

# DOSSIER E5

Jean POULARD BORE

## Sommaire

Présentation de l'entreprise.....	4
Projet I Mise en place d'un BIOS personnalisé sur PC embarqués.....	9
I. Contexte.....	9
II. Étude des solutions.....	9
III. Proposition (POC).....	10
III.1 – Analyse de l'existant :.....	10
III.2 – Mise en évidence des contraintes :.....	10
III.3 – Premier déploiement en environnement contrôlé :.....	11
III.4 – Mise en place en production :.....	12
1. Stratégie de déploiement.....	12
2. Plan de retour arrière.....	13
3. Limites du périmètre (lien avec l'affichage OS).....	13
III.5 – Rédaction de la procédure :.....	14
IV. Mise en place / PoC.....	15
1. Environnement de test.....	15
2. Outils utilisés.....	16
3. Déroulement du PoC.....	16
a) [REDACTED].....	16
b) [REDACTED].....	16
c) Vérification de l'image BIOS.....	17
d) Première tentative de flash.....	17
e) Seconde tentative de flash (validée).....	18
4. Limites constatées et ouverture vers la suite.....	19
V. Livraison :.....	20
Livrables.....	20
Critères d'acceptation.....	21
Projet II Rotation OS embarqué.....	22
I. Contexte.....	22
II. Analyse de l'existant côté OS.....	23

III. Étude des solutions d'orientation.....	24
III.1 [REDACTED] (maquette).....	24
a) Recensement des sorties vidéo et du périphérique tactile.....	24
b) Rotation de l'affichage et calibration du tactile.....	26
III.2 Limites [REDACTED]	28
III.3 Piste [REDACTED]	29
IV. Mise en place du PoC.....	30
IV.1 [REDACTED]	30
IV.2 [REDACTED]	31
V. Mise en place en production.....	33
VI. Rédaction de la procédure / VII. Livraison.....	34
Projet III Déploiement d'images sur PC embarqués.....	35
I. Contexte.....	35
II. Étude des solutions.....	37
Options considérées : .....	37
III. Proposition (POC).....	39
Architecture technique : .....	39
Processus simplifié : .....	40
RACI : .....	40
IV. Mise en place / PoC.....	41
Étapes réalisées : .....	41
Procédure d'exploitation (résumé) : .....	41
V. Livraison.....	42
Livrables : .....	42
Critères d'acceptation : .....	42
Plan de passage en production (phase 2) : .....	42
VI. Annexes : .....	43

## Présentation de l'entreprise

E-COBOT est une entreprise industrielle française créée en 2016, spécialisée dans la robotique collaborative et l'intelligence artificielle. Basée à Carquefou, près de Nantes, elle conçoit et intègre des solutions robotiques destinées principalement à l'industrie et à la logistique, avec l'objectif d'automatiser les flux tout en améliorant l'ergonomie et la sécurité des opérateurs.

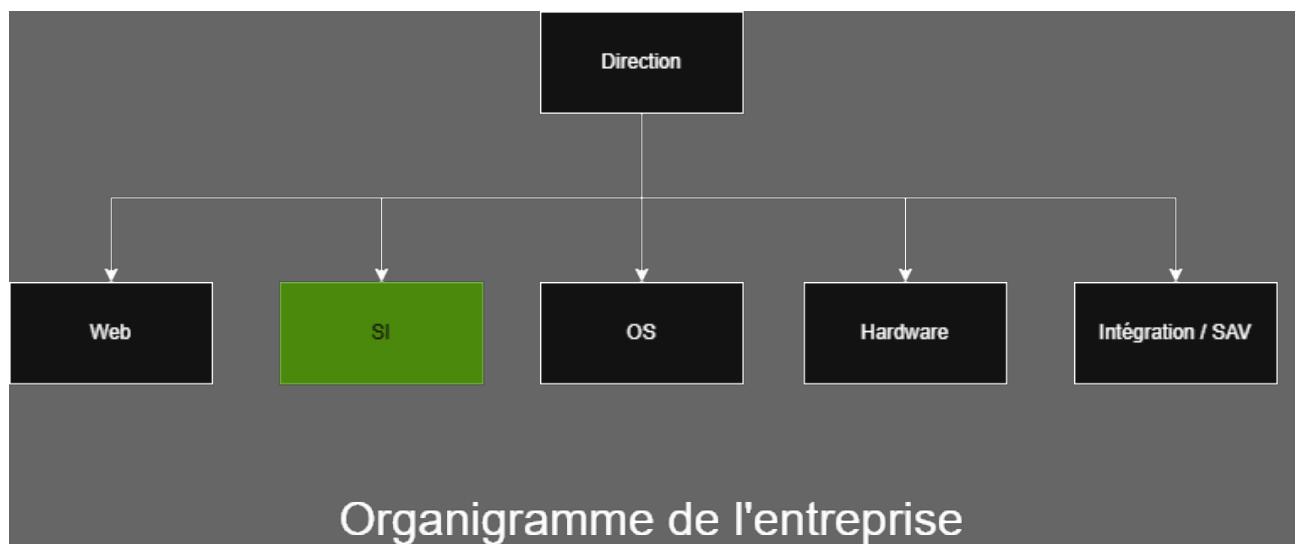


L'entreprise compte une trentaine de collaborateurs aux compétences variées (robotique, informatique, mécanique, ergonomie, etc.) et s'organise autour de plusieurs pôles qui travaillent en étroite collaboration. On retrouve notamment un pôle d'ingénierie et de conception, chargé de définir les solutions techniques, un pôle de développement logiciel et d'intelligence artificielle, un pôle d'intégration et de mise en service chez les clients, ainsi que des fonctions support (commerce, administration, ressources humaines, informatique).

