Compte Rendu de Projet :

Table des matières

[I. Introduction : 2](#_Toc40643065)

[II. Réalisation du projet : 2](#_Toc40643066)

[a) Rappel de la tâche de l’étudiant 2](#_Toc40643067)

[b) Communication entre les différents éléments 3](#_Toc40643068)

[c) Communication 4](#_Toc40643069)

[d) Câblage et choix des PIN 4](#_Toc40643070)

[e) Chronogrammes 5](#_Toc40643071)

[f) Système socket 6](#_Toc40643072)

[g) Code du client 7](#_Toc40643073)

[III. Réalisation de l’application : 7](#_Toc40643074)

[a) Récupération de l’UID 8](#_Toc40643075)

[b) Allumage des LEDs et déclanchement du module relais 8](#_Toc40643076)

[c) Bibliothèque RFID 10](#_Toc40643077)

[IV. Technologie RFID 12](#_Toc40643078)

[V. Fiches recettes 13](#_Toc40643079)

[a) Test des LED 13](#_Toc40643080)

[b) Ouverture de la porte et mettre fin à une partie 14](#_Toc40643081)

[c) Redémarrage à distance du sous-système médaillons 15](#_Toc40643082)

[VI. Conclusion 16](#_Toc40643083)

[a) Communication avec le groupe 16](#_Toc40643084)

[b) Difficultés liées au confinement dû au COVID-19 17](#_Toc40643085)

[c) Connaissances acquises 17](#_Toc40643086)

[d) Ce qui reste à faire 17](#_Toc40643087)

[e) Poursuite d’étude 17](#_Toc40643088)

[VII. Annexe 18](#_Toc40643089)

# Introduction :

#### **Rappel du cahier des charges :**

Afin que la partie se déroule correctement, un système pour les « *médaillons*» est à fournir.

Ce système sera au cœur même de la salle de l’escape game et devra prendre en charge les demandes suivantes :

* Le lecteur RFID doit pouvoir lire un UID (*User Identifier*) à travers une plaque de contreplaqué de 3mm.
* Utiliser le bus SPI (*Serial Peripheral Interface*) de la Raspberry pour le contrôle des lecteurs RFID.
* Le superviseur doit pouvoir récupérer UID (*User Identifier*) afin de le changer facilement pour le programme principal dans le cas d’une perte de l’un des médaillons.
* Le joueur doit pouvoir constater l’allumage ou non d’une LED dans la seconde en fonction de l’état d’un lecteur RFID (si un badge ou « *médaillon*» valide est posé dessus ou non).
* Dans le cas où le joueur place correctement quatre badges, la gâche électrique doit s’activer, permettant l’ouverture de la porte. Un ordre de fin de partie doit également être envoyer via un système de client socket.
* L’application de supervisions doit pouvoir détecter l’ordre de fin de partie via un système serveur socket.

# Réalisation du projet :

## Rappel de la tâche de l’étudiant

Au sein de ce projet d’escape game, ma partie consistait à créer un système permettant :

