

19/05/2025

Rapport de projet DIANGO Akouman Projet ACCÈS LYCÉE

Badge RFID et Interface de gestion



Table des matières

Introduction Générale	2-3
But du projet.....	4
Diagramme de Gantt	5
RÉPARTITION DES TACHES	6
ANALYSE UML.....	7
DIAGRAMME.....	8
Diagramme de déploiement.....	8-9
Diagramme de cas d'utilisation.....	10-11
ARCHITECTURE GLOBALE DU SYSTEME.....	12
PILOTAGE DU BARIONET.....	13
Présentation du BARIONET100.....	13-14
Configuration et Mise en Œuvre.....	15-21
MISE EN OEUVRE DES DIFFÉRENTS LECTEURS DE BADGES	22
Présentation du lecteur MINOVA MCR04.....	22-28
Méthode de Communication entre le Lecteur MCR04 et le Serveur.....	29-31
Présentation du lecteur RFID Inveo USB Desk.....	32-33
ANNEXES.....	34

INTRODUCTION GÉNÉRALE

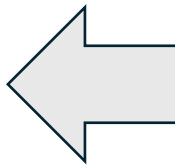
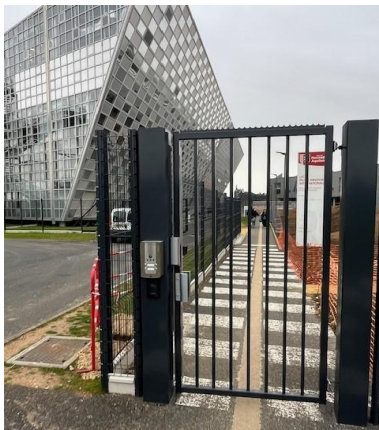


Le BTS CIEL est une formation de deux ans qui te prépare à travailler dans le secteur de l'informatique. Un des aspects cruciaux de cette formation est le Projet E6. C'est un travail pratique où tu mets en œuvre les compétences acquises durant notre formation. C'est également l'occasion de faire valoir notre esprit d'analyse et notre sens de l'organisation.

Présentation

Dans un monde où la sécurité et la gestion des accès sont devenues des enjeux majeurs, il est essentiel de mettre en place des systèmes performants et fiables.
















Ce projet vise à reproduire et développer un système de gestion des accès basé sur le lycée du Pilote Innovant International.



Matériels Utilisés :

Désignation	Caractéristiques
<ul style="list-style-type: none"> • 2 lecteurs de badges et des badges RFID • Volet • Une caméra IP • Un capteur de présence photoélectrique 	<ul style="list-style-type: none"> • De la marque Invéo et Minova • Matériel de récupération • De la marque D-Link DCS-5000L • De la marque OSIRIS

Logiciels Utilisés :

Désignation	Caractéristiques
<p>Base de données :</p>  <p>Pilotage du Barionet :</p>     <p>Création des interfaces :</p>   	<p>Moteur : Application :</p>   <p>Application :</p>     <p>Application :</p> 

BUT DU PROJET

L'objectif principal est d'automatiser l'identification des utilisateurs grâce à des badges RFID, tout en permettant une gestion centralisée des accès via un serveur. Ce système devra être capable de :

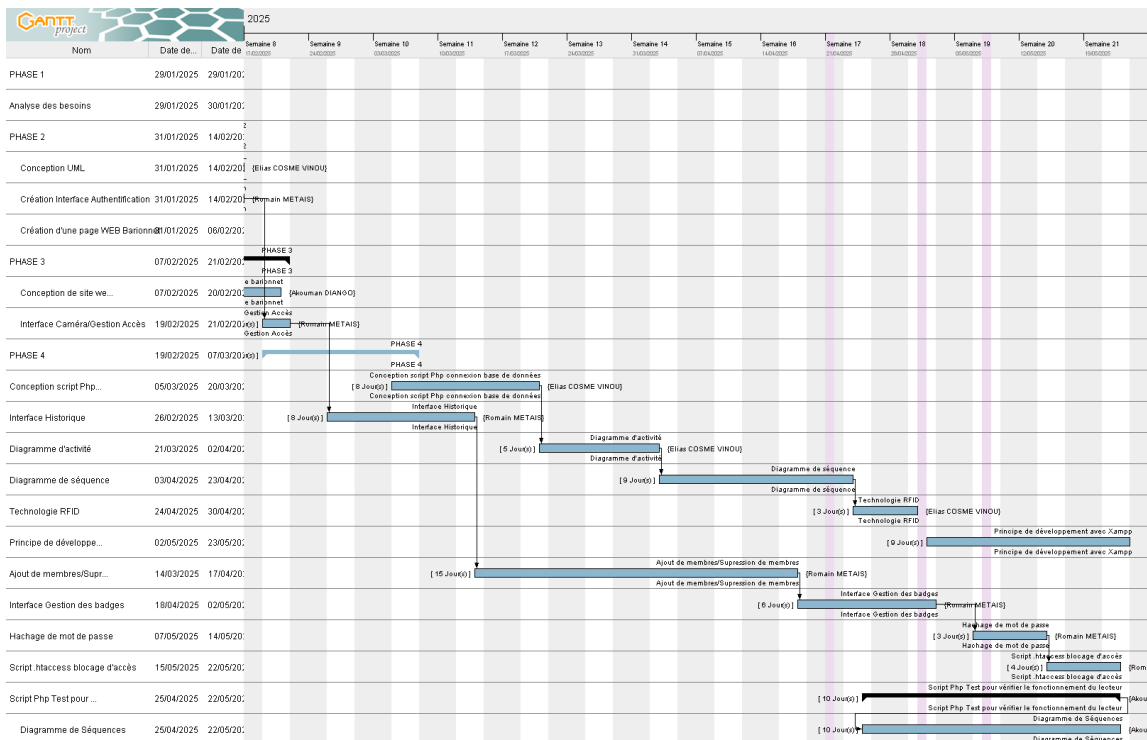
- Lire et vérifier les informations contenues dans un badge RFID
- Accorder ou refuser l'accès en fonction des droits attribués à l'utilisateur
- Enregistrer les événements d'accès dans une base de données pour assurer un suivi
- Intégrer une interface de gestion pour les administrateurs et techniciens.

Pour mener à bien ce projet, nous avons adopté une approche structurée en plusieurs étapes :

1. Analyse des besoins et conception du système, en nous basant sur les diagrammes UML pour modéliser l'architecture et les interactions.
2. Mise en place de l'infrastructure matérielle, incluant des lecteurs RFID, un serveur, une base de données et un module Barionnet100 permettant de piloter les actionneurs.
3. Développement du système, avec l'implémentation du logiciel en utilisant des technologies comme PHP et MySQL.
4. Tests et validation, afin d'assurer le bon fonctionnement du dispositif et sa robustesse.

Ce document détaille l'ensemble des aspects du projet, depuis la conception jusqu'à l'intégration finale du système, en passant par les choix techniques et les contraintes rencontrées.

DIAGRAMME DE GANTT



Le diagramme de Gantt est un outil qui permet de représenter visuellement l'état l'avancement des différentes tâches qui constituent le projet.

Celui ci-dessus est séparé en 4 phases et relate l'avancement de chaque tâche qu'on a ou est en train d'effectuer sur le projet :

- Analyse des besoins (Phase 1)
- Conception UML (Phase 2)
- Conception Interface Authentification (Phase 2)
- Création de Page Web Sorties Barionet (Phase2)
- Conception Site Web Technicien/Agent Barionet (Phase3)
- Création Interface Gestion d'Accès (Phase3)
- Conception Base de Données (Phase4)
- Ouverture des Portes via les badges (Phase4)
- Conception Script PHP(Phase4)
- Interface Historique (Phase4)
- Technologie RFID (Phase4)
- Principe de développement (Phase4)
- Ajout de membres/Suppression (Phase4)
- Interface de Gestion des badges (Phase4)
- Hachage de mots de passe (Phase4)
- Script .htaccess blocage d'accès (Phase4)
- Script Php test pour vérifier le fonctionnement du lecteur Minova(Phase4)
- Diagramme de séquence (Phase4)

RÉPARTITION DES TACHES

Etudiant	Tâches
Akouman DIANGO	<ul style="list-style-type: none">• Pilotage du Barionet 100 via un site web• Mise en place du lecteur RFID Invéo• Configuration et mise en œuvre du lecteur RFID MINOVA MCR04
Elias COSME VINOUE	<ul style="list-style-type: none">• Conception UML• Mise en place de la base de données• Communication avec le système RFID
Romain METAIS	<ul style="list-style-type: none">• Développement de l'interface de l'agent d'accueil• Développement de l'interface du technicien• Communication avec le système RFID

ANALYSE UML

Langage UML

Le langage de modélisation unifié (UML) est le langage standard que de nombreux ingénieurs logiciels et de nombreuses entreprises utilisent pour avoir une vue d'ensemble de systèmes complexes.

Les différents diagrammes que nous allons vous proposer sont des diagrammes créés avec ce langage.



DIA

Dia est un logiciel libre de création de diagramme.

Dia est conçu de manière modulaire avec plusieurs paquetages de formes pour des besoins différents : diagramme de flux, diagramme de circuit électrique, diagramme UML, etc.

A partir de ce logiciel nous avons conçu les différents diagrammes un à un. Les diagrammes UML les plus adaptés à notre projet sont :



- Diagramme de déploiement
- Diagramme de cas d'utilisation
- Diagramme d'activité
- Diagramme de séquences

Un algorithme a été conçu afin d'avoir une représentation détaillée de notre projet avec le tout cumulé.

