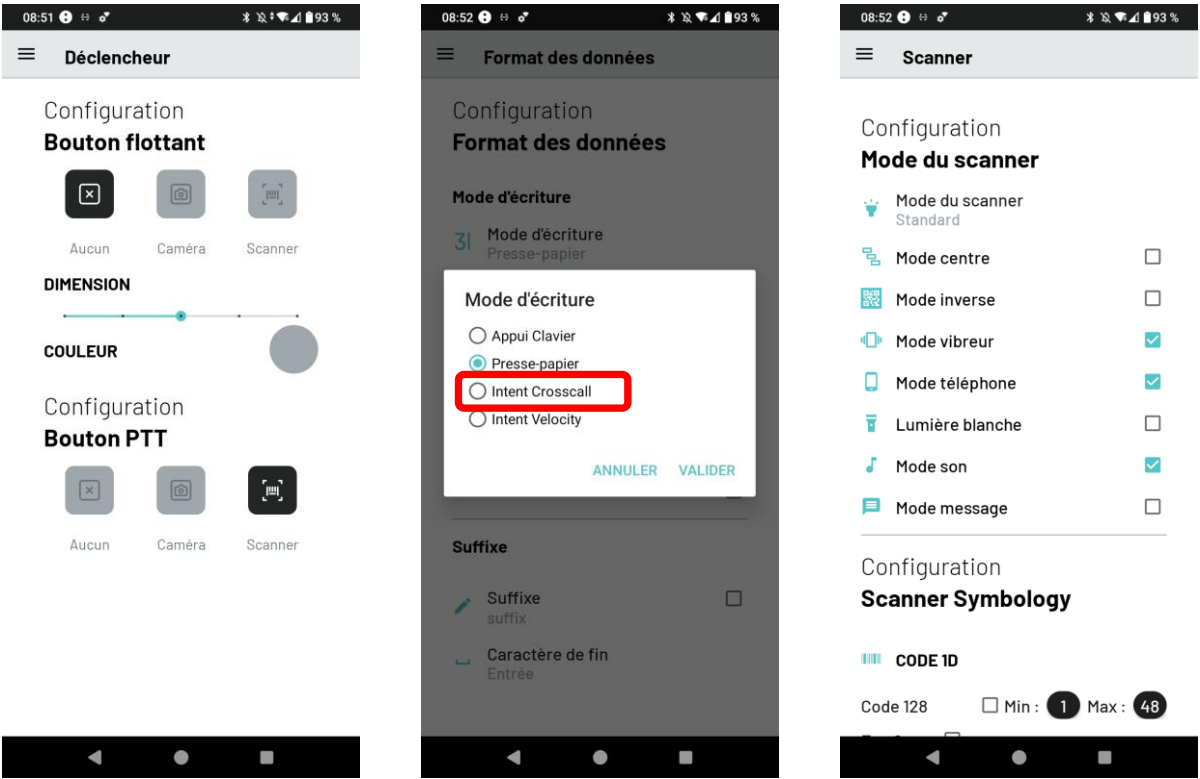
VARIABLE : `com.crosscall.xtrack.data_string`