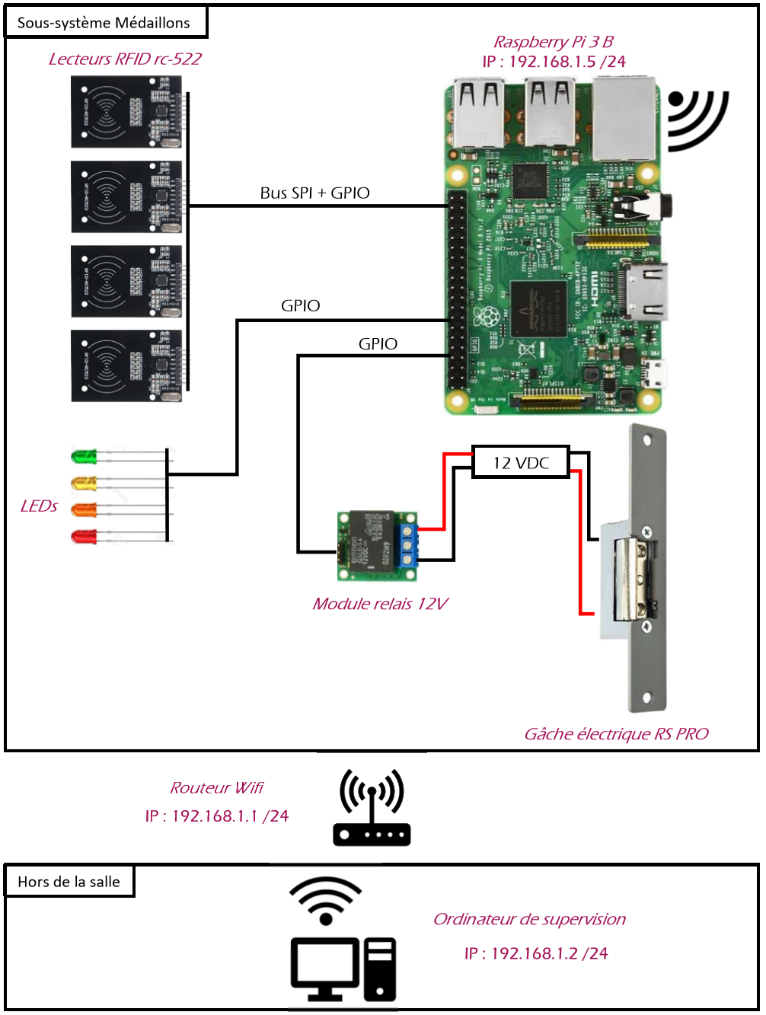
* + La détections des « *médaillons* ».
  + L’allumage des différentes LED.
  + L’ouverture de la porte.
  + La communication entre système « *médaillons*» et l’application de supervision

Le langage python est utilisé avec différentes librairies, comme « *pi-RC522* » qui permet la gestion des lecteurs RFID, et « *socket*» qui permet l’utilisation des services socket. Concernant l’IDE, j’ai opté pour le logiciel « *Thonny* », puisqu’il est très simple d’utilisation.

Le système doit être développé sur une carte Raspberry sous Raspbian. Le contrôle du superviseur de cette dernière sera possible via l’application « *Anydesk* » qui est compatible avec les deux environnements, Raspbian et Windows. Elle permet le contrôle à distance via un protocole TCP.

## Communication entre les différents éléments

***Synoptique***



L’ordinateur de supervision, le routeur wifi ainsi que Raspberry sont adressés en IP statiques.

## Communication

Le bus SPI (*Serial Peripheral Interface*) est un bus de données série synchrone en full-duplex. Il fonctionne sous un schéma maître-esclave. Dans ce cas le maître est la Raspberry et les esclaves sont les lecteurs RFID RC522.

Ce modèle possède 8 interfaces dont 4 qui sont dédiés au bus SPI :

* SCK – Consacré à l’horloge et généré par le maître.
* MOSI – Master Output, Slave Input (généré par le maître).
* MISO – Master Input, Slave Output (généré par l’esclave).
* NSS – La sélection de l’esclave (généré par l’esclave).

Les 4 interfaces restantes sont :

* VCC – Alimente le module avec un voltage de 3,3 Volt.
* GND – La terre.
* RST – Permet la réinitialisation et la mise hors tension du module.
* IRQ – Permet une interruption, cela alerte le module quand un tag RFID se trouve à proximité

## Câblage et choix des PIN

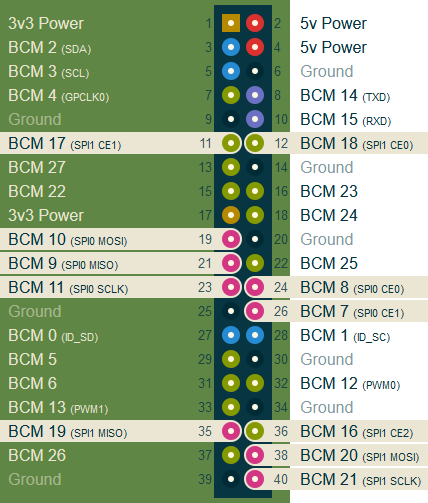
Cette partie du projet demande un câblage plus ou moins complexe, une recherche et des choix ont donc été nécessaires.

