

République du Sénégal



Un Peuple – Un But – Une Foi

Ministère de l'Enseignement Supérieur et de la Recherche

Université Alioune DIOP de Bambey



UFR Sciences Appliquées et Technologies de l'Information et de la
Communication (SATIC)

Département Technologies de l'Information et de la Communication (TIC)

Licence Professionnelle

Systèmes, Réseaux et Télécommunications (SRT)

PROJET DE MÉMOIRE

*Mise en place d'un système de
reconnaissance des plaques
d'immatriculation pour le
contrôle de l'accès au parking*

Encadrant(s) :

Pr Abdou Khadre DIOP

Présenté par :

Mouhamed Niang

&

Ngagne Niang

Année académique 2020-2021

Remerciement

*Tous d'abord nous tenons à remercier **Dieu** tout puissant pour nous avoir accorder la force et la patience de terminé ce modeste travail.*

*Ainsi que notre encadreur **Abdou Khadre DIOP** pour son aide et ses conseils tous au long de ce travail.*

Nous tenons à remercier aussi les membres de jury pour accepter d'évaluer ce modeste travail, ainsi que tous les enseignants du département de TIC (Technologie de l'Information et de la Communication) pour leur soutien durant toute l'année.

Sans oublier toute ma famille et mes amis pour leur soutien inconditionnel.

Dédicaces

A nos chers parents, pour tous leurs sacrifices, leur amour, leur tendresse, leur soutien et leurs prières tout au long de nos études, A nos chers frères et sœurs pour leurs encouragements permanents, et leur soutien moral, A toute nos familles pour leur soutien tout au long de nos parcours, à tous nos amis et ceux qui ont partagé avec nous tous les moments d'émotion lors de la réalisation de ce travail.

Que ce travail soit l'accomplissement de vos vœux tant allégués, et le fruit de votre soutien infailible, Merci d'être toujours là pour nous.

Table des matières

Introduction générale.....	6
CHAPITRE 1 : Architecture de reconnaissance des plaques et le contrôle d'accès au parking... 8	8
1) Descriptions matérielle et logicielle de la réalisation :	8
1.1. Matériels :.....	8
1.2. Logiciels :	14
2) Fonctionnement de l'architecture :	24
Chapitre 2 : Mise en place du système de reconnaissance des plaques d'immatriculation	28
1. INSTALLATION DE LOGICIEL :	28
1.1. Mise en œuvre d'une installation node-red :.....	28
1.2. Mise en œuvre d'une installation motion :	31
1.4. Mise en œuvre d'une installation OpenALPR :.....	34
1.4. Mise en œuvre d'une installation PhpMyAdmin :.....	37
2. Mise en place d'un programme sur node red :.....	42
3. Mise en place d'application web et base de données :	51
Conclusion générale.....	58
Bibliographie.....	60

Table légende

Figure 1 : Raspberry Pi.....	9
Figure 2 : composantes du Raspberry Pi.....	10
Figure 3 : Système d'exploitation Raspberry Pi	12
Figure 4 : Camera	13
Figure 5 : servomoteur	14
Figure 6 : Interface phpMyAdmin	22
Figure 7 : Interface node-red.....	23
Figure 8 : Architecture de detection des plaques d'immatriculation pour le contrôle d'accès au parking	25
Figure 9 : présentation de la voiture à l'entre du parking avec le positionnement du camera ...	25

Introduction générale

Depuis la naissance de la voiture jusqu'à aujourd'hui, il y'a une augmentation importante de sa production vue qu'elle rend beaucoup de service et facilite le déplacement dans plusieurs domaine « Commerce, Transport... ». Le fait que la production de voiture ne cesse d'augmenter a engendré le besoin de beaucoup d'espace pour leur stationnement dans les parkings. Ce dernier est un ensemble des voitures immatriculées stationne pendant un temps déterminé dans une surface clôturée.

Le nombre de voitures augmente par le fait du besoin croissant de se déplacer. Chaque voiture est caractérisée par son numéro de plaque d'immatriculation. Il est nécessaire d'identifier les plaques d'immatriculation des véhicules pour plusieurs raisons tel que la sécurité. L'identification des véhicules doit se faire de manière très efficace en utilisant des ressources limitées. L'un des meilleurs moyens est de mettre en place un système de reconnaissance des plaques d'immatriculation. Le numéro de plaque d'immatriculation est un code numérique utilisé pour identifier de manière unique les véhicules et permet le contrôle de l'accès au parking.

De ce fait, une problématique peut être posée de la façon suivante :

- ✚ Quels sont les outils matériels et logiciels à exploiter pour la reconnaissance des plaques d'immatriculation et le contrôle de l'accès au parking ?
- ✚ Comment le système de reconnaissance des plaques d'immatriculation favorise-t-il le contrôle de l'accès au parking ?

L'objectif de notre projet est de rendre plus moderne le secteur du transport en concevant d'une part une application web pour renseigner efficacement les informations du véhicule à la base de données, et d'autre part en concevant un système intelligent de détection des plaques d'immatriculation grâce à une caméra qui sera positionnée afin de capter la plaque du véhicule lorsque ce dernier présentera à l'entrée du parking et une barrière automatisée qui est fermée par défaut et s'ouvre si la plaque du véhicule est reconnue.

Notre travail a pour finalité de :

- ✓ Détection des plaques d'immatriculation
- ✓ Automatisation d'accès parking
- ✓ Augmentation de la sécurité

Le présent document est organisé comme suit :

Le chapitre 1 on se charge de détailler l'architecture qui permet de reconnaître les plaques et le contrôle d'accès au parking. Dans cette partie on va essayer de faire une description matérielle et logicielle de la réalisation puis le fonctionnement de l'architecture.

Et enfin nous présentons dans le chapitre 2 une mise en place du système de reconnaissance des plaques d'immatriculation.

CHAPITRE 1 : Architecture de reconnaissance des plaques et le contrôle d'accès au parking

1) Descriptions matérielle et logicielle de la réalisation :

Le but de notre projet de fin d'étude est de concevoir un système de reconnaissance des plaques immatriculation. Et vu qu'on a besoin des matérielles et des logiciels pour des entrées des signaux de commande en sorties pour le contrôle d'accès au parking.

1.1. Matériels :

1.1.1. Raspberry Pi :

Le Raspberry Pi est un mini-ordinateur mono-carte, souvent utilisé avec le système d'exploitation linux mais il peut supporter d'autres systèmes d'exploitation comme Windows par exemple. Il est destiné à des applications d'informatique embarquée. Le cœur de l'ordinateur est un FPGA (Broadcom2836) intégrant un processeur quad-core ARM11 Cortex A7 cadencé à 700 MHz, de 256 Mo jusqu'à 4 Go de RAM selon la version et de nombreux périphériques. Le Raspberry Pi peut être directement connecté à une IHM classique, souris/clavier/écran HDMI ou vidéo composite. Le connecteur d'extension supporte les entrées/sorties parallèles ainsi que la plupart des bus de communication comme le bus CAN, Ethernet, I2C... C'est un support particulièrement économique et puissant qui peut être facilement mis en œuvre dans des systèmes nécessitant un accès au monde physique par des capteurs/actionneurs disposant d'interfaces numériques.

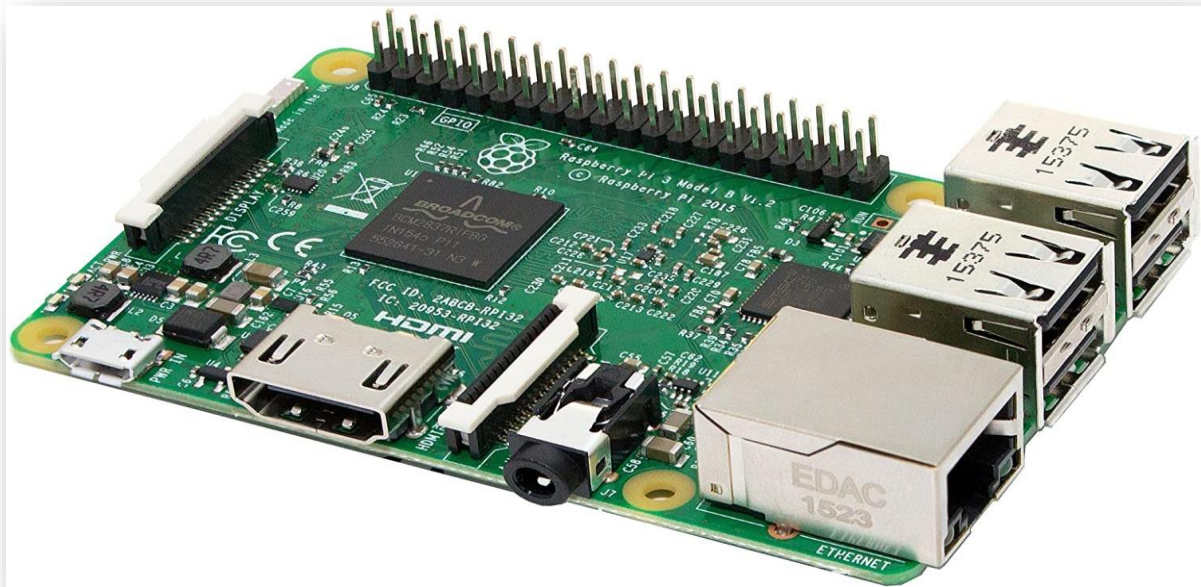


Figure 1 : Raspberry Pi

1.1.1.1. Historique :

Le Raspberry PI, ce petit ordinateur, était créé par des étudiants qui voulaient faire l'ordinateur le moins cher du monde afin d'encourager l'apprentissage de la programmation. En 2006, Eben Upton étudiant de la faculté des sciences informatiques de l'université de Cambridge en Angleterre et ses collègues ont réalisés le premier prototype sur l'Atmel AtMega 664. Ce premier prototype était de taille d'une clé USB, et contient qu'un port USB et HDMI comme périphériques d'entrées/sorties. Les premières cartes sont produites en Mars 2012 à Taiwan et en Chine. Ils étaient équipés d'une RAM de 256Mo et un port RJ45. Puis la production est passée au Royaume-Uni en 2013-2014. Aujourd'hui la fondation Raspberry PI annonce plus de 6 millions de Raspberry PI qui ont été vendus dans le monde entier.

1.1.1.2. Composants de base :

Le Raspberry Pi 4 model B est le dernier produit de la gamme d'ordinateurs Raspberry Pi. Il offre des augmentations révolutionnaires de la vitesse du processeur, des performances multimédias, de la mémoire et de la connectivité par rapport au Raspberry Pi 3 model B+ de la

génération précédente, tout en conservant une compatibilité descendante et une consommation d'énergie similaire. Pour l'utilisateur final, le Raspberry Pi 4 modèle B offre des performances de bureau comparables aux systèmes PC x86 d'entrée de gamme. C'est pour cela que nous l'avion choisie. Les principales caractéristiques de ce produit sont :

- Un processeur 64 bits quad-core ARM Cortex A72 cadencé à 1.5GHz intégré dans FPGA Broadcom2711.
- La prise en charge du double affichage à des résolutions allant jusqu'à 4K via une paire de ports micro-HDMI.
- Le décodage vidéo matériel jusqu'à 4Kp60
- Une Capacité de RAM jusqu'à 8 Go. • Double bande LAN sans fil 2.4/5.0 GHz + Bluetooth 5.0
- Gigabit Ethernet.
- Capacité PoE (via un module complémentaire PoE HAT distinct).

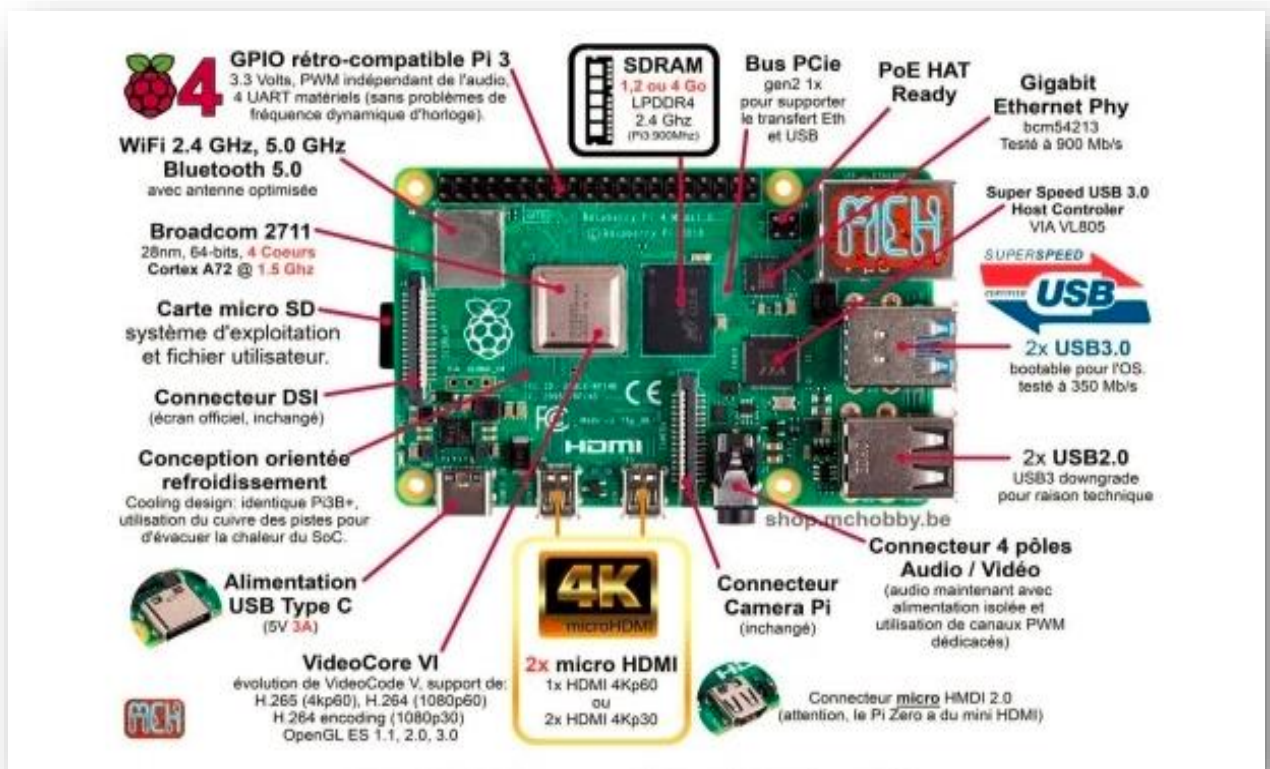


Figure 2 : composantes du Raspberry Pi

1.1.1.3. Raspberry Pi OS :

Raspberry Pi OS (anciennement nommé **Raspbian**) est un système d'exploitation libre et gratuit basé sur Debian optimisé pour fonctionner sur les différents Raspberry Pi.