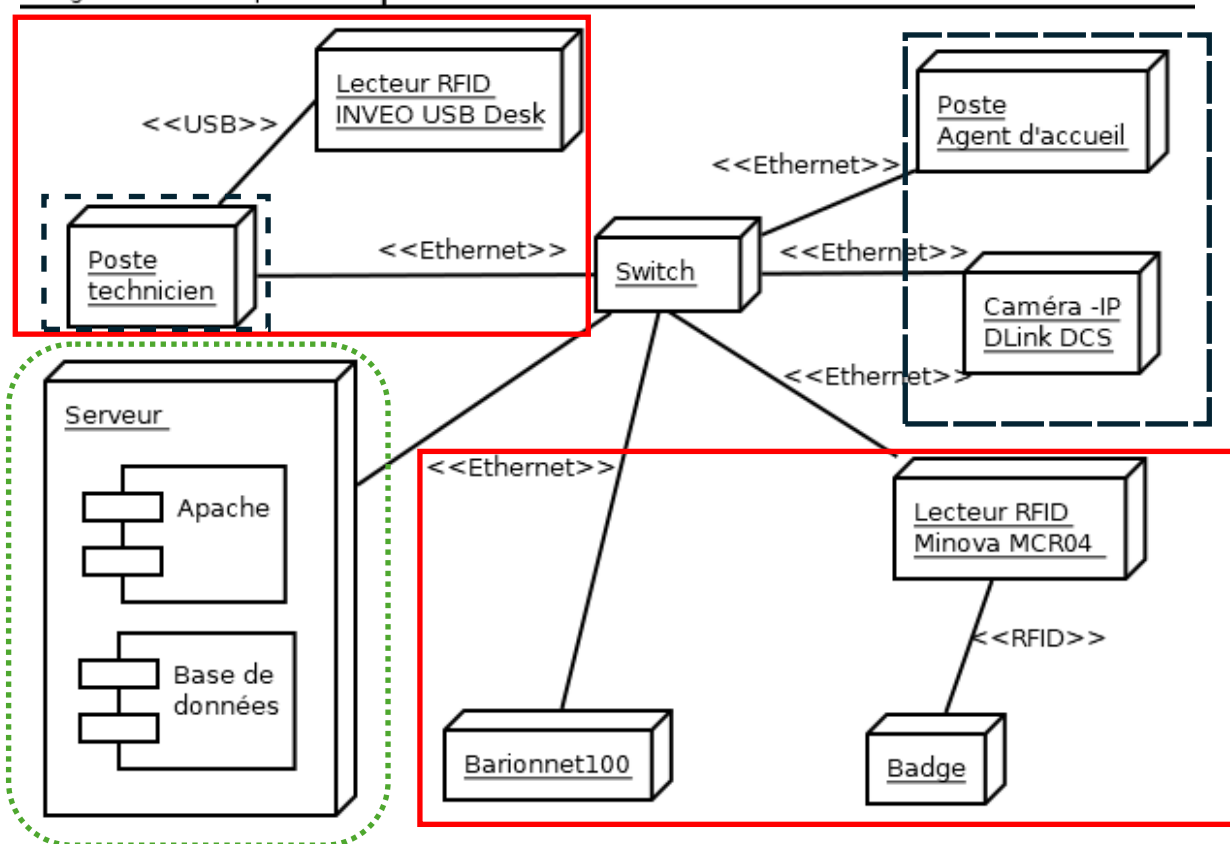
DIAGRAMMES

Diagramme de déploiement

En UML, les diagrammes de déploiement modélisent l'architecture physique d'un système. Ces derniers affichent les relations entre les composants logiciels et matériels du système, d'une part, et la distribution physique du traitement, d'autre part.

Ils présentent la disposition physique des nœuds dans un système réparti, les artefacts qui sont stockés sur chaque nœud et les composants et autres éléments que les artefacts implémentent. Les nœuds représentent des périphériques matériels tels que des ordinateurs, des détecteurs et des imprimantes, ainsi que d'autres périphériques qui prennent en charge l'environnement d'exécution d'un système. Les chemins de communication et les relations de déploiement modélisent les connexions dans le système.

Diagramme de déploiement



- Partie de Akouman DIANGO
- Partie de Elias COSME VINO
- - - Partie de Romain METAIS

Ce diagramme de déploiement représente l'architecture d'un système utilisant des lecteurs RFID, un serveur, des postes de travail et des périphériques connectés via un réseau Ethernet. Voici une description détaillée de chaque composant :

Serveur (en vert, partie de Elias COSME VINOUE)

- Héberge Apache (pour gérer l'application web).
- Contient une Base de données (stockage des informations RFID et autres données du système).

Poste technicien (en bleu pointillé, partie de Romain METAIS)

- Connecté via Ethernet au réseau.
- Relié à un Lecteur RFID INVEO USB Desk via USB, permettant la lecture des badges RFID.

Poste agent d'accueil (en bleu pointillé, partie de Romain METAIS)

- Connecté au réseau via Ethernet.
- Associé à une Caméra IP D-Link DCS pour la surveillance.

Switch (élément central du réseau)

- Connecte les différents équipements via Ethernet.

Barionnet100 (en rouge, partie de Akouman DIANGO)

- Connecté au réseau Ethernet.

Lecteur RFID Minova MCR04 (en rouge, partie de Akouman DIANGO)

- Connecté au réseau via Ethernet.
- Utilisé pour lire les Badges RFID.

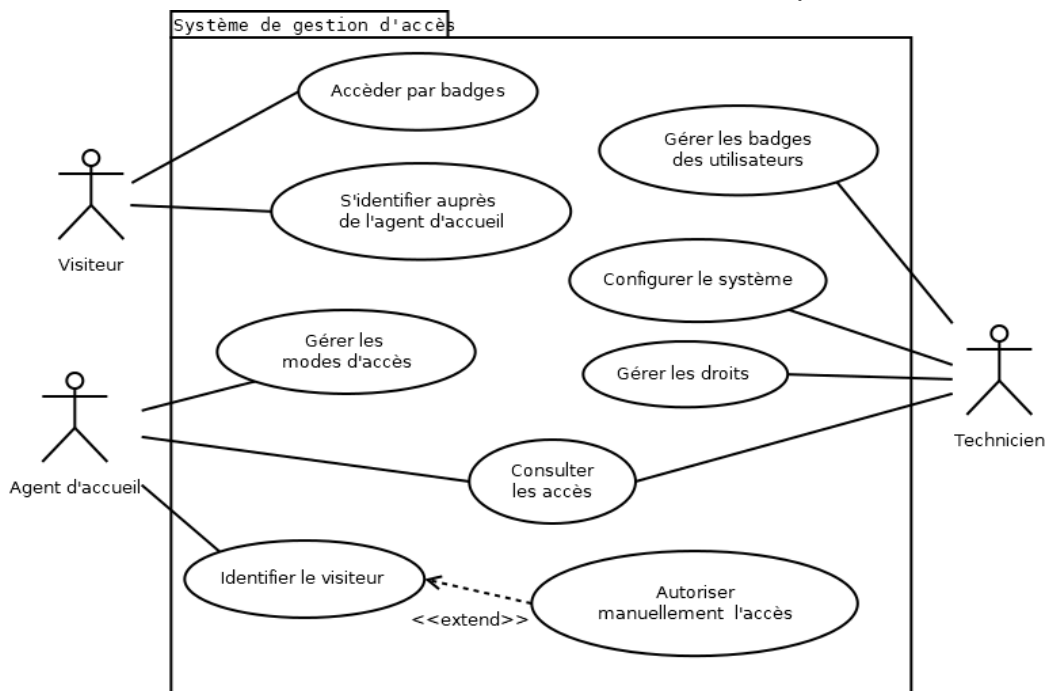
Diagramme de cas d'utilisation

En langage de modélisation unifié (UML), un diagramme de cas d'utilisation peut servir à résumer les informations des utilisateurs de votre système (également appelés acteurs) et leurs interactions avec ce dernier. La création de ce type de diagramme UML requiert un ensemble de symboles et de connecteurs spécifiques. Lorsqu'ils sont bien conçus, les diagrammes de cas d'utilisation peuvent aider notre équipe à collaborer et représenter :

- Les scénarios dans lesquels notre système ou application interagit avec des personnes, des organisations ou des systèmes externes ;
- Les objectifs que notre système ou application permet aux entités (appelées acteurs) d'atteindre ;
- La portée de notre système.

Les diagrammes sont généralement composés :

- **D'acteurs** : utilisateurs qui interagissent avec un système. Un acteur peut être une personne, une organisation ou un système externe qui interagit avec votre application ou votre système. Il s'agit nécessairement d'objets externes qui produisent ou consomment des données.
- **Du système** : séquence spécifique d'actions et d'interactions entre les acteurs et le système. Un système peut également être appelé scénario.
- **Des objectifs** : résultat final de la plupart des cas d'utilisation. Un diagramme réussi doit décrire les activités et les variantes utilisées pour atteindre l'objectif.



Dans notre diagramme de cas d'utilisation les acteurs sont :

- Le visiteur
- L'agent d'accueil
- Le technicien

La balise **<<extend>>** présente dans le diagramme indique une relation optionnelle entre deux cas d'utilisation. Elle signifie qu'un cas d'utilisation (appelé extension) peut être inséré, sous certaines conditions, dans l'exécution d'un autre cas d'utilisation (appelé de base).

Dans notre diagramme, le cas d'utilisation « Identifier le visiteur » possède une extension vers le cas d'utilisation « Autoriser manuellement l'accès ». Cette relation signifie que, lors de l'identification d'un visiteur, il est parfois nécessaire pour l'agent d'accueil d'autoriser manuellement l'accès. Mais cette action n'est pas systématique : elle ne se produit que si une condition particulière est remplie (par exemple, un visiteur non reconnu par le système ou nécessitant une autorisation exceptionnelle).

On peut retenir que :

- Le cas d'utilisation « Identifier le visiteur » se déroule normalement.
- Mais, dans certains scénarios (non automatiques), une action supplémentaire (« Autoriser manuellement l'accès ») peut être exécutée, déclenchée par le cas principal.
- La relation **<<extend>>** permet donc de modéliser des options supplémentaires ou des "ajouts" qui ne sont pas toujours réalisés.

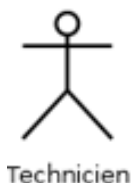
Ainsi, l'utilisation de **<<extend>>** rend le diagramme plus flexible et plus précis en illustrant les exceptions ou déroulés alternatifs possibles.



Peut accéder par badges et aussi s'identifier auprès de l'agent d'accueil.