Au sein de l'entreprise, ces services collaborent pour mettre en place des systèmes robotiques complets : de l'analyse du besoin client, à la conception de la solution, en passant par l'installation sur site et l'accompagnement des utilisateurs au quotidien. Cette organisation permet à E-COBOT de proposer des projets « clé en main » et de suivre ses clients sur toute la durée de vie de leurs équipements.

Au moment de mon arrivée, l'entreprise ne disposait pas de service Système d'Information interne : la majorité de la gestion informatique (maintenance des postes, serveurs, sauvegardes, réseau, etc.) était confiée à un prestataire d'infogérance. Mon poste a donc été créé afin d'internaliser progressivement ces missions et de structurer un véritable service SI. Je suis ainsi le seul référent informatique en interne et j'ai pour rôle d'assurer le bon fonctionnement du parc informatique, d'accompagner les utilisateurs au quotidien, de contribuer à la sécurisation des systèmes et de mettre en place les outils nécessaires au développement de l'activité.

Mon référent au sein de l'entreprise est le responsable de l'équipe de développement web, avec lequel je travaille en étroite collaboration pour faire le lien entre les besoins métiers (production, R&D, web, administratif, etc.) et les solutions techniques déployées. Cette organisation permet de mieux intégrer les enjeux informatiques au cœur des projets de l'entreprise, tout en gardant une continuité avec l'ancien modèle d'infogérance, encore présent pour certaines opérations spécifiques.



Avant d'intégrer E-COBOT, je me suis d'abord orienté vers le développement web. Cette première formation m'a permis d'acquérir des bases solides en programmation, en conception d'applications et en compréhension des besoins utilisateurs. Au fil du temps, j'ai toutefois ressenti le besoin d'élargir ma vision au-delà du code : comprendre comment les applications s'intègrent dans une infrastructure complète, comment les réseaux sont organisés, comment les données sont stockées, protégées et mises à disposition des utilisateurs.

C'est dans cette optique que je me suis tourné vers une formation en administration systèmes et réseaux (BTS SIO orienté SISR), afin d'acquérir des compétences plus transverses : gestion de parc, services réseau, virtualisation, sauvegardes, annuaire, etc. À travers ce parcours, mon objectif est de me spécialiser progressivement en cybersécurité, en comprenant à la fois le fonctionnement technique des systèmes (systèmes d'exploitation, réseaux, services) et les risques auxquels ils sont exposés. Mon expérience actuelle au sein du service SI d'E-COBOT s'inscrit directement dans ce projet : elle me permet de confronter mes connaissances théoriques à des problématiques concrètes de production, de support et de sécurité.

Dans le cadre de mon poste, j'assure au quotidien le bon fonctionnement du système d'information d'E-COBOT. Mes missions couvrent à la fois le support aux utilisateurs et la gestion de l'infrastructure. Elles peuvent se regrouper en plusieurs volets :

**Support et assistance utilisateurs** : résolution des incidents du quotidien, accompagnement à l'utilisation des outils, prise en main des nouveaux arrivants.

**Gestion du parc informatique** : installation, configuration et maintenance des postes de travail, suivi du matériel et des logiciels utilisés dans l'entreprise.

**Administration des systèmes et des services** : gestion des comptes et des droits, participation à la configuration des serveurs et des services réseau nécessaires à l'activité.

**Sécurité et fiabilité** : sensibilisation aux bonnes pratiques, participation à la mise en place de sauvegardes, de solutions d'authentification et de mesures de protection des données.

**Documentation et structuration du SI** : rédaction de procédures, organisation des informations techniques et contribution à la structuration d'un service SI pérenne.

Certains projets plus techniques, comme le flash de firmware ou la mise en place de déploiements automatisés via PXE, sont détaillés dans la suite de ce document.

## **Projet I Mise en place d'un BIOS personnalisé sur PC embarqués**

### **I. Contexte**

Dans le cadre d'une configuration personnalisé de l'affichage graphique du système d'exploitation mis à disposition de nos clients. Nous avons contacté le fournisseur logiciel du matériel utilisé afin qu'il nous mettes à disposition une version de leur BIOS personnalisé à flasher sur leurs appareils.

### **II. Étude des solutions**

Dans les échanges avec le fournisseur de l'image BIOS, il a été défini que l'update du BIOS doit se faire par notre SI via les outils mis à disposition sur leur site « AMI ». Compte tenu de leur politique interne (NDA) ils ne peuvent nous fournir une procédure complète ainsi que les outils à employés.

Cette contrainte nous amène à une nécessité d'étude et de création d'une procédure interne. Compte tenu du caractère critique de la mise en place il est obligatoire de passer par un « Proof of Concept » avant toute mise en production.