Etant donné les ressources limitées des nano-ordinateurs pour lesquels Raspberry Pi OS est principalement destiné, il intègre des logiciels réputés pour être légers et peu gourmands en ressources :

- L'environnement de bureau par défaut est PIXEL
- Le navigateur web par défaut est Chromium

Raspberry Pi OS est fourni avec les environnements de programmation suivants :

- Java IDE : environnement de programmation Java
- Geany : éditeur de texte léger avec fonctions de développement
- Green foot Java IDE : logiciel de développement d'applications et de jeux sous plateforme Java.
- Mathématique : logiciel de calcul scientifiques.
- Node-RED (en) : logiciel de programmation avec interface visuelle
- Python 2 et 3 (IDLE).
- Scratch 1.0 et 2.0 : logiciels de programmation visuelle.
- Simulateur 3D de la carte Sense HAT fait pour simuler des codes en Python
- Sonic Pi : logiciel permettant de créer des mélodies en Ruby.
- Python IDE : environnement de programmation Python

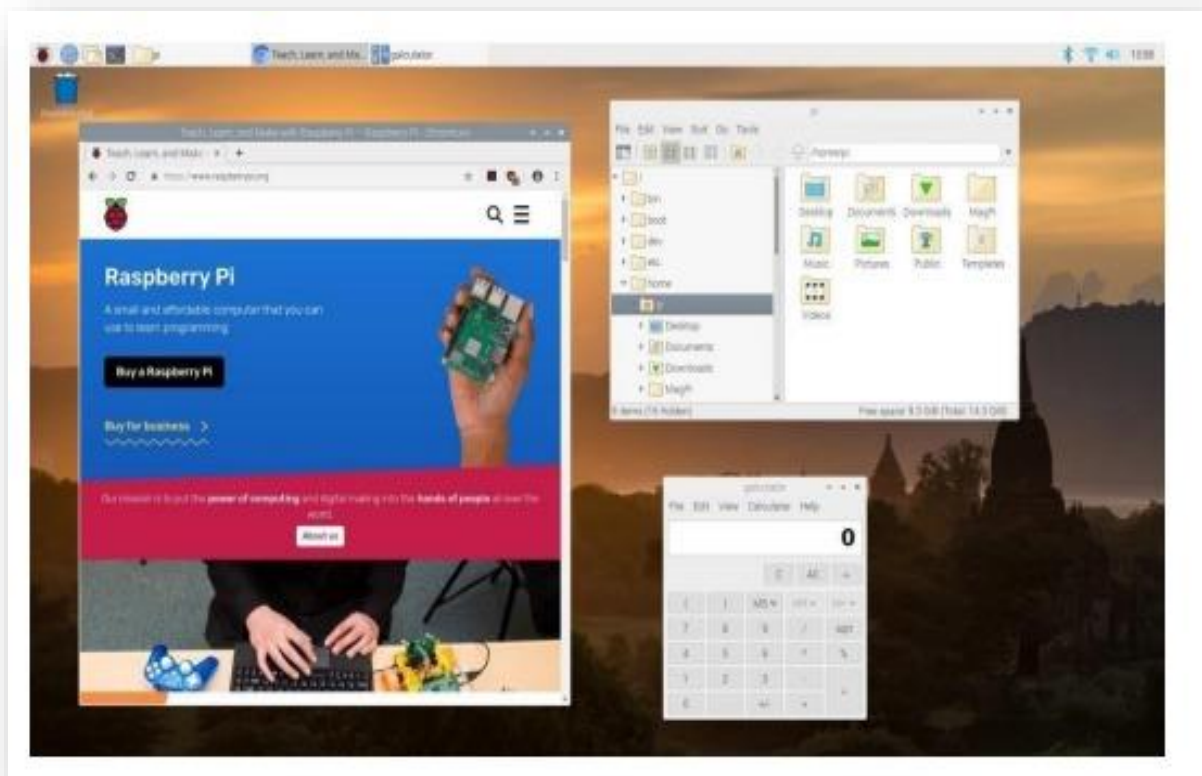


Figure 3 : Système d'exploitation Raspberry Pi

1.1.2. Camera :

Une **caméra IP** ou **caméra réseau** est une caméra de surveillance utilisant le Protocole Internet pour transmettre des images et des signaux de commande via une liaison Fast Ethernet. Certaines caméras IP sont reliées à un numérique (DVR) ou un enregistreur vidéo en réseau (NVR) pour former un système de surveillance vidéo.

L'avantage des caméras IP est qu'elles permettent aux propriétaires et aux entreprises de consulter leurs caméras depuis n'importe quelle connexion internet via un ordinateur portable ou un smartphone

Une caméra IP peut être câblée avec du RJ45 vers un routeur ou « box ADSL », ce qui lui permet à la fois d'être alimentée et les images visionnées sur le réseau, ou alors par WI-FI (une alimentation en courant électrique devient alors nécessaire). Contrairement aux WEBCAMS USB, la compatibilité avec les logiciels de VISIOCONFERENCE n'est pas toujours garantie.

Les caméras de surveillance IP tendent à se démocratiser, plusieurs modèles sur le marché sont à la disposition des particuliers. Les modèles cloud sont basés sur des services payants, les caméras autonomes fonctionnent avec une interface web ainsi que des modèles fixes ou motorisés.

Il est nécessaire de configurer son routeur pour pouvoir y accéder depuis l'extérieur, de manière à rediriger un ou plusieurs ports déterminés vers l'adresse IP locale de la caméra.

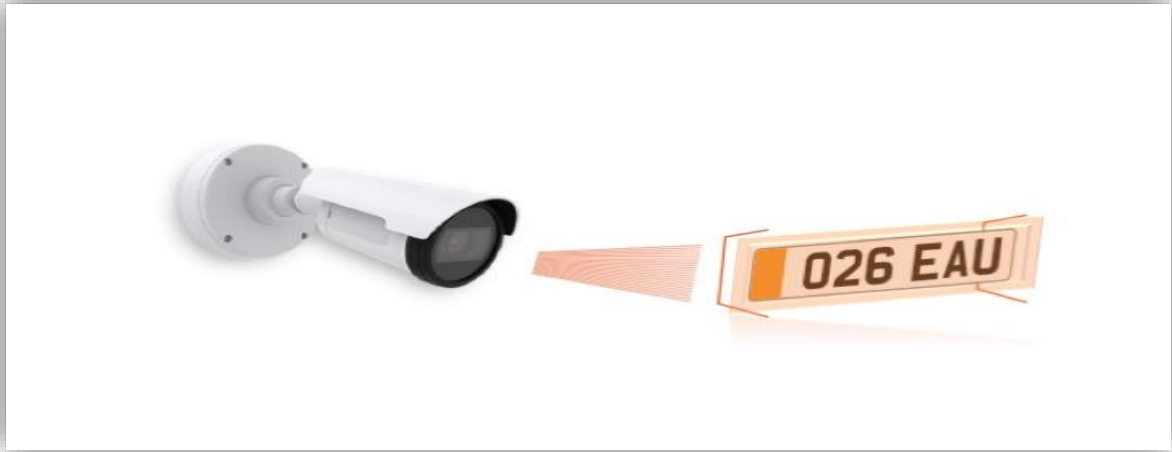


Figure 4 : Camera

1.1.3. Servomoteur :

Un servomoteur est un système motorisé capable d'atteindre des positions prédéterminées, puis de les maintenir. La position est : dans le cas d'un moteur rotatif, une valeur d'angle et, dans le cas d'un moteur linéaire une distance.

On utilise des moteurs électriques (continu, asynchrone) aussi bien que des moteurs hydrauliques. Le démarrage et la conservation de la position prédéterminée sont commandés par un système de réglage.

Pour un ajustement précis de la position, le moteur et son réglage sont équipés d'un système de mesure qui détermine la position courante (p. ex. l'angle de rotation parcouru relatif à une position de départ) du moteur. Cette mesure est effectuée sur un réglage rotatif comme un résolveur, un réglage incrémental ou un réglage absolu (réalisable par un potentiomètre).



Figure 5 : servomoteur

1.2.Logiciels :

1.2.1. OpenALPR :

OpenALPR est une bibliothèque de reconnaissance automatique de plaques d'immatriculation écrite en C++ . Le logiciel est distribué à la fois dans une version commerciale basée sur le cloud et dans une version open source .

Il peut être exécuté en tant qu'utilitaire de ligne de commande, bibliothèque autonome ou processus d'arrière-plan. Le logiciel s'intègre également aux systèmes de gestion vidéo (VMS) tels que Milestone XProtect.

OpenALPR a été initialement développé par une équipe de deux hommes dirigés par **Matt Hill**. Le logiciel open source est devenu disponible en téléchargement gratuit fin 2015.En mars 2016, OpenALPR a lancé le service Cloud API payant et en février 2017 a introduit l'agent OpenALPR pour les caméras Axis Communications .

En août 2017, un développeur Web australien Tait Brown s'est fait connaître en créant une alternative à un projet de 86 millions AUD de la police de Victoria en utilisant OpenALPR. En mars 2018, Programmable Web a ajouté OpenALPR à sa liste d'API de reconnaissance.

OpenALPR utilise les bibliothèques Open CV et Tesseract OCR pour la reconnaissance des caractères de la plaque immatriculation.

1.2.1.1. Open CV :

Open CV (Open Source Computer Vision Library) est une bibliothèque de logiciels et de logiciels d'apprentissage automatique. Il a été conçu pour divers objectifs tels que l'apprentissage automatique, la vision par ordinateur, l'algorithme, les opérations mathématiques, la capture vidéo, le traitement d'images, ... etc., Android, iOS).

En outre, il a enveloppé dans divers langages de programmation renommés. Dans le cadre du contrat de licence, les entreprises peuvent accéder au code et le modifier. La bibliothèque contient plus de 2500 algorithmes optimisés, qui ont une excellente précision en termes de performances et de vitesse. Ces algorithmes peuvent être utilisés pour détecter et reconnaître des visages, identifier des objets, classer des actions humaines dans des vidéos, suivre des mouvements de caméra, suivre des objets en mouvement, extraire des modèles 3D d'objets, produire des nuages de points 3D à partir de caméras stéréo image d'une scène entière, trouver des images similaires à partir d'une base de données d'images, supprimer les yeux rouges des images prises, suivre les mouvements oculaires, reconnaître les décors et établir des marqueurs pour la superposer à la réalité augmentée. , les développeurs et les chercheurs, le nombre est plus de 47 000 et le nombre estimé de téléchargements dépasse 7 millions. La bibliothèque est largement constituée de sociétés professionnelles, de groupes de recherche et d'autres groupes. De nombreuses entreprises bien établies telles que Google, Yahoo, Microsoft, Intel, IBM, Sony, Honda et Toyota qui emploient la bibliothèque, ainsi que de nombreuses startups telles que Applied Minds, VideoSurf et Zeitera, utilisent largement Open CV. Les utilisations déployées d'Open CV s'étendent de l'assemblage d'images StreetView, à la détection d'intrusions dans la vidéo de surveillance en Israël, à la surveillance des équipements miniers en Chine, à la détection des accidents de noyade en Europe et à l'art interactif. L'Espagne et New York vérifient les pistes en Turquie et inspectent les étiquettes des produits dans les usines du monde entier pour détecter rapidement les visages au Japon.

Il possède des interfaces C ++, C, Python, Java et MATLAB et supporte Windows, Linux, Android et Mac OS. Open CV s'appuie principalement sur les applications de vision en temps réel et tire parti des instructions MMX et SSE lorsqu'elles sont disponibles. Des interfaces complètes CUDA et Open CL sont en cours de développement. Il existe plus de 500 algorithmes et environ 10 fois plus de fonctions qui composent ou supportent ces algorithmes. Open CV est écrit nativement en C ++ et possède une interface basée sur des modèles qui fonctionne de manière transparente avec les conteneurs STL.

1.2.1.1.1. Algorithmes implémentés

Nous allons présenter ci-dessous, sous format algorithmique, les modules du système reconnaissance des plaques que nous avons fait leur conception. Le premier algorithme (Algorithme 1) permet de préparer chaque trame reçue depuis la caméra à l'étape de localisation de la plaque d'immatriculation. Le prétraitement de l'image passe par les étapes de redimensionnement, conversion en niveau de gris et l'élimination des bruits en utilisant un filtre bilatéral.

Algorithme 1 : Prétraitement et amélioration de la trame lue

Algorithme Prétraitement(trame_lue)

Entrées :

Trame_lue : image représentant une trame vidéo

Résultats :

Image_filtrée : image qui a subis un ensemble de pré-traitements.

Début

Redimensionner (trame_lue, largeur=500)

Convertir l'image redimensionnée en niveaux de gris.

Appliquer un filtre bilatéral sur l'image en niveau de gris (pour éliminer les bruits)

Renvoyer Image_filtrée

L'algorithme 2 sert à localiser une plaque d'immatriculation s'il existe dans l'image prétraitée. Cela sera effectué en suivant les étapes suivantes : Conversion de l'image en noir et blanc, l'extraction de la liste des contours fermés, le tri des contours par ordre de surface et on va choisir enfin celui qui a quatre coins et qui va vérifier un facteur largeur/hauteur entre 3.5 et 6.

Algorithme 2 : Localisation de la plaque d'immatriculation

Algorithme Localiser (image_entrée)

Entrées :

Image_entrée : image prétraite

Résultats

DéTECTÉE : **Booléen** indiquant **si** une plaque est détectée,

PlaqueImg : Image extraite de l'image d'entrée représentant la plaque,

PLargeur, PHauteur : Les dimensions de la plaque.

Début

Appliquer un seuillage adaptatif sur l'image d'entrée pour avoir une image noire et blanc

Appliquer un filtre de Canny sur l'image noir et blanc Extraire, de la liste des contours, seulement les contours fermés.

Trier la liste des contours extraites selon leurs surfaces du plus grand

Vers le plus petit

DéTECTÉE \leftarrow faux

Pour chaque contour de la liste faire

Si le contour possède quatre (04) coins **Alors**

Extraire les coordonnées du rectangle englobant le contour

Calculer la largeur et la hauteur du contour : PLargeur, PHauteur

Rapport \leftarrow PLargeur / PHauteur

Si Rapport ≥ 3.5 et Rapport ≤ 6 **Alors**

Déecté = vrai Sortir de la boucle

FinSi

FinSi

FinPour

Si Déectée **Alors**

Extraire l'image de la plaque déectée soit : PlaqueImg

FinSi

Renvoyer (Déectée, PlaqueImg, PLargeur, PHauteur)

Fin

L'algorithme suivant (Algorithme 3) permet de segmenter l'image de la plaque déectée en se basant sur les contours et d'identifier les chiffres du matricule. Pour la liste des contours déectés on va choisir celui qui ont un rapport Largeur/Hauteur entre 0.25 et 1 et un rapport Hauteur/Hauteur plaque entre 0.5 et 0.8. La zone candidate sera envoyée à un algorithme de reconnaissance de k plus proches voisins (kNN) qui va nous renvoyer un chiffre correspondant. Les chiffres correspondant aux zones segmentées vont construire le matricule de la voiture.

Algorithme 3 : Segmentation de la plaque d'immatriculation et reconnaissance des chiffres Algorithme SegNumEtConvert (PlaqueImg, PLargeur, PHauteur)

Entrées :

PlaqueImg : image de la plaque déectée

PLargeur, PHauteur : Largeur et hauteur de la plaque

Résultats :

Matricule : Chaîne de caractère représentant les chiffres définissant le matricule

Reconnu.

Début

Matricule \leftarrow " "

Conversion de l'image de la plaque en niveaux de gris

Appliquer sur l'image en niveau de gris un filtre bilatéral

Appliquer sur l'image filtrée un seuillage adaptatif pour avoir une image noire et blanc

Agrandir la taille de l'image de la plaque

Appliquer une autre fois un seuillage normal pour confirmer l'élimination de tout bruit

Appliquer un filtre de Canny pour extraire les contours

Trouver la liste des contours fermés

Trier la liste des contours trouvés du gauche vers la droite

Pour tous les contours triés faire

Calculer pour le rectangle englobant le contour : Hauteur et Largeur Rapport \leftarrow Largeur / Hauteur

RapportChiffrePlaque \leftarrow Hauteur / PIHauteur

Si Rapport < 1 et Rapport > 0.25 et RapportChiffrePlaque > 0.5 et RapportChiffrePlaque et RapportChiffrePlaque > 0.8 **Alors**

Extraire depuis l'image binaire imgROI englobée par le rectangle calculé

Redimensionner imgROI à une taille de 20 x 30

Construire depuis imgROI un vecteur uni-dimensionnel

Trouver le chiffre approprié en utilisant l'algorithme de "k plus proches voisins"

Ajouter le chiffre reconnu à (Matricule) la liste des chiffres de la plaque

FinSi

FinPour

Renvoyer Matricule

Fin

Le dernier algorithme représente la suite des étapes à suivre pour accomplir le processus de la reconnaissance automatique des plaques d'immatriculation.

