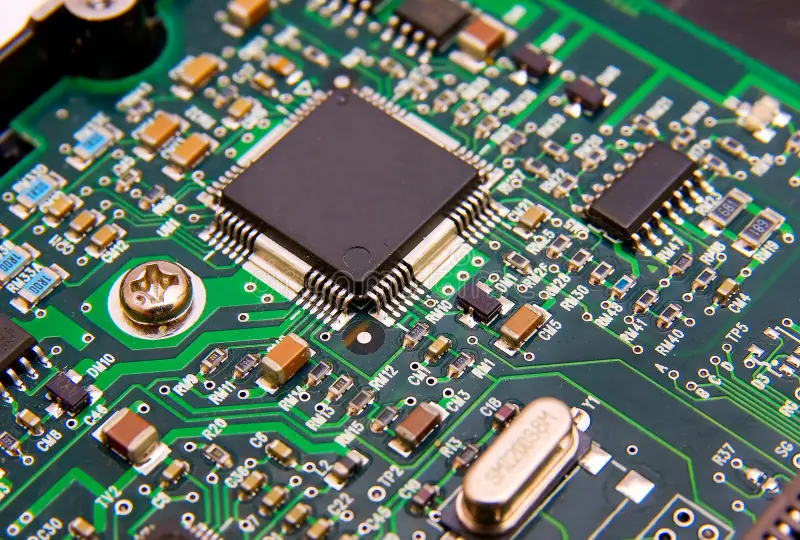
**LIVRABLE 4 : FULL STRONGBOX 3000**

****

**Membres du groupe :**

**Doudou Manil (Chef du groupe)**

**Rami Mohamed Amine.**

**Bouyacoub Rayan.**

**Mobarek Malik.**

Date et lieu de rédaction du livrable : 11/11/2023, CESI Exia Algérie.

**INTRODUCTION :**

Je vous présente la partie finale de notre projet, une extension naturelle de notre premier projet "Algo 3000". Ce nouveau livrable, intitulé "Full STONGBOX 3000", nous permettra de mettre en œuvre l'algorithme que nous avons conçu précédemment sur une carte Arduino.

Dans ce projet, nous allons transformer notre algorithme d'authentification robuste en un programme réel qui peut être exécuté sur une carte Arduino. Cela implique non seulement la traduction de l'algorithme en code, mais aussi la gestion des interactions matérielles avec la carte Arduino.

Un aspect clé de ce projet est l'importance accordée à la maintenabilité du code. Nous allons structurer notre programme en utilisant des fonctions, ce qui rendra notre code plus lisible, plus facile à déboguer et à maintenir.

Le livrable final de ce projet comprendra le montage final du prototype, avec toutes les connexions nécessaires à la carte Arduino, ainsi que le programme complet du prototype.

Par la suite nous pourrons attaquer la dernière partie de ce projet, la soutenance qui aura lieu ce lundi devant les jurys ou nous vous présenterons tout les livrables que nous avons envoyé et comment nous avons fait le code et le montage de la maquette de ce livrable.

**Nous présenterons ce livrable en 2 parties importantes : La partie du code puis la partie de la conception de la maquette du circuit.**

1. **L’écriture du code Arduino-C à partir de nos logigrammes :**

L’écriture du code des logigrammes que nous avons fait pendant le livrable 3 est la première étape de ce livrable afin que notre montage de circuit électronique puisse marcher correctement, afin de créer nos codes et faire le programme complet du prototype nous nous sommes inspirés des logigrammes que nous avons conçus précédemment dans le livrable 3 puis nous les avons vérifié sur Arduino IDE et testé sur Tinkercad afin d’être sur que tout marche correctement puis grâce au suivi de projet nous avons corrigé quelques erreurs qu’il y avait dans notre programme pour enfin attaquer la partie pratique de ce projet qui est le montage.

