**Partie individuelle**

Projet CA25

Etudiant 3

Accession au local

Raphael RODRIGUEZ GLISE

**Remerciements**

Je tiens à remercier Monsieur **HERZOG**, mon professeur de physique, qui m’a beaucoup aidé dans la réalisation de ce projet.

Son expertise en électricité m’a été d’une grande aide, notamment lors du montage, en particulier pour la mise en place du relais et de la gâche électromécanique.

Je remercie également Monsieur **MAURIN**, mon professeur d’informatique et de réseaux, pour son accompagnement tout au long du projet.

Il m’a grandement facilité la tâche en me fournissant la documentation technique de chaque composant utilisé, ce qui m’a permis de gagner un temps précieux et de mieux comprendre le fonctionnement de l’ensemble du système.

Enfin, je souhaite remercier Monsieur **FOURNIER**, également professeur d’informatique et de réseaux, pour son aide précieuse.

Il m’a notamment conseillé sur la structuration de mon code Python, en m’orientant vers une organisation plus claire et efficace, en utilisant différentes classes réparties dans plusieurs fichiers.

**Sommaire**

**1 Introduction**

* 1. Présentation du candidat
  2. Présentation du projet
  3. Diagramme informel
  4. Attente du point de vu client pour le projet Contrôle d’accès
  5. Diagramme de déploiement
  6. Base de données
  7. Fiche Recette

1. **Présentation de ma partie**
   1. Mes taches
   2. Diagramme de cas d’utilisation
   3. Diagramme de séquence accès au local
   4. Diagramme de classe
   5. Matériel utilisé
   6. Langage utilisé
   7. Logiciel utilisé
2. **Montage sur Raspberry**
   1. Raspberry
   2. Lecteur RFID
   3. Relais mécanique et LED
   4. Gâche électromécanique
   5. Détecteur d’ouverture de porte filaire
   6. Capteur de présence
   7. Montage Global
   8. Partie physique
3. **Amélioration du projet**
   1. Conclusion
4. **Annexe**
   1. Lecteur RFID
   2. Relais mécanique et LED
   3. Gâche électromécanique
   4. Détecteur d’ouverture de porte filaire
   5. Capteur de présence
   6. Code complet du projet

**Présentation du candidat**

Je m’appelle Raphael RODRIGUEZ GLISE j’ai 19 ans, et je suis actuellement en 2 -ème année de mon BTS cybersécurité, informatique et réseaux, électronique (C.I.E.L).

J’ai fait un BAC général avec comme spécialité physique chimie, spécialité NSI (numérique science et informatique) et option math complémentaire.

Lors de ma première année de BTS j’ai dû chercher un stage dans le but de valider ma première année de BTS.

J’ai finalement été pris dans la mairie de ville la grand pendant une durée de 8 semaines.

Dans le cadre du projet CA25 que j’ai dû réaliser pendant la deuxième année de mon BTS j’ai choisi la partie accéder au local qui consiste à permettre à l’utilisateur de badger à l’aide d’une carte NFC.

**Présentation du projet**

CA25 = control accès 2025

L’accès au local technique informatique du bâtiment A du Lycée Charles Poncet se fait grâce à une clé mécanique.

Pour faciliter la gestion de l’accès à ce local, il est envisagé de mettre en place un système de contrôle d’accès informatisé.

Les étudiants de BTS CIEL2 sont chargés de développer ce système de contrôle d’accès à ce local utilisant une carte NFC du lycée pour chaque personne autorisée.

Contraintes financières (budget alloué) :

Budget alloué :

130 € pour un lecteur de carte adapté au RPI,

Un détecteur de présence adapté au RPI, une alimentation assez puissante pour le RPI, le reste du matériel étant fourni.

Utilisation des équipements de la section BTS CIEL du lycée.

Contraintes de développement (matériel t/ou logiciel imposé / technologies utilisées) :

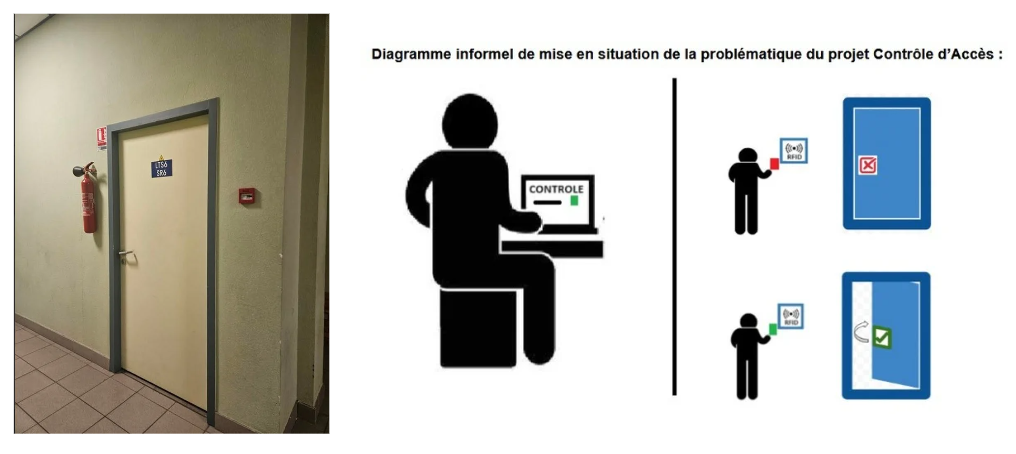
Système embarqué : RPI + capteur et actionneurs OS RPI : Libre EDI : Libre Communications : Ethernet Développements : orientés objet, langages au choix Serveur physique : fourni par le lycée OS serveur Linux : Serveur Web au choix Serveur BDD : MySQL

Solution opérationnelle pour fin mai 2025.

Conforme aux règles de l’art.

**Diagramme informel**

Porte qui s’ouvre avec une clé mécanique



L’administrateur définit les droits d’accès et supervise les accès

L’employé est autorisé ou non à déverrouiller l’accès

**Attente du point de vue du client pour le projet Contrôle d’accès**

Pour la partie Général du projet

L’employé sollicite l’accès au local à partir d’un lien mis à disposition sur l’ENT.

L’administrateur définit les employés ayant le droit d’accès au local informatique et leur fournit une carte NFC nominative.

Seuls les employés autorisés par l’administrateur peuvent déverrouiller la porte d’accès à ce local informatique avec la carte NFC fournie.

Pour des raisons de sécurité, la porte doit pouvoir être déverrouillée de l’intérieur et un détecteur de présence doit informer de la présence ou non de quelqu’un dans le local.

L’administrateur peut être informé en temps réel des tentatives d’accès et des accès, de l’état de la porte et de la présence ou non de quelqu’un dans le local.

**Diagramme de déploiement**

****

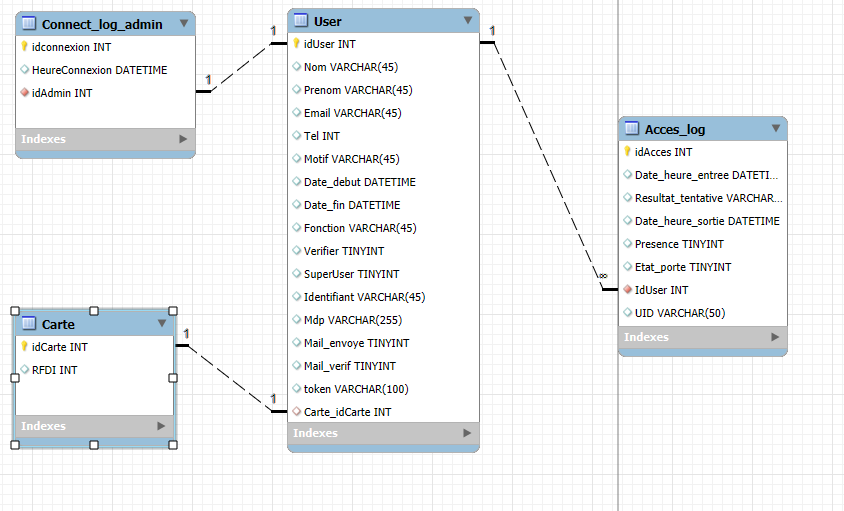
Nous pouvons définir et documenter les différents types de système qui sont utilisée pour le projet CA25.