Algorithme 4 : Programme principale de détection des plaques

Algorithme detectionPlaque ()

Début Lancer l'apprentissage pour l'algorithme de kNN (k plus proches voisins)

Pour toutes les images capturées faire

Image_filtrée=Pré-traitement (Trame courrente)

(Détectée, Image_Plaque, PLargeur, PHauteur) = Localizer (Image_filtrée)

Si Détectée **Alors**

Matricule=SegNumEtConvert (Image_Plaque, PLargeur, PHauteur)

Afficher (Matricule)

FinSi

FinPour

Fin

1.2.1.2. Tesseract OCR :

OCR est l'acronyme de Reconnaissance Optique de Caractères. Cette technologie permet de reconnaître automatiquement les caractères grâce à un mécanisme optique. Dans le cas des êtres humains, nos yeux sont un mécanisme optique. L'image vue par les yeux est une entrée pour le cerveau. La capacité à comprendre ces entrées varie chez chaque personne en fonction de plusieurs facteurs. L'OCR est une technologie qui fonctionne comme une capacité de lecture humaine, bien que l'OCR ne puisse pas rivaliser avec les capacités de lecture l'être humain.

Parmi les techniques existantes nous pouvons citer : Tesseract Engine et l'algorithme de k-plus proches voisins (KNN en anglais k nearest neighbour).

1.2.1.2.1. Moteur Tesseract :

Tesseract est un moteur de reconnaissance optique de caractères pour différents systèmes d'exploitation. C'est un logiciel libre, publié sous la licence Apache, version 2.0, et le développement est sponsorisé par Google depuis 2006. Tesseract est considéré comme l'un des moteurs de OCR open source les plus précis actuellement disponibles.

1.2.1.2.2. Algorithme KNN :

L'algorithme KNN figure parmi les plus simples algorithmes d'apprentissage artificiel. Dans un contexte de classification d'une nouvelle observation x , l'idée fondatrice simple est de faire voter les plus proches voisins de cette observation.

La classe de x est déterminée en fonction de la classe majoritaire parmi les k plus proches voisins de l'observation x . La méthode KNN est donc une méthode à base de voisinage, non-paramétrique ; Ceci signifiant que l'algorithme permet de faire une classification sans faire d'hypothèse sur la fonction $y=f(x_1, x_2, \dots, x_p)$ qui relie la variable dépendante aux variables indépendantes. La méthode du plus proche voisin est une méthode non paramétrique où une nouvelle observation est classée dans la classe d'appartenance de l'observation de l'échantillon d'apprentissage qui lui est la plus proche, au regard des covariables utilisées. La détermination de leur similarité est basée sur des mesures de distance. On va essayer d'utiliser une des deux techniques pour faire la reconnaissance des caractères. Donc, les caractères segmentés vont être envoyés à un algorithme de reconnaissance, qui renvoie l'ASCII de chaque numéro de la plaque.

1.2.2. PhpMyAdmin :

phpMyAdmin (PMA) est une application Web pour les systèmes de gestion de base de données MySQL et Maria DB, réalisée principalement en PHP et distribuée sous licence GNU GPL.

Il s'agit de l'une des plus célèbres interfaces pour gérer une base de données MySQL sur un serveur PHP. De nombreux hébergeurs, gratuits comme payants, le proposent ce qui évite à l'utilisateur d'avoir à l'installer.

Cette interface graphique permet d'exécuter, très facilement et sans grandes connaissances en bases de données, des requêtes comme les créations de table de données, insertions, mises à jour, suppressions et modifications de structure de la base de données, ainsi que l'attribution et la révocation de droits et l'import/export. Ce système permet de sauvegarder commodément une base de données sous forme de fichier.sql et d'y transférer ses données, même sans connaître SQL.

Les requêtes SQL restent possibles, ce qui permet de les tester interactivement lors de la création d'un site pour les utiliser ensuite en batch (c'est-à-dire en différé) une fois au point.

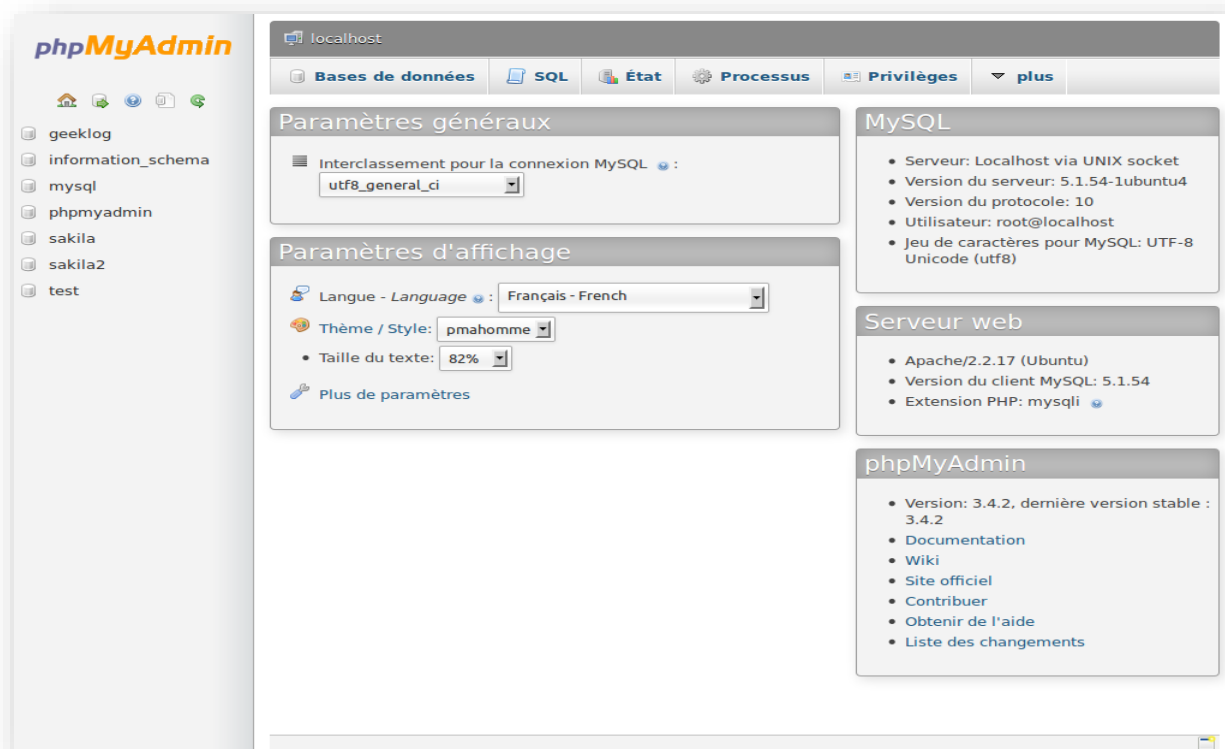


Figure 6 : Interface phpMyAdmin

1.2.3. NODE-RED :

Node-red est un outil puissant pour construire des applications de l'Internet des Objets en mettant l'accent sur la simplification de la programmation qui se fait grâce à des blocs de code prédéfinis, appelés nodes pour effectuer des tâches.

Il utilise une approche de programmation visuelle qui permet aux développeurs de connecter les blocs de code ensemble. Les nœuds connectés, généralement une combinaison de nœuds d'entrée, de nœuds de traitement et de nœuds de sortie, lorsqu'ils sont câblés ensemble, constituent un flow. Node-red est construit sur Node.js, tirant pleinement parti de son modèle non bloquant piloté par les événements. Cela le rend idéal pour fonctionner en périphérie du réseau sur un serveur d'application qui peut être un matériel à faible coût tel que le Raspberry Pi ou un serveur dans le cloud.

C'est un logiciel permettant de gérer des flows d'événements, des suites de traitements à effectuer suite à la réception de messages ou événements. Il contient un certain nombre de fonctionnalités de base, mais la plupart des fonctionnalités utiles dans notre cas devront être installées par la suite. Dans Node-RED, une « fonctionnalité » est représentée sous la forme d'une node, un élément pouvant être placé dans votre flow, reliée à d'autres nodes en entrée ou en sortie. Le flow représente l'ensemble des nodes. Il n'est pas linéaire et une node reliée à aucune autre peut quand même s'activer si les conditions sont réunies.

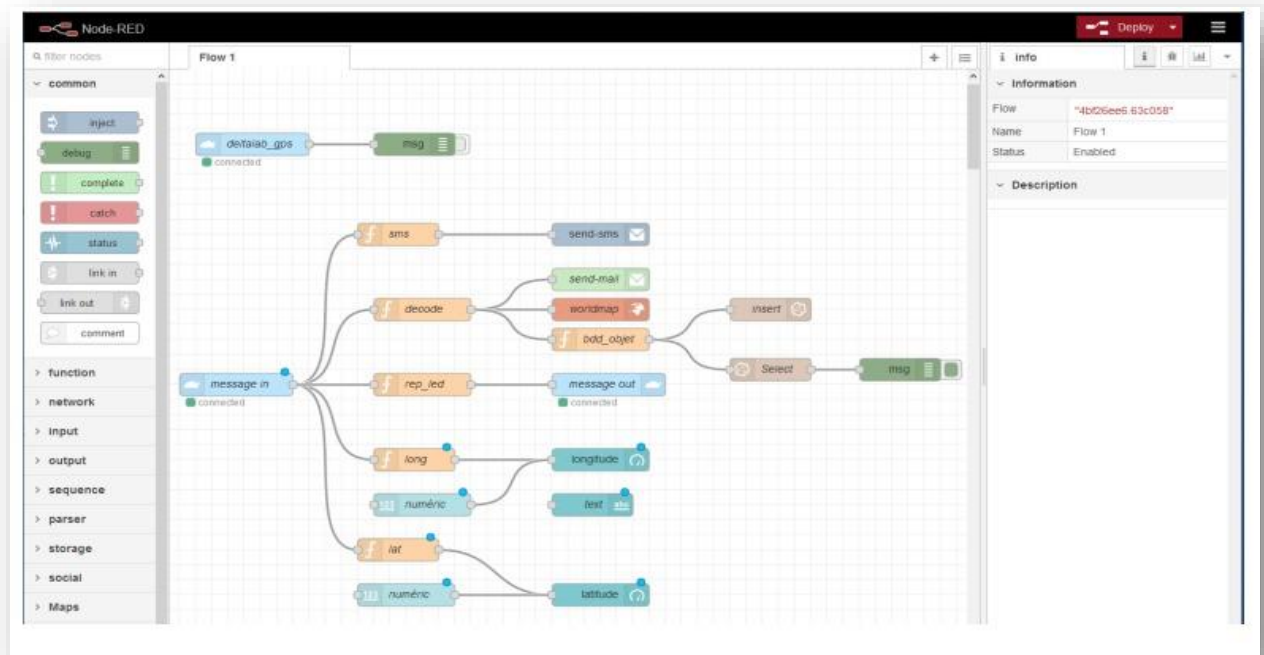


Figure 7 : Interface node-red

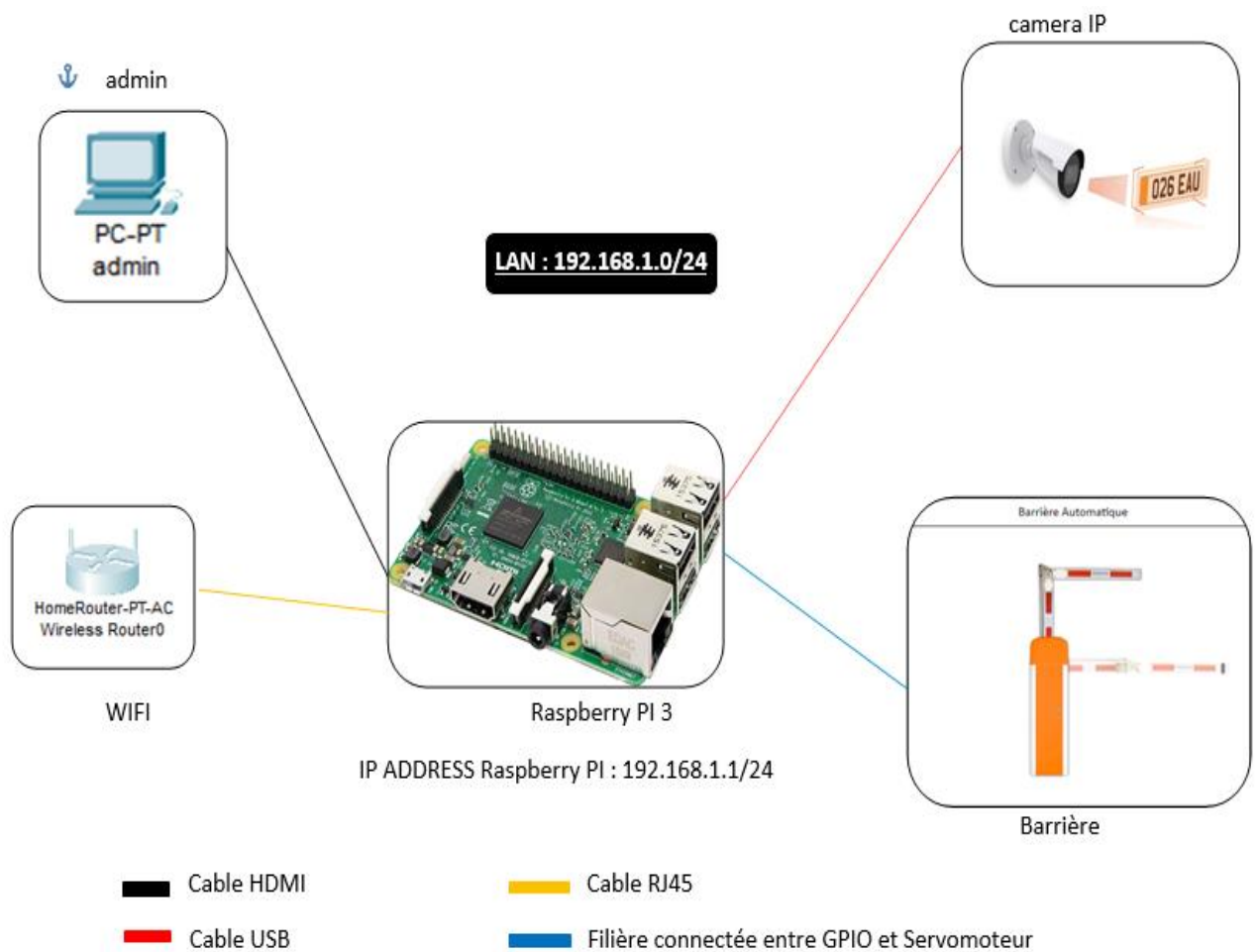
1.2.4. MOTION :

Un logiciel pour la surveillance vidéo et l'accès en ligne, disponible pour la **Raspberry Pi**. La fonction première de **Motion** est de faire de la vidéo-surveillance avec diffusion en ligne, en temps réel, via une ou plusieurs caméras. Motion permet également de faire de la « motion capture » c'est-à-dire que Motion est capable de comparer la dernière image prise avec la nouvelle pour faire de la détection de mouvements.

Motion est un logiciel très simple d'utilisation disponible pour Raspbian et parfaitement adapté à la vidéo-surveillance avec la Raspberry Pi.

2) Fonctionnement de l'architecture :

Le fonctionnement du projet doit se base sur une bonne architecture qui est une représentation abstraite de la réalité. Cette représentation doit être clair et compréhensible par l'être humain qui sera comme un cahier de charge pour une représentation concrète. Dans notre cas nous avons produit une architecture de detection des plaques d'immatriculation pour le contrôle d'accès au parking qui sera détaillé comme suite :



Architecture de detection des plaques d'immatriculation pour le contrôle d'accès au parking

Figure 8 : Architecture de detection des plaques d'immatriculation pour le contrôle d'accès au parking



Figure 9 : présentation de la voiture à l'entre du parking avec le positionnement du camera

Le véhicule s'arrête en face de l'entrée juste avant la barrière. Une caméra dédiée automatiquement reconnaît et enregistre le numéro de la plaque. Les données captures seront envoyées vers la Raspberry Pi grâce à la bibliothèque OpenALPR qui va détecte le numéro de la plaque et va interroger la base de données si la plaque est déjà inscrite.

- Si c'est le cas La barrière d'entrée s'élève automatique permettant l'accès du conducteur au parking. La barrière diminue automatiquement, une fois qu'un véhicule passe le capteur de sécurité.
- Si ce n'est pas le cas La barrière d'entrée reste intacte.