### III. Proposition (POC)

La procédure de mise en place du POC repose sur plusieurs étapes logique afin d'éviter toutes erreurs destructrices sur le matériel (brick). Ces étapes suivent un ordre classique d'étude de solution :

#### III.1 – Analyse de l'existant :

Cette étape est crucial pour le bon dérouler de l'intervention, en effet elle permet un instantané stable des versions antérieures fonctionnels. Cela servira de base de référence pour la mise à jour du BIOS. Doit être référencé toutes les informations critiques et disponibles de la version :

- Identifiant des composants matériels
- Reprise des versions du firmware
- Reprise des identifiants logiciels et matériels

#### III.2 – Mise en évidence des contraintes :

Compte tenu du caractère critique et potentiellement destructif de l'intervention il est obligatoire de remonter toutes contraintes ou paramètres non-maîtrisés qui seraient susceptibles d'avoir des effets indésirables comme l'indisponibilité matériels (brick) ou une configuration défectueuse (défauts liaison matériel-logiciel).

*Les contraintes potentiels sont :*

Criticité du matériel ciblé : équipement en production, absence de redondance ou de solution de repli en cas de panne.

Incertitude sur le périmètre matériel : doute sur le modèle exact, la révision de carte, la version de chipset ou la compatibilité du binaire fourni.

Incertitude sur l'outil et les paramètres de flash : [REDACTED] non validée par le fournisseur, options de flash non documentées (régions ME, NVRAM, bootblock, EC, etc.).

Présence de mécanismes de sécurité non pris en compte : [REDACTED]  
[REDACTED] dont les clés ou procédures de récupération ne sont pas disponibles.

Ces contraintes seront à écarter lors de l'analyse de l'existant et à prendre en considération lors de l'étape de déploiement en environnement contrôler.

**III.3 – Premier déploiement en environnement contrôlé :**

Pour le déploiement en environnement contrôler, une machine cible avec les mêmes paramétrages que les cibles en production à été mise à disposition du SI.

Des sauvegardes de la machines seront effectuées en amont afin d'assurer un retour à l'état initial en cas d'effets indésirables lors du premier déploiement.

### **III.4 – Mise en place en production :**

L'objectif est de disposer d'une procédure reproductible et maîtrisée permettant de déployer le BIOS modifié sans dégradation du service ni risque majeur pour le matériel.

#### **1. Stratégie de déploiement**

La mise à jour sera réalisée en plusieurs étapes :

##### **Lot pilote**

- Sélection de quelques platines représentatives (même modèle, même usage).
- Application de la procédure de flash définie dans le document annexe.
- Vérification systématique :
  - de l'accessibilité du BIOS après redémarrage ;
  - de l'affichage pivoté à 180° au niveau BIOS ;
  - de l'absence de comportements anormaux au démarrage (erreurs, instabilité...).

##### **Validation interne**

- Présentation des résultats du lot pilote aux équipes concernées (SI / métier).
- Si besoin, ajustement de la procédure (pré-requis, points de contrôle, temps d'intervention).

## Généralisation

- Planification des interventions sur l'ensemble des platines concernées (par vague, en dehors des heures de production si nécessaire).
- Application de la procédure de mise à jour BIOS sur chaque platine, en respectant :
  - la préparation (sauvegarde des paramètres critiques) ;
  - l'ordre des étapes [REDACTED]  
[REDACTED]
  - la traçabilité (noter la date, la version de BIOS avant/après, et le résultat).

## 2. Plan de retour arrière

En cas de problème lors d'une mise à jour (erreur au flash, comportement instable après redémarrage) :

- la platine pourra être restaurée à partir :
  - du BIOS d'origine (image sauvegardée lors du PoC),
  - et/ou des paramètres BIOS initiaux relevés lors de l'analyse de l'existant ;
- la procédure de rollback devra être documentée et testée sur une platine de test avant d'être appliquée en production.

## 3. Limites du périmètre (lien avec l'affichage OS)

La mise à jour du BIOS permet bien de **forcer l'orientation de l'affichage au niveau firmware**.

Cependant, au cours du PoC, il est apparu que **le système d'exploitation embarqué applique sa propre configuration graphique**, indépendante du BIOS.

### **III.5 – Rédaction de la procédure :**

À l’issue du PoC, une procédure technique formalisée a été rédigée afin de permettre la **reproduction maîtrisée** de la mise à jour du BIOS sur les platines ciblées.

Cette procédure a pour objectifs principaux de :

- **sécuriser l’intervention** en rappelant les prérequis (sauvegardes, environnement de test, matériel nécessaire) ;
- **standardiser les manipulations** [REDACTED]
- **garantir la traçabilité** (version de BIOS avant/après, date, résultat de l’intervention) ;
- **prévoir un retour arrière** en cas de dysfonctionnement (réutilisation du BIOS d’origine, restauration des paramètres).

La procédure détaillée de mise à jour du BIOS est fournie en annexe sous la référence :

- **Annexe X – Procédure P-BIOS-01 : Mise à jour du BIOS personnalisé (NDA)**

Elle est destinée à être utilisée par l’équipe système lors des futures interventions sur ces matériels, que ce soit dans le cadre de nouveaux déploiements ou de maintenances planifiées.