Sur ce diagramme de déploiement vous pouvez y retrouver les matériaux utilisés, exemple (gâche, lecteur de carte etc..).

Mais nous pouvons voir aussi les différents langages utilisés à la réalisation de ce projet comme par exemple python.

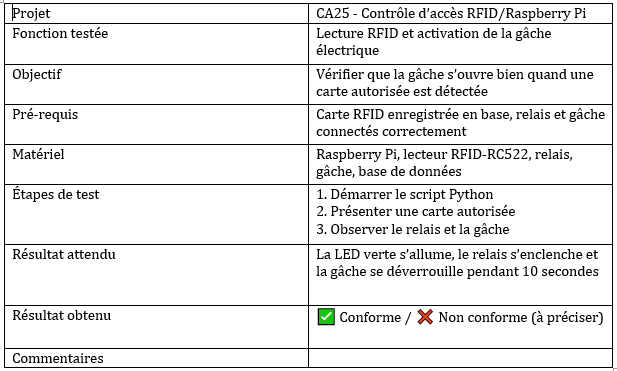
Enfin il y a les logiciels à utiliser, dans notre cas c’est AAPanel et MySQL.

**Base de données**

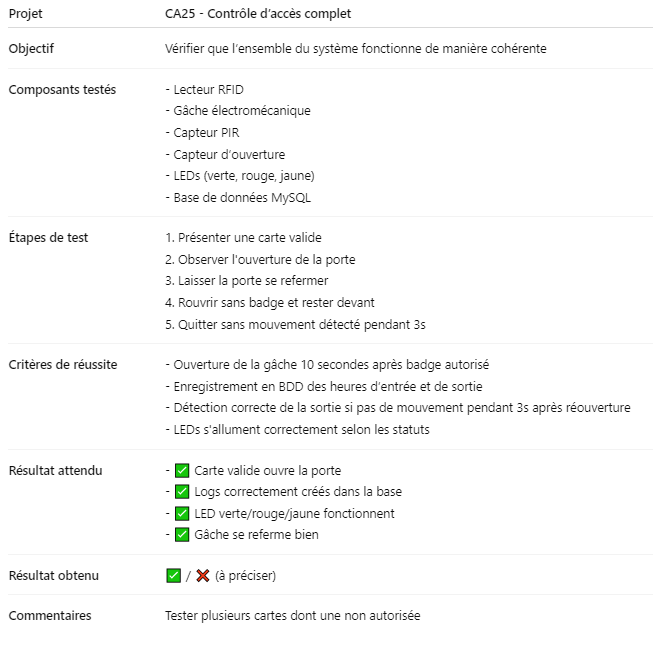
****

**Fiche Recette**

Fiche de recette unitaire

****

Fiche de recette d’intégration

****

**Mes taches**

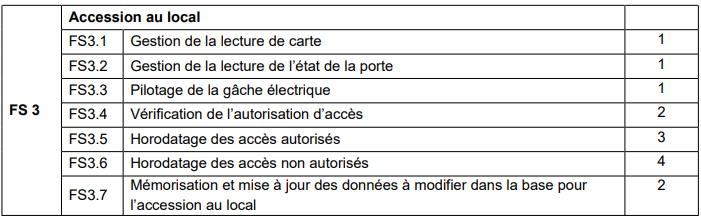
Pour la partie accéder au local

L’administrateur peut aussi voir l’historique des accès ou tentatives d’accès au local sur la période de son choix.

L’employé présente ça carte devant le lecteur de carte.

S’il est autorisé, la porte est déverrouillée, il peut ouvrir la porte et pénétrer dans le local et ressortir une fois son intervention terminée.

Le système horodate et mémorise l’accès de l’employé : horodatage, durée, N carte. Si l’accès n’est pas autorisé, le système horodate et mémorise la tentative d’accès avec le n de la carte utilisée.

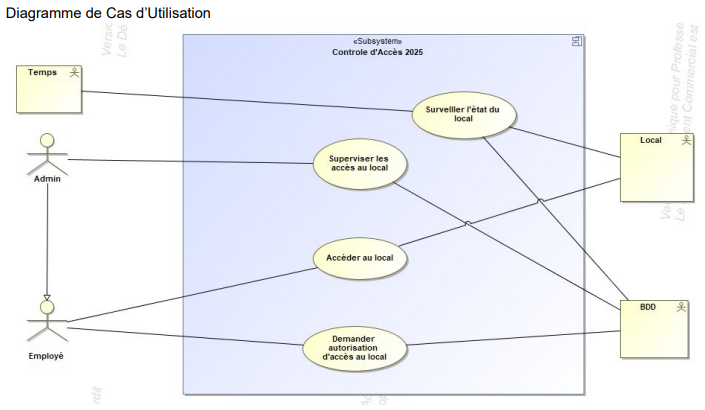


Les taches sont triées par ordre de priorité, des plus importantes aux moins importantes.

Horodatage = Lorsque l'on enregistre la date et l'heure d'un événement.

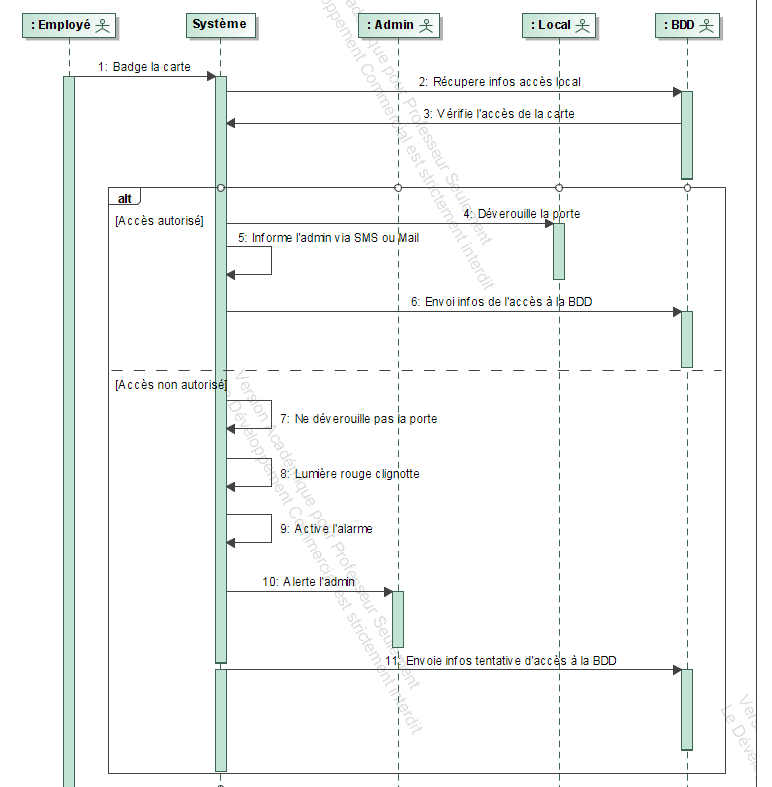
Dans notre cas je veux savoir le jour, l’heure et le résultat qu’a retourné le lecteur de carte suite à une carte badgée (carte autorisé ou carte non autorisé).