L'application tierce intercepte ce message et implémente la logique associée



OPTION 2 : INTENT CROSSCALL

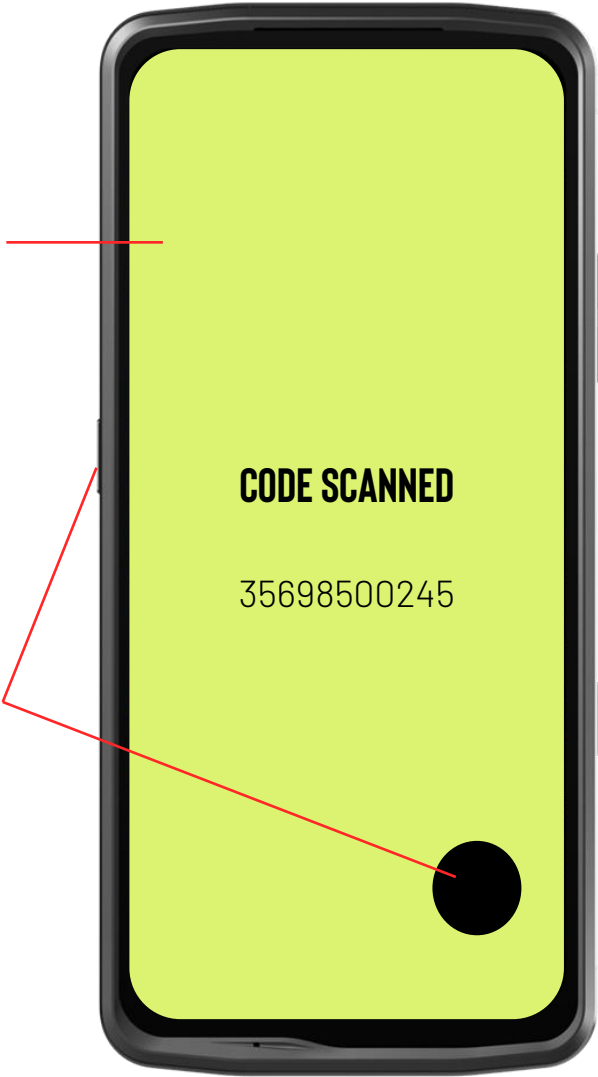
Expérience Utilisateur



Configuration du X-SCAN dans l'application X-TRACK

L'application tierce intercepte l'intent et implémente la logique associée

Appui sur le bouton programmable ou le bouton flottant





OPTION 3 : INTÉGRATION SDK

Principe

Le X-SCAN est totalement géré par l'application tierce:

- La configuration de l'appareil
- Le déclenchement du scan
- Le traitement du code scanné

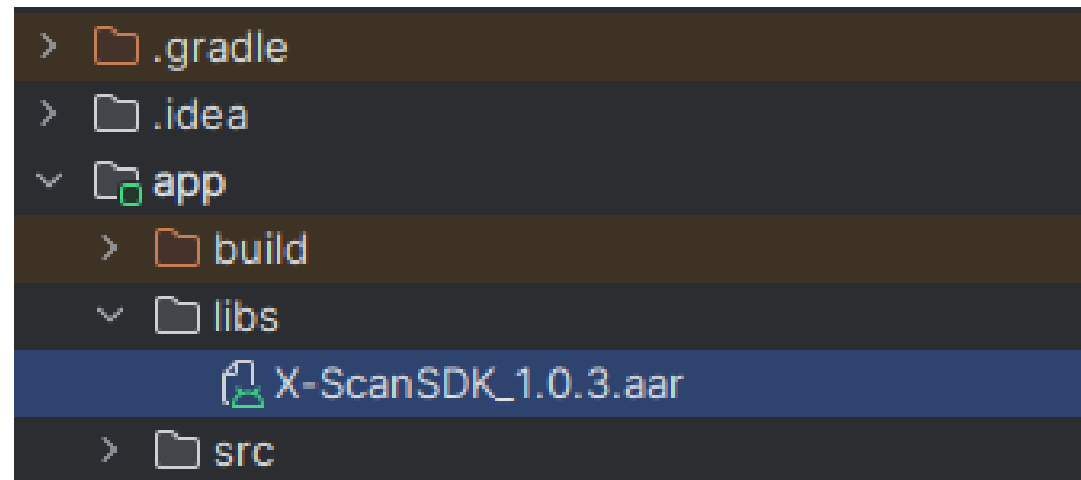
L'application X-TRACK n'est pas nécessaire dans le cas de l'utilisation du SDK



OPTION 3 : INTÉGRATION SDK

Intégration de la librairie

Ajout du fichier *.AAR addition dans Android STUDIO





OPTION 3 : INTÉGRATION SDK

Initialisation = Configuration du SDK et du X-SCAN

- Symbolologies (Types & Longueur)
- Mode de lecture (Level, Pulse, Sense, Batch)
- Mode centre
- Réglages audio et vibration
- Sensibilité

```
private val xScanManager = XScanManager(appContext, listener: this)

init {
    xScanManager.initialize(
        disconnectedBeep = false,
        scanBeep = false,
        connectBeep = false,
        showToast = false,
        vibration = true
    )
}
```

```
override fun startConnexion() {
    startUsbConnection()
}
```