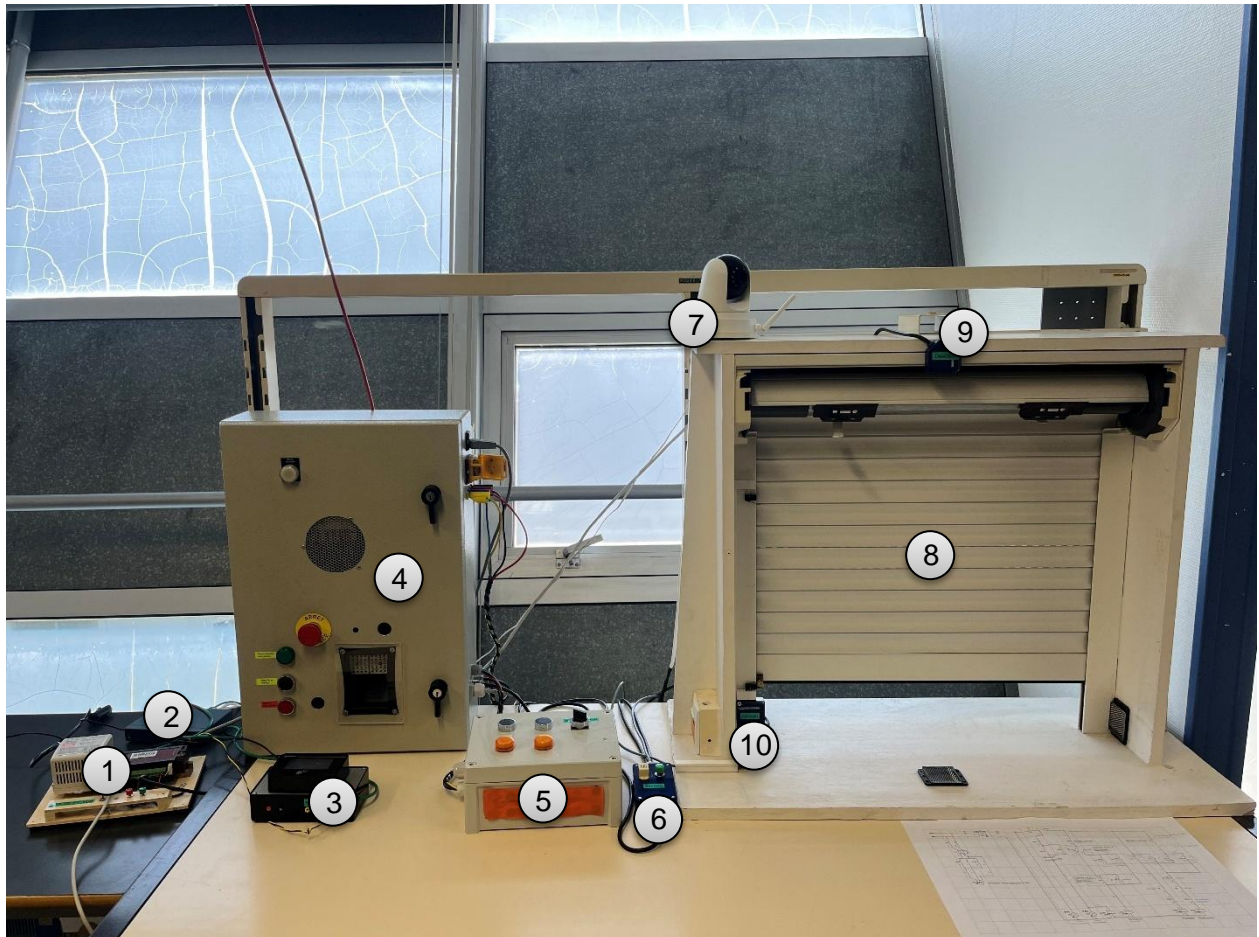
Peut gérer les modes d'accès, consulter les accès, identifier le visiteur et il peut optionnellement autoriser manuellement l'accès.



Peut gérer les badges utilisateurs, configurer le système, gérer les droits et il peut aussi consulter les accès.

ARCHITECTURE GLOBALE DU SYSTÈME

Voici la disposition matérielle complète du système, des lecteurs RFID aux actionneurs, en passant par les éléments de contrôle et de sécurité.



1. Barionet 100
2. Switch
3. Lecteur RFID Minova MCRO4
4. Armoire électrique
5. Boîtier de commande manuelle
6. Sonnette
7. Caméra de IP D-Link
8. Volet roulant (Porte)
9. Capteur de présence Osiris
10. Capteur de collision Osiris

PILOTAGE DU BARIONET

Présentation du BARIONET100

Le Barionet de Barix est une gamme de contrôleurs d'automatisation programmables et compatibles réseau permettant d'interfacer une grande variété d'appareils et de systèmes avec des réseaux IP pour les applications domotiques et industrielles.

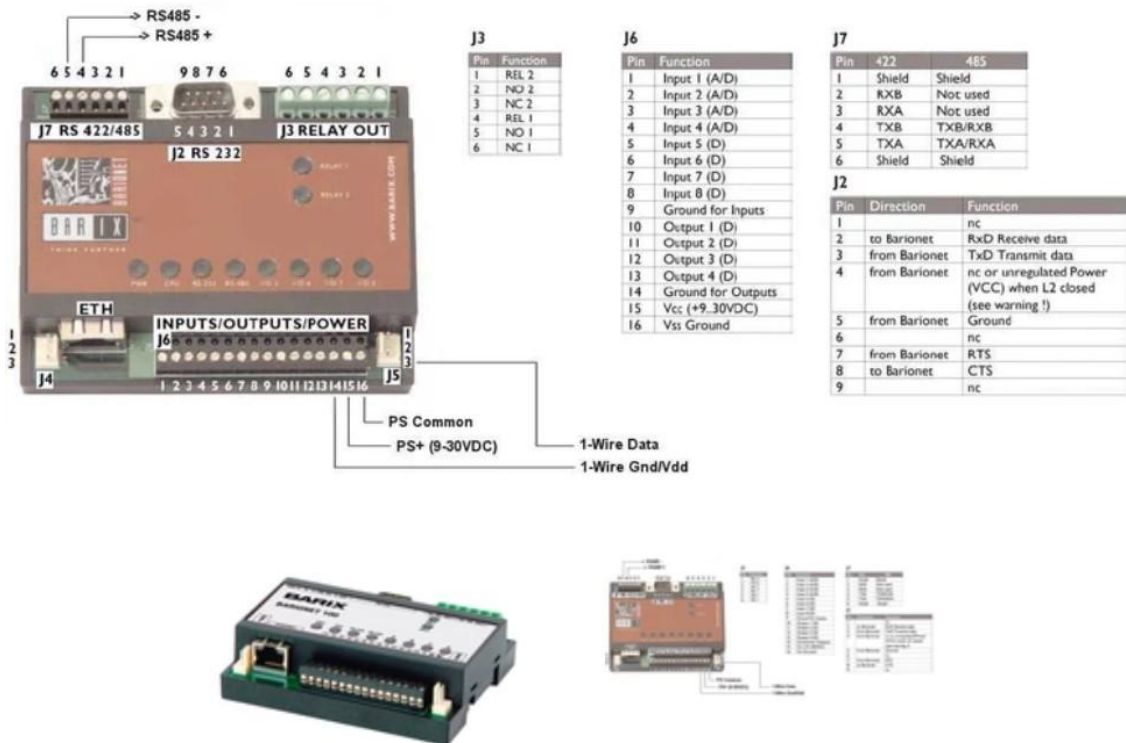
Avec Barionet, la plupart des appareils peuvent être connectés en réseau pour la surveillance et le contrôle via un navigateur web, ainsi que d'autres systèmes d'automatisation standardisés, tels que SNMP et Modbus.

Barionet offre une variété d'interfaces matérielles et logicielles standard, ainsi que des entrées et sorties polyvalentes pour les applications de contrôle et de surveillance.

Deux modèles Barionet sont actuellement disponibles : le Barionet 100 (le Barionet original) et le nouveau Barionet 50. Ces deux modèles se distinguent principalement par le nombre d'entrées et de sorties disponibles.

Caractéristiques Techniques

Fonctionnalité	Détails
Entrées/Sorties	4 sorties numériques, 4 entrées analogiques/numériques, 2 relais
Connectivité	Ethernet, RS-422/485
Protocoles	HTTP (CGI), Modbus-TCP, SNMP
Alimentation	10-24VDC
Modèles	Barionet 100 (plus complet)



Le Barionet 100 sert de :

- **Contrôleur central** : Gère les relais pour l'ouverture/fermeture des portes.
- **Interface réseau** : Communication avec le site web via HTTP/CGI.
- **Système polyvalent** : Basculer entre modes manuel/automatique.

Configuration et Mise en Œuvre

Pour la configuration du Barionet 100, j'ai opté pour un pilotage via un site web grâce aux commandes CGI intégrées à mon programme HTML, plus précisément la commande CGI est imbriquée dans du Javascript. Cette approche offre plusieurs avantages:

- **Flexibilité d'accès:** L'interface web permet un contrôle depuis n'importe quel poste connecté au réseau
- **Simplicité d'implémentation:** Pas besoin d'installer de logiciel spécifique sur les postes clients
- **Intégration facilitée** avec le système d'information existant du lycée

```
// Contrôle une sortie (ON, OFF)
function controlOutput(output, value) {
    const ip = getBarionetIP(); // Récupère l'adresse IP
    const url = "http://".concat(ip, "/rc.cgi?o=", output, ",",
value); // Construit l'URL pour la requête HTTP
```


Utilisation des commandes CGI

Une commande CGI (Common Gateway Interface) est un protocole standard qui permet à un serveur web d'exécuter des programmes externes pour générer des pages web dynamiques.