**Diagramme de cas d’utilisation**

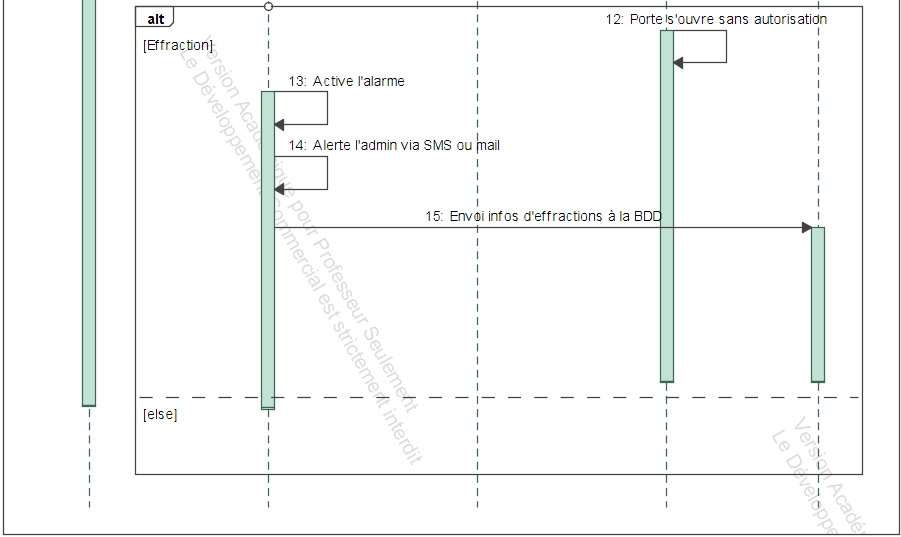


Voici la partie que je me suis occupée durant tout le projet

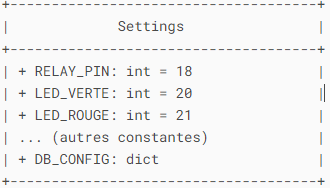
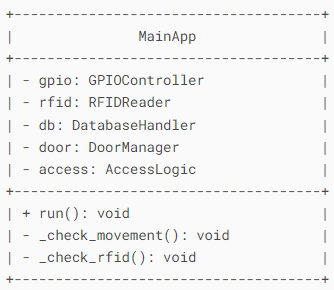
**Diagramme de séquence accès au local**

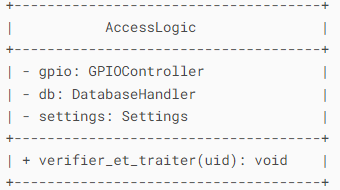


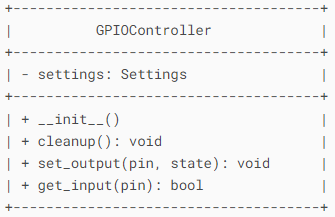
**Diagramme de séquence accès au local effraction**

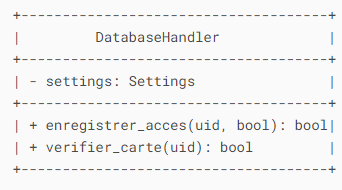


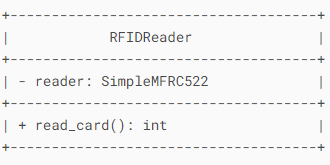
**Diagramme de classe**

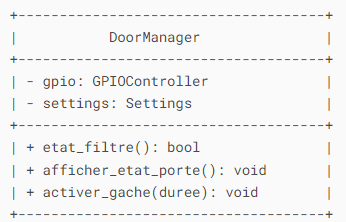
****



****







**Matériel utilisé**

Raspberry : nano-ordinateur pouvant se connecter à un moniteur.

Il a le rôle principal de notre projet car tous les programme seront dessus ainsi que tous les équipements utiles pour faire marcher la gâche ou qui sont demander pour réaliser le projet.



****

Breadboard ou platine de prototypage : son est de pouvoir faire des connexions simplement entre des composants.

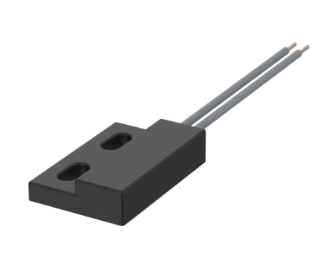
****

Gâche électromécanique : permet l'ouverture ou la fermeture de la porte quand une carte sera badgée.

****

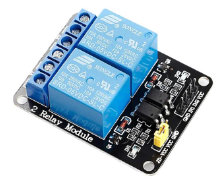
RFID : Radio Frequency Identification.

Nous allons utiliser ce lecteur pour accéder au local.



Détecteur d’ouverture de porte filaire :

Permet de savoir en direct l’état de la porte (ouvert ou fermée).



Relais mécanique :

Permet de distribuer la puissance et gère l'ouverture et la fermeture d'un circuit électrique



Capteur de présence :

Permet de mesurer et signaler une activité dans une zone de détection précise (notre local).



Carte NFC :

Nous allons utiliser ces cartes pour ouvrir le local et surtout pour remplacer la clé mécanique.



LED :

Elles ont comme rôle d’afficher l’état du local (rouge = fermé, vert = ouvert, jaune = occupé).

**Langage utilisé**

Pourquoi avoir choisi python ?

J’ai choisi le Python est préinstallé sur Raspbian (le système du Raspberry).

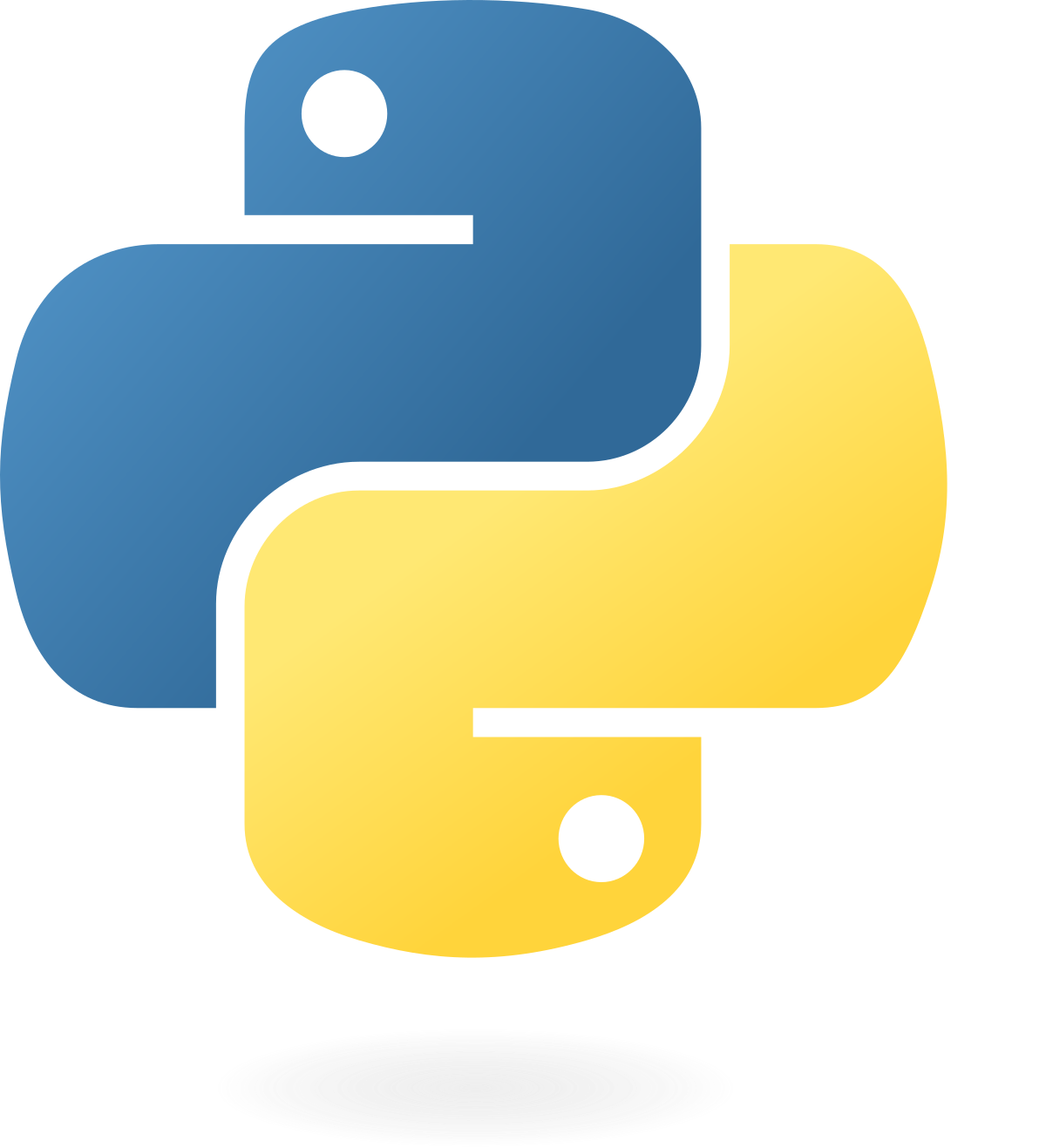
Il gère très bien les capteurs : RFID, PIR, relais, LEDs, etc.