1. **Le montage de la maquette du circuit électronique :**

Après avoir fini le code, nous nous sommes attaqué directement à la conception de la maquette du circuit électronique du projet final, nous avons essayé de comprendre comment devait fonctionner le circuit à partir de l’énoncé du projet et grâce a l’aide de l’intervenant nous avons pu concevoir la maquette correctement malgré quelque problèmes au niveau de la valeur de la tension de sortie du circuit, au final nous avons testé le circuit avec l’Arduino et tout marcher correctement, nous vous expliquerons le montage du circuit et son fonctionnement complet pendant la soutenance du lundi.

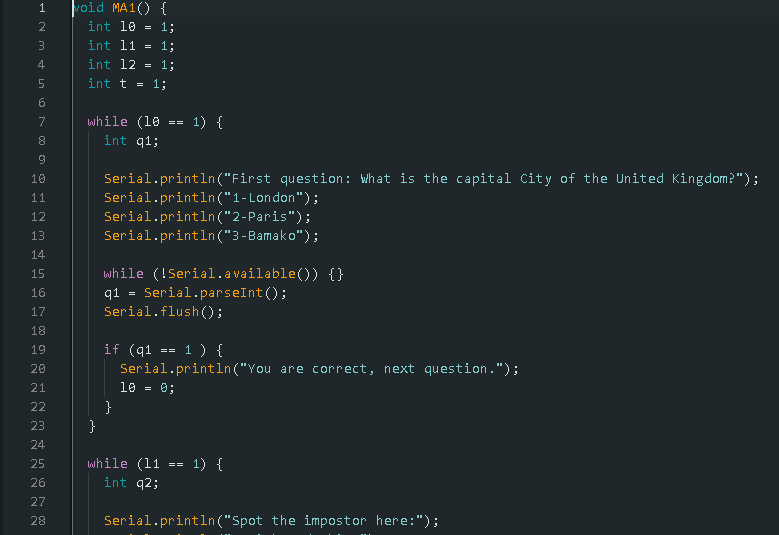
**Maintenant après avoir donner les grands titres nous allons vous présentait chaque programme de chaque modèle que nous avons concocter pour faire marcher notre maquette par la suite :**

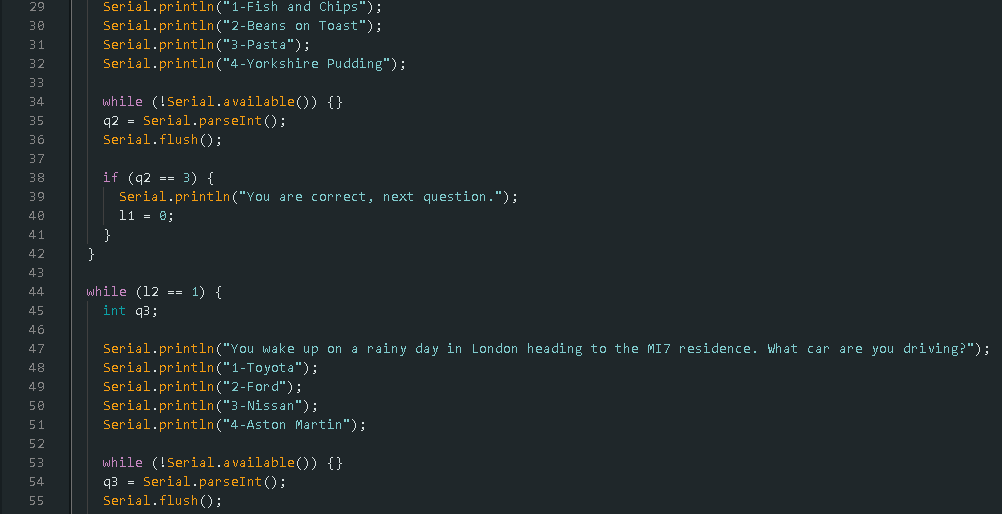
**MA1 : Authentification par Question/Réponse :**

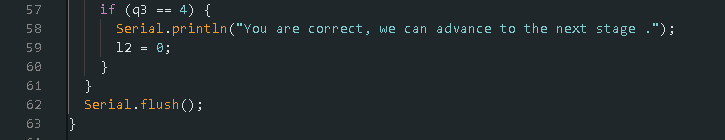
Un ensemble de questions est formulé, avec des réponses sous forme de choix multiples. Dans notre prototype, l'agent répondra simplement en sélectionnant le numéro correspondant à la réponse d'une question à choix multiples.