Le modèle du lecteur RFID RC522 a besoin de 8 PIN pour pouvoir fonctionner. Le bus spi nécessite 4 d’entre eux : le MOSI, le MISO, le SCK (qui peuvent être communs pour tous les lecteurs) et enfin, le NSS (CE) (qui est unique pour chacun d’entre eux). Ces PIN ont des emplacements dédiés sur la Raspberry. Les 4 interfaces restantes peuvent être communes pour tous les lecteurs, dont une alimentation et une terre.

Pour les LEDs, un port GPIO (*General Purpose Input/Output*) est nécessaire pour chacune d’entre elles, ainsi qu’une résistance de 90Ω du voltage des ports GPIO de 3,3V. Elles ont cependant une terre commune.

Le module relais a besoin quant à lui d’un port GPIO pour le contrôler, un port d’alimentation de 5V et enfin une terre.

Un schéma préliminaire du câblage est disponible en annexe.



GPIO

SPI

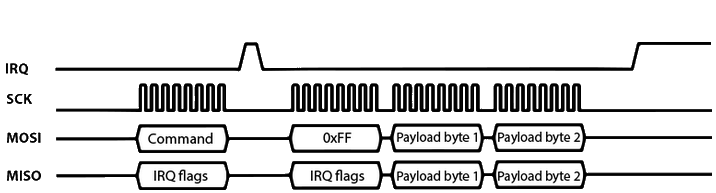
Terre

5V

3,3V

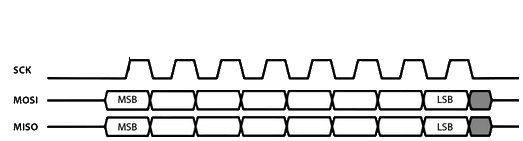
## Chronogrammes

Dans le premier chronogramme, nous voyons que lorsqu’un badge passe à proximité, l’IRQ signale le passage d’un tag RFID. La première transaction qui passe avant le signal de l’IRQ, c’est l’octet (8 bites) de commande. Dans la seconde transaction, 24 bits sont transmis. Les 16 derniers bits transmis sont l’UID (User Identifier) qui est propre à chaque taf RFID.



Chronogramme 1

Dans le second chronogramme on voit comment se passe la transaction d’un octet, le bite de poids fort (MSB) est transmis en premier.



Chronogramme 2

## Système socket

Pour envoyer l’ordre de fin de partie, un système de socket a été choisi avec comme « *client* » pour l’application du sous-système médaillons et un serveur pour l’application de supervision. Ainsi, le serveur sera en « *écoute* » en attendant que le client envoie un message qui confirmera la réussite de l’équipe qui participera à la session de jeu de l’escape game.

Pour ce faire les informations suivantes sont requises :

Pour le client :

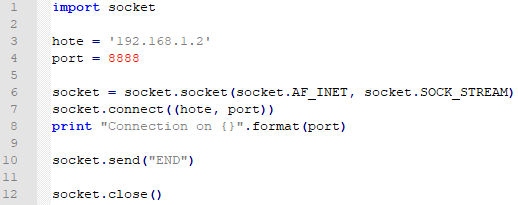
* Adresse IP : **192.168.1.2** (adresse IP l’ordinateur de supervision)
* Port : **8888**

Pour le serveur :

* Adresse IP : **192.168.1.5** (adresse IP de la Raspberry)
* Port : **8888**

## Code du client

Afin que l’application puisse envoyer l’ordre de fin de partie par un simple message « END », il faut utiliser le code suivant en python :



La méthode *import*, ligne 1, permet d’accéder à différents modules. Dans ce cas, il faut importer le module socket.