Dans l'architecture chaque partie joue un rôle important comme :

- **Camera** : elle sera positionnée afin de capter la plaque d'immatriculation du véhicule lorsque ce dernier présentera à l'entrée du parking.
- **Raspberry Pi** : 70% du projet s'effectue sur la Raspberry pi grâce à des logiciels et des bibliothèques installaient dans la Raspberry Pi OS qui joue un rôle important sur la reconnaissance des plaques et le contrôle d'accès au parking. Les bibliothèques sont **OpenALPR** avec ces dépendances comme (**open CV** et **Tesseract OCR**), **Motion** pour faire du streaming via le camera et les logiciels comme **Node-red** et **PhpMyAdmin** pour la base de donne MySQL. Les bibliothèques sont en relation avec la caméra et un inter-logiciel permettant de faire communiquer les bibliothèques et les logiciels appelées Node-red. Il permet d'interconnecter des applications hétérogènes devant échanger des informations. On parle d'échange de message. Ce dernier joue un rôle important sur le projet en communicant OpenALPR et affiche le numéro de la plaque si elle est détectée, communique aussi la base de données PhpMyAdmin pour la vérification du numéro de la plaque, et en fin il communique aussi avec le servomoteur pour la barrière automatique.
- **Barrière** : Cette barrière automatique est assurée par un **servomoteur** qui reçoit des instructions qui lui permet de faire des mouvements. Elle est fermée par défaut et s'ouvre si la plaque du véhicule est reconnue.

- **Administrateur :** administrateur joue un rôle très important, car c'est lui qui a l'accès du système d'exploitation Raspberry Pi grâce à un câble HDMI sur un écran ou via l'accès à distance sur le Raspberry pi par le logiciel VNC Viewer. Il a le pouvoir d'installer des logiciels et des bibliothèques sur le terminal. C'est lui qui lance la commande de détection des numéros de plaque d'immatriculation dans le terminal : alprd.

Chapitre 2 : Mise en place du système de reconnaissance des plaques d'immatriculation

Pour que le camera détecte la plaque et que le que la barrière s'élève automatiquement à l'entrée du parking il est nécessaire de mettre en place les systèmes de reconnaissance des plaques a la base de données.

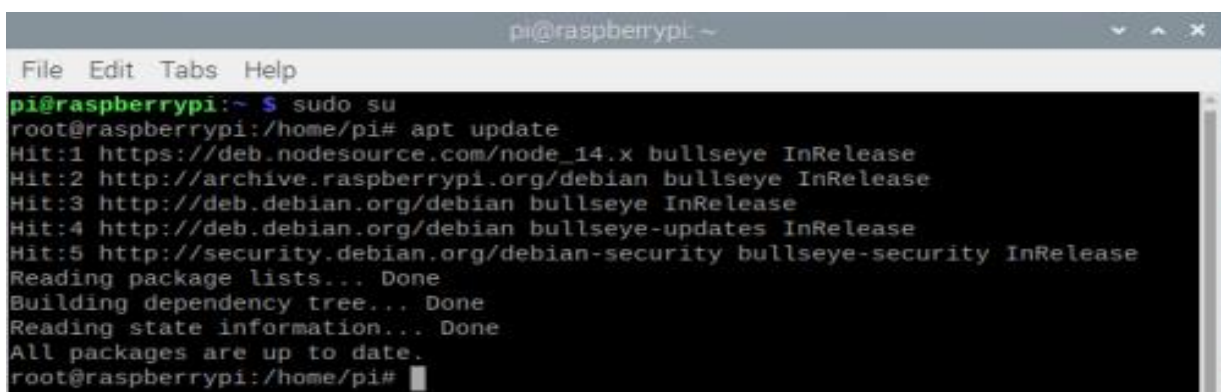
1. INSTALLATION DE LOGICIEL :

Les logiciels sont installés dans la Raspberry Pi et seront mise en œuvre ci-dessous :

1.1.Mise en œuvre d'une installation node-red :

Node-red est un outil de programmation open source base sur les flux conçu pour nous aider à connecter des périphériques, matériels, des API et des services en ligne. Nous allons installer node-red sur notre Raspberry.

Dans un premier temps, on va faire mise en jour des paquets à l'aide des deux commandes et basculer dans le mode administrateur :



```
pi@raspberrypi: ~  
File Edit Tabs Help  
pi@raspberrypi:~ $ sudo su  
root@raspberrypi:/home/pi# apt update  
Hit:1 https://deb.nodesource.com/node_14.x bullseye InRelease  
Hit:2 http://archive.raspberrypi.org/debian bullseye InRelease  
Hit:3 http://deb.debian.org/debian bullseye InRelease  
Hit:4 http://deb.debian.org/debian bullseye-updates InRelease  
Hit:5 http://security.debian.org/debian-security bullseye-security InRelease  
Reading package lists... Done  
Building dependency tree... Done  
Reading state information... Done  
All packages are up to date.  
root@raspberrypi:/home/pi#
```

```
root@raspberrypi:/home/pi# apt upgrade
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
Calculating upgrade... Done
The following package was automatically installed and is no longer required:
  libfuse2
Use 'sudo apt autoremove' to remove it.
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
root@raspberrypi:/home/pi#
```

Nous devons installer le package « build-essential ». Ce package est utilisé pour créer tous les modules requis par npm :

```
root@raspberrypi:~# sudo apt-get install -y nodejs build-essential
```

Après on installe git :

```
root@raspberrypi:/home/pi# sudo apt-get install git
```

On fait la vérification la version de nodejs et npm en utilisant la commande :

```
root@raspberrypi:~# node -v
v14.19.0
root@raspberrypi:~# npm -v
6.14.16
root@raspberrypi:~#
```

Une fois la version vérifiée on installe « **curl** » en utilisant la commande :

```
root@raspberrypi:/home/pi# sudo apt-get install curl
```

Après installation « curl » on installe « **node.js** » en utilisant la commande :

```
root@raspberrypi:/home/pi# curl -sL https://deb.nodesource.com/setup_14.x | sudo -E bash
```

On installe « **node-red** » en utilisant la commande :

```
root@raspberrypi:~# sudo npm install -g --unsafe -perm node-red
```

Pour démarrer node red on tape la commande : **sudo node-red**

```
root@raspberrypi:~# node-red
12 Mar 01:08:01 - [info]

Welcome to Node-RED
=====

12 Mar 01:08:01 - [info] Node-RED version: v2.2.2
12 Mar 01:08:01 - [info] Node.js version: v14.19.0
12 Mar 01:08:01 - [info] Linux 5.10.103-v8+ arm64 LE
12 Mar 01:08:02 - [info] Loading palette nodes
12 Mar 01:08:07 - [info] Dashboard version 3.1.6 started at /ui
12 Mar 01:08:09 - [info] Settings file : /root/.node-red/settings.js
12 Mar 01:08:09 - [info] Context store : 'default' [module=memory]
12 Mar 01:08:09 - [info] User directory : /root/.node-red
12 Mar 01:08:09 - [warn] Projects disabled : editorTheme.projects.enabled=false
12 Mar 01:08:09 - [info] Flows file : /root/.node-red/flows.json
12 Mar 01:08:09 - [info] Server now running at http://127.0.0.1:1880/
12 Mar 01:08:09 - [warn]
```

```
-----
Your flow credentials file is encrypted using a system-generated key.
```

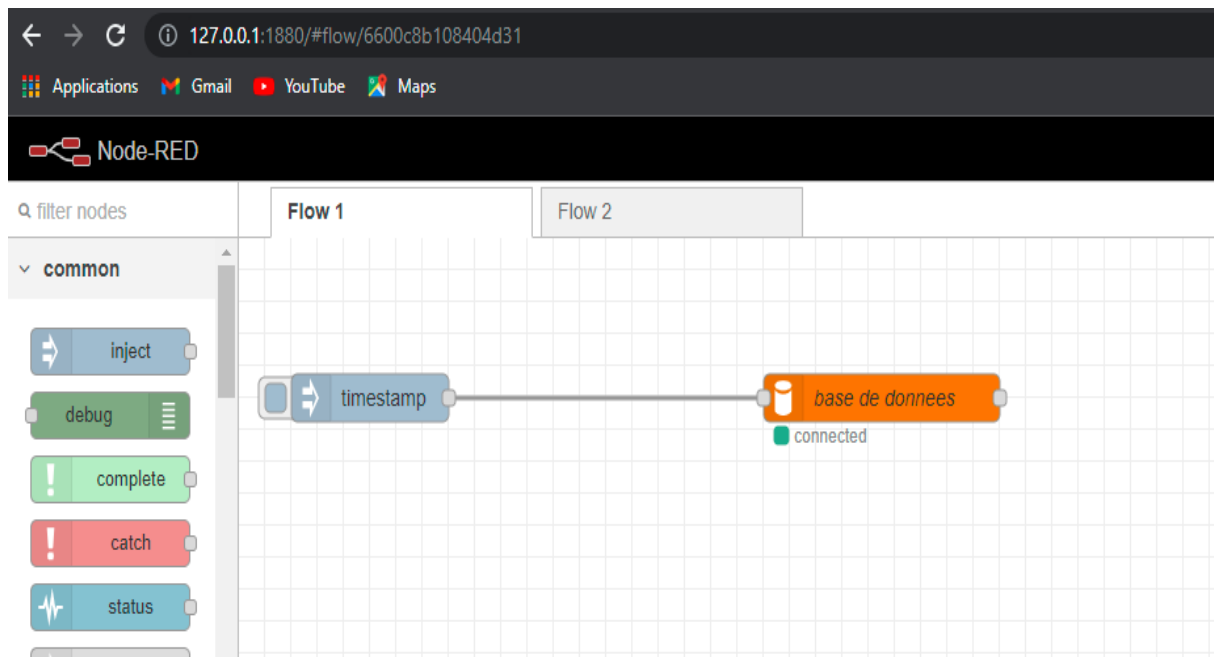
```
If the system-generated key is lost for any reason, your credentials
file will not be recoverable, you will have to delete it and re-enter
your credentials.
```

```
You should set your own key using the 'credentialSecret' option in
your settings file. Node-RED will then re-encrypt your credentials
file using your chosen key the next time you deploy a change.
```

```
-----
```

```
127.0.0.1
```

Puis on va dans notre moteur de recherche et on tape sur url :**127.0.0.1 :1880**



1.2. Mise en œuvre d'une installation motion :

C'est un logiciel très simple à utiliser disponible pour le Raspbian et parfaitement adapté à la vidéo-surveillance avec Raspberry pi. Maintenant nous allons installer motion sur notre Raspberry.

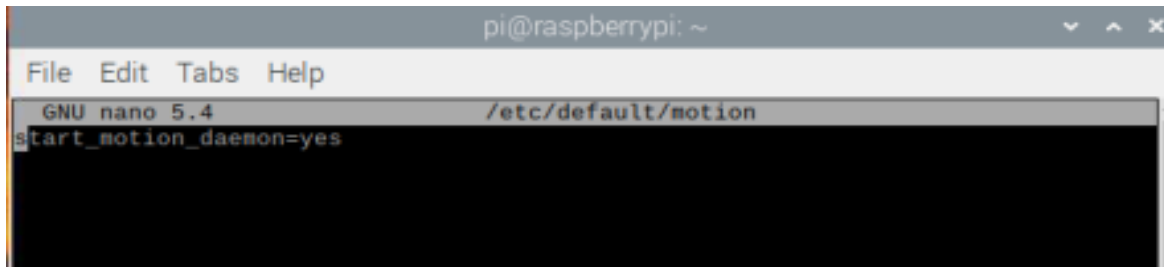
Dans un premier temps, on va faire mise à jour des paquets à l'aide des deux commandes suivantes :

Sudo apt-get update et sudo apt-get upgrade

Une fois de mettre les paquets de Raspberry pi à jour, On installe motion en utilisant la ligne de commande :

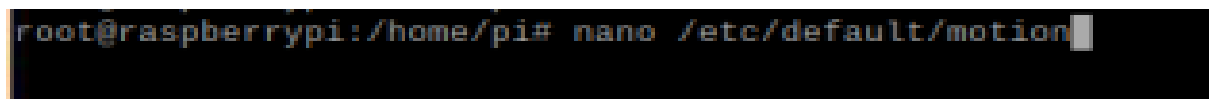
```
root@raspberrypi:/home/pi# apt-get install motion
```

Maintenant motion est installé, nous allons donc configurer motion sur notre Raspberry pi, en éditant le fichier « **/etc/default/motion** » et modifier la ligne « **start_motion_daemon=yes** » en :



```
pi@raspberrypi: ~
File Edit Tabs Help
GNU nano 5.4 /etc/default/motion
start_motion_daemon=yes
```

Maintenant, nous modifions un certain nombre de points de la configuration de motion, enfin de régler au mieux notre système de vidéo. Pour cela on édite notre fichier « **/etc/motion.conf** » dans certains cas ce fichier se trouvera plutôt dans « **/etc/motion/motion.conf** »



```
root@raspberrypi:/home/pi# nano /etc/default/motion
```

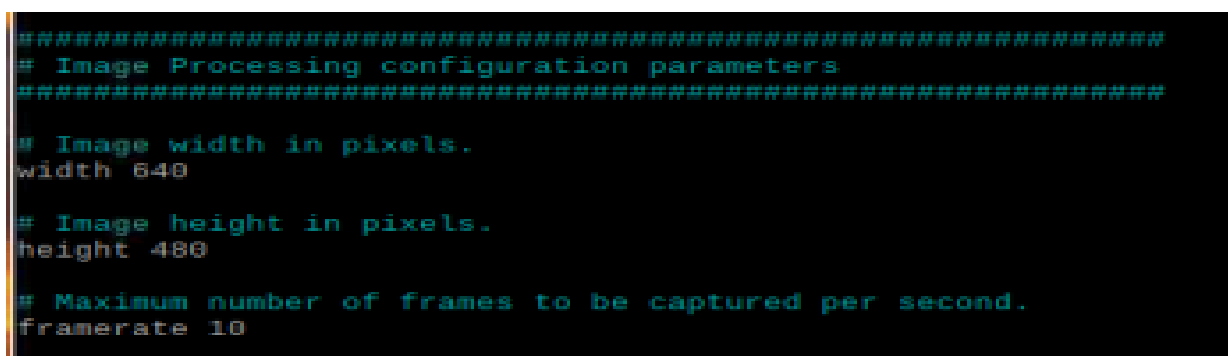
Nous modifions certaine ligne de ce fichier de façon à ce qu'elles correspondent aux lignes ci-dessous :

On commence par mettre « **on** » sur « **daemon** »



```
daemon on
```

Ensuite on choisit les dimensions de l'image « **640** » et « **480** » :



```
#####
# Image Processing configuration parameters
#####

# Image width in pixels.
width 640

# Image height in pixels.
height 480

# Maximum number of frames to be captured per second.
framerate 10
```



```
# The number of pre-captured (buffered) pictures from before motion.
pre_capture 2

# Number of frames to capture after motion is no longer detected.
post_capture 5
```

```
#####
# Picture output configuration parameters
#####

# Output pictures when motion is detected
picture_output off
```

Ensuite le « **movie_quality** » c'est la qualité de visionnage :

```
# The encoding quality of the movie. (0=use bitrate. 1=worst quality, 100=best)
movie_quality 100
```

```
# Restrict webcontrol connections to the localhost.
webcontrol_localhost off
```