## IV. Mise en place / PoC

### 1. Environnement de test

Le PoC a été réalisé sur une platine de test représentative de la production, avec les caractéristiques suivantes :

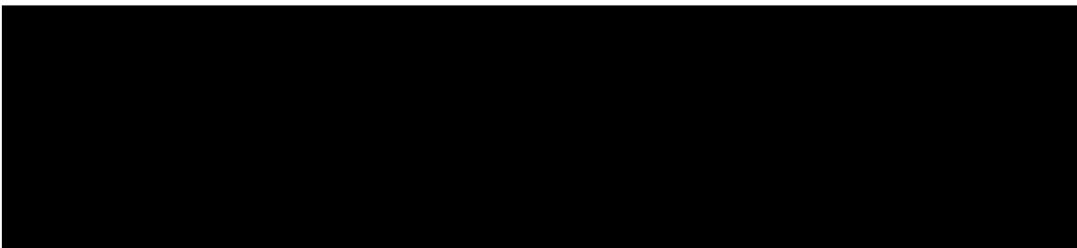
- **Carte embarquée** : modèle identique aux platines destinées à la production.
- **Affichage** : écran intégré à la platine (scénario final des clients).
- **Alimentation** : branchement sur alimentation dédiée, hors poste utilisateur.
- **Isolation** : PoC réalisé sur un banc de test, sans impact sur les machines en production.

Un poste d'administration a été utilisé pour :

- [REDACTED]
- [REDACTED]
- [REDACTED]

## 2. Outils utilisés

Les outils principaux utilisés pour le PoC sont :



- **Fichier BIOS personnalisé** : image fournie par le constructeur, adaptée à l'affichage 180°.
- **Outils de contrôle** :
  - relevé des paramètres BIOS avant et après mise à jour (CPU, boot, ME, Secure Boot, etc.) ;
  - calcul de hash de l'image BIOS (intégrité du fichier) ;
  - export de configuration (DSF) pour comparaison.

## 3. Déroulement du PoC

Le PoC s'est déroulé en plusieurs étapes.

### a) Préparation

- [Redacted]
- [Redacted]
- [Redacted]

### b) Démarrage

- [Redacted]
- [Redacted]

### c) Vérification de l'image BIOS

Avant toute écriture sur la puce, une vérification de l'image a été effectuée avec :



L'objectif est de contrôler l'intégrité de l'image et sa compatibilité, **sans flasher** le BIOS.

Les retours de l'utilitaire n'ayant remonté aucune erreur, l'image a été considérée comme exploitable.

### d) Première tentative de flash

Une première mise à jour a été lancée avec la commande :



Les paramètres utilisés ont les rôles suivants :

- 
- 
- 
- 
- 

Sur cette plateforme, l'option [ ] n'est toutefois **pas supportée** :

l'utilitaire a retourné un code d'erreur de type :

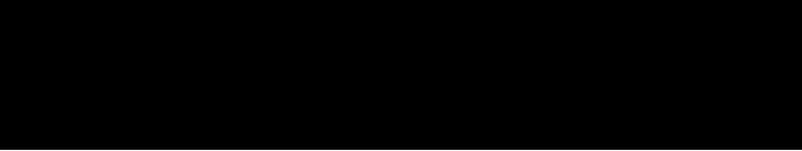


Cette erreur indique que la fonction de préservation automatique de la configuration ne peut pas être utilisée.

La fiabilité de cette première tentative n'étant pas garantie, il a été décidé de relancer la mise à jour avec des paramètres adaptés.

**e) Seconde tentative de flash (validée)**

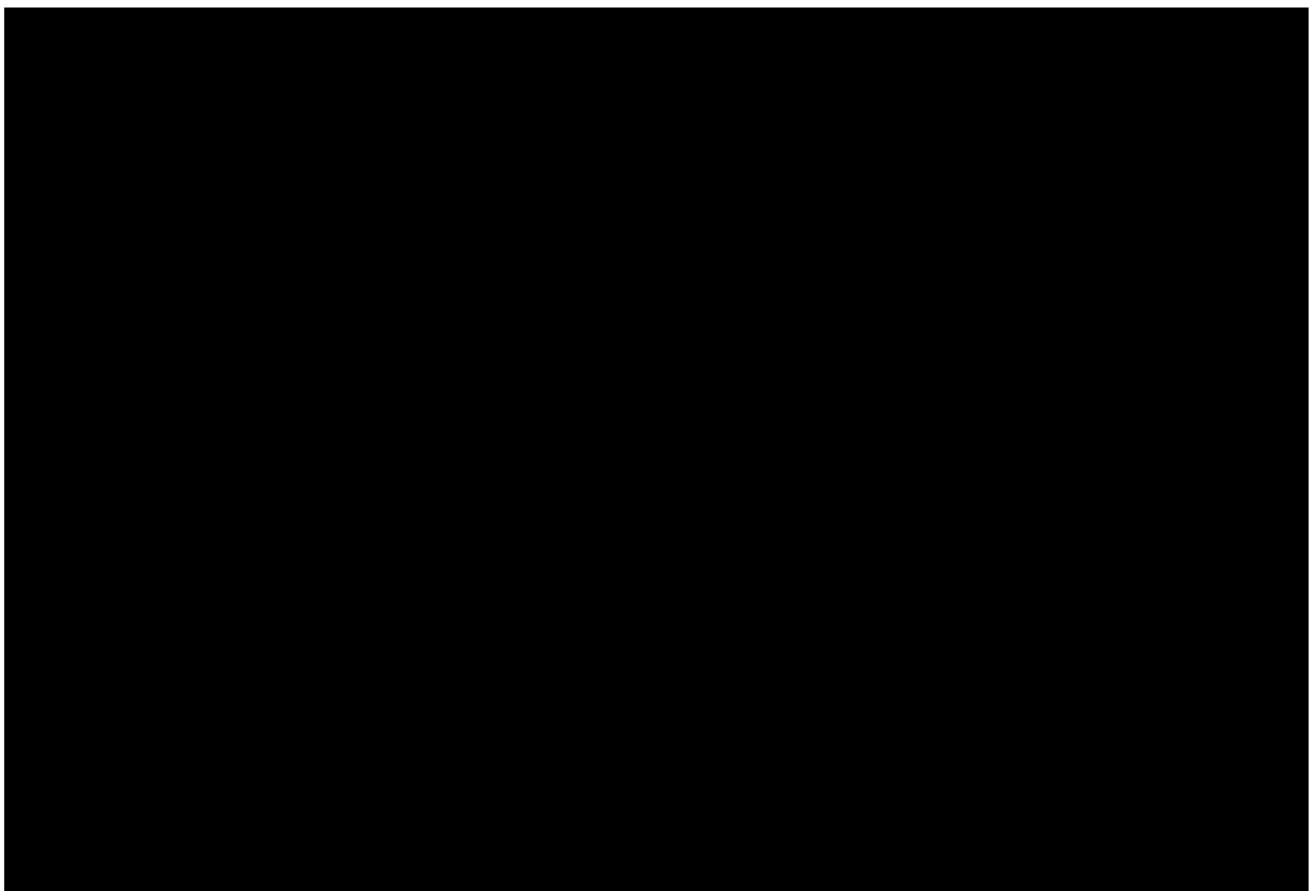
Une seconde mise à jour a donc été réalisée avec une commande simplifiée,