Pas besoin de drivers compliqués ou d’interfaces bas niveau.

Il possède de nombreuse librairie puissantes et simple a utilisée, Tout ce qu’il faut est déjà disponible, documenté, et simple à intégrer.

Si tu veux sécuriser ton accès avec des rôles, des logs avancés, ou envoyer des emails Python le gère très bien.

Mais c’est aussi que j’ai codé en python de la première à la terminale et donc je me sentais plus à l’aise avec ce langage de programmation.



**Logiciel utilisé**

J’ai commencé par installer Visual studio sur le Raspberry code pour pouvoir coder en python.

L’inconvénient est que Visual studio code est beaucoup trop lourd pour un simple Raspberry pi 4.

Suite a ça l’écran que j’avais branché aux Raspberry pour coder ne marcher plus à cause de Visual Studio Code installé et j’ai dû le désinstaller.

Suite a ça il a fallu que je cherche un logiciel plus léger qui fonctionne avec le python.

Ducoup j’ai trouvé comme logiciel mu pour pouvoir codé en python.

Mu est un éditeur python simple pour les programmeurs débutant.

Grâce à Mu, vous pouvez écrire du code Python et Micro Python dans l'éditeur pour contrôler Halocode et d'autres modules électroniques.

Outre les fonctionnalités de base de Mu, cette version prend en charge davantage de fonctionnalités.

L’avantage de mu par rapport à Visual studio c’est qu’il est beaucoup plus léger à installer et il est très simple utiliser.

Car par rapport à Visual studio code c’est que nous n’avons pas beaucoup d’option dans cette interface.



**Raspberry**

Le Raspberry est le cœur de notre projet car tous les matériaux utilisés seront branchés sur lui plus précisément sur ces pins.

Tous les pins ne sont pas toute pareils.

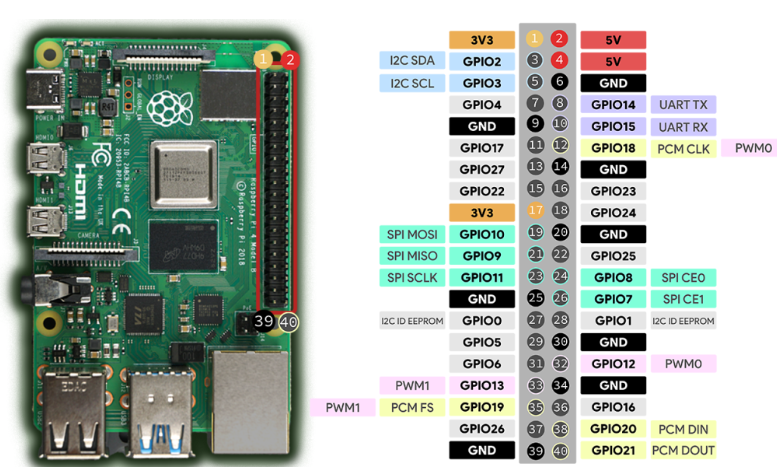
Certaine ont pour fonction d’alimenter avec du 5 volts (par exemple pour alimenter le ventilo sur le Raspberry).

D’autre son des GPIO : General Purpose Input/Output

Ce sont des ports d’entrée et de sortie.

Il y’a aussi des pins GND : Ground

Un chemin de retour commun ou partagé du courant électrique vers la source d'alimentation, permettant la fermeture du circuit, fait référence à la terre dans les circuits électriques et électroniques.

****

**Configuration des pins du Raspberry pour le projet**

LED jaune

Masse du détecteur d’ouverture de porte filaire

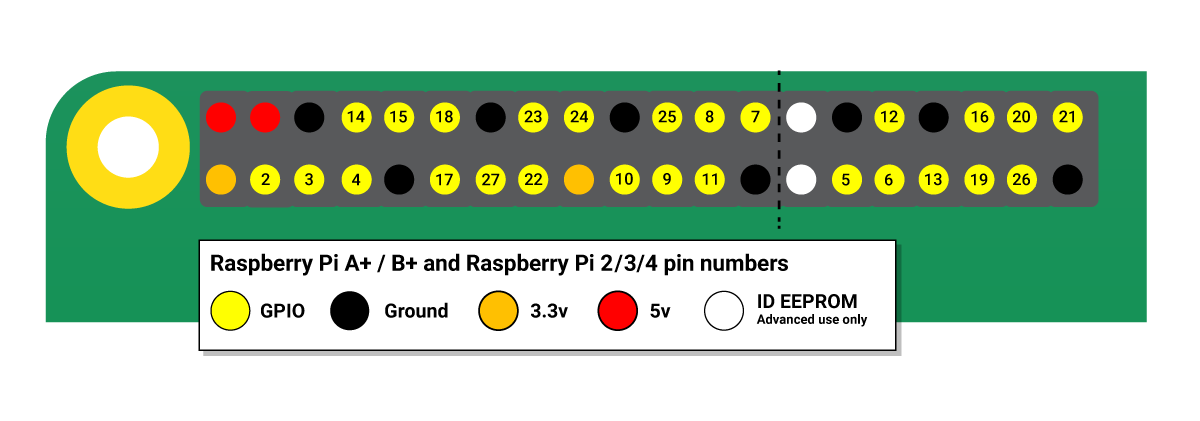
Relier au lecteur RFID a la pin SDA

Ground de la platine de prototypage

Alimentation 5 volts branché sur la platine de prototypage

Relier au lecteur RFID a la pin RST

LED verte



Ground du capteur de présence

Relier au relais numéro 1(IN1)

Relier au lecteur RFID a la pin 3.3 volts

Relier au lecteur RFID a la pin GND

Masse du capteur de présence

Relier au lecteur RFID a la pin SCK

Relier au capteur de présence

LED rouge

Relier au détecteur de porte filaire

Relier au lecteur RFID a la pin MISO

Relier au lecteur RFID a la pin MOSI

**Lecteur RFID**

Le lecteur RFID a pour objectif de lire la carte badger et d’envoyer les informations sur cette carte à la base de données (ID, si la carte est autorisée etc…).

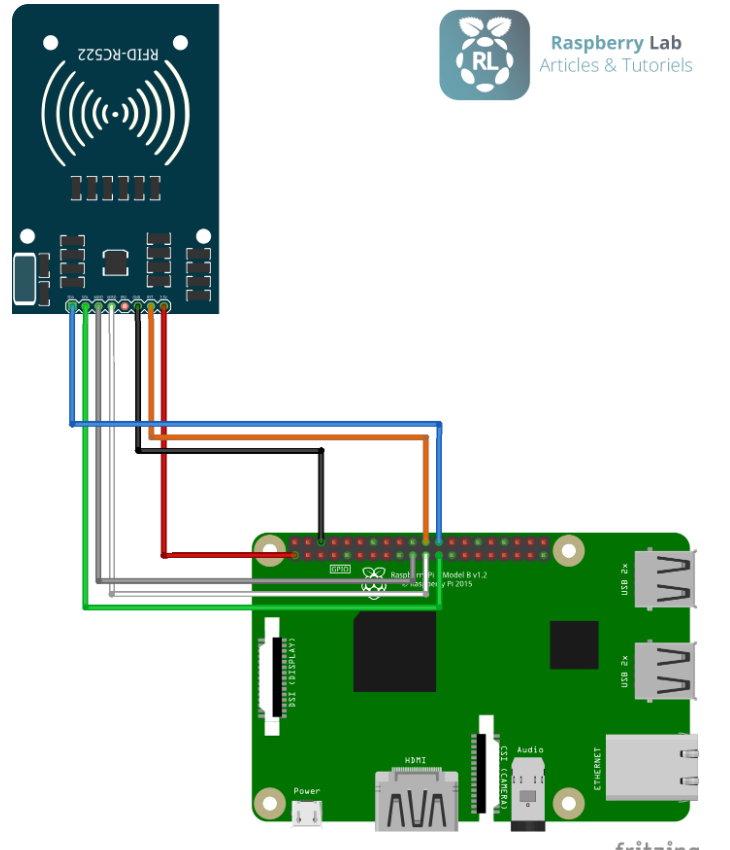
Dans l’idée la différence avec le lecteur NFC c’est que lui il doit juste dire s’il carte est autoriser ou pas.