Par la suite, il est nécessaire de procéder comme suit :

* Définir l’hôte et le port de communication ;
* Ensuite, créer le socket avec la méthode *socket(socket.AF\_INET, socket.SOCK\_STREAM)*, ligne 6 ;
* Puis, envoyer une requête de connexion au serveur avec la méthode *socket.connect()*, ligne 7 ;
* Si la connexion n’est pas rejetée, envoyer le message de fin avec la méthode *socket.send()*, ligne 10 ;
* Enfin, fermer le socket avec *socket.close()*, ligne 12.

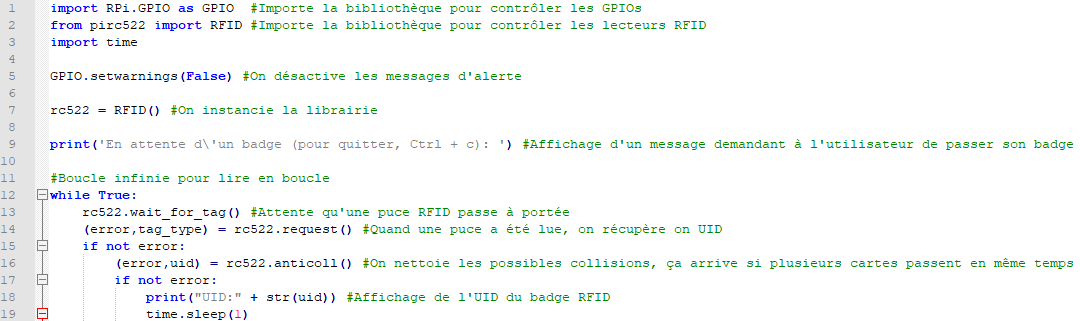
# Réalisation de l’application :

Afin que la partie se déroule, il est nécessaire de fournir une application qui permettra de détecter les médaillons, allumer les LEDs, déclencher le module relais et enfin envoyer l’ordre de fin partie si les médaillons sont bien placés.

## Récupération de l’UID

En premier lieu, pour le superviseur, il faut pouvoir récupérer l’UID d’un badge RFID dans le cas d’une perte et/ou vol.

Pour ce faire, le code suivant est utilisé :

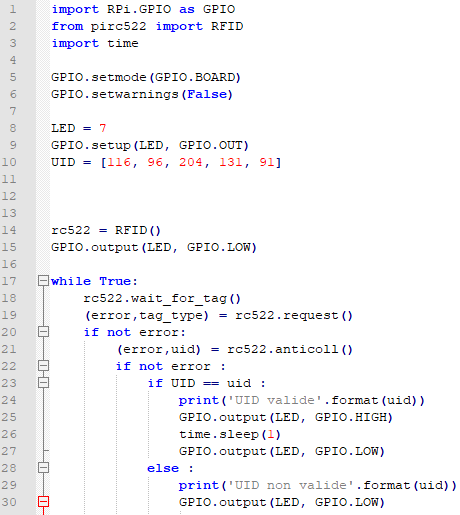


Dans un premier temps, importer le module RFID depuis la librairie « pi-rc522 », et instancier la librairie avec *rc522 = RFID()*, ligne 2.

Ensuite, il faut créer une boucle infinie pour lire en continu. Dans cette boucle, on utilise la fonction *wait\_for\_tag()*, ligne 13, pour attendre qu’un badge RFID passe à portée de lecture. Lorsqu’un badge passe à proximité, son UID est récupérée avec *request()*, ligne 14. Les possibles collisions sont nettoyées avec *anticoll()*, ligne 16 ; ce qui arrive si plusieurs badges passent à portée en même temps. Ensuite, nous affichons l’UID unique du badge avec *str()*, ligne 18. Pour la fin, il faut attendre 1 pour ne pas lire le badge des centaines de fois en quelques millisecondes avec *sleep(1)*, ligne 19. 1 étant 1 seconde.