OPTION 3 : INTÉGRATION SDK

Initialisation = Configuration du SDK et du X-SCAN

```
private fun startUsbConnection() {  
    val formats: MutableList<IBarcodeFormat> = mutableListOf()  
    xScanManager.connect()  
    formats.add(BarcodeFormatAztec(IBarcodeFormat.Activation.Enable))  
    formats.add(BarcodeFormatCode39(IBarcodeFormat.Activation.Enable))  
    formats.add(BarcodeFormatCode93(IBarcodeFormat.Activation.Enable))  
    formats.add(BarcodeFormatCode128(IBarcodeFormat.Activation.Enable))  
    formats.add(BarcodeFormatDataMatrix(IBarcodeFormat.Activation.Enable))  
    formats.add(BarcodeFormatEan8(IBarcodeFormat.Activation.Enable))  
    formats.add(BarcodeFormatEan13(IBarcodeFormat.Activation.Enable))  
    formats.add(BarcodeFormatFebraban(IBarcodeFormat.ActivationFebraban.Enable))  
    formats.add(BarcodeFormatGS1_128(IBarcodeFormat.Activation.Enable))  
    formats.add(BarcodeFormatGS1Databar(IBarcodeFormat.Activation.Enable))  
    formats.add(BarcodeFormatIndustrial25(IBarcodeFormat.Activation.Enable))  
    formats.add(BarcodeFormatISBN(IBarcodeFormat.Activation.Enable))  
    formats.add(BarcodeFormatISSN(IBarcodeFormat.Activation.Enable))  
    formats.add(BarcodeFormatITF20f5(IBarcodeFormat.Activation.Enable))  
    formats.add(BarcodeFormatITF6(IBarcodeFormat.ActivationITF.EnableAndTransmitCheckCharacter))  
    formats.add(BarcodeFormatITF14(IBarcodeFormat.ActivationITF.EnableAndTransmitCheckCharacter))  
    formats.add(BarcodeFormatMatrix20f5(IBarcodeFormat.Activation.Enable))  
    formats.add(BarcodeFormatMicroQrCode(IBarcodeFormat.Activation.Enable))  
    formats.add(BarcodeFormatMSIPlessey(IBarcodeFormat.Activation.Enable))  
    formats.add(BarcodeFormatPdf417(IBarcodeFormat.Activation.Enable))  
    formats.add(BarcodeFormatPlessey(IBarcodeFormat.Activation.Enable))  
    formats.add(BarcodeFormatQr(IBarcodeFormat.Activation.Enable))  
    formats.add(BarcodeFormatStandard25(IBarcodeFormat.Activation.Enable))  
    formats.add(BarcodeFormatUPCA(IBarcodeFormat.Activation.Enable))  
    formats.add(BarcodeFormatUPCE(IBarcodeFormat.Activation.Enable))  
    xScanManager.setBarcodeFormatEnable(formats)  
}
```



OPTION 3 : INTÉGRATION SDK

Activation (en fonction du mode sélectionné)

- Level Mode : Capture instantanée / Actif pendant 5 sec
- Pulse mode : Capture instantanée / Actif tant que le bouton est maintenu appuyé
- Sense mode : Détection de mouvement
- Batch mode : Détection de lot / Exclusion de réplica

```
override fun startScan() {  
    xScanManager.startScan()  
}
```

Une unique commande est utilisée pour lancer quel que soit le mode sélectionné. Le mode est préalablement défini dans les réglages



OPTION 3 : INTÉGRATION SDK

Traitement des callbacks

Traitement des événements de scan propre à chaque application tierce

```
override fun onScanResult(result: String) {  
    scope.launch { this: CoroutineScope  
        _usbEventFlow.emit(result)  
    }  
}
```



OPTION 3 : INTÉGRATION SDK

Désactivation

Déconnexion du X-SCAN

```
override fun stopConnexion() {  
    xScanManager.close()  
}
```



SYNTHÈSE

X-TRACK	OPTION 1 (INJECTION)	OPTION 2 (INTENT)	OPTION 3 (SDK)
Configuration du X-SCAN	Géré par X-TRACK	Géré par X-TRACK	Géré par l'application tierce
Déclenchement du X-SCAN	Géré par X-TRACK	Géré par X-TRACK	Géré par l'application tierce
Traitement du code scanné	Géré par X-TRACK	Géré par l'application tierce	Géré par l'application tierce
Effort d'intégration	Nul	Limité (1 à 2 jours)	Simple (2 à 3 jours)
Avantages	<ul style="list-style-type: none">Aucune modification de l'application tierce requisePeut-être utilisé avec n'importe quelle application Android ou web dès lors qu'elle dispose d'un champ texte	<ul style="list-style-type: none">Impact limité pour l'application tierceFonctionne sans obligation d'avoir un champ texte dans l'application	<ul style="list-style-type: none">X-TRACK non nécessaireMeilleures performances de scanContrôle total et dynamique de la configuration du X-SCAN (Gestion torche, mode écran...)Contrôle total de l'UX
Cible	<ul style="list-style-type: none">Petite/Moyenne entreprises sans équipe de développementTests du X-SCAN	<ul style="list-style-type: none">Entreprises avec équipe de développement	<ul style="list-style-type: none">Entreprises avec équipe de développement