Stream_localhost : On met « **off** » sur « **picture_output** » pouvoir accéder à la caméra depuis l'extérieur du réseau local :

```
#####
# Live stream configuration parameters
#####

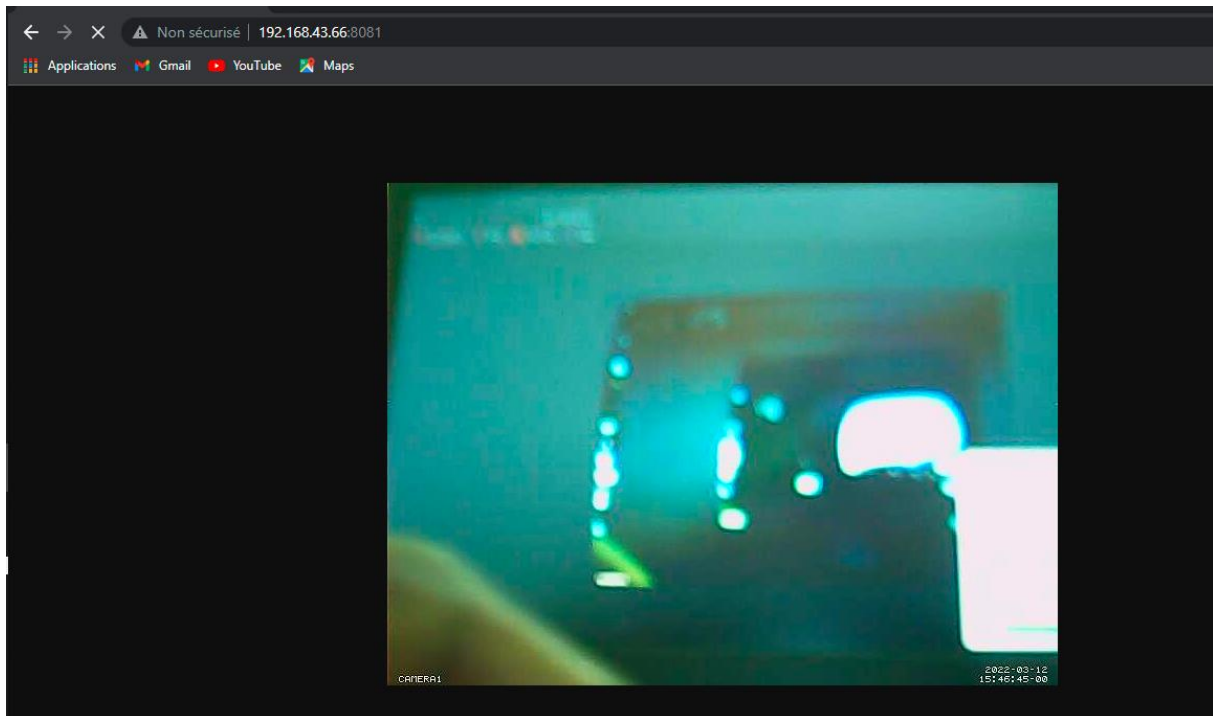
# The port number for the live stream.
stream_port 8081

# Restrict stream connections to the localhost.
stream_localhost off
```

On sauvegarde avec **ctrl-o**, valider, puis quitter avec **ctrl-x**

On démarre motion avec « **sudo motion** » et « **sudo reboot** » pour redémarrer le système

On va aller dans url et tapé @ IP Raspberry **192.168.43.66 :8081**



1.4. Mise en œuvre d'une installation OpenALPR :

OpenALPR est un logiciel de reconnaissance de plaque d'immatriculation. Ce logiciel s'appuie sur la bibliothèque open CV et le logiciel de reconnaissance de caractères tesseract. Avant d'installer OpenALPR on d'abord installe les dépendances tel qu'on tape la commande :

```
pi@raspberrypi:~$ sudo su
root@raspberrypi:/home/pi# sudo apt-get install libopencv-dev libtesseract-dev git cmake build-essential libleptonica-dev curl libcurl4-openssl-dev beanstalkd openjdk-17-jdk liblog4cplus-dev
```

Après on entre dans le fichier **cd /usr/src/**

```
pi@raspberrypi:~$ sudo su
root@raspberrypi:/home/pi# cd /usr/src/
```

Une fois dans le fichier on Install openALPR

```
root@raspberrypi:/usr/src# git clone https://github.com/openalpr/openalpr.git
```

On tape la commande **ls** pour voir le contenu du fichier

```
root@raspberrypi:/usr/src# ls
openalpr  sense-hat
```

Après on entre dans le fichier « openalpr/ » en faisant **cd openalpr/**

```
root@raspberrypi:/usr/src# cd openalpr/
```

On utilise la commande « **ls** » pour voir le contenu et créer aussi dossier « **build** »

```
root@raspberrypi:/usr/src# cd openalpr
root@raspberrypi:/usr/src/openalpr# ls
cla.txt  config  CONTRIBUTING.md  distros  doc  Dockerfile  LICENSE  README.md  runtime_data  src
root@raspberrypi:/usr/src/openalpr# cd src
root@raspberrypi:/usr/src/openalpr/src# ls
bindings  CMakeLists.txt  daemon  inc  misc_utilities  plate_push.py  tclap  video
build  cmake_modules  daemon.cpp  main.cpp  openalpr  statedetection  tests
root@raspberrypi:/usr/src/openalpr/src# mkdir build
```

Sur le dossier build on installe le paquet « **CMAKE et MAKE** » en tapant la commande :

```
root@raspberrypi:/usr/src/openalpr/src# cd build
root@raspberrypi:/usr/src/openalpr/src/build# cmake -D CMAKE_INSTALL_PREFIX:PATH=/usr -D CMAKE_INSTALL_SYSCONFDIR:PATH=/etc ..
```

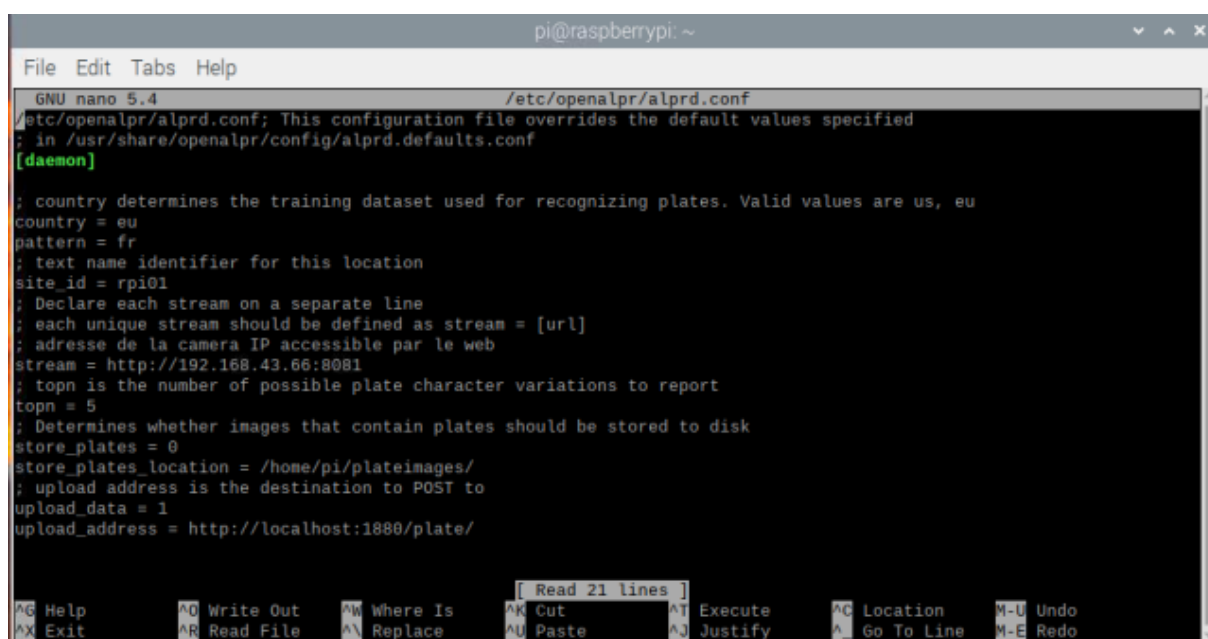
```
root@raspberrypi:/usr/src/openalpr/src/build# make
```

```
root@raspberrypi:/usr/src/openalpr/src/build# make install
```

Après l'installation on commence à faire une sauvegarde du fichier de configuration (openalprd.conf) avec la commande : ctrl+o

```
root@raspberrypi:~# nano /etc/openalpr/alprd.conf
```

La configuration du serveur se fait à partir du fichier `/etc/openalpr/alprd.conf`. Et indique le chemin d'accès au flux vidéo de la caméra IP.



```
pi@raspberrypi: ~
File Edit Tabs Help
GNU nano 5.4 /etc/openalpr/alprd.conf
/etc/openalpr/alprd.conf; This configuration file overrides the default values specified
; in /usr/share/openalpr/config/alprd.defaults.conf
[daemon]

; country determines the training dataset used for recognizing plates. Valid values are us, eu
country = eu
pattern = fr
; text name identifier for this location
site_id = rpi01
; Declare each stream on a separate line
; each unique stream should be defined as stream = [url]
; adresse de la camera IP accessible par le web
stream = http://192.168.43.66:8081
; topn is the number of possible plate character variations to report
topn = 5
; Determines whether images that contain plates should be stored to disk
store_plates = 0
store_plates_location = /home/pi/plateimages/
; upload address is the destination to POST to
upload_data = 1
upload_address = http://localhost:1880/plate/

[ Read 21 lines ]
^G Help      ^O Write Out  ^W Where Is   ^K Cut        ^T Execute    ^C Location   M-U Undo
^X Exit      ^R Read File  ^\ Replace    ^U Paste      ^J Justify    ^_ Go To Line  M-E Redo
```

Après la configuration on lance la commande « **alprd -f** » et placer une image comportant une plaque d'immatriculation devant le camera et on tape la commande :

« **Alpr -c eu/home/pi/downloads/d.jpeg** » pour le résultat :

```
root@raspberrypi:~# alpr -c eu /home/pi/Downloads/d.jpeg
plate0: 10 results
- DK163M      confidence: 84.3131
- DK163       confidence: 82.4487
- 1DK163M     confidence: 81.7383
- DK163M      confidence: 79.9251
- 1DK163      confidence: 79.874
- DK163M      confidence: 78.6074
- DK1673M     confidence: 77.9893
- DK163W      confidence: 77.5796
- 1DK163H     confidence: 77.3504
- DK163M      confidence: 76.5124
root@raspberrypi:~#
```

1.4. Mise en œuvre d'une installation PhpMyAdmin :

PhpMyAdmin est une application Web open source qui offre une interface Web pour gérer et accéder directement aux bases de données MySQL ou Maria DB de n'importe où/à distance à l'aide d'un navigateur Web. L'utilisateur peut utiliser l'interface utilisateur graphique Web fournie par celui-ci pour interagir avec les bases de données sans avoir une connaissance approfondie des commandes.

Dans un premier temps, on va faire mise en jour des paquets à l'aide des deux commandes suivantes :

Sudo apt-get update et sudo apt-get upgrade :

Installez Apache, PHP et Maria DB

Eh bien, phpMyAdmin nécessite que le serveur Web Apache soit servi via un navigateur Web en utilisant le réseau local ou Internet. Par conséquent, nous devons installer la même chose sur votre serveur :

```
pi@raspberrypi:~$ sudo su
root@raspberrypi:/home/pi# sudo apt install apache2
```

Une fois installer « apache2 » on vérifie l'état du serveur Web en utilisant la commande :

```

root@raspberrypi:/home/pi# sudo systemctl enable --now apache2
Synchronizing state of apache2.service with SysV service script with /lib/systemd/systemd-sysv-install.
Executing: /lib/systemd/systemd-sysv-install enable apache2
root@raspberrypi:/home/pi# systemctl status apache2
● apache2.service - The Apache HTTP Server
   Loaded: loaded (/lib/systemd/system/apache2.service; enabled; vendor prese
   Active: active (running) since Fri 2022-03-11 18:51:38 GMT; 6h ago
     Docs: https://httpd.apache.org/docs/2.4/
    Main PID: 611 (apache2)
      Tasks: 6 (limit: 948)
         CPU: 940ms
    CGroup: /system.slice/apache2.service
            └─611 /usr/sbin/apache2 -k start
              └─674 /usr/sbin/apache2 -k start
                └─675 /usr/sbin/apache2 -k start
                  └─676 /usr/sbin/apache2 -k start
                    └─677 /usr/sbin/apache2 -k start
                      └─678 /usr/sbin/apache2 -k start

mar 11 18:51:34 raspberrypi systemd[1]: Starting The Apache HTTP Server...
mar 11 18:51:37 raspberrypi apachectl[519]: AH00557: apache2: apr_sockaddr_info
mar 11 18:51:37 raspberrypi apachectl[519]: AH00558: apache2: Could not reliabl
mar 11 18:51:38 raspberrypi systemd[1]: Started The Apache HTTP Server.
lines 1-19/19 (END)

```

Comme **phpMyAdmin** est basé sur PHP, nous devons donc l'installer avec les extensions courantes de php.

```

root@raspberrypi:/home/pi# sudo apt install php php-cgi php-pear php-mbstring libapache2-mod-php php-common php-ph
seclib php-zip php-curl php-xml php-mbstring php-zip php-gd -y

```

Si tu **déjà** avoir une base **de données** installé sur votre serveur Debian 11 où vous installez phpMyAdmin où vous avez une base de données sur tout autre serveur que vous souhaitez gérer à l'aide de phpMyAdmin puis laissez ces étapes sinon installez **Maria DB/MySQL** :

```

root@raspberrypi:/home/pi# sudo apt install mariadb-server

```

On vérifie l'état :

```

root@raspberrypi:/home/pi# systemctl status mariadb
● mariadb.service - MariaDB 10.5.12 database server
   Loaded: loaded (/lib/systemd/system/mariadb.service; enabled; vendor preset: enabled)
   Active: active (running) since Fri 2022-03-11 18:51:41 GMT; 6h ago
     Docs: man:mariadb(8)
           https://mariadb.com/kb/en/library/systemd/
   Main PID: 596 (mariabdd)
    Status: "Taking your SQL requests now..."
     Tasks: 10 (limit: 948)
        CPU: 4.191s
   CGroup: /system.slice/mariadb.service
           └─596 /usr/sbin/mariabdd

mar 11 18:51:40 raspberrypi mariabdd[596]: 2022-03-11 18:51:40 0 [Note] InnoDB: Loading buffer pool(s) from /var/lib/
mar 11 18:51:40 raspberrypi mariabdd[596]: 2022-03-11 18:51:40 0 [Note] Plugin 'FEEDBACK' is disabled.
mar 11 18:51:40 raspberrypi mariabdd[596]: 2022-03-11 18:51:40 0 [Note] Server socket created on IP: '127.0.0.1'.
mar 11 18:51:41 raspberrypi mariabdd[596]: 2022-03-11 18:51:41 0 [Note] InnoDB: Buffer pool(s) load completed at 20
mar 11 18:51:41 raspberrypi mariabdd[596]: 2022-03-11 18:51:41 0 [Note] Reading of all Master_info entries succeed
mar 11 18:51:41 raspberrypi mariabdd[596]: 2022-03-11 18:51:41 0 [Note] Added new Master_info '' to hash table
mar 11 18:51:41 raspberrypi mariabdd[596]: 2022-03-11 18:51:41 0 [Note] /usr/sbin/mariabdd: ready for connections.
mar 11 18:51:41 raspberrypi mariabdd[596]: Version: '10.5.12-MariaDB-0+deb11u1' socket: '/run/mysqld/mysqld.sock'
mar 11 18:51:41 raspberrypi systemd[1]: Started MariaDB 10.5.12 database server.
mar 11 18:51:41 raspberrypi /etc/mysql/debian-start[661]: Upgrading MySQL tables if necessary.
lines 1-22/22 (END)

```

Une fois terminer, on installe « phpMyAdmin » en utilisant la commande :

```

root@raspberrypi:/home/pi# sudo apt install phpmyadmin

```

Serveur de base de données sécurisé :