Cette fois, la mise à jour du BIOS s'est déroulée **sans erreur**.

Après redémarrage, les paramètres critiques du BIOS

[REDACTED] ont été **revérifiés et réappliqués** manuellement en s'appuyant sur le relevé effectué lors de l'analyse de l'existant.



## 4. Limites constatées et ouverture vers la suite

Le PoC a permis de confirmer que :

- le BIOS personnalisé fonctionne correctement : l'affichage est bien pivoté à 180° au niveau firmware et la carte reste pleinement fonctionnelle après mise à jour ;
- la procédure de flash [REDACTED] est maîtrisée et reproductible.

En revanche, le PoC a également mis en évidence une limite importante :

- l'OS embarqué [REDACTED] ne suit pas automatiquement l'orientation définie dans le BIOS ;
- [REDACTED]

Cette limite a conduit à ouvrir un **deuxième chantier distinct** (documenté séparément) visant à :

- adapter la configuration [REDACTED] (paramètre orientation de la [REDACTED]) ;
- garantir que l'affichage et le tactile restent cohérents avec la rotation définie dans le BIOS.

Ce document se concentre sur la partie **mise à jour du BIOS**.

La partie **adaptation de l'OS** [REDACTED] fait l'objet d'une documentation dédiée.

## V. Livraison :

### Livrables

À l'issue du projet, les livrables suivants sont mis à disposition :

- **Une platine de référence** avec le BIOS personnalisé installé et fonctionnel, permettant :
  - de vérifier l'affichage pivoté à 180° au niveau BIOS ;
  - de servir de modèle pour les futurs déploiements (lot pilote).
- **Une procédure formalisée de mise à jour du BIOS personnalisé**, destinée à l'équipe système, comprenant :
  - [REDACTED]
  - [REDACTED]
  - les points de contrôle après flash (vérifications dans le BIOS, comportement au démarrage) ;
  - les éléments de retour arrière en cas de problème.

Cette procédure est fournie en annexe sous la référence :

- **Annexe X – Procédure P-BIOS-01 : Mise à jour du BIOS personnalisé (NDA)**

## Critères d'acceptation

La livraison est considérée comme conforme si :

- la carte démarre normalement après mise à jour du BIOS ;
- la version de BIOS correspond à la version personnalisée attendue ;
- l'affichage au niveau BIOS est bien pivoté à 180° conformément au besoin initial ;
- aucun dysfonctionnement critique n'est constaté (erreurs au POST, instabilité, impossibilité d'accéder au BIOS) ;
- la procédure de mise à jour peut être appliquée à l'identique sur une seconde platine de test avec le même résultat.

L'adaptation de l'affichage au niveau du système d'exploitation [REDACTED] et la gestion de la rotation côté OS ne font pas partie du périmètre de ce livrable et sont traitées dans un document séparé.

## Projet II Rotation OS embarqué

### I. Contexte

Dans la continuité d'un premier projet portant sur la mise en place d'un BIOS personnalisé (rotation de l'affichage à 180° au niveau firmware), un second besoin est apparu : garantir que **le système d'exploitation embarqué** présente lui aussi l'affichage dans le bon sens, de manière cohérente avec le BIOS et avec l'écran tactile intégré à la platine.

En effet, après la mise à jour du BIOS, il a été constaté que l'OS remisait parfois l'écran “à l'endroit”, indépendamment de la rotation firmware. Le projet décrit dans ce document vise donc à :

- comprendre le fonctionnement de la couche graphique utilisée par l'OS embarqué ;
- valider une solution de rotation au niveau OS (affichage + tactile) ;
- l'industrialiser sous forme de procédure et/ou de script d'installation.