Le Barionet dispose d'un serveur web intégré qui implémente des commandes CGI principales que j'ai exploitées :

Syntaxe générale :

```
http://<adresse_IP_barionet>/rc.cgi?o=<adresse_E/S>,<valeur>[&L=<page_réponse>]
```

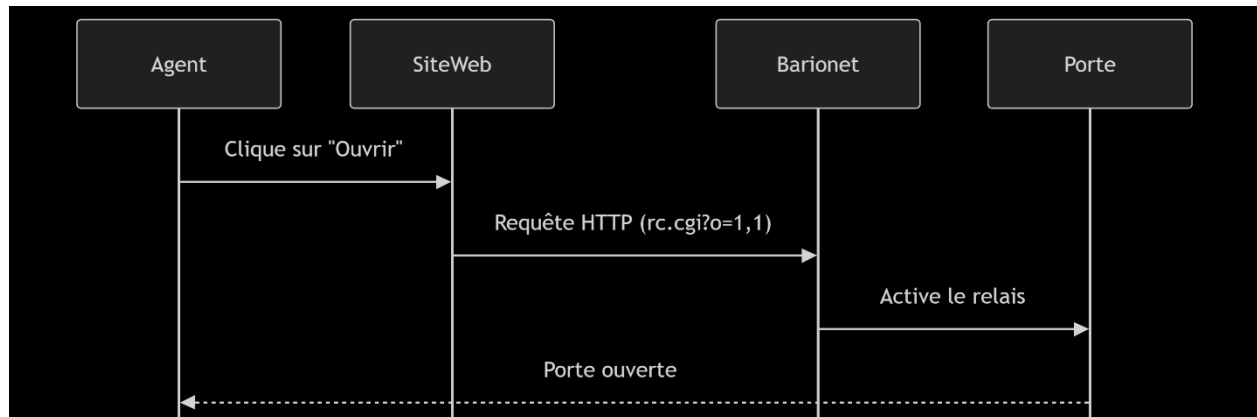
Commande	Description
<code>http://172.22.21.164/rc.cgi?o=1,1</code>	Activer le relais 1 (ouverture de la porte)
<code>http://172.22.21.164/rc.cgi?o=1,0</code>	Désactiver le relais 1 (fermeture de la porte)
<code>http://172.22.21.164/rc.cgi?o=1,10</code>	Activer le relais 1 pendant 1 seconde (ouverture temporisée)
<code>http://172.22.21.164/rc.cgi?o=2,0</code>	Configurer en mode "Badge ou Agent"
<code>http://172.22.21.164/rc.cgi?o=2,1</code>	Configurer en mode "Auto"

Ouverture de la porte via la page web par l'agent d'accueil

La communication mise en place est la suivante :

- **Côté agent** : L'interface web est chargée dans le navigateur de l'agent d'accueil (il dispose d'un bouton ouvrir pour laisser passer une personne sans badge).
- **Requêtes http** : Lorsque l'agent clique sur un bouton, le JavaScript génère une requête HTTP vers le Barionet
- **Traitement CGI** : Le serveur web du Barionet reçoit la requête et exécute la commande CGI correspondante
- **Action matérielle** : Le Barionet active ou désactive les sorties physiques en

fonction de la commande reçue



Modes de fonctionnement

Le système de contrôle d'accès au lycée est conçu pour fonctionner dans deux modes principaux gérés par le relai 2 du Barionet, permettant une flexibilité et une adaptabilité selon les besoins et les situations.

```

<!-- Contrôle de la sortie 2 -->
<div class="output-group">
  <h3>Modes de fonctionnement</h3>
  <button onclick="controlOutput(2, 0)" class="btn on">Badge ou Agent</button>
  <button onclick="controlOutput(2, 1)" class="btn off">Mode Auto</button>
</div>

```

Dans le cas où le bouton "Badge ou Agent" est cliqué, le relais est à l'état 0, une requête HTTP est envoyée au Barionet pour la mise en mode badge ou agent.

Dans le cas où le bouton "Mode Auto" est cliqué, le relais est à l'état 1, une requête HTTP est envoyée au Barionet pour la mise en mode automatique du système.

Voici une description de ces modes :

Mode "Badge ou Agent" (Relais 2 = État 0)

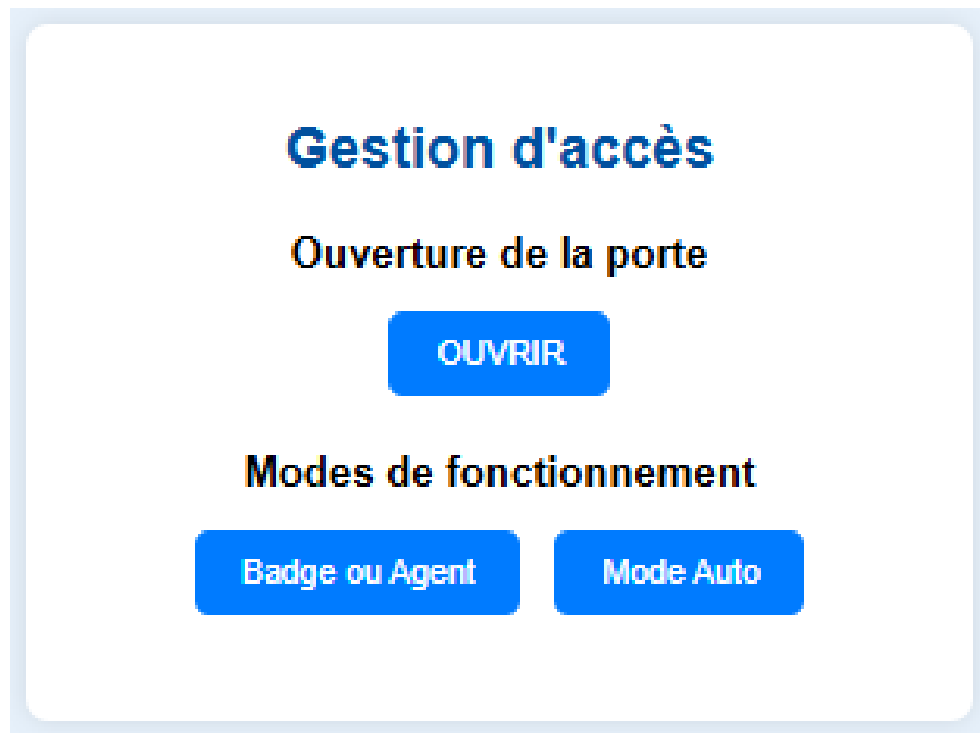
Dans ce mode, l'accès au lycée est contrôlé soit par la présentation d'un badge RFID valide, soit par l'intervention manuelle d'un agent d'accueil.

- **Accès par Badge** : Les utilisateurs (élèves, enseignants, personnel) présentent leur badge RFID devant le lecteur. Si le badge est reconnu et autorisé, le système active le relais pour ouvrir la porte.
- **Accès par Agent** : En cas d'absence de badge ou pour les visiteurs sans badge, l'agent d'accueil peut manuellement autoriser l'accès via une interface web. L'agent clique sur un bouton pour générer une requête HTTP qui active le relais de la porte.
- **Utilisation** : Ce mode est utilisé pendant les heures normales d'ouverture du lycée, assurant un contrôle strict et sécurisé de l'accès.

Mode "Auto" (Relais 2 = État 1)

Ce mode permet une ouverture automatique de la porte sans nécessiter l'utilisation de badges ou l'intervention d'un agent.

- **Détection de Présence** : Des capteurs de présence sont activés pour détecter l'approche des individus. Lorsqu'une présence est détectée, un signal est envoyé au contrôleur Barionet.
- **Ouverture Automatique** : Le Barionet active automatiquement le relais pour ouvrir la porte, permettant un accès fluide et sans entrave.
- **Utilisation** : Ce mode est particulièrement utile lors d'événements spéciaux tels que les journées portes ouvertes ou les réunions parents-professeurs, où un grand nombre de visiteurs est attendu et où un accès simplifié est souhaité.

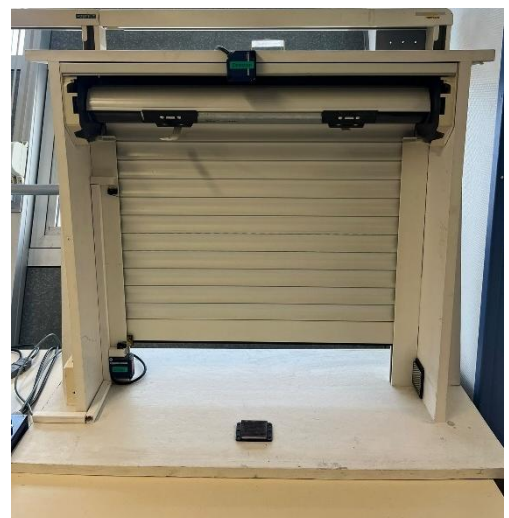


Présentation des Capteurs de Présence et de Collision Osiris

Lors d'événements spéciaux tels que des réunions parents-professeurs ou des journées portes ouvertes, le système de contrôle d'accès peut être configuré pour offrir une expérience fluide et accueillante aux visiteurs. Pendant ces événements, la porte principale est programmée pour s'ouvrir automatiquement à l'approche d'un individu, sans nécessiter l'utilisation de badges RFID.