Notre code comprendra chaque question et les choix associés, qui seront affichés sur notre moniteur série en utilisant la fonction `Serial.println()`. Les entrées seront alors l'entier contenant la réponse correcte à chaque question, en utilisant les fonctions `Serial.available()`, `Serial.Read()` et `Serial.parseInt()`, le tout en utilisant des conditions `if`.

Il est important de ne pas oublier la fonction `Serial.begin(9600)`, qui assure une communication série efficace avec le moniteur série. Le code est présenté ci-dessous.







**MA2 : Authentification par un code qui change avec le temps :**

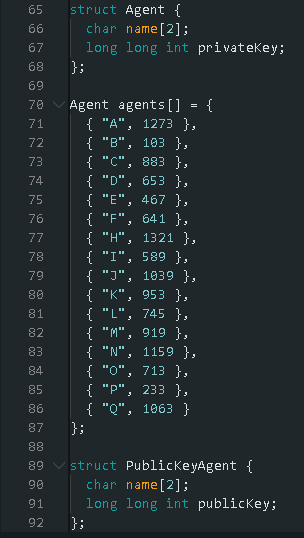
Ce niveau d'authentification fait appel à l'algorithme RSA pour générer un code aléatoire M qui varie à chaque tentative d'authentification. On va crypter ce message en utilisant la fonction de chiffrement qui prend cette opération mathématique comme paramètre : .

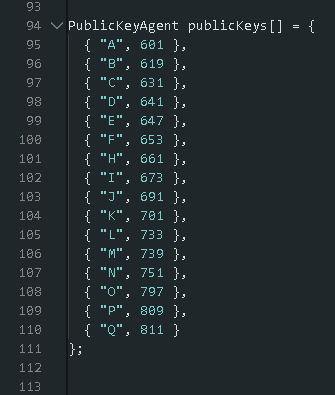
Ici, E représente la clé publique de l'agent. Par la suite, le programme nous invitera à entrer notre clé publique afin de déchiffrer le message crypté en utilisant ce processus : .

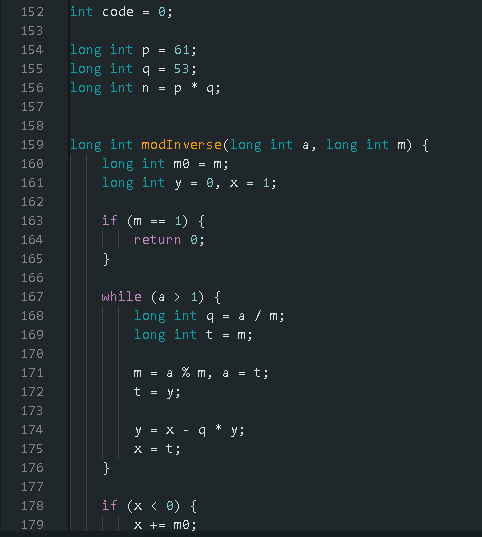
À la fin, le programme comparera M avec Mp et nous donnera accès au contenu du coffre uniquement si M=Mp.