Tan disque que le lecteur NFC s’occupe juste de donner l’accès a certaine carte badger.

Ce lecteur sera branché directement sur le Raspberry là où se situera le programme python qui permettra de l’activer et lire les informations sur la carte ou le badge à proximité de lui.

Dans le cadre notre projet nous avons l’obligation d’utiliser des carte NFC et pas de badge (car lorsque vous acheter un lecteur RFID ils vous fournissent un badge qui marche déjà avec le lecteur RFID).

Voici un schéma de comment se branche le lecteur RFID sur le Raspberry.



**Relais mécanique et LED**

Le relais va être relier au Raspberry et à la gâche.

C’est lui aussi qui va nous permettre d’allumer les LED qui sont brancher sur la platine de prototypage qui est alimenter par le Raspberry.

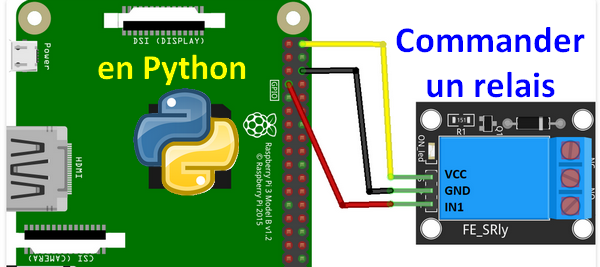
L’objectif est qu’au lancement du programme les LED soient alimentées mais qu’elle ne s’allume pas.

Il faut qu’elle s’allume que dans 3 possibilités (pas en même temp).

Première possibilité la porte est ouverte donc LED verte s’allume et les autres sont éteinte.

Deuxième possibilité la porte est fermée la LED rouge s’allume et les autres sont éteinte.

Troisième possibilité le capteur de présence détecte un mouvement la LED jaune s’allume et les autres sont éteinte.





**Gâche électromécanique**

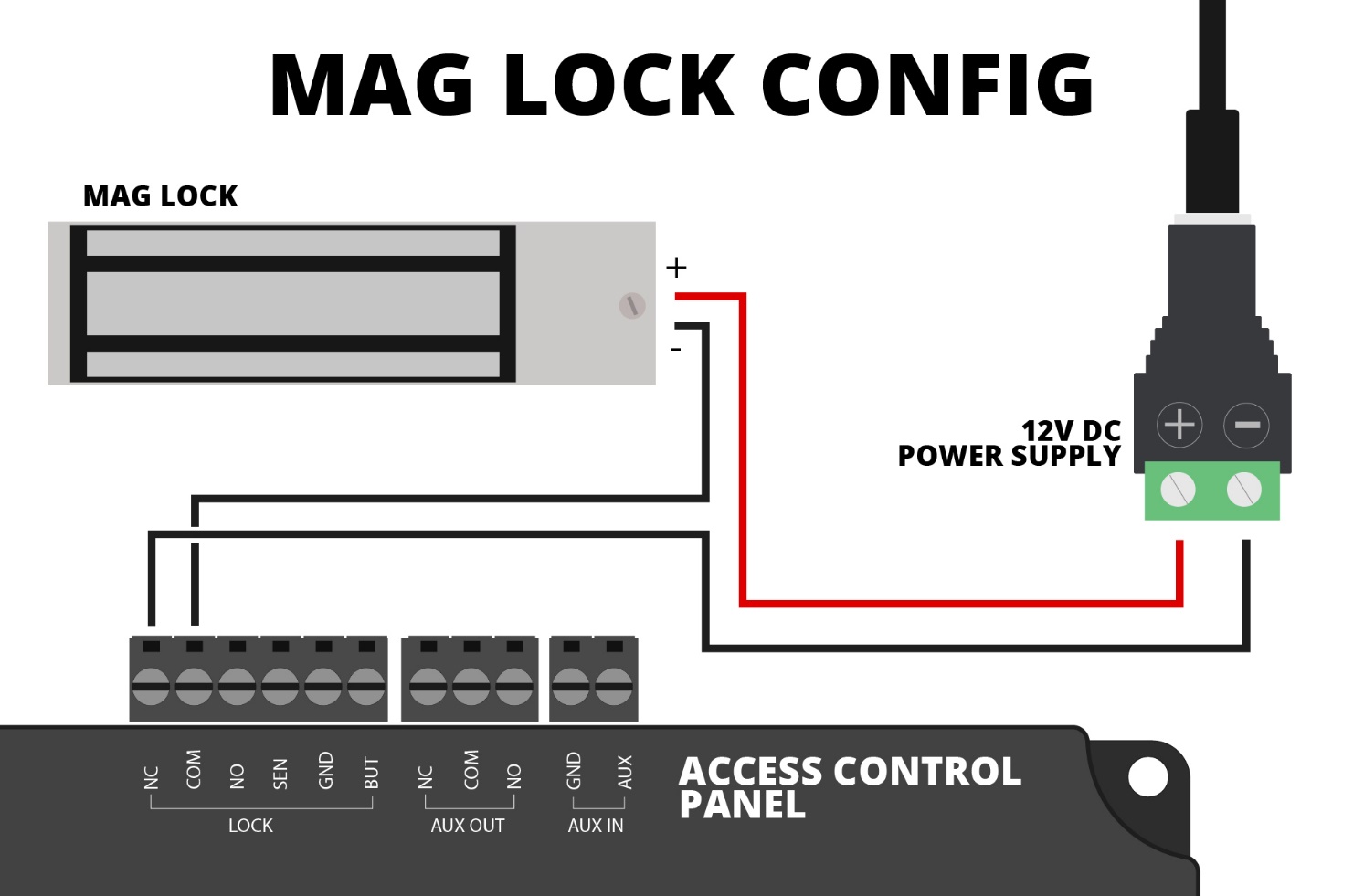
La gâche c’est ce qui va permettre l’ouverture ou la fermeture de la porte de notre local.

Elle sera alimentée grâce notre Raspberry et le relais sera brancher directement sur notre gâche pour la dirigée.

Attention si trop de courent passe ou que on la laisse alimenter sans interruption la gâche peut commencer a chauffé très rapidement et ne plus du tout marcher.

Comme solution il faut ajouter 2 résistances sur la platine de prototypage.

Le but de faire c’est de faire passer le minimum de courant pour que la gâche ne chauffe pas mais assez pour que la gâche marche.



**Détecteur d’ouverture de porte filaire**

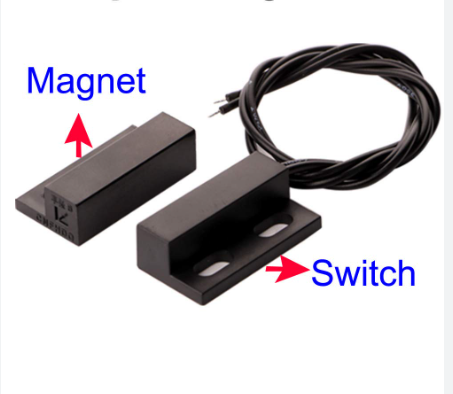
Un des détecteur (celui qui a un fils attacher à lui) sera installer sur un mur du local et le deuxième (celui qui n’a pas de fils) sera installer sur la porte.

L’objectif du détecteur de porte et d’informer l’administrateur en continue de la situation du local et de la porte.

Tout cela sera possible grâce à notre montage, surtout grâce au magnet qui sera coller coté porte et le switch qui sera installer coté mur du local.

Une fois coller si les 2 capteurs son assez proche ça nous affichera porte fermé et s’ils sont à une certaine distance ça nous affichera porte ouverte.

La distance sur laquelle on veut que ça nous affiche porte ouverte peut varier à cause de différent facteur (le programme qui le commande, la qualité de l’aimant ou la qualité de notre détecteur de porte filaire).