Les capteurs **Osiris** (marque Télémécanique/Square D) utilisés jouent un rôle clé dans le système de contrôle d'accès du lycée. Deux types de capteurs sont utilisés :

- Capteurs de présence
- Capteurs de collision



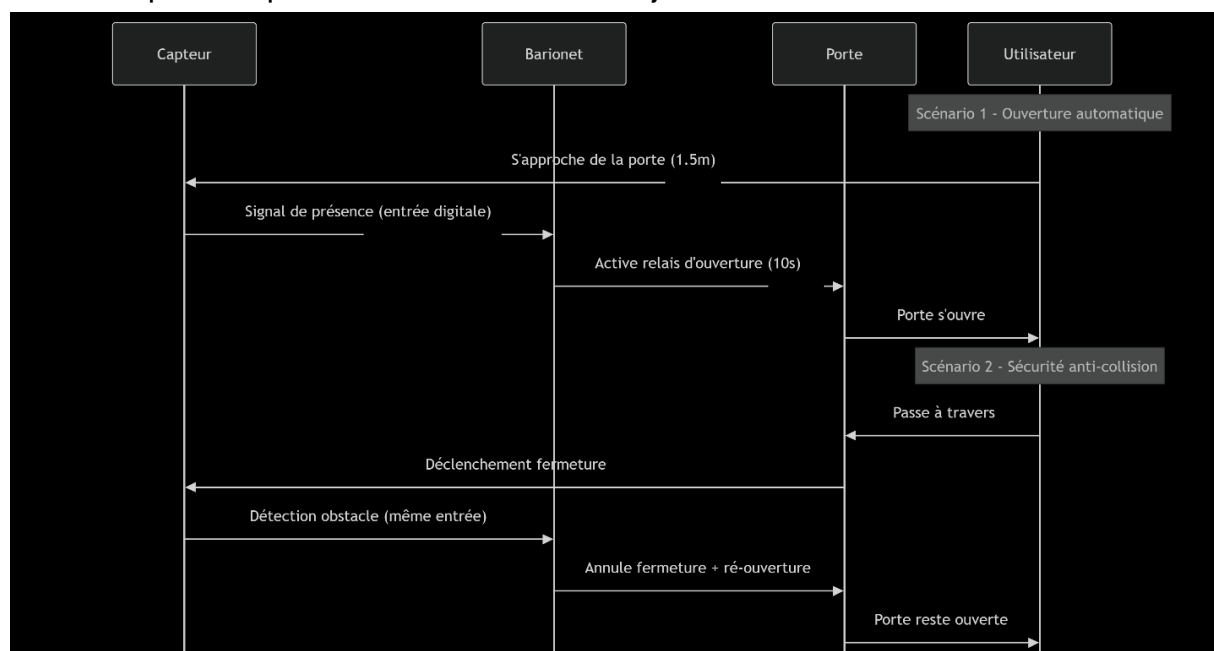
Ces capteurs partagent la même technologie et marque, assurant une intégration homogène avec le système.

Caractéristiques techniques des capteurs

Paramètre	Description
Type	Photoelectric sensors (XUK0ARCTL2)
Portée	5 m
Alimentation	24...240 V AC/DC
Connectique	Raccordement électrique par câble
Usage	Détection d'individus à l'approche ou d'obstacle

Le Barionet est programmé pour activer le relais de la porte dès qu'un signal de présence est reçu. La porte s'ouvre automatiquement, permettant ainsi un accès sans entrave pour les visiteurs.

Dans le cas où une présence est détectée pendant la fermeture de la porte c'est dans ce cas-là que le capteur de collision rentre en jeu.

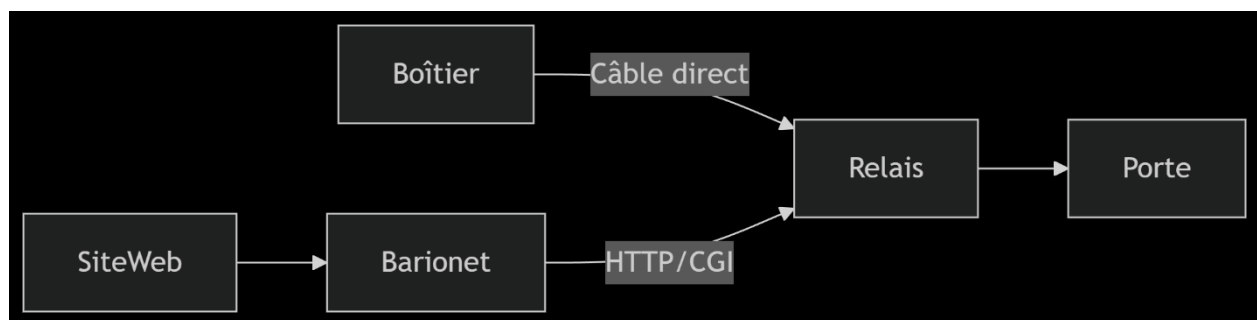


La porte reste ouverte pendant une durée prédéfinie grâce à un contacteur avec une temporisation, suffisante pour permettre le passage des visiteurs.

À la fin de l'événement, le système est reconfiguré pour revenir au mode de fonctionnement normal, nécessitant l'utilisation de badges RFID pour l'accès.

Gestion Alternative des Accès

Pour garantir une accessibilité permanente, même en cas de défaillance du système informatique, le système de contrôle d'accès intègre **un boîtier de commande manuelle**. Ce dispositif physique permet d'ouvrir la porte indépendamment du site web ou du réseau, assurant ainsi la continuité du service en toutes circonstances.



L'automatisation web offre confort et modernité, tandis que le mode manuel garantit la permanence du service – même dans les pires scénarios.



MISE EN OEUVRE DES DIFFÉRENTS LECTEURS DE BADGES

Présentation du lecteur MINOVA MCR04

Pour l'identification ainsi que l'accès au lycée, nous avons opté pour le capteur Minova MCR04. Ce lecteur de badge Ethernet est spécialement conçu pour les applications de contrôle d'accès et offre une solution complète grâce à son écran LCD intégré qui permet d'afficher des informations aux utilisateurs.

Le MCR04 est un lecteur sans contact haute performance compatible avec la technologie MIFARE et la norme ISO14443-A. Sa robustesse et sa flexibilité en font un choix idéal pour notre environnement scolaire, où la fiabilité et la facilité d'utilisation sont essentielles.



Caractéristiques clés du lecteur MINOVA MCR04 :

Fonctionnalité	Détails
Technologie	RFID MIFARE (ISO 14443-A)
Connectivité	Ethernet TCP/IP (intégré)
Interface	Écran LCD rétroéclairé (messages personnalisables)
Entrées/Sorties	4 entrées numériques + 2 relais de sortie
Alimentation	12V DC

Installation Physique

Fixation :

- Monté près de la porte d'entrée à une hauteur accessible.
- Fixation sécurisée avec vis.

Câblage :

- **Réseau** : Câble Ethernet vers le switch local.
- **Alimentation** : Alimenté via un câble d'alimentation 12V DC dédiée.
- **Relais** : Connecté au système d'ouverture de la porte.

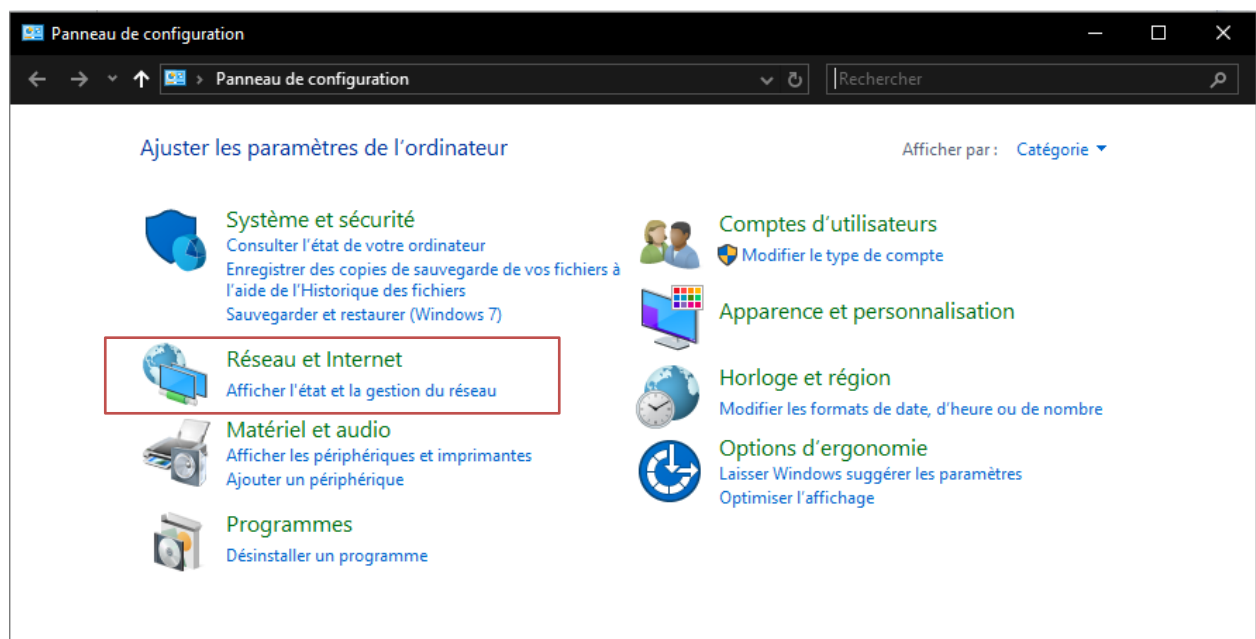
Configuration Logicielle du Lecteur Minova MCR04

Prérequis

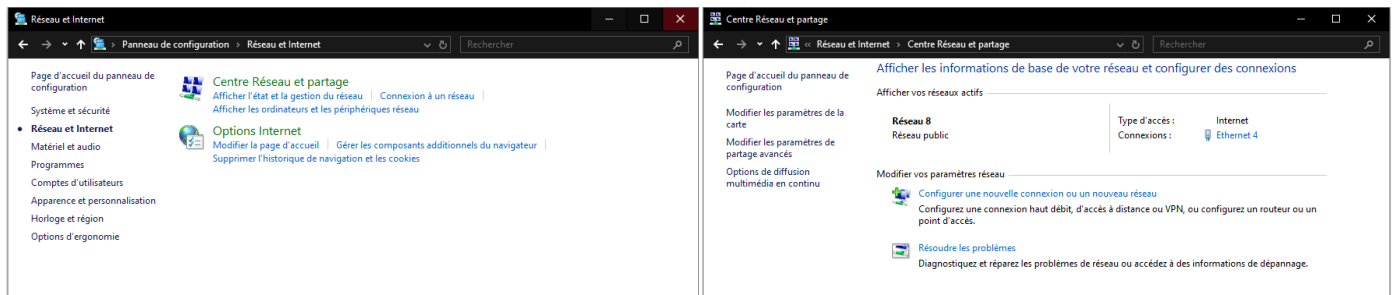
Avant de configurer le lecteur, vérifiez que :

- Le PC dispose du logiciel Mifinder
- Le PC dispose du logiciel XAMPP avec Apache **activé** (Les ports TCP 80 (HTTP))
- Le PC et le MCR04 sont sur le même réseau.

Avant toutes chose nous allons nous rendre dans le panneau de configuration de notre PC.