## Allumage des LEDs et déclanchement du module relais

L’allumage de la LED doit pouvoir se faire dans la seconde en fonction de l’état du capteur. C’est-à-dire, si un badge RFID est à proximité ou non. Pour cela, le code suivant est utilisé :



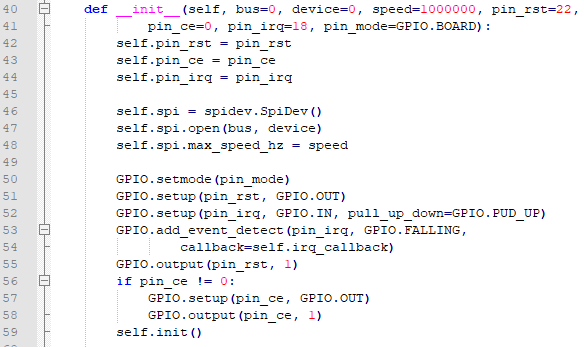
Une fois qu’un UID a été récupéré, ligne 10, il peut être utilisé pour faire un comparatif avec l’UID récupéré par le lecteur RFID. S’il est valide, nous pouvons allumer la LED. Pour ce faire, il faut définir le numéro du port GPIO qui va contrôler la LED (ici port n°7) et activer le contrôle du GPIO avec *setup(LED,GPIO.OUT)*, ligne 9. Puis, nous initialisons ce port à l’état bas (LED éteinte) avec *output(LED,GPIO.LOW)*, ligne 15. Ensuite, il faut reprendre la boucle d’écoute décrite précédemment et faire la comparaison des deux UID. Si l’UID est valide, on allume la LED avec *output(LED,GPIO.HIGH)*, ligne 25. Il faut ensuite attendre 1 seconde pour éteindre la LED. Si le badge reste à portée du lecteur RFID, la LED restera allumée car la boucle est infinie.

Pour le déclenchement du module relais, il nous faudra réaliser sensiblement la même démarche. Seul le temps d’attente sera différent pour permettre aux joueurs d’avoir un délai suffisant pour sortir de la salle.

## Bibliothèque RFID

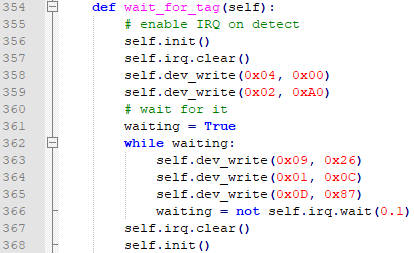
***Constructeur***

La bibliothèque principale utilisée pour le contrôle des lecteurs RFID est « pi-rc522 ». Cette libraire est basée sur une autre librairie, la « MFRC522 » qui n’est malheureusement plus à jour.



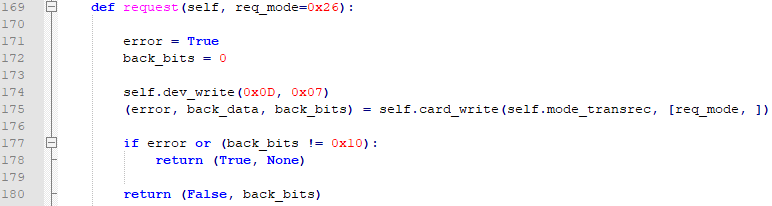
Sur la ligne 40, créer notre constructeur avec *\_\_init\_\_*, *self* en premier argument, permet d’avoir des méthodes non statiques. Nous pouvons donc les changer pour l’adapter à notre câblage. Le *bus=0* définit, si on utilise, le bus SPI 1 ou 2, puisqu’un Raspberry peut en gérer deux. Le *device=0* définit le pin de contrôle du lecteur RFID (NSS). Grâce à ce constructeur, nous pouvons choisir quel lecteur nous voulons contrôler.

***Fonction wait\_for\_tag***



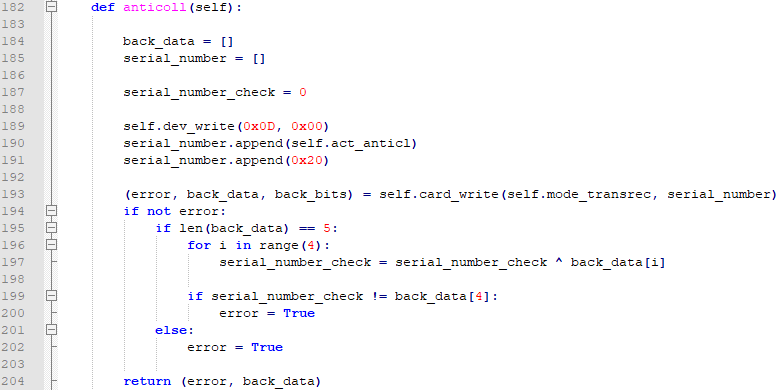
Cette fonction créée avec *def* permet l’utilisation de l’IRQ, qui permet à son tour de détecter si un tag RFID passe à proximité.