## II. Analyse de l'existant côté OS

L'OS embarqué fourni avec la platine est basé sur [REDACTED] complété par un script d'installation qui déploie :

- [REDACTED]
- [REDACTED]

Dans un premier temps, l'OS a été abordé comme un environnement [REDACTED]

- [REDACTED]
- [REDACTED]

Les premiers tests ont cependant montré l'absence :

- [REDACTED]
- [REDACTED]
- [REDACTED]

À ce stade, la technologie graphique réellement utilisée par l'OS embarqué n'était pas encore clairement identifiée, ce qui a motivé la mise en place [REDACTED] dans un environnement contrôlé pour explorer les solutions possibles côté OS.

### III. Étude des solutions d'orientation

#### III.1 [REDACTED] (maquette)

Dans un premier temps, [REDACTED] a été mise en place sur une platine de test afin de :

- comprendre le comportement de l'écran en rotation 180° ;
- valider les possibilités de correction au niveau OS dans un environnement graphique standard.

Les étapes principales ont été :

- [REDACTED]
- [REDACTED]

#### a) Recensement des sorties vidéo et du périphérique tactile

Avant de modifier quoi que ce soit, une phase de reconnaissance matériel a été réalisée pour identifier :

- la **sortie vidéo** correspondant à l'écran intégré ;
- le **périphérique d'entrée tactile** associé.

Les commandes suivantes ont été utilisées :

# Liste des sorties vidéo disponibles



# Liste des périphériques d'entrée (claviers, souris, tactile, etc.)



Cela a permis de confirmer que :

- l'écran intégré utilise la sortie vidéo [REDACTED] avec une résolution de [REDACTED]
- l'écran tactile apparaît comme un périphérique de type pointer [REDACTED]

Cette étape est importante car :

- la rotation doit être appliquée à la **bonne sortie vidéo** [REDACTED]
- la transformation doit être appliquée au **bon périphérique tactile** [REDACTED]

### b) Rotation de l'affichage et calibration du tactile

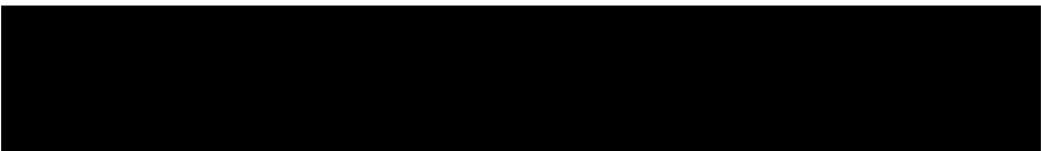
Une fois les périphériques identifiés, les tests suivants ont été réalisés :

- **Rotation de l'affichage**

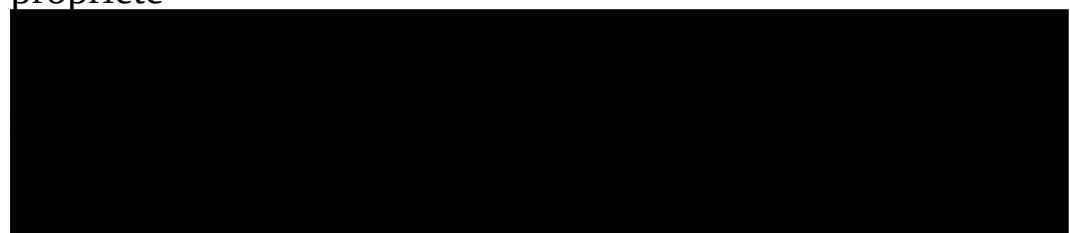
Visuellement, l'écran est bien pivoté à 180°.  
En revanche, à ce stade, le tactile reste dans son repère “normal” : les coordonnées envoyées ne correspondent plus à ce qui est affiché.

- **Identification et transformation du tactile**

1. Récupération du nom exact du périphérique tactile avec :



2. Application d'une **matrice de transformation 3x3** sur la propriété



-1 0 1 \

0 -1 1 \

0 0 1

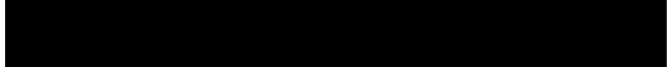
3. Cette matrice permet de transformer les coordonnées ( $x, y$ ) produites par l'écran tactile de la même façon que l'affichage a été pivoté :

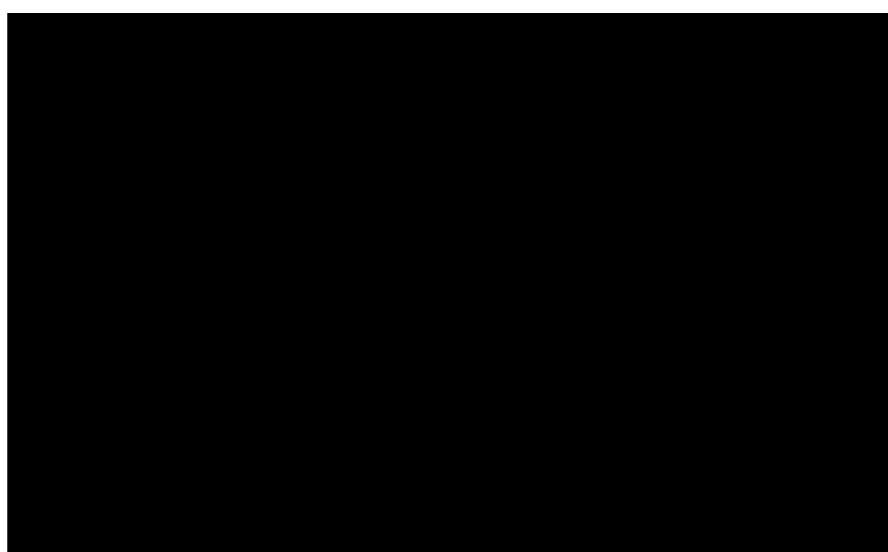
- les valeurs -1 appliquent une inversion sur les axes ;
- les 1 dans la dernière colonne compensent cette inversion pour que l'origine reste dans le coin attendu de l'écran.