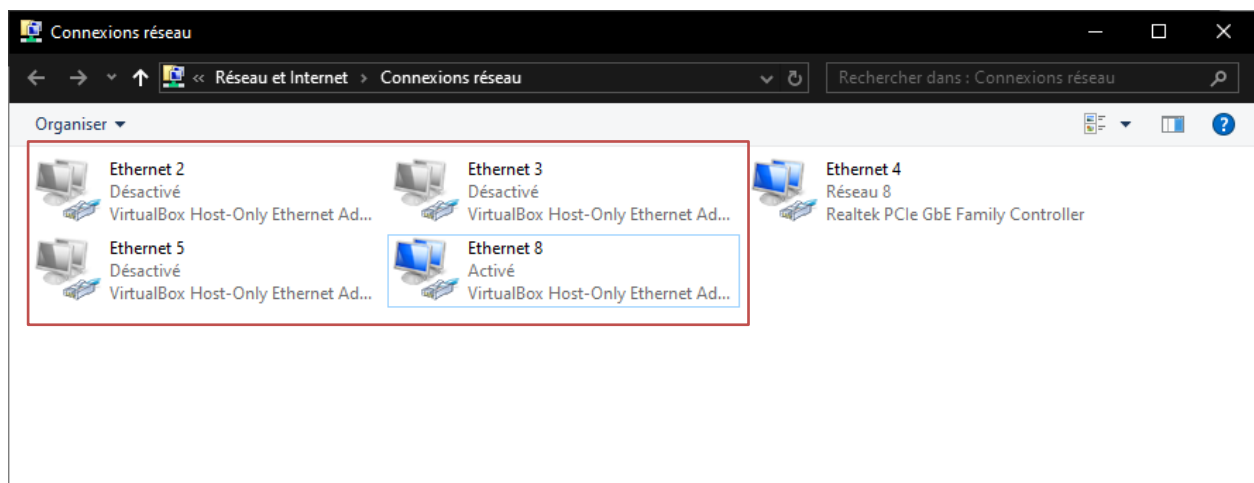


Une fois dans le panneau de configuration nous allons « **Réseau et Internet > Centre Réseau et partage** ».



Ensuite dans « **Modifier les paramètres de la carte** », nous allons procéder à la désactivation des interfaces inutiles tel que :

- Ethernet 2 (VirtualBox)
- Ethernet 3 (VirtualBox)
- Ethernet 5 (VirtualBox)
- Ethernet 8 (VirtualBox)

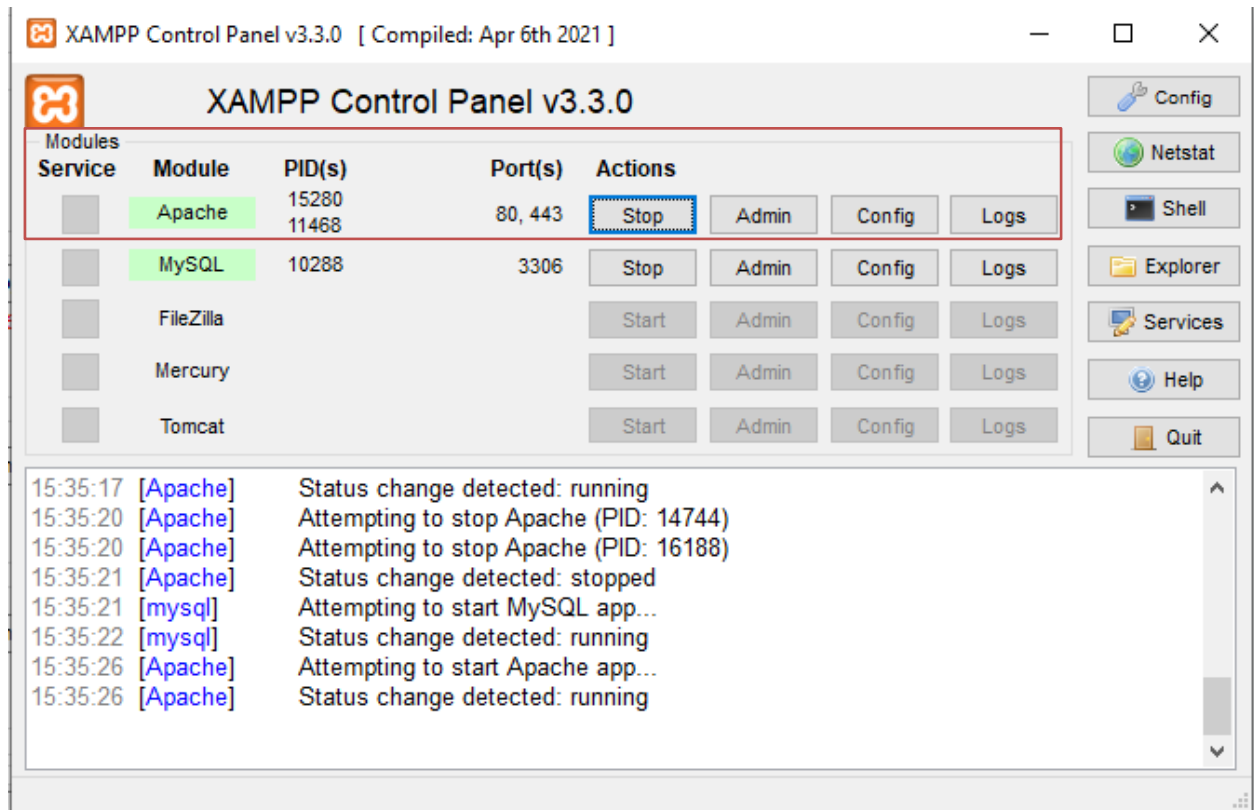


Nous allons garder uniquement « **Ethernet 4** », qui correspond à l'interface principale de notre PC.

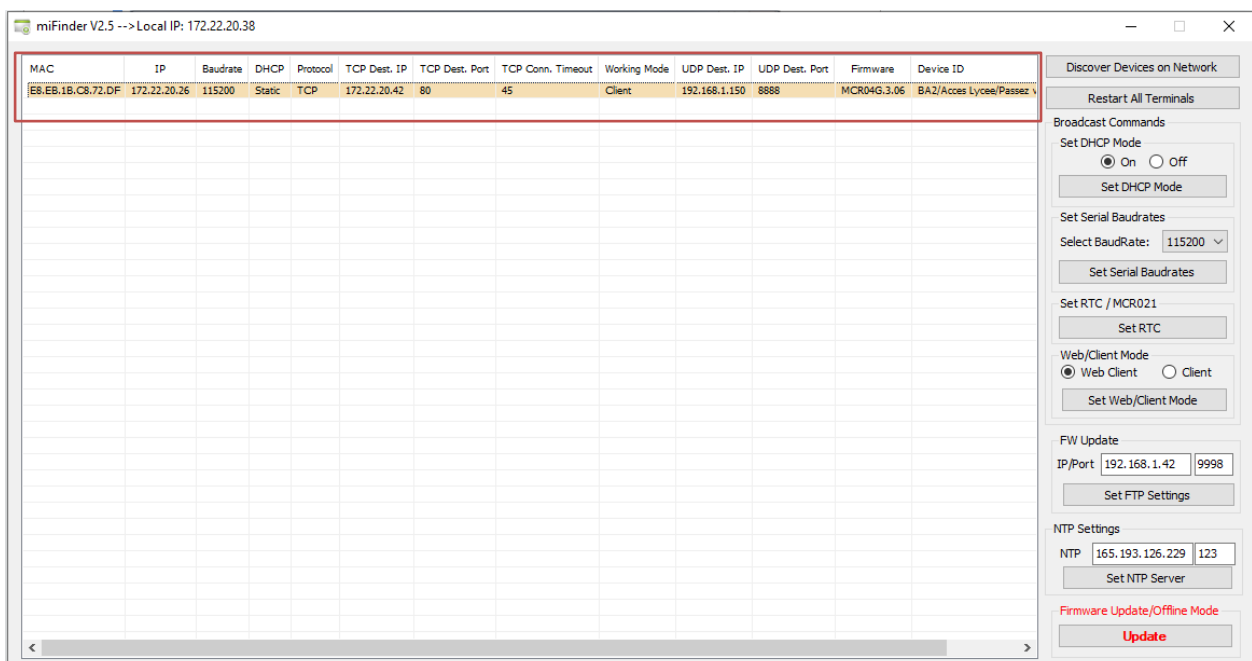
Dans le cas où aucune interface n'aurait été désactivée, le logiciel **MiFinder** pourrait scanner le mauvais réseau (ex: VirtualBox au lieu du réseau physique), empêchant la découverte du Lecteur Minova MCR04.

Après cela, nous allons ensuite ouvrir le logiciel XAMPP et activer le serveur APACHE (serveur HTTP).