Une fois l'installation terminée, exécutez la commande donnée pour sécuriser notre serveur de base de données :

```

root@raspberrypi:/home/pi# mysql_secure_installation

NOTE: RUNNING ALL PARTS OF THIS SCRIPT IS RECOMMENDED FOR ALL MariaDB
SERVERS IN PRODUCTION USE! PLEASE READ EACH STEP CAREFULLY!

In order to log into MariaDB to secure it, we'll need the current
password for the root user. If you've just installed MariaDB, and
haven't set the root password yet, you should just press enter here.

Enter current password for root (enter for none):
OK, successfully used password, moving on...

Setting the root password or using the unix_socket ensures that nobody
can log into the MariaDB root user without the proper authorisation.

You already have your root account protected, so you can safely answer 'n'.

Switch to unix_socket authentication [Y/n] n
... skipping.

You already have your root account protected, so you can safely answer 'n'.

Change the root password? [Y/n] n
... skipping.

By default, a MariaDB installation has an anonymous user, allowing anyone
to log into MariaDB without having to have a user account created for
them. This is intended only for testing, and to make the installation
go a bit smoother. You should remove them before moving into a
production environment.

Remove anonymous users? [Y/n] n
... skipping.

Normally, root should only be allowed to connect from 'localhost'. This
ensures that someone cannot guess at the root password from the network.

Disallow root login remotely? [Y/n] n
... skipping.

By default, MariaDB comes with a database named 'test' that anyone can
access. This is also intended only for testing, and should be removed
before moving into a production environment.

Remove test database and access to it? [Y/n] n
... skipping.

Reloading the privilege tables will ensure that all changes made so far
will take effect immediately.

Reload privilege tables now? [Y/n] y
... Success!

Cleaning up...

All done! If you've completed all of the above steps, your MariaDB
installation should now be secure.

Thanks for using MariaDB!
root@raspberrypi:/home/pi# █

```

Une fois MySQL installé, on utilise la commande « **mysql -u root -p** » pour sécuriser l'accès au phpMyAdmin.

```

root@raspberrypi:/home/pi# mysql -u root -p
Enter password:
Welcome to the MariaDB monitor. Commands end with ; or \g.
Your MariaDB connection id is 33
Server version: 10.5.12-MariaDB-0+deb11u1 Debian 11

Copyright (c) 2006, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]> GRANT ALL PRIVILEGES on *.* to 'admin'@'localhost' IDENTIFIED
BY 'admin';
Query OK, 0 rows affected (0.014 sec)

MariaDB [(none)]> FLUSH PRIVILEGES;
Query OK, 0 rows affected (0.003 sec)

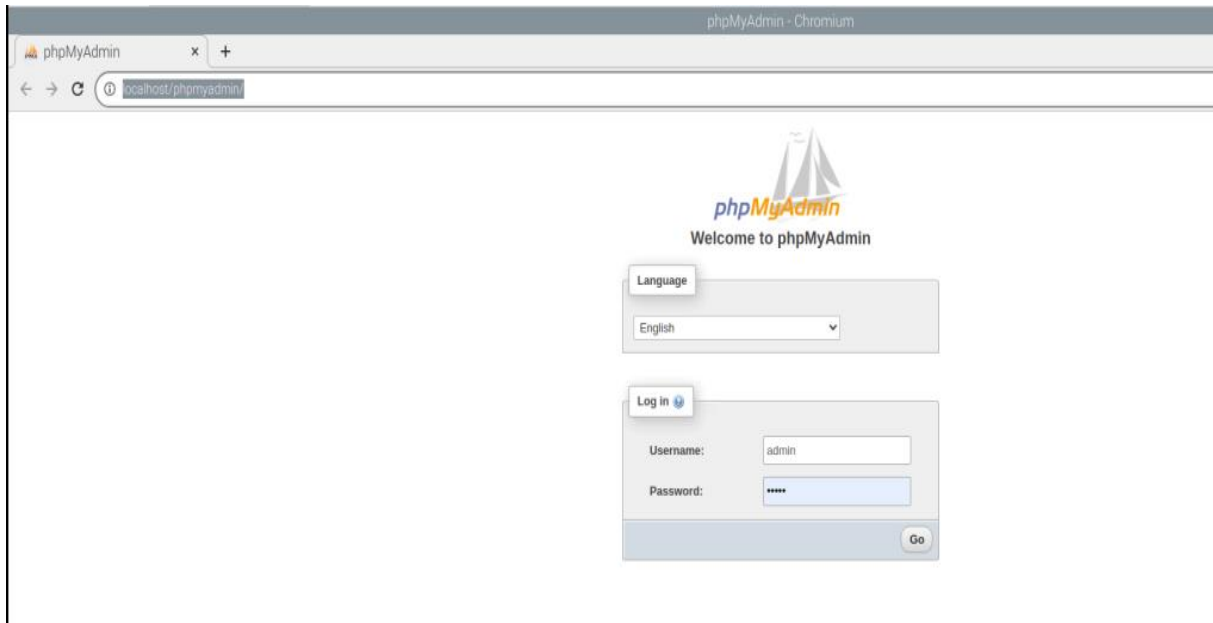
MariaDB [(none)]> \q
Bye
root@raspberrypi:/home/pi# █

```

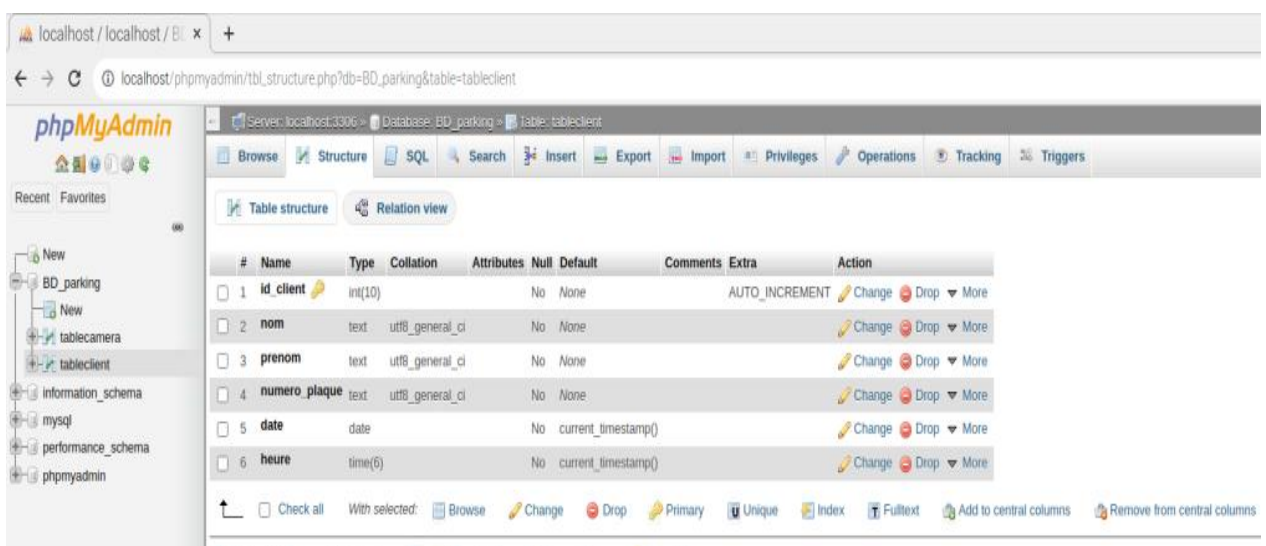

Pour démarrer le serveur on tape la commande :

```
root@raspberrypi:/home/pi# service mysql start
```

Une fois le serveur démarré, il affiche l'interface

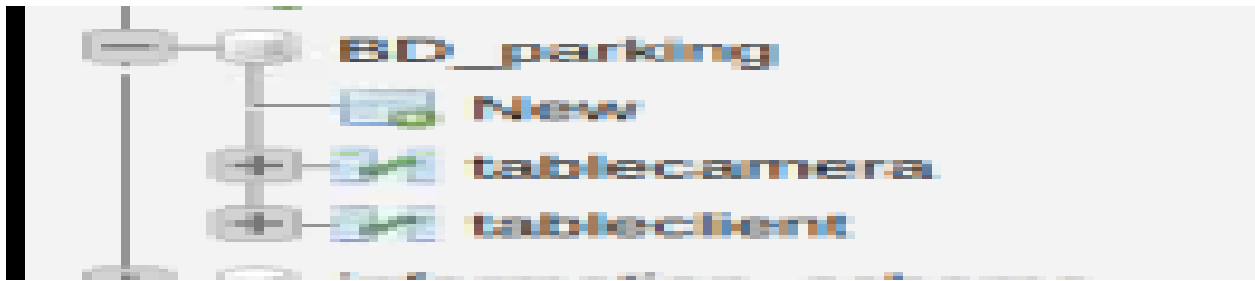


Après on entre le mot de passe et le nom utilisateur :



#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	id_camera	int(10)			No	None		AUTO_INCREMENT	Change Drop More
2	nom	text	utf8_general_ci		No	None			Change Drop More
3	plaque	text	utf8_general_ci		No	None			Change Drop More
4	date	date			No	current_timestamp()			Change Drop More
5	heure	time(6)			No	current_timestamp()			Change Drop More
6	autorisation_camera	varchar(6)	utf8_general_ci		No	None			Change Drop More

Voici les tables :



Pour stopper le serveur on utilise la commande :

```
root@raspberrypi:/home/pi# service mysql stop
```

2. Mise en place d'un programme sur node red :

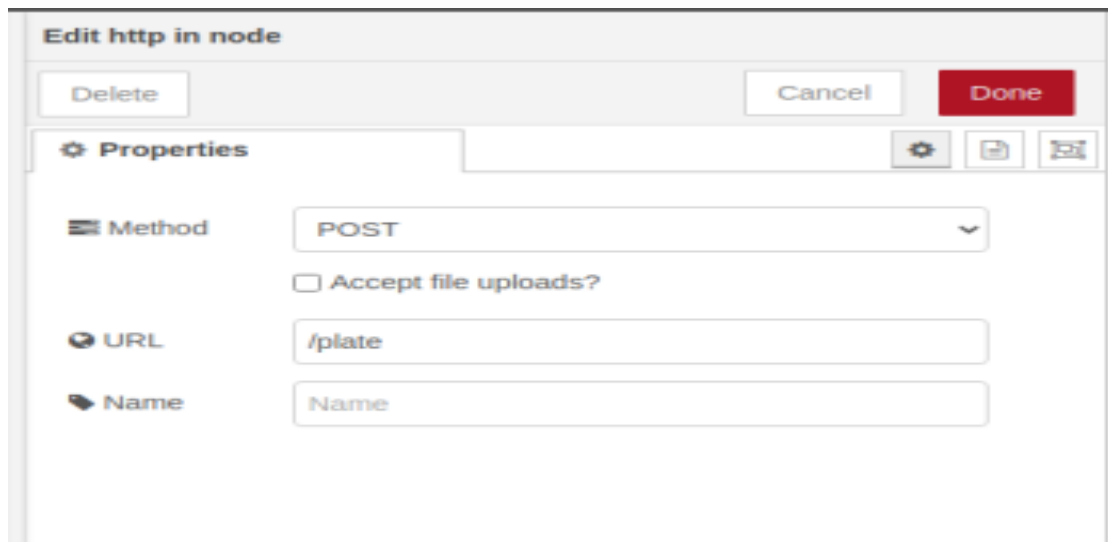
Dans cette partie nous allons mettre en place un programme qui permet de communiquer les différentes applications comme OpenALPR, PhpMyAdmin lors du detection de la plaque d'immatriculation qui permet aussi le servomoteur de levé la barrière de manière automatique grâce à sa connexion au GPIO du Raspberry.

Ce programme s'effectue sur node red grâce à des nœuds et des flux. Ce dernier se rattache entre eux qui permet de véhiculer l'information d'un nœud à un autre nœud. Les nœuds peuvent être différencier en trois catégories qui sont :

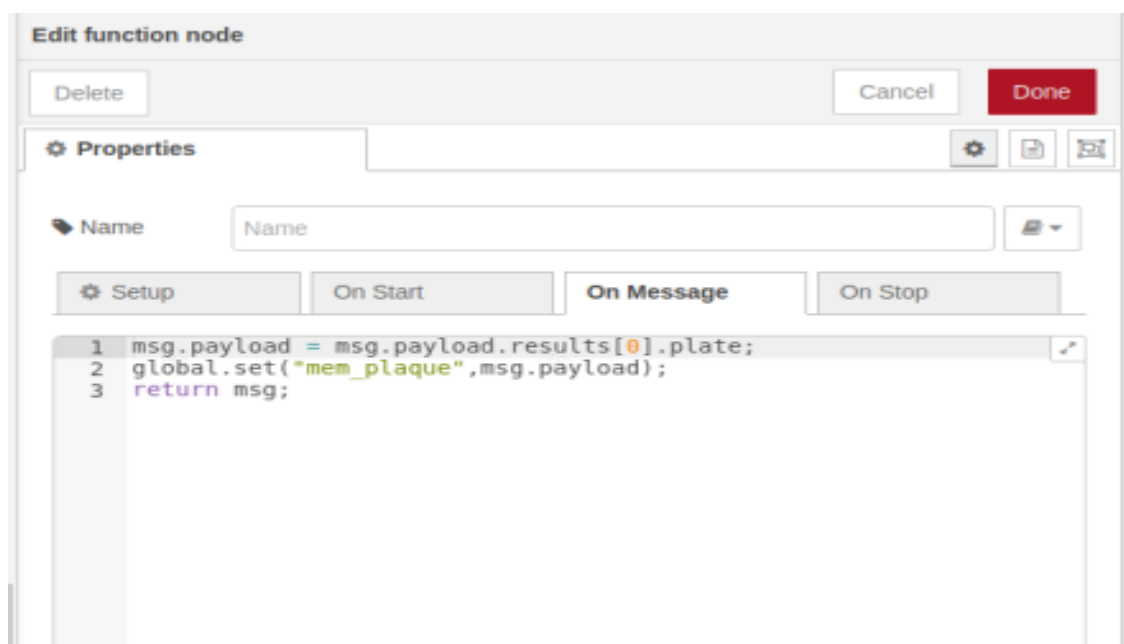
- Les nœuds qui prennent en entrée (in) et en sortie(out)
Exemple : le nœud fonction ou bien le nœud MySQL.
- Les nœuds qui prennent en sortie (out) seulement
Exemple : le nœud http out, Button.
- Les nœuds qui prennent en entrée (in) seulement
Exemple : le nœud http in, Gmail, notification.

Pour tout le long du projet 70% repose sur node red. C'est-à-dire que node red est le point central ou tous les systèmes se connectent. Ainsi nous avons réalisé un programme qui permet d'interconnecter le OpenALPR, PhpMyAdmin, motion et le servomoteur.

Voir ci-dessous :

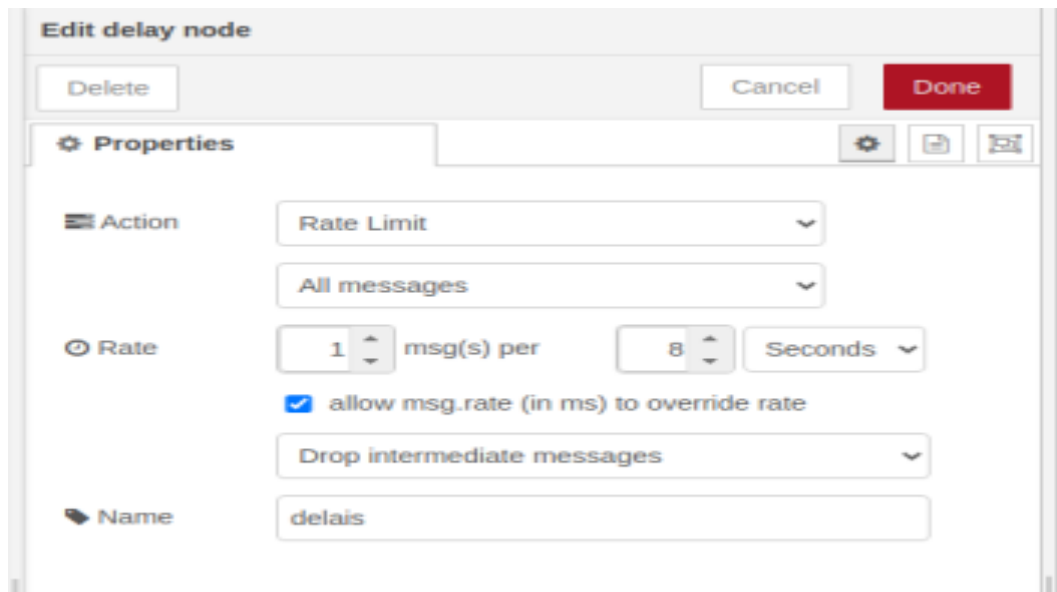


Après la detection de la plaque les informations seront envoyées par le nœud fonction qui permet d'extraire les informations inutiles afin de récupérer le numéro de la plaque spécifiquement. « Msg. Payload » permet de reçoit ou d'envoyé des informations à un autre nœud.