Après quelques essais et ajustements, les tests ont montré que :

- les coins de l'écran correspondent bien aux coins de la zone tactile
- le déplacement du doigt suit correctement les éléments affichés, malgré la rotation 180°.

Une configuration **persistante** a ensuite été préparée dans

- 
- démarrer directement avec l'affichage en 180° ;
  - appliquer automatiquement la transformation du tactile au démarrage, sans commande manuelle.



Dossier E5 – Jean POULARD BORE

Cette maquette [REDACTED]

- l'affichage et le tactile partagent une même logique de coordonnées ;
- toute rotation d'écran doit être accompagnée d'une transformation équivalente du tactile pour garder un système exploitable.

### III.2 Limites [REDACTED]

Cette approche [REDACTED] correction logicielle de la rotation et du tactile est techniquement maîtrisée dans un environnement graphique standard.

Cependant, elle présente une limite majeure dans le contexte du projet :

- l'OS embarqué en production [REDACTED]  
[REDACTED]
- les fichiers de configuration [REDACTED]

La maquette [REDACTED]

- comme **base de compréhension technique** (rotation + matrice tactile) ;
- mais elle ne peut pas être utilisée telle quelle en production sur les platines clients.

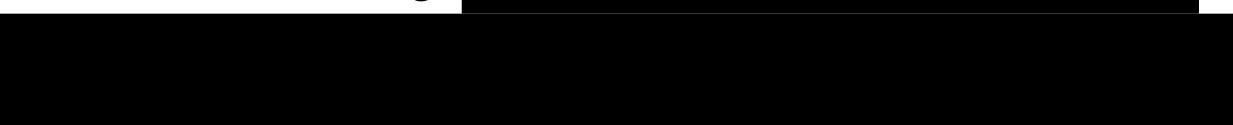
Cette limite a conduit à approfondir l'analyse de l'OS embarqué pour [REDACTED]

### III.3 Piste [REDACTED]

L'analyse détaillée du script d'installation [REDACTED] sur la machine a permis d'identifier les composants suivants :

- [REDACTED]
- [REDACTED]

Autrement dit, l'affichage [REDACTED]



- [REDACTED]
- [REDACTED]

L'export de cette configuration a mis en évidence un paramètre orientation par sortie [REDACTED] avec les valeurs possibles :

- [REDACTED]

L'objectif du PoC est donc d'adapter ce paramètre orientation côté [REDACTED] pour  **sur l'orientation 180° déjà choisie au niveau BIOS, tout en vérifiant que le tactile reste cohérent.**

## IV. Mise en place du PoC

### IV.1 PoC [REDACTED]

Sur la base de la maquette décrite précédemment, [REDACTED]  
de valider que :

- [REDACTED]
- [REDACTED]
- [REDACTED]

Cette étape sert de **référence technique** : elle a permis de comprendre  
comment l'orientation et le tactile doivent être gérés, [REDACTED]

## IV.2 PoC [REDACTED]

Le PoC a ensuite été transposé sur la stack réelle en agissant [REDACTED]

### 1. Export de la configuration d'affichage

- [REDACTED]
- [REDACTED]

### 2. Modification de l'orientation

Le paramètre a été adapté comme suit :

### 3. Réimport de la configuration

## 4. Résultats

- l'interface graphique [REDACTED] désormais affichée dans le bon sens (alignée avec le BIOS) ;
- le tactile reste cohérent (les points de contact suivent correctement l'orientation), [REDACTED]

Le PoC a donc validé que la rotation de l'OS peut être gérée **proprement** via la configuration [REDACTED]

## V. Mise en place en production

Pour la mise en production, l'objectif est d'intégrer cette adaptation directement au **script d'installation** de l'OS, afin que :

- l'orientation soit automatiquement réglée sur les platines ciblées ;
- aucune intervention manuelle ne soit nécessaire après installation.

Concrètement, le script :

- 
- 
- 
- 

Une évolution possible consiste à conditionner cette modification à la **version du BIOS**

## **VI. Rédaction de la procédure / VII. Livraison**

- [REDACTED]
- livrables :
  - une platine de référence avec OS + affichage + tactile OK,
  - une procédure documentée,
  - éventuellement un script d'installation mis à jour.

## Projet III Déploiement d'images sur PC embarqués

### I. Contexte

Dans le cadre de la mise à jour du service web [REDACTED]  
[REDACTED] sur des robots industriels, il a été décidé de revoir le système  
de déploiement [REDACTED]

- [REDACTED]
- [REDACTED]

#### Limites constatées :

- [REDACTED]
- [REDACTED]
- [REDACTED]
- [REDACTED]
- [REDACTED]

Le **SI** a été sollicité pour industrialiser le processus, mutualiser les compétences, réduire les temps d'immobilisation des robots et sécuriser la chaîne de déploiement.