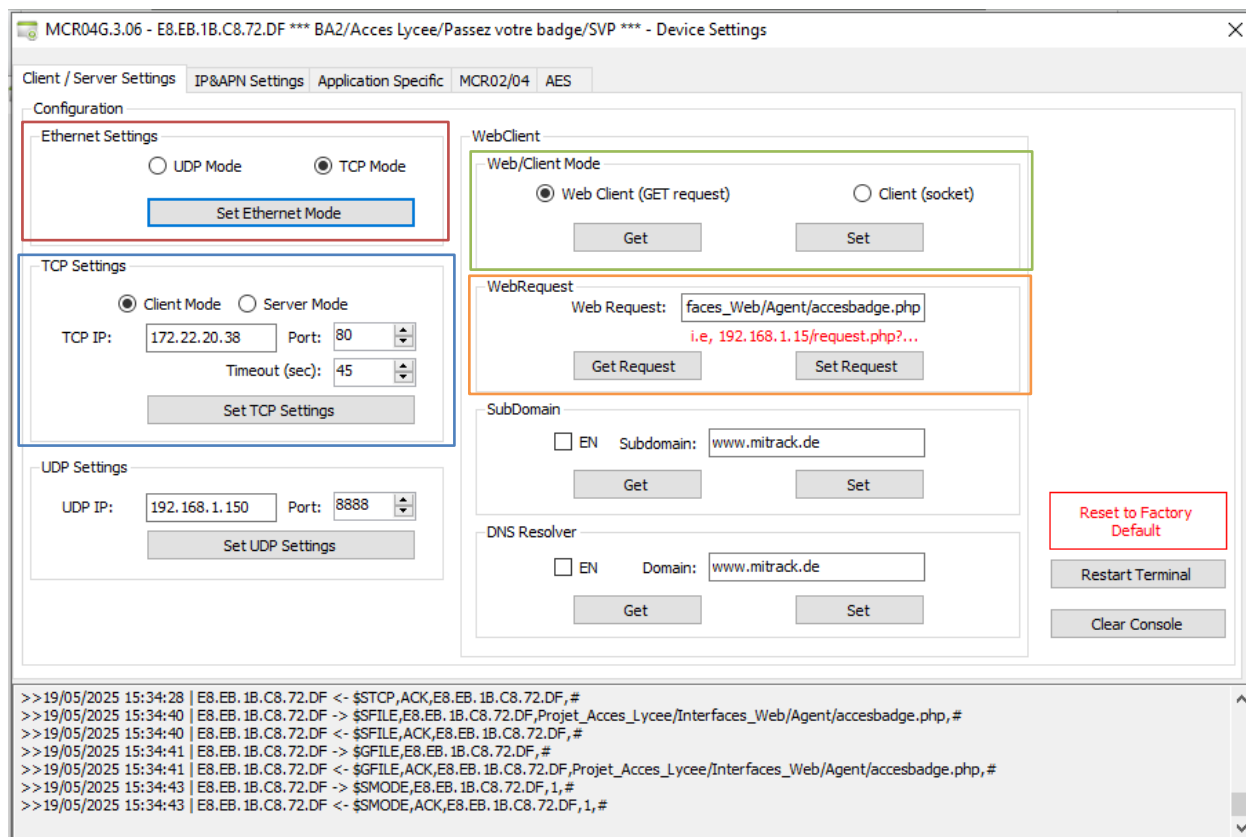
S



Une fois APACHE, lancé nous allons ouvrir le logiciel MiFinder afin de vérifier si le lecteur est détecté et ensuite procéder aux différentes configurations de celui-ci.



Pour la détection du lecteur nous allons appuyer sur le bouton « Discover Devises on Network ».

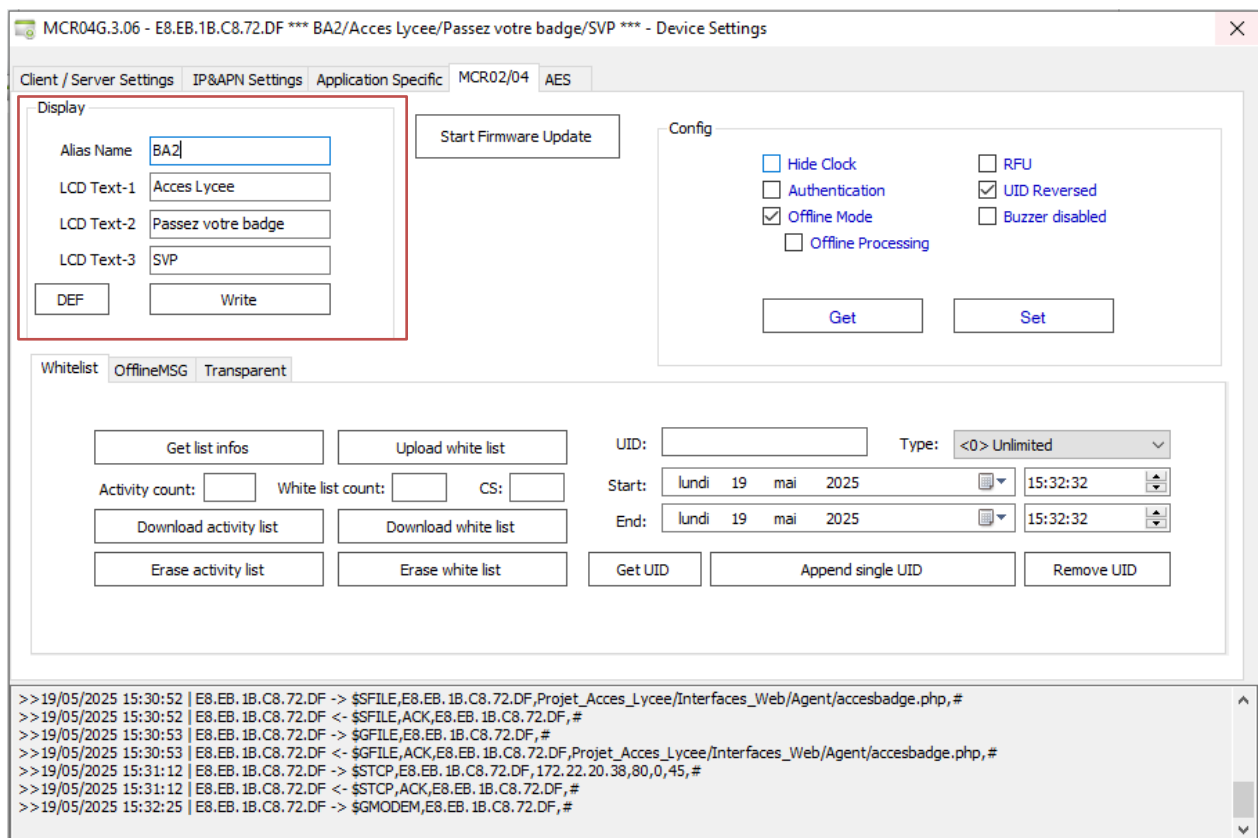


La configuration Ethernet est **TCP Mode**, le lecteur utilise le protocole TCP/IP pour une communication fiable et connectée.

Le mode « **Web Client** » est activé car, le lecteur se comporte comme un client qui initie les connexions vers le serveur, il envoie les données des badges via des **requêtes HTTP GET** au serveur.

Pour le paramétrage **TCP (Client)**, nous configurons l'adresse IP du serveur (**IP du PC**) ainsi que le Port utilisé.

Enfin pour assurer la communication avec **la base de données** nous renseignons le chemin vers le **script PHP** qui doit être exécuter lorsqu'un badge est passé.



Cette partie de l'interface permet de **personnaliser l'affichage du lecteur MINOVA MCR04** pour guider les utilisateurs.

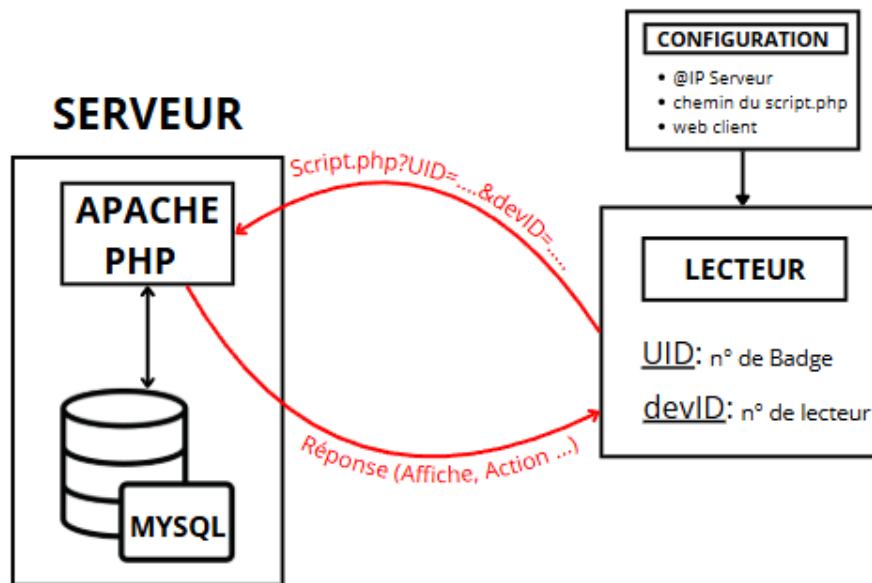


Méthode de Communication entre le Lecteur MCR04 et le Serveur

Protocole de Communication

Le lecteur MCR04 communique avec le serveur via des **requêtes HTTP GET** contenant :

- L'UID du badge (UID=xxx)
- L'identifiant du lecteur (devID=xxx)



```

// Vérification si au moins une ligne a été retournée
if($reponse->rowCount() > 0)
{
    $badge_info = $reponse->fetch(PDO::FETCH_ASSOC);

    // Activation du Barionet
    $barionet_ip = "172.22.21.164"; // Adresse IP du Barionet
    $urlOn = "http://$barionet_ip/rc.cgi?o=1,10";
    file_get_contents($urlOn);

    // Message de bienvenue simple envoyé sur l'afficheur
    echo $dev_id.',LCDCLR,LCDSET;0;5;3;Badge
valide,LCDSET;0;25;3;Acces autorise';
}
else {
    echo $dev_id.',LCDCLR,LCDSET;0;10;0;Badge invalide';
}
  
```

Réponse du Serveur (Format Spécifique au lecteur)

Le serveur répond avec des **commandes textuelles** interprétables par le lecteur MCR04 pour :

Effacer l'écran : `LCDCLR`

Afficher un message : `LCDSET;<ligne>;<position>;<couleur>;<texte>`

Quelques détails des paramètres du format de réponse du lecteur

Paramètre	Description	Valeurs possibles
<ligne>	Numéro de ligne (0 à 2)	0 (Ligne 1), 1 (Ligne 2)...
<position>	Position horizontale (en pixels)	5, 25, etc.
<couleur>	Couleur du texte	0=Rouge, 3=Vert
<texte>	Message à afficher	Max 16 caractères

Intégration avec le Barionet

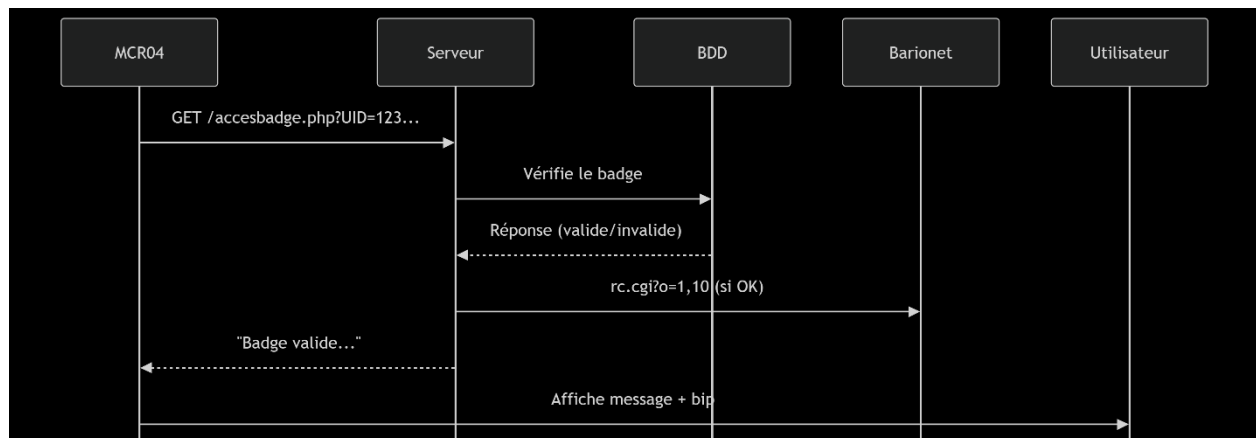
Quand un badge est valide :

Le script PHP :

- Active le relais du Barionet via `rc.cgi?o=1,10` (ouvre la porte 1 sec).
- Envoie la réponse formatée au MCR04.