**Capteur de présence**

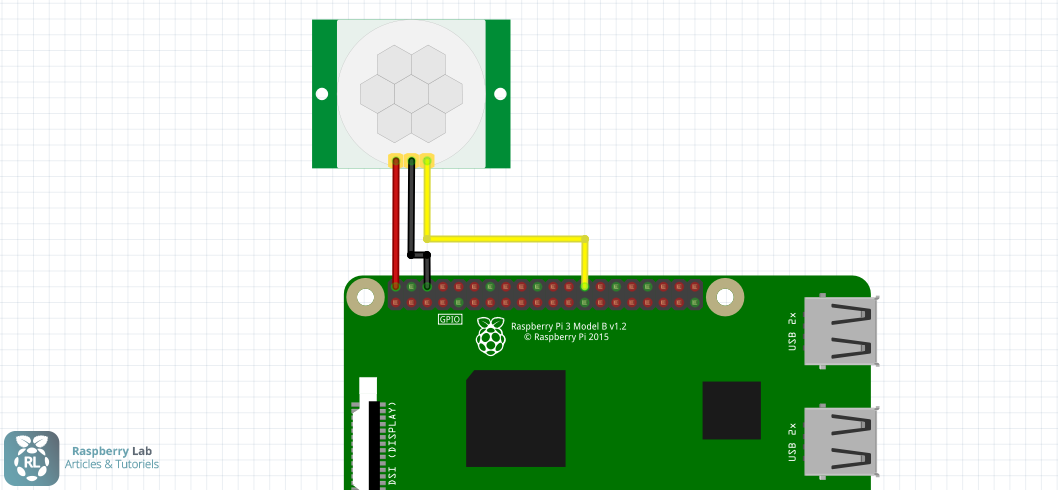
Le capteur de présence sera directement branché au Raspberry.

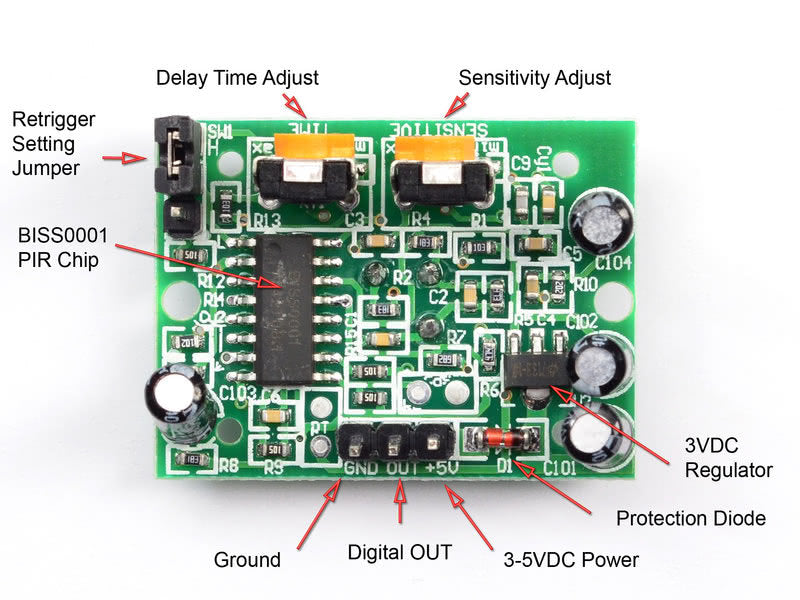
Son but sera d’avertir l’administrateur s’il y a quelqu’un dans le local au moment présent et à n’importe quel autre moment dans la journée.

Une fois un mouvement détecter ça enverra l’information a l’administrateur et en même temp notre LED jaune s’allumera.

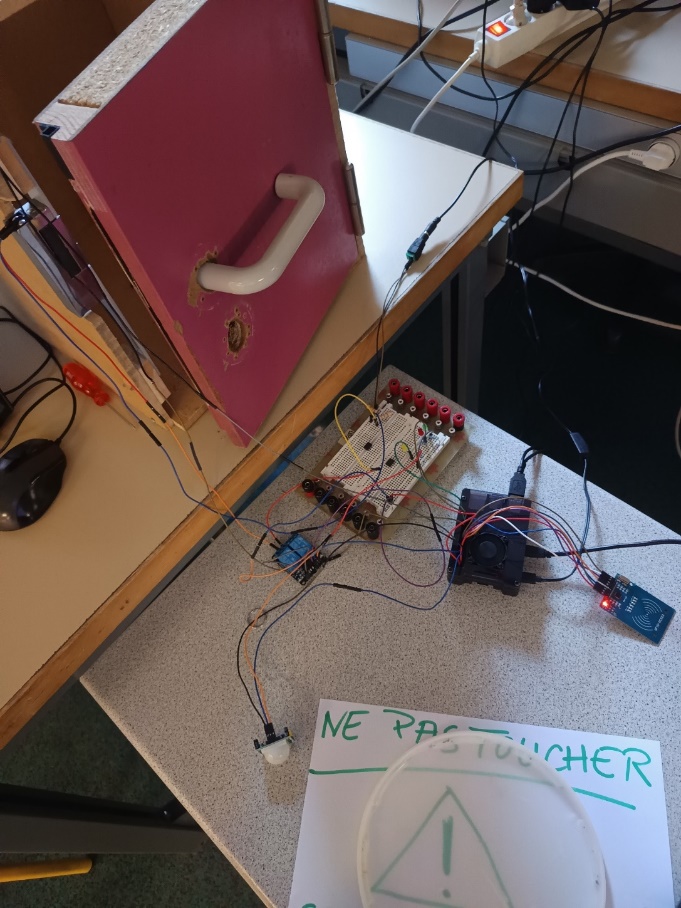
Il est possible de régler la sensibilité du capteur mais aussi le délai de temp.

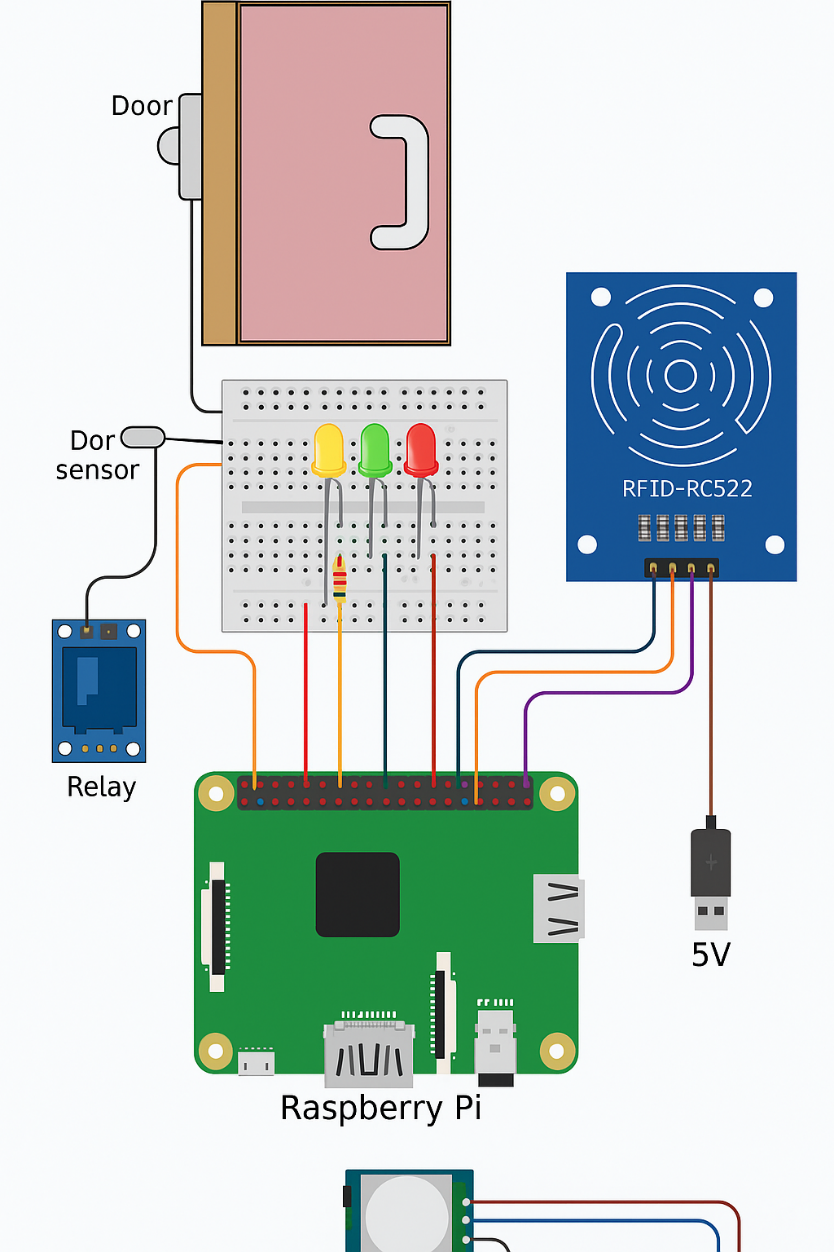
S’il ne détecte plus de mouvement dans une intervalle de temps continue (dans notre cas 10 seconde), il envoie l’information aux Dashboard de l’administrateur et au même moment la LED jaune s’éteint.