**Périmètre** : création d'une plate-forme d'imagerie réseau permettant la capture d'une machine de référence et le déploiement simultané sur plusieurs hôtes compatibles ( [REDACTED] )

**Hors périmètre** : industrialisation définitive en production multi-sites (phase 2), intégration AD, supervision centralisée.

**Parties prenantes** : Équipe WEB, SI, Intégrateurs

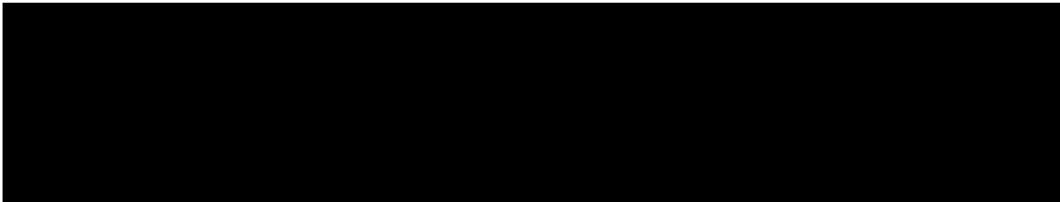
**Objectifs mesurables** :

- Déployer **≥ 5 postes/robots en parallèle**.
- Temps d'installation par machine < **20 min** (hors transfert d'image).
- Traçabilité : log de déploiement par hôte (date/heure, image utilisée, opérateur).
- Zéro impact sur le LAN d'entreprise durant la phase PoC.

## II. Étude des solutions

Options considérées :

1.



2. **FOG Project** (open-source PXE/iPXE, TFTP, NFS)

Deux modes :

- **ProxyDHCP** (dnsmasq) sur le LAN d'entreprise :
  - n'interfère pas avec DHCP existant ; – plus complexe, sensible au réseau (VLAN, filtrage, course DHCP).
- **LAN isolé** avec FOG qui fait DHCP :
  - zéro risque pour le LAN, débug simple, reproductible ; – nécessite un petit switch et câblage dédié.  
→ **Retenu pour le PoC : LAN isolé.**

3. **Solutions commerciales** (MDT/WDS, Ivanti, Acronis Snap Deploy...)

- Support éditeur, intégrations
  - Coûts licences + délais d'approvisionnement → **non retenu** court terme.

**Critères de choix** : sécurité (pas d'impact LAN), coûts (0€ logiciel), rapidité de mise en place, scalabilité, traçabilité.

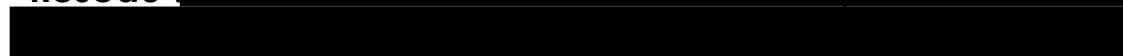
**Décision : FOG en LAN isolé** pour le PoC ; ProxyDHCP envisagé en **phase 2** après validation et garde-fous réseau.

### III. Proposition (POC)

#### Architecture technique :

(cf. Annexe : 1.1-Archi-PXE)

- Réseau :

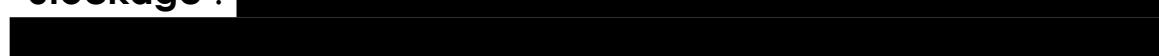


- Serveur FOG :

rôles :

- [Redacted]
- [Redacted]
- [Redacted]
- [Redacted]

- Stockage :



- Sécurité :

- [Redacted]
- [Redacted]
- [Redacted]

- Bonnes pratiques image :

- [Redacted]
- [Redacted]

**Processus simplifié:**

(cf. Annexe : 1.2-Procédure-Simplifié)

1. **Capture** d'une machine de référence (Upload).
2. **Déploiement** sur cibles (Download), en parallèle.
3. **Traçabilité** : FOG garde l'historique par hôte (imaged, tasks).

**RACI :**

- **SI/Infra** : design, installation FOG, exploitation.
- **Web** : fourniture de la machine master, validation fonctionnelle.
- **Intégration** : tests à l'usage.

## IV. Mise en place / PoC

Étapes réalisées :

- Choix [REDACTED]
- Installation FOG  
[REDACTED]
- Vérifications :

- [REDACTED]
- [REDACTED]
- [REDACTED]

- Boot PXE client → **menu FOG, Full Host Registration** ok.
- **Capture** : [REDACTED]

**Procédure d'exploitation (résumé) :**

(cf. Doc : Procédure-Déploiement-PXE) (NDA)

1. **Créer l'Image** [REDACTED]
2. **Enregistrer le host** (Full Host Registration).
3. **Assigner l'image** au host.
4. **Tasks** → **Upload** (capture) → vérifier /images/<nom> et colonne *Imaged*.
5. **Déployer** : Assign Image + **Tasks** → **Download** sur cibles.

## V. Livraison

Livrables :

- Serveur FOG opérationnel [REDACTED]
- Stockage [REDACTED]
- 1 image de référence **capturée & testée** [REDACTED]
- Dossier projet (présent document) + **runbook** d'exploitation.

Critères d'acceptation :

- Boot PXE → menu FOG visible en < 10 s.
- Enregistrement hôtes OK ; Tasks Upload/Download fonctionnelles.
- Déploiement sur **≥ 2 cibles en parallèle** sans erreur.
- PV de recette fonctionnelle signé ([REDACTED])

Plan de passage en production (phase 2) :

- Option A : [REDACTED]
- Option B : [REDACTED]

- [REDACTED]
- [REDACTED]
- [REDACTED]

## VI. Annexes :