Le MCR04 :

- Affiche les messages de confirmation (accès autorisé ou pas).
- Joue un bip sonore.



Le diagramme explique son intégration avec la base de données.

Présentation des Badges RFID Utilisés



Les badges RFID employés dans le système de contrôle d'accès sont des **badges passifs MIFARE Classic 1K**, conformes à la norme **ISO/IEC 14443-A**.

Spécification	Détail
Fréquence	13,56 MHz (HF - Haute Fréquence)
Mémoire	1 Ko
Portée	5–10 cm
Sécurité	UID unique
Format	Porte-clés

Présentation du lecteur RFID Inveo USB Desk

Le lecteur de badges Inveo RFID USB Desk est un dispositif polyvalent conçu pour lire divers types de badges RFID et transmettre les informations directement à un ordinateur via une connexion USB. Ce lecteur est particulièrement utile pour les applications nécessitant une saisie rapide et précise des données, telles que l'enregistrement des utilisateurs dans une base de données.



Caractéristiques Techniques

Fonctionnalité	Détails
Compatibilité RFID	MIFARE (ISO 14443-A)
Connectivité	USB Type-B (plug-and-play, alimenté par le port USB)
Modes de sortie	Émulation clavier (HID)
Retours utilisateur	LEDS (statut/alimentation) + buzzer intégré
Plage de lecture	3–8 cm (selon le type de badge)

Le lecteur de badges Inveo RFID USB Desk est un outil précieux pour l'enregistrement des utilisateurs dans une base de données. Sa capacité à fonctionner comme un émulateur de clavier facilite grandement le processus de saisie des données, tout en offrant une grande flexibilité et efficacité. Il constitue une solution idéale pour les environnements nécessitant une gestion précise et rapide des informations des utilisateurs.

ANNEXE

Programme principale de la gestion des modes d'accès (HTML, CSS, JAVASCRIPT)

```
<!DOCTYPE html>
<html lang="fr">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Contrôle Barionet 100</title>
  <link rel="stylesheet" href="ModeAcces.css">
  <style>
    .success { color: green; }
    .error { color: red; }
    .hidden {
      display: none; /* Masque l'élément */
    }
  </style>

  <script>
    // Récupère l'adresse IP saisie par l'utilisateur
    function getBarionetIP() {
      return document.getElementById('ipAddress').value;
    }

    // Contrôle une sortie (ON, OFF)
    function controlOutput(output, value) {
      const ip = getBarionetIP(); // Récupère l'adresse IP
      const url = "http://".concat(ip, "/rc.cgi?o=", output, ",", value);
      // Construit l'URL pour la requête HTTP
      fetch(url) // Envoie la requête HTTP
        .then(response => {
          if (response.ok) {
            showStatus(`Commande envoyée : Sortie ${output} - Valeur
${value}`, 'success'); // Affiche un message de succès si la réponse est OK
          } else {
            showStatus('Erreur lors de l\'envoi de la commande',
'error'); // Affiche un message d'erreur sinon
          }
        })
        .catch(error => {
          showStatus('Erreur de communication', 'error');
          console.error('Erreur:', error);
        });
    }
  </script>
</head>
<body>
  <div class="container">
    <div class="row">
      <div class="col">
        <div class="card">
          <div class="card-header">
            <h3>Contrôle Barionet 100</h3>
          </div>
          <div class="card-body">
            <div class="form">
              <div class="input">
                <input type="text" value="192.168.1.100" />
              </div>
              <div class="button">
                <button type="button" value="Contrôler">Contrôler</button>
              </div>
            </div>
            <div class="status">
              <div class="text">Statut: Inconnu</div>
              <div class="text">Mode: Inconnu</div>
            </div>
          </div>
        </div>
      </div>
      <div class="col">
        <div class="card">
          <div class="card-header">
            <h3>Paramètres</h3>
          </div>
          <div class="card-body">
            <div class="form">
              <div class="input">
                <input type="text" value="192.168.1.100" />
              </div>
              <div class="button">
                <button type="button" value="Paramétrer">Paramétrer</button>
              </div>
            </div>
            <div class="status">
              <div class="text">Statut: Inconnu</div>
              <div class="text">Mode: Inconnu</div>
            </div>
          </div>
        </div>
      </div>
    </div>
  </div>
</body>
</html>
```

```

    </script>
</head>
<body>
    <div class="container">
        <h1>GESTION D'ACCÈS</h1>

        <!-- Section pour configurer l'adresse IP du Barionet -->
        <div class="config-section">
            <div class="input-group hidden">
                <label for="ipAddress">Adresse IP:</label>
                <input type="text" id="ipAddress" value="172.22.21.164"> <!--
Champ pour entrer l'adresse IP -->
            </div>
        </div>

        <!-- Section pour contrôler les sorties du Barionet -->
        <div class="section">
            <div class="controls">
                <!-- Contrôle de la sortie 1 -->
                <div class="output-group">
                    <h3>Commande AGENT</h3>
                    <button onclick="controlOutput(1, 10)" class="btn
on">OUVRIR</button> <!-- Activer la sortie 1 pendant 1s -->
                </div>

                <!-- Contrôle de la sortie 2 -->
                <div class="output-group">
                    <h3>Modes de fonctionnement</h3>
                    <button onclick="controlOutput(2, 0)" class="btn on">Badge ou
Agent</button> <!-- Bouton pour désactivé sortie 2 -->
                    <button onclick="controlOutput(2, 1)" class="btn off">Mode
Auto</button> <!-- Bouton pour activé sortie 2 -->
                </div>
            </div>
        </div>
    </div>
</body>
</html>

```

Script PHP d'authentification RFID avec Base de Données et Commande Barionet.

```
<?php
// Récupération des paramètres GET
$code_badge = htmlspecialchars($_GET["UID"]); // Code unique du badge
$dev_id = htmlspecialchars($_GET["devID"]); // Identifiant du Lecteur Minova

// Inclusion de la configuration de la base de données
require '../config.php';

try {
    // Connexion à la base de données avec PDO
    $conn = new PDO(DB_DSN, DB_USER, DB_PASS, $pdo_options);

    // Vérification de l'existence du badge actif
    $sql = "SELECT b.id_badge, b.code_badge, u.nom, u.prenom
            FROM badge b
            JOIN utilisateur u ON b.id_utilisateur = u.id_utilisateur
            WHERE b.code_badge = :code_badge AND b.etat = 1";

    $reponse = $conn->prepare($sql);
    $reponse->bindValue(':code_badge', $code_badge, PDO::PARAM_STR);
    $reponse->execute();

    // Si un badge actif est trouvé
    if ($reponse->rowCount() > 0) {
        $badge_info = $reponse->fetch(PDO::FETCH_ASSOC);
        $id_badge = $badge_info['id_badge'];

        // Enregistrement de l'accès avec statut 1 (accepté)
        $insert = $conn->prepare("INSERT INTO historique (id_badge, date, heure,
statut)
VALUES (:id_badge, CURDATE(), CURTIME(), 1)");
        $insert->bindValue(':id_badge', $id_badge, PDO::PARAM_INT);
        $insert->execute();

        // Activation du Barionet pour ouvrir la porte
        $barionet_ip = "172.22.21.164";
        $urlOn = "http://$barionet_ip/rc.cgi?o=1,10";
        file_get_contents($urlOn);

        // Affichage du message de succès sur l'écran LCD
        echo $dev_id.',LCDCLR,LCDSET;0;13;5;Badge valide,LCDSET;8;35;3;Acces
autorise';
    } else {
        // Si le badge n'est pas actif, vérification s'il existe dans la base
```

```

        $getId = $conn->prepare("SELECT id_badge FROM badge WHERE code_badge =
:code_badge");
        $getId->bindValue(':code_badge', $code_badge, PDO::PARAM_STR);
        $getId->execute();
        $id_badge = $getId->fetchColumn();

        // Si le badge existe mais n'est pas actif
        if ($id_badge) {
            // Enregistrement de la tentative avec statut 0 (refusé)
            $insert = $conn->prepare("INSERT INTO historique (id_badge, date,
heure, statut) VALUES (:id_badge, CURDATE(), CURTIME(), 0)");
            $insert->bindValue(':id_badge', $id_badge, PDO::PARAM_INT);
            $insert->execute();
        }

        // Affichage du message d'erreur sur l'écran LCD
        echo $dev_id.',LCDCLR,LCDSET;0;13;4;Badge invalide,LCDSET;8;35;3;Acces
refuse';
    }
} catch (PDOException $e) {
    // En cas d'erreur de base de données
    echo $dev_id.',LCDCLR,LCDSET;0;10;0;ERREUR BD';
    error_log("Erreur de base de données : " . $e->getMessage());
}
?>

```


Affichage du lecteur dans les différents cas d'utilisations



Intérieur de l'armoire électrique