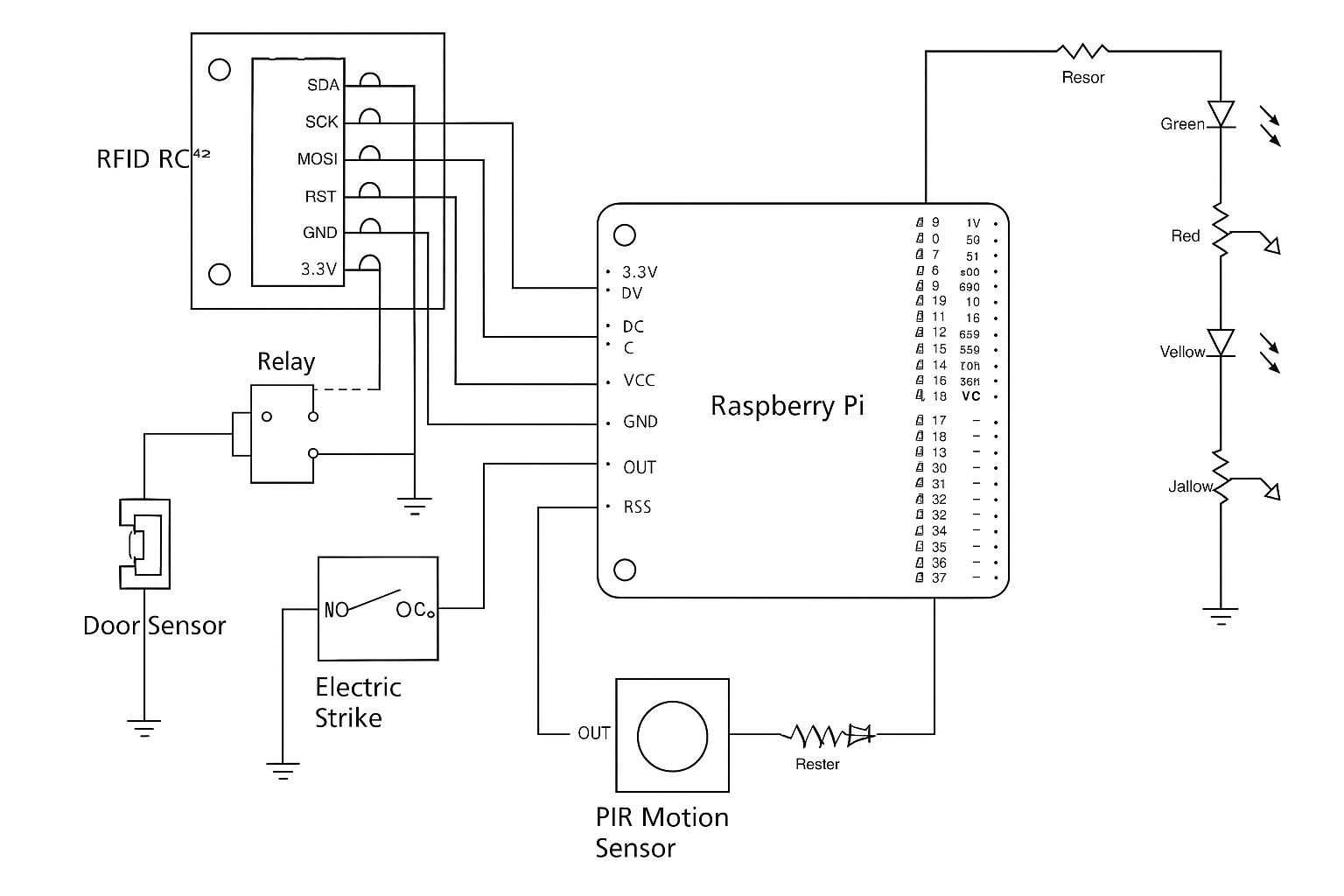
**Montage Global**

****

****

**Partie physique**

Voici un schéma électrique du montage global vu précédemment du projet CA25

****

**Conclusion**

Ce projet s’est révélé particulièrement enrichissant à plusieurs niveaux. J’ai trouvé le sujet très stimulant, en particulier la partie liée à l’accès au local, qui a représenté un véritable défi technique et organisationnel.

Travailler sur ce thème m’a permis de mieux comprendre les enjeux liés à la sécurisation des accès physiques à l’aide de dispositifs électroniques, tels que le lecteur RFID ou la gâche électromécanique.

Grâce à ce projet, j’ai pu consolider mes compétences en programmation, notamment en Python, en mettant en œuvre des solutions concrètes pour interagir avec des composants matériels.

J’ai également approfondi mes connaissances en schémas électriques, ce qui m’a permis de mieux appréhender l’interconnexion entre les différents éléments du système.

Au cours de cette réalisation, je me suis régulièrement confronté à des difficultés techniques et à des imprévus.

Chaque obstacle a été l’occasion de développer ma capacité d’analyse et de résolution de problèmes, en recherchant des solutions efficaces et adaptées aux contraintes du projet. Cette démarche a renforcé ma rigueur et ma persévérance dans le travail.

Par ailleurs, ce projet m’a offert une expérience de travail hybride : d’un côté, j’ai pu collaborer efficacement au sein d’une équipe, en partageant les tâches et en communiquant sur l’avancement des travaux ; de l’autre, j’ai su faire preuve d’autonomie dans la réalisation de certaines parties, ce qui m’a permis de développer ma capacité à gérer seul des missions techniques.

En conclusion, ce projet m’a permis d’acquérir de nouvelles compétences, tant techniques que méthodologiques, tout en m’initiant à une démarche de projet structurée et professionnelle.

Il a constitué une étape importante dans mon parcours de formation et m’a conforté dans mon intérêt pour les systèmes embarqués et les solutions d’automatisation.

**Amélioration sur le projet**

Comme amélioration suggérer pour ce projet il y’a la réalisation d’une nouvelle maquette.

En effet pour réaliser nos tests sur la gâche nous possédions une maquette mais il avait plusieurs problèmes.

Comme par exemple le fait qu’elle soit montée à l’envers (la porte doit s’ouvrir en la poussant une fois badger et tiré une fois que la personne veut quitter le local, et la maquette faisait l’inverse).

Nous pourrions aussi faire comme amélioration de remplacer le capteur de présence par une caméra.

Le problème du capteur du présence c’est qu’il informe l’administrateur de s’il y’a du mouvement dans le local mais nous ne savions pas ce qu’elle fait est c’est un gros problème.

Si quelqu’un réussi à rentrer dans le local avec des intentions malveillantes nous ne pourrions pas savoir ce qu’il fait et il aura tout le temps qu’il faut pour saboter.

Enfin comme autre amélioration suggérer c’est l’installation d’un 2 -ème lecteur RFID à l’intérieur du local.