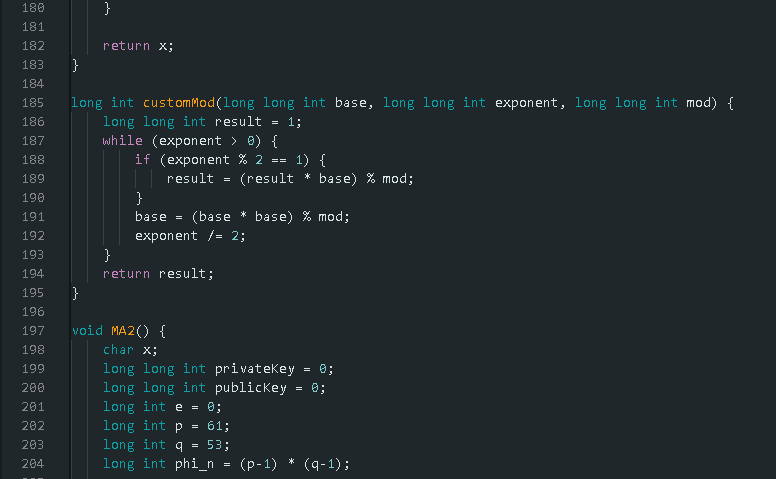
Le code sera composé de plusieurs fonctions, comme la fonction de chiffrement et la fonction de déchiffrement, et des tableaux pour stocker les lettres des agents et les clés publiques.

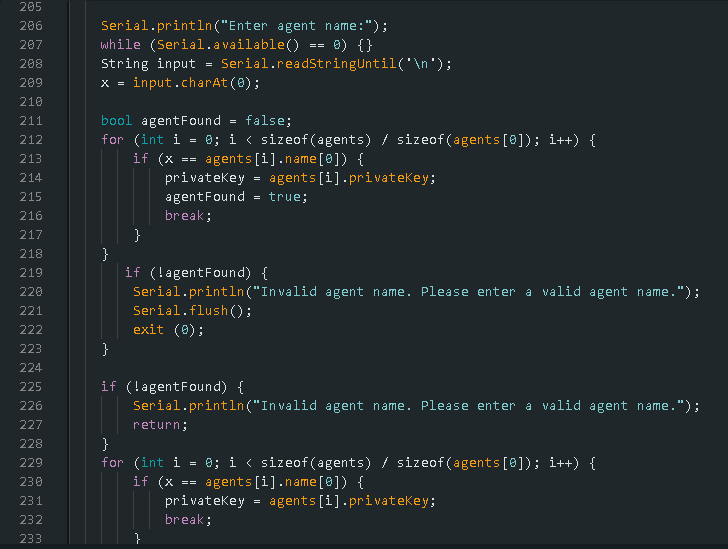
On nous demandera d'abord de nous identifier, puis il générera un entier aléatoire M<2881, qui sera crypté avec la clé publique associée à notre nom d'agent en appelant la fonction de chiffrement. Ensuite, on nous demandera d'entrer notre clé privée pour déchiffrer le message crypté en appelant la fonction de déchiffrement, puis on utilise la condition `if` pour comparer l'entier généré M avec le code déchiffré Mp comme nous allons le montrer ci-dessous.