Pour le nœud Délais, le camera détecte une plaque et sera mise en attente tous les 8 seconds

Pour que les informations ne soient pas envoyées de façon répétitive par seconde.



Nous passons à la fonction vérification qui permet d'interroger la base de données SQL par une requête SELECT et permet de vérifier si la plaque détectée par le camera « msg. Payload » existe dans la base de données.



Sur le nœud MSQL Implémentons toutes les informations concernant la base de données comme user, password, le nom de la base de données « BD_parking », le numéro port et le localhost.

Edit mysql node > Edit MySQLdatabase node

Delete Cancel Update

Properties

Host 127.0.0.1

Port 3306

User admin

Password

Database BD_parking

Timezone ±hh:mm

Charset UTF8

Name Name

Tip: The timezone should be specified as ±hh:mm or leave blank for 'local'.

Le nœud du fonction confirmation nous permet de pose des conditions sur le chiffre qu'il retourne de la base de données, si elle a trouvé la plaque en posant une condition if et else ce qui permet lors de l'exécution retourne 0 s'il ne trouve pas et retourne 8 en cas contraire.

Edit function node

Delete Cancel Done

Properties

Name confirmation

Setup On Start On Message On Stop

```

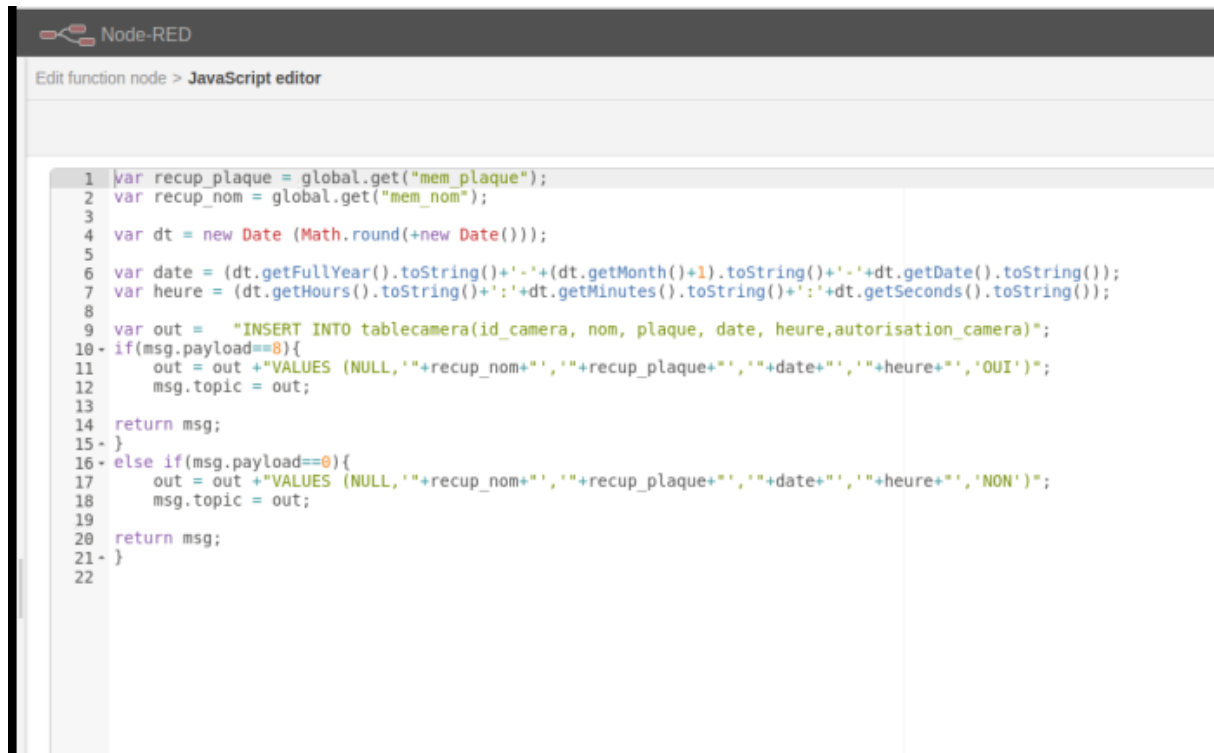
1- if(msg.payload==0){
2-   msg.payload = 0;
3-   return msg;
4- }
5-
6- }
7- else{
8-
9-   var value = msg.payload[0]['nom'];
10-  global.set("mem_nom",value);
11-  msg.payload=8;
12-  return msg;
13-
14-
15-
16- }

```

Concernant le nœud du fonction camera on a realise une insertion sur la table camera en se basant sur le retourne du fonction confirmation 8 et 0 .

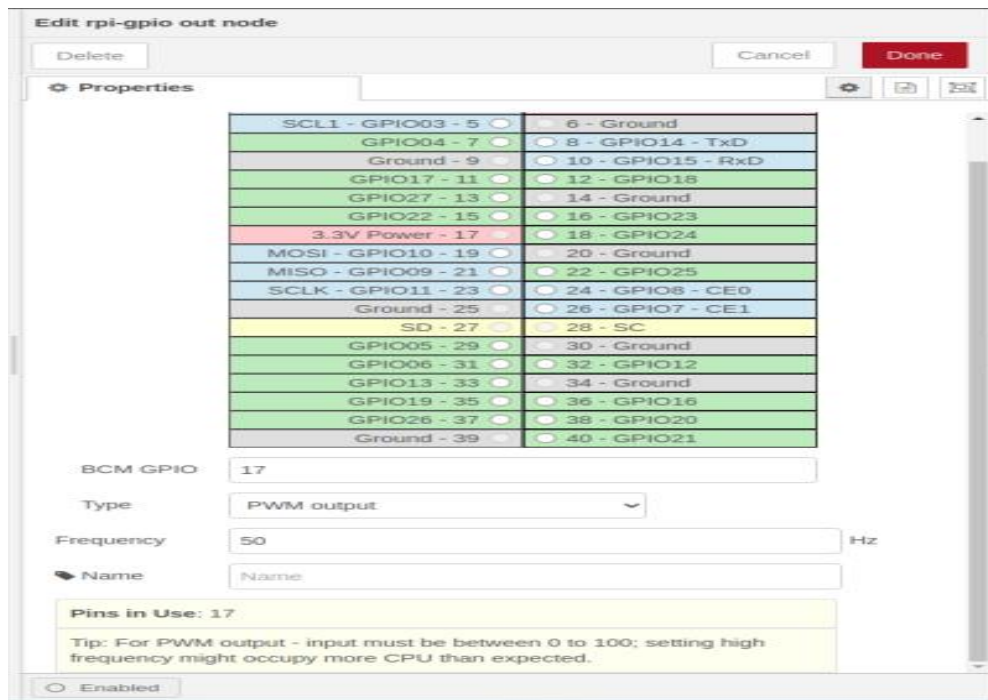
Dans le cas ou la fonction retourne 8 qui montre que la plaque detecte est present dans la base de donnees ce qui permet une autorisation d'accès au parking « OUI » avec la date et l'heure d'accès.

Dans le cas ou la fonction retourne 0 qui montre que la plaque a mal detecte ou ce qu'il a detecte n'est pas dans la base de donnees , ce qui permet une non autorisation d'accès au parking « NON » avec la date et l'heure de refuse.

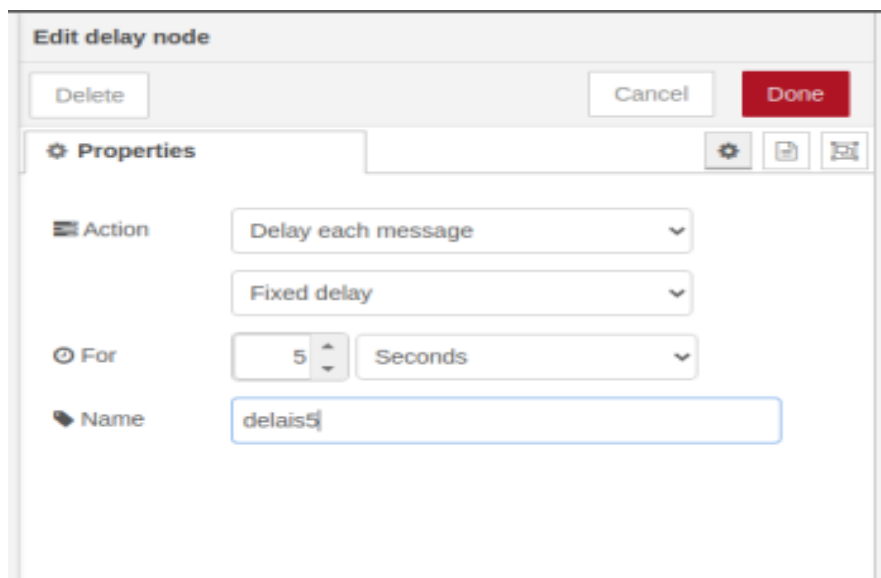
The image shows a screenshot of the Node-RED web interface. At the top, there's a header with the Node-RED logo and the text "Node-RED". Below that, a tab says "Edit function node > JavaScript editor". The main area contains a JavaScript code editor with the following code:

```
1 var recup_plaque = global.get("mem_plaque");
2 var recup_nom = global.get("mem_nom");
3
4 var dt = new Date (Math.round(+new Date()));
5
6 var date = (dt.getFullYear().toString()+'-'+(dt.getMonth()+1).toString()+'-'+dt.getDate().toString());
7 var heure = (dt.getHours().toString()+'-'+dt.getMinutes().toString()+'-'+dt.getSeconds().toString());
8
9 var out = "INSERT INTO tablecamera(id_camera, nom, plaque, date, heure,autorisation_camera)";
10 if(msg.payload==8){
11     out = out +"VALUES (NULL,"+recup_nom+", "+recup_plaque+", "+date+", "+heure+", 'OUI')";
12     msg.topic = out;
13 }
14 return msg;
15 }
16 else if(msg.payload==0){
17     out = out +"VALUES (NULL,"+recup_nom+", "+recup_plaque+", "+date+", "+heure+", 'NON')";
18     msg.topic = out;
19 }
20 return msg;
21 }
22 }
```

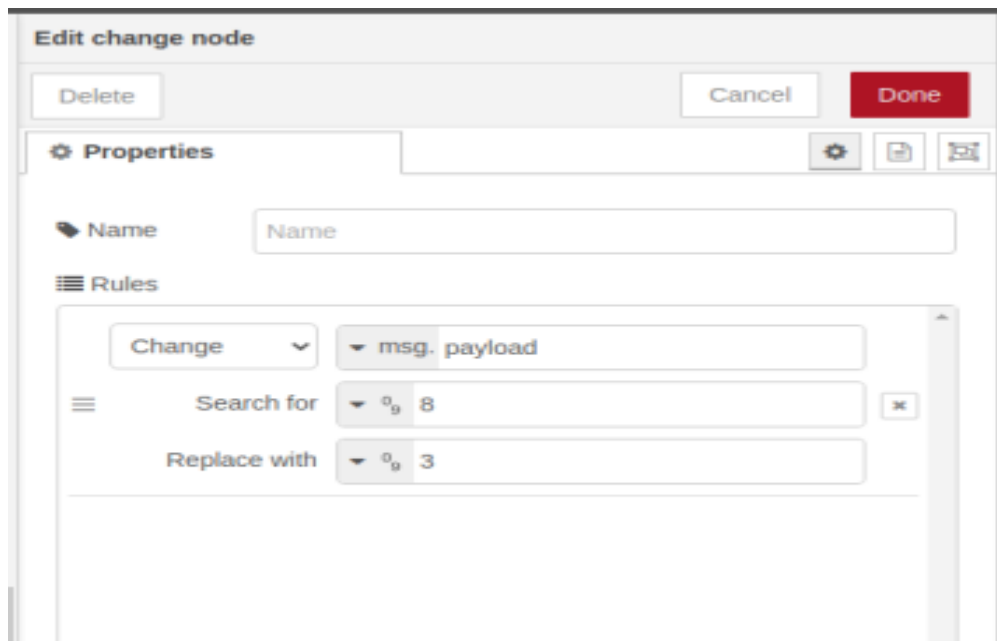
Le nœud GPIO joue un rôle très important sur le levé de la barrière. Il s'associe directement avec le servomoteur par des filières qui permet de l'échange des signaux. Le nœud confirmation retourne le chiffre 8 sur le GPIO qui lui permet de le véhiculé au servomoteur et fait une rotation de 8 pas qui correspond au 90 ° pour que la voiture accède au parking. Le GPIO 17 est responsable de la sortie du signal vers le servomoteur sur une fréquence de 50 Hz.



Ici le nœud délais joue un rôle très important car il reçoit en entre le chiffre 8 qui sera Mise en attente pour une dure de 5 secondes pour que le GPIO travail sur le levé de la barrière.



Après le délai de 5 secondes le chiffre 8 sera envoyé par le nœud échange qui permet d'échanger le 8 par le 3 qui sera l'origine de la baisse de la barrière après une recherche trouve par la base de données.



Ce nœud fonction qui reçoit d'information sur le nœud de confirmation et permet de donner un message dans le cas où la plaque a été trouvé dans la base de données avec le chiffre 8 qu'il retourne.

Dans le cas contraire il retourne le chiffre 0.



Le message a été envoyé directement par une compte « **niangfamily148006@gmail.com** » qui permet de reçoit en temps réel.

Des informations concernant le parking. Certaines informations du compte doivent être remplies sur le nœud email comme l'utilisateur et le mot de passe du compte.

3. Mise en place d'application web et base de données :

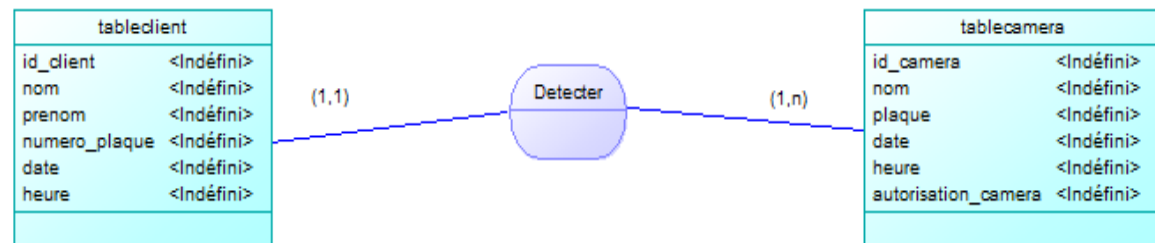
La plaque détectée par la caméra sera stockée dans une base de données qui permet de faire des études sur le fonctionnement du parking. Avant l'entrée du parking, le conducteur doit s'inscrire sur l'administrateur pour y avoir l'accès. Il doit fournir des informations telles que le nom, le prénom et le numéro de la plaque d'immatriculation qui sera enregistré dans une base de données et une application web qui permet d'inscrire et d'afficher les informations dans une table.