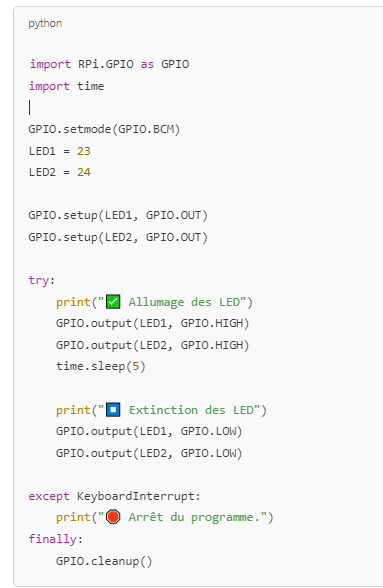
Car ça renforcerait la sécurité du local si quelqu’un a réussi à entrer par effraction ou s’il a trouvé un autre endroit pour entrer dans le local.

**Annexes**

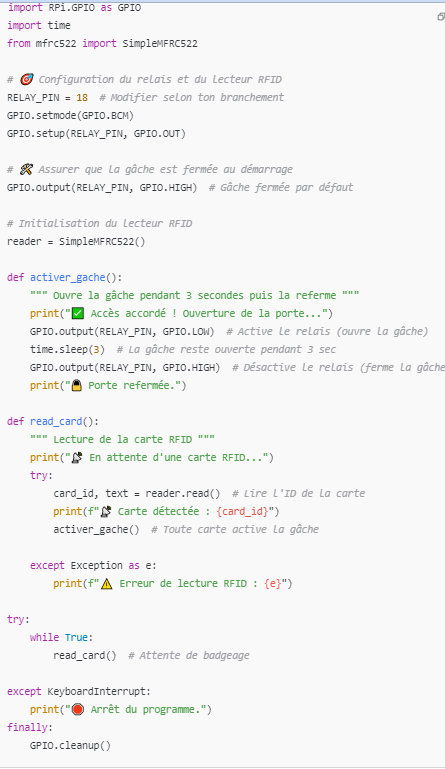
Lecteur RFID

****

**Relais mécanique et LED**



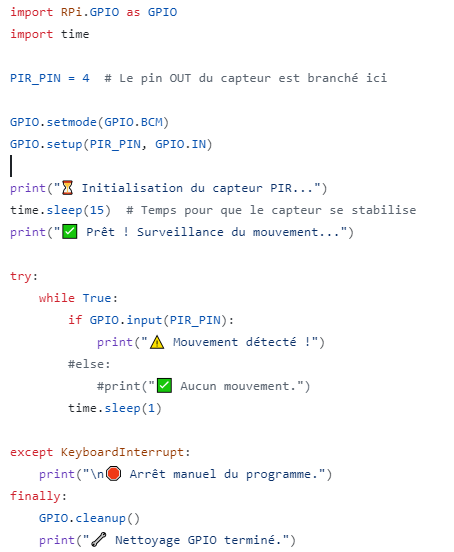
**Gâche électromécanique**



**Détecteur d’ouverture de porte filaire**

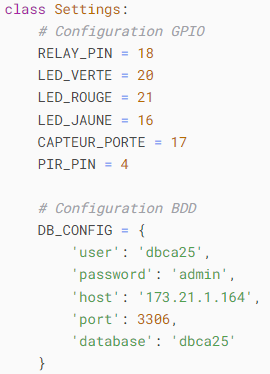


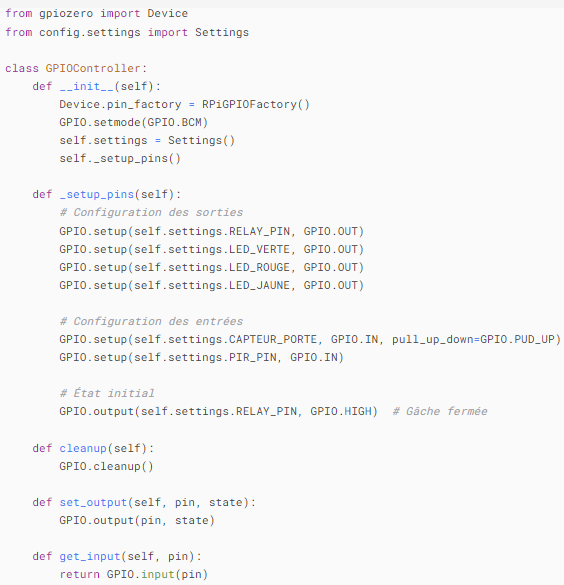
**Capteur de présence**



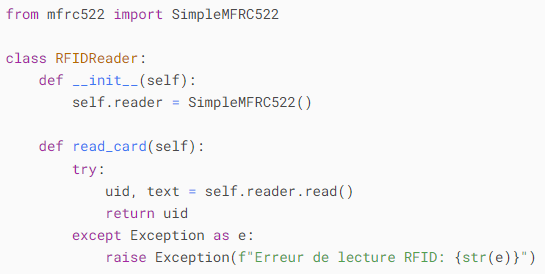
**Code complet du projet**

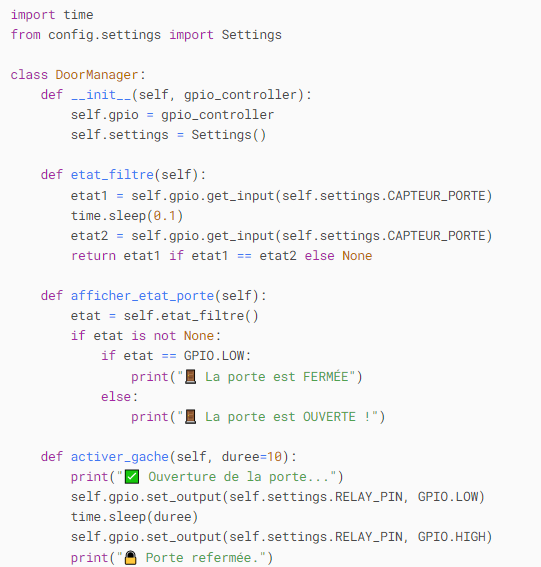
Classe paramètre

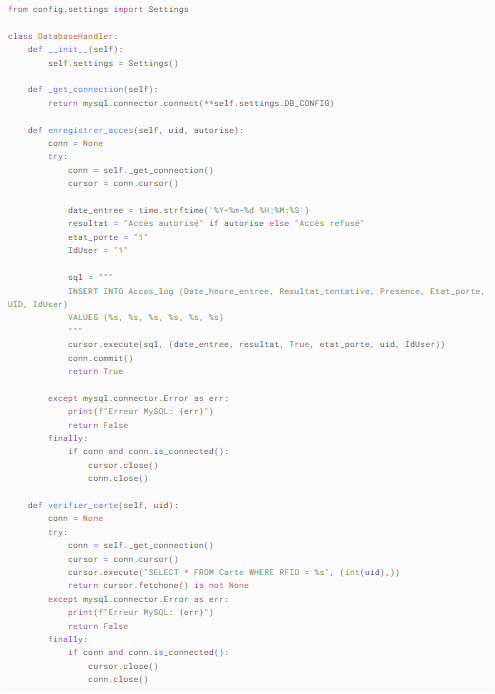


Classe contrôleur GPIO

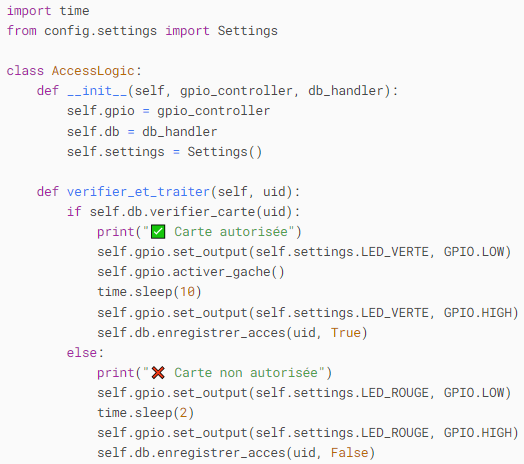
Classe lecture du RFID



Classe Gestion de la porte

Classe gestionnaire de la base de données

Classe accès



Le Main