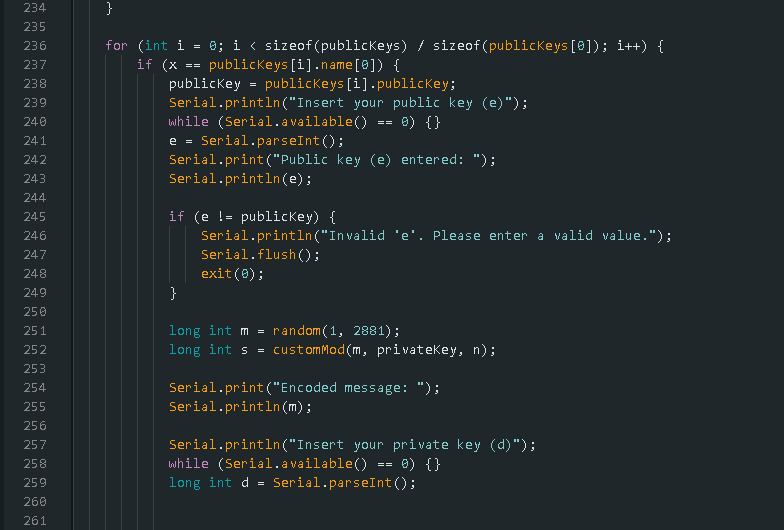


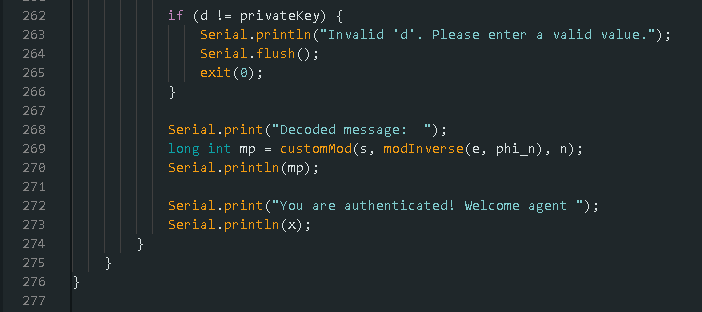






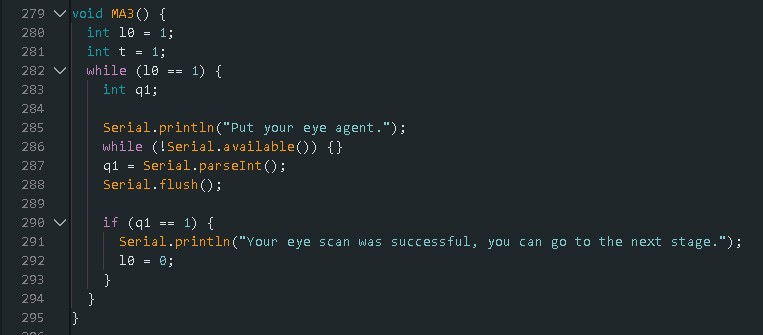




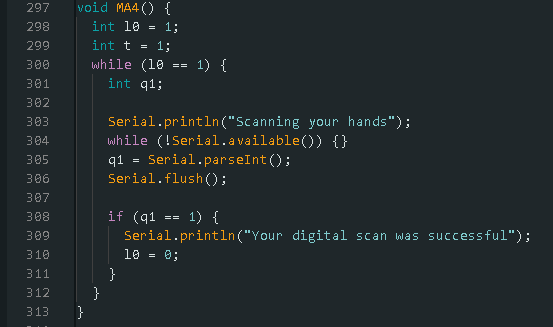
****

**MA3 : Scan rétinien, MA4 : Scan digital :**

**MA3 :** Dans ce niveau d’authentification, nous utiliserons un nombre entier pour représenter le résultat du scan. Par conséquent, l’accès n’est autorisé que si ce nombre entier est égal à 1, voilà le code :

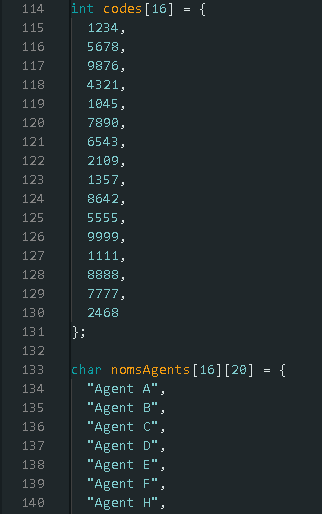


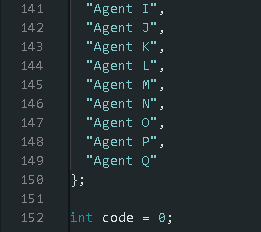
**MA4 :** Les explication sont identique à MA3, on remplace seulement l’empreinte rétinienne par un empreinte digitale, voilà le code :