***Fonction request***



La fonction permet de récupérer l’UID du tag RFID qui est à proximité du lecteur.

***Fonction anticoll***



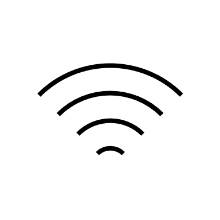
Cette fonction est nécessaire pour éviter les collisions entre les données récupérées dans le cas où 2 tags RFID sont à proximité de lecture. Après quelques tests, cette distance est d’environ 5cm. Pour différencier les 2 tags, le numéro de série est récupéré et peut ainsi être comparé.

# Technologie RFID

La technologie RFID (Radio Frequency Identification) permet de récupérer des données à distance de différents « *tag RFID* » en utilisant un champ magnétique.

Pour cela le principe suivant est utilisé : le système de radio-identification est composé de deux parties qui communiquent l’une avec l’autre. L’échange des données débute par un transfert d’énergie électromagnétique entre le tag et l’émetteur ou lecteur RFID.

Signal





Tag RFID

Energie

Lecteur RFID

Signal

Le modèle RC522 est un lecteur dit à haute fréquence. Il fonctionne avec une longueur d’onde de 13,56 MHz, qui a un déploiement difficile à travers les environnements métalliques.

Le tag est dit « *passif* » de classe 2. Il permet la lecture et l’écriture de la mémoire, cette mémoire correspond à l’UID unique composé entre 7 à 14 caractères. Il répond à la norme ISO 14443.

Cette norme décrit plusieurs couches, 4 au total. La première couche a pour dénomination «*Physical Layer* » : elle décrit la partie physique du tag. La deuxième couche, « *Radio frequence signal interface*», traite de la fréquence à laquelle l’émetteur et le récepteur doivent fonctionner (ici 13,56MHz), ainsi que le débit de transfert (ici 25kb/s). La troisième couche, « *Activation and anti-collision*» décrit le principe d’anti-collisions qui permet de gérer le cas de plusieurs récepteurs qui se trouveraient en même temps dans le champ magnétique, d’où la fonction *anticoll()* décrite précédemment. La quatrième et dernière couche est appelé « *Transmission protocol* ». Elle permet d’y décrire le format des données, la communication durant une transaction.

# Fiches recettes

Dans cette partie, deux fiches recettes sont destinées au client. Elles permettent de valider le fonctionnement du système.

## Test des LEDs

|  |  |
| --- | --- |
| **Nom** : Lancer le sous-système médaillons | |
| **Recette** : Technique | |
| **Objectif** | Savoir si toutes les LED fonctionnent. |
| **Elément à tester** | LED |
| **Pré-requis** | Avoir branché l’alimentation de la Raspberry et avoir attendu quelques secondes pour que le système démarre. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Scénario** | | | | |
| **ID** | **Démarche** | **Données** | **Comportement attendu** | **OK** |
| **1** | Placer le tag RFID n°1 sur le lecteur RFID n°1. |  | La LED n°1 doit s’allumer dans la seconde où le tag est placé. | ❑ |
| **2** | Retirer le tag RFID n°1 sur le lecteur RFID n°1. |  | La LED n°1 doit s’éteindre dans la seconde où le tag est enlevé. | ❑ |
| **3** | Placer le tag RFID n°2 sur le lecteur RFID n°2. |  | La LED n°2 doit s’allumer dans la seconde où le tag est placé. | ❑ |
| **4** | Retirer le tag RFID n°2 sur le lecteur RFID n°2. |  | La LED n°2 doit s’éteindre dans la seconde où le tag est enlevé. | ❑ |
| **5** | Placer le tag RFID n°3 sur le lecteur RFID n°3. |  | La LED n°3 doit s’allumer dans la seconde où le tag est placé. | ❑ |
| **6** | Retirer le tag RFID n°3 sur le lecteur RFID n°3. |  | La LED n°3 doit s’éteindre dans la seconde où le tag est enlevé. | ❑ |
| **7** | Placer le tag RFID n°4 sur le lecteur RFID n°4. |  | La LED n°4 doit s’allumer dans la seconde où le tag est placé. | ❑ |
| **8** | Retiré le tag RFID n°4 sur le lecteur RFID n°4. |  | La LED n°4 doit s’éteindre dans la seconde où le tag est enlevé. | ❑ |