Ainsi nous avons réalisé une modélisation de la base de données appelée **BD_parking** constituée de deux tables qui sont :

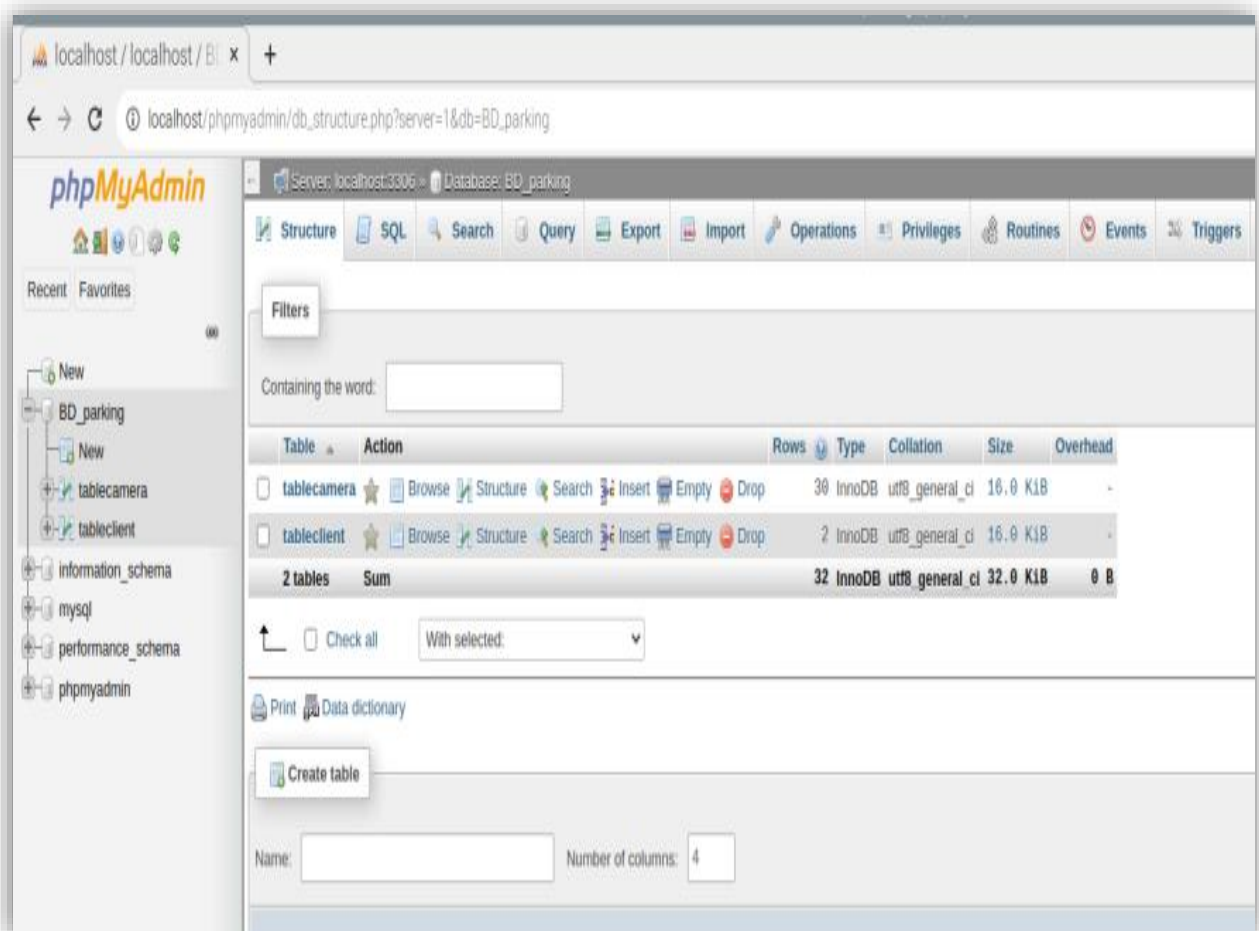
Tableclient (id client, nom, prénom, numero_plaque, date, heure)

Tablecamera (id_camera, nom, plaque, date, heure, autorisation_camera).

Pour les cardinalités, on peut dire que le numero_plaque de la tableclient a été détecté par une et une **id_camera** et que la tablecamera peut être détectée une ou plusieurs plaques.



Dans le phpMyAdmin on voit que le nom de la base de données BD_parking et les deux tables qui sont : tableclient et tablecamera.



Nous avons créé une application qui permet d'inscrire des conducteurs pour accéder au parking.

Dans cette application nous allons donner un exemple d'inscription voir ci-dessous.

The screenshot displays a web application interface with a blue header bar containing the text 'Inscription Client' and a hamburger menu icon. The main content area is white and features a registration form titled 'Inscription' in blue text. The form includes three input fields: 'nom' with the value 'niang', 'prenom' with the value 'mouhamed', and 'plaque' with the value 'AA001AA'. Below the input fields are two blue buttons: 'ENREGISTRE' and 'ANNULE'.

≡ Inscription Client

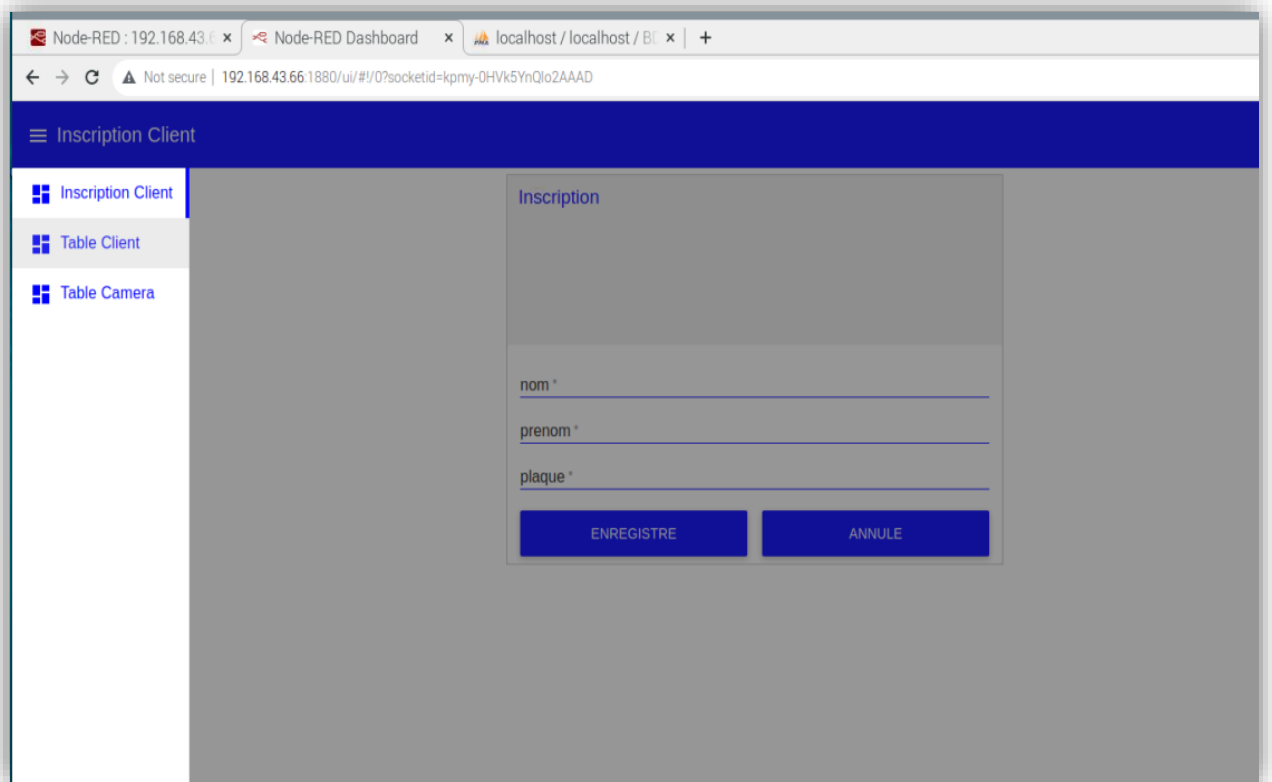
Inscription

nom *
niang

prenom *
mouhamed

plaque *
AA001AA

ENREGISTRE ANNULE



Pour la table client nous avons constaté que la personne (**niang, Mouhamed, AA001AA**) a été inscrit sur la base de données.

De même il s'affiche sur la table client de l'application en cliquant sur actualise.

Node-RED : 192.168.43.1 x Node-RED Dashboard x localhost / localhost / B x +

Not secure | 192.168.43.66:1880/ui/#/1?socketid=kpmY-0HVk5YnQIo2AAAD

Table Client

Table Client

ACTUALISE

id_client	nom	prenom	numero_plaque	date	heure
1	mouhamed	niang	AA001AA	2022-03-11T00:00:00.000Z	13:51:15
2	niang	ngagne	AA999AA	2022-03-12T00:00:00.000Z	14:54:21
4	MOUSSA	DIA	AA012BC	2022-03-14T00:00:00.000Z	15:46:28
5	BABACAR	DIA	AA234AZ	2022-03-14T00:00:00.000Z	15:52:15
6	MOUSTAPHA	MANE	AA234AQ	2022-03-14T00:00:00.000Z	18:50:43
7	ASTOU	BA	AA367AB	2022-03-14T00:00:00.000Z	18:54:54
8	ALSANE	SANE	AB089AA	2022-03-14T00:00:00.000Z	19:01:38

Recherche

plaque *

RECHERCHE ANNULE

Supprime

id_client *

SUPPRIMER ANNULE

Modifier

id_client *

nom *

prenom *

numero_plaque *

MODIFIER ANNULE

Sur la table camera on reçoit tous les informations concernant la detection des plaques par le camera.

On visualise dans cette table les plaques autoris   et les plaques non autoris  . En ce qui nous concerne l'inscription fait par la personne (**niang, Mouhamed, AA001AA**) on a vu que la plaque a   t   autoris  e « **OUI** » et que le camera a d  tect   la plaque **AA001AA**.

Table Camera

Table Camera

ACTUALISE

id_camera	nom	plaque	date	heure	autorisation_camera
1	mouhamed	AA001AA	2022-03-12T00:00:00.000Z	14:51:42	OUI
2	niang	AA999A	2022-03-12T00:00:00.000Z	15:16:27	NON
3	niang	AA012BC	2022-03-12T00:00:00.000Z	15:16:56	NON
4	niang	AA999	2022-03-12T00:00:00.000Z	15:17:14	NON
5	niang	AA999A	2022-03-12T00:00:00.000Z	15:17:24	NON
6	niang	A001A	2022-03-12T00:00:00.000Z	15:17:32	NON
7	niang	AA001AE	2022-03-12T00:00:00.000Z	15:17:44	NON
8	mouhamed	AA001AA	2022-03-12T00:00:00.000Z	15:17:54	OUI
9	mouhamed	AA001AA	2022-03-12T00:00:00.000Z	15:18:02	OUI
11	niang	AA999AA	2022-03-12T00:00:00.000Z	15:27:46	OUI
12	mouhamed	AA001AA	2022-03-12T00:00:00.000Z	15:47:34	OUI

Recherche

plaque *

RECHERCHE ANNULER

RECHERCHE

date *

RECHERCHE ANNULER

Supprime

id_camera *

SUPPRIMER ANNULER

Cela affiche la même chose sur la base de données ce qui prouve que le bon fonctionnement de la base de données. Voir ci-dessous

Node-RED : 192.168.43.6 x Node-RED Dashboard x localhost / localhost / BI x +

localhost/phpmyadmin/sql.php?server=1&db=BD_parking&table=tablecamera&pos=0

phpMyAdmin

Récentes Préférences

Nouvelle base de données

BD_parking

Nouvelle table

tablecamera

tableclient

information_schema

mysql

performance_schema

phpmyadmin

Parcourir Structure SQL Rechercher Insérer Exporter Importer Privileges Opérations Suivi

✓ Affichage des lignes 0 - 10 (total de 11, traitement en 0.0176 seconde(s).)

SELECT * FROM `tablecamera`

Profilage [Éditer en ligne] [Éditer]

☐ Tout afficher Nombre de lignes : 25 Filtrer les lignes: Chercher dans cette table Trier par clé : Aucun(e)

+ Options

	id_camera	nom	plaque	date	heure	autorisation_camera
<input type="checkbox"/> Éditer Copier Supprimer	1	mouhamed	AA001AA	2022-03-12	14:51:42	OUI
<input type="checkbox"/> Éditer Copier Supprimer	2	niang	AA999A	2022-03-12	15:16:27	NON
<input type="checkbox"/> Éditer Copier Supprimer	3	niang	AA012BC	2022-03-12	15:16:56	NON
<input type="checkbox"/> Éditer Copier Supprimer	4	niang	AA999	2022-03-12	15:17:14	NON
<input type="checkbox"/> Éditer Copier Supprimer	5	niang	AA999A	2022-03-12	15:17:24	NON
<input type="checkbox"/> Éditer Copier Supprimer	6	niang	A001A	2022-03-12	15:17:32	NON
<input type="checkbox"/> Éditer Copier Supprimer	7	niang	AA001AE	2022-03-12	15:17:44	NON
<input type="checkbox"/> Éditer Copier Supprimer	8	mouhamed	AA001AA	2022-03-12	15:17:54	OUI
<input type="checkbox"/> Éditer Copier Supprimer	9	mouhamed	AA001AA	2022-03-12	15:18:02	OUI
<input type="checkbox"/> Éditer Copier Supprimer	11	niang	AA999AA	2022-03-12	15:27:46	OUI
<input type="checkbox"/> Éditer Copier Supprimer	12	mouhamed	AA001AA	2022-03-12	15:47:34	OUI

☐ Tout cocher Avec la sélection : Éditer Copier Supprimer Exporter

La réalisation de cette application est essentielle sur notre projet car elle sert à l'administrateur de s'inscrire des personnes désirantes accéder au parking et de savoir les personnes qui sont autorisés ou non autorisés à l'entrée du parking.

Conclusion générale

Depuis longtemps, l'homme a cherché un moyen afin de lutter contre les insécurités et les infractions routières.

Avec le développement de la science et des nouvelles technologies, les moyens rudimentaires sont remplacés par des techniques modernes.

L'homme s'est penché sur des techniques permettant à des machines d'accomplir des tâches et de résoudre des problèmes normalement réservés aux humains.

La vision par ordinateur est utilisée afin de localiser et de reconnaître des plaques d'immatriculation pour pouvoir automatiser

Les surveillances dans des stations de services, dans des autoroutes, dans des parking, etc...

Dans ce travail, nous avons pour objectif de développer un système de détection et de reconnaissance de plaques d'immatriculation.

La phase de reconnaissance se base sur les différents traitements d'images et les différentes méthodes de reconnaissance afin d'avoir des résultats appropriés. La phase de conception et de réalisation du logiciel a pour but principal de bien reconnaître les plaques d'immatriculation en temps réel avec une grande précision sur les chiffres et les lettres formant le matricule du véhicule. En plus, ce projet nous a été très bénéfique car il nous a permis de maîtriser plusieurs techniques comme les traitements d'image, Il nous a aussi permis de manipuler des outils très complexes tels que les librairies OpenALPR avec ses dépendances (Open CV, OCR, Motion). Mais encore, une bonne connaissance concernant Node-Red.

Enfin, ce projet était une bonne occasion pour réaliser un travail très concret, avec des objectifs clairs et bien définis et de se familiariser avec un environnement de développement professionnel.

Il faut dire que des améliorations peuvent encore être apportées à ce système, notamment pour une interface graphique qui pourrait satisfaire l'utilisateur en lui facilitant l'accès dans l'utilisation du logiciel ou encore l'ajout d'une base de données afin de stocker les identités des propriétaires. Ces améliorations peuvent faire l'objet d'une autre étude.

Toutefois, des perspectives intéressantes sont envisagées pour rendre le système plus complet. Par exemple, on pourrait envisager un système de paiement en ligne en temps réel grâce à une application mobile.

Bibliographie

- 1) <https://www.rgot.org/controle-dacces-par-lecture-de-plaques-dimmatriculation-alpr/>
- 2) <http://www.bajart.fr/index.php/2020/11/14/controle-dacces-lecture-de-plaques/>
- 3) [https://www.reddit.com/r/raspberry_pi/comments/baxwz5/how to install o
penalpr on raspberry pi/](https://www.reddit.com/r/raspberry_pi/comments/baxwz5/how_to_install_openalpr_on_raspberry_pi/)
- 4) [https://deltalabprototype.fr/wp-content/uploads/2019/10/Serveur-
NodeRED.pdf](https://deltalabprototype.fr/wp-content/uploads/2019/10/Serveur-NodeRED.pdf)
- 5) [https://www.how2shout.com/linux/how-to-install-phpmyadmin-on-debian-
11-bullseye-apache/](https://www.how2shout.com/linux/how-to-install-phpmyadmin-on-debian-11-bullseye-apache/)
- 6) <https://pfrlju.com/archives/313/>
- 7) www.youtube.com : Lecture plaque d'immatriculation avec ALPRD et Raspberry Pi