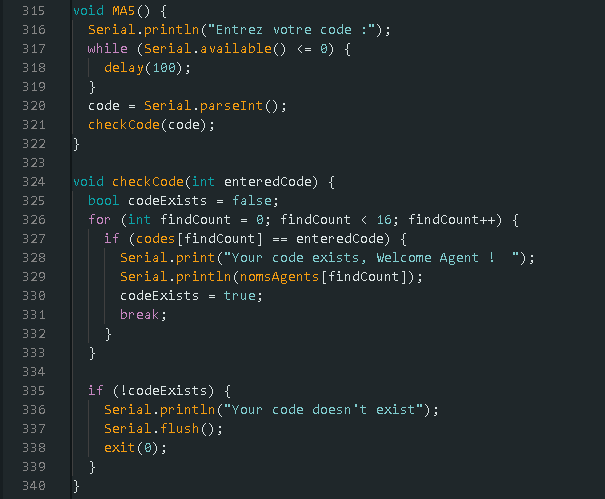


**MA5 : CARD\_ID**

Dans ce programme, l’agent est tenu d’entrer sa lettre (A, B, C, etc.) et le mot de passe associé à cette lettre. Cependant, pour contourner certaines complexités dans le code, nous avons substitué les lettres par des chiffres. Le programme présentera aux agents une alternative de cette façon : 0)A, 1)B, 2)C. Les différents mots de passe ont été conservés dans un tableau où l’index de chaque élément représente la lettre de l’agent. Par exemple, l’index 0 correspond à l’agent A, l’index 1 à l’agent B, et ainsi de suite. Voici le programme :







**Et pour finir vous trouverais le code principal avec et sans tension de sortie avec lequel nous allons brancher l’Arduino de notre maquette (notre circuit électronique) du projet en tant que pièce jointe du livrable :**

Le code final qui réunit les 5 Méthodes d’Authentification (MA) fonctionnerait de la manière suivante :

-MA1 : L’agent sélectionne une combinaison d’interrupteurs et insère sa carte dans le système. Le système récupère 4 signaux en sortie de la carte dépendant de la combinaison des 4 interrupteurs et de la carte introduite. Le système interne du coffre calcule la sortie attendue selon la combinaison d’interrupteurs, puis la compare avec ce signal.

-MA2 : L’agent est invité à entrer sa clé publique pour déchiffrer un message crypté. Le programme compare ensuite le message original avec le message déchiffré. Si les deux correspondent, l’accès est autorisé.

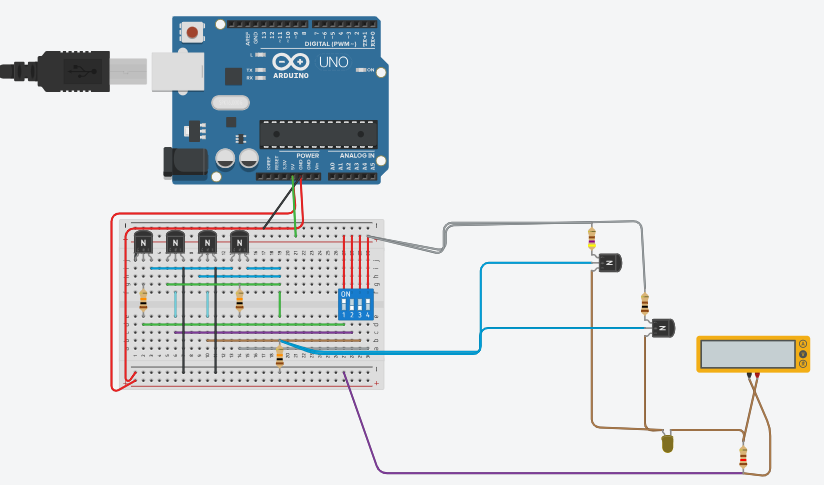
-MA3 : Un nombre entier est utilisé pour simuler le résultat du scan. L’accès est autorisé uniquement si ce nombre entier est égal à 1.

-MA4 : L’agent doit saisir sa lettre (A, B, C…) et le mot de passe correspondant à sa lettre. Les différents mots de passe sont stockés dans un tableau et l’index de chaque case représente la lettre de l’agent.