|  |  |  |  |
| --- | --- | --- | --- |
| **Rapport de test** | **Testé par : Etudiant 1** | | **Le : 17/05/2020** |
| Conformité  ❑Excellente  ❑Moyenne  ❑Faible | | | |
| Commentaire : | | Approbation : | |

## Ouverture de la porte et mettre fin à une partie

|  |  |
| --- | --- |
| **Nom** : Ouverture de la porte et fin de partie | |
| **Recette** : Technique | |
| **Objectif** | Savoir si toute les LED fonctionnent. |
| **Elément à tester** | Gâche électrique et déclenchement de fin de partie. |
| **Pré-requis** | Avoir branché l’alimentation de la Raspberry et avoir attendu quelques secondes pour que le système démarre. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Scénario** | | | | |
| **ID** | **Démarche** | **Données** | **Comportement attendu** | **OK** |
| 1 | Placer le tag RFID n°1 sur le lecteur RFID n°1. |  | La LED n°1 doit s’allumer dans la seconde où le tag est placé. | ❑ |
| 2 | Placer le tag RFID n°2 sur le lecteur RFID n°2. |  | La LED n°2 doit s’allumer dans la seconde où le tag est placé. | ❑ |
| 3 | Placer le tag RFID n°3 sur le lecteur RFID n°3. |  | La LED n°3 doit s’allumer dans la seconde où le tag est placé. | ❑ |
| 4 | Placer le tag RFID n°4 sur le lecteur RFID n°4. |  | La LED n°4 doit s’allumer dans la seconde où le tag est placé. Le déclenchement de la gâche doit se déclencher pour une durée de 10 secondes. Dans l’application de supervision, la partie lancée doit se terminer. | ❑ |

|  |  |  |  |
| --- | --- | --- | --- |
| **Rapport de test** | **Testé par : Etudiant 1** | | **Le : 17/05/2020** |
| Conformité  ❑Excellente  ❑Moyenne  ❑Faible | | | |
| Commentaire : | | Approbation : | |

## Redémarrage à distance du sous-système médaillons

|  |  |
| --- | --- |
| **Nom** : Redémarrage du sous-système médaillons | |
| **Recette** : Technique | |
| **Objectif** | Redémarrer pour palier à un bug de non-détection des tags RFID |
| **Elément à tester** | Détection des tags RFID |
| **Pré-requis** | Raspberry allumé et tag RFID non-détecté |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Scénario** | | | | |
| **ID** | **Démarche** | **Données** | **Comportement attendu** | **OK** |
| 1 | Démarrer « *Anydesk* » sur l’ordinateur de supervision. |  | *Anydesk* démarre avec une seule possibilité de connexion. | ❑ |
| 2 | Sélectionner cette possibilité et entrer le mot de passe quand il est demandé. | Mot de passe : roottel44 | La connexion s’établit et le bureau de la Raspberry apparaît. | ❑ |
| 3 | Cliquer sur la « Framboise » en haut à gauche de l’écran |  | Plusieurs choix apparaissent. | ❑ |
| 4 | Cliquer sur « *Shutdown…* » |  | Une fenêtre apparait avec 3 choix possibles. | ❑ |
| 5 | Cliquer sur « *Reboot* » |  | Une fenêtre apparaît disant « *Connexion fermée* » | ❑ |
| 6 | Attendre quelques secondes et tester l’allumage d’une LED. |  | La LED s’allume. | ❑ |

|  |  |  |  |
| --- | --- | --- | --- |
| **Rapport de test** | **Testé par : Etudiant 1** | | **Le : 17/05/2020** |
| Conformité  ❑Excellente  ❑Moyenne  ❑Faible | | | |
| Commentaire : | | Approbation : | |

# Conclusion

## Communication avec le groupe

Dès le début du projet, nous avons mis en place différents moyens pour garder une cohésion de groupe. Cette cohésion était d’autant plus importante pendant la période de confinement. Pour cela, plusieurs outils et méthodes ont été utilisés :

* Un diagramme de Gantt : cela nous a permis de bien définir nos tâches avec une estimation de leurs durées ainsi que de les répartir correctement entre nous et dans le bon ordre, au début du projet.
* La plateforme GitHub et le logiciel GitHub Desktop ou GitKraken : cela nous a permis d’échanger nos documents et programmes ainsi que de voir l’avancement de chacun tout au long du projet.
* Nous avons créé en début de projet une charte graphique pour nos documents Word et PowerPoint.
* En début de projet, j’ai eu l’idée de créer un journal d’activités avec Excel. Cela nous a permis de voir ce que chaque personne faisait chaque jour ainsi que le temps qu’il y avait consacré avec un total du nombre d’heures faites sur le projet.
* En début de confinement, nous avons créé un groupe Teams. Ainsi, nous avons pu faire nos réunions de groupe dans de bonnes conditions.
* Nous avons également utilisé la suite Office 365, que nous fournit notre établissement, principalement OneDrive, qui permet la rédaction en simultané de différents documents.
* Nous avons établi des réunions de groupe toutes les semaines avec Mr Angibaud, afin de savoir ou en était rendu le projet.
* Par soucis de praticité, nous avons utilisé différentes messageries instantanées comme celle de Teams ou Messenger.

Toutes ces méthodes m’ont permis de visualiser de manière plus détaillée le travail en équipe et l’organisation qui en découlait.

Je tiens à remercier mes collègues, Nathan Guigand ainsi que Cyril Dohin pour leur participation et implication dans le projet.

## Difficultés liées au confinement dû au COVID-19

A partir de Mars 2020 nous avons été confiné. Cela s’est produit au début de la phase de développement du projet. Nous avons donc dû récupérer tous les matériels nécessaires chez nous afin de poursuivre notre travail.

Nous avons donc dû travailler dans un environnement moins propice au travail, ainsi qu’un manque de matériel (jumper). J’ai fait une commande de ceux-ci par internet. L’attente pour la livraison a été grandement augmentée. Il a fallu attendre plus de deux semaines pour faire réception du colis et enfin faire des tests unitaires.

## Connaissances acquises

J’ai pu découvrir le langage python car nous n’avions jusque-là jamais codé avec ce langage.

J’ai également pu développer dans un environnement que je ne connaissais pas encore très bien. Raspbian avec une Raspberry permet de réaliser toute sorte de projets que je réutiliserais certainement dans l’avenir pour mon usage personnel.

## Ce qui reste à faire

Au moment de la rédaction de ce rapport, il me reste à faire :

* Mettre en place plusieurs lecteurs RFID en simultané.
* La connexion entre la Raspberry et l’application de supervision via un système socket client/serveur.
* Un test en grandeur nature avec la gâche électrique.

## Poursuite d’étude

Après des tests et un entretien, j’ai été accepté à l’ENI pour la formation « Administrateur Système et Réseau » en alternance. L’entreprise où j’avais fait mon stage de première année m’avait fait savoir qu’ils seraient prêts à m’embaucher en tant qu’alternant. Je pourrais donc rejoindre l’entreprise « TIGS » (Technique Informatique Global pour la Santé) pour une durée de 2 ans, durée de la formation, avec une possibilité de continuer à l’ENI avec la formation « Expert en Sécurité Digitale ».

# Annexe

***Schéma de câblage***