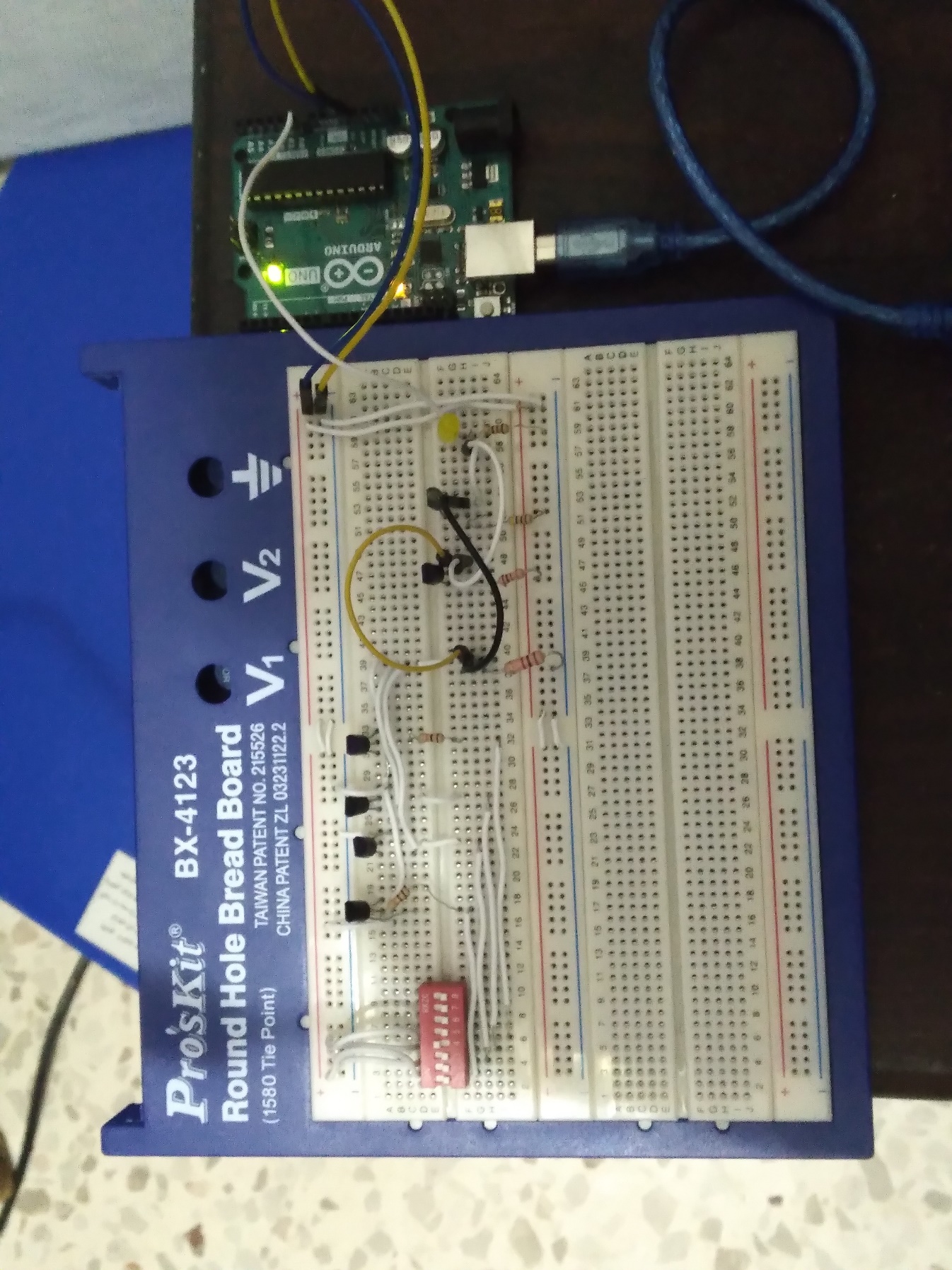
-MA5 : L’agent doit saisir son CARD\_ID.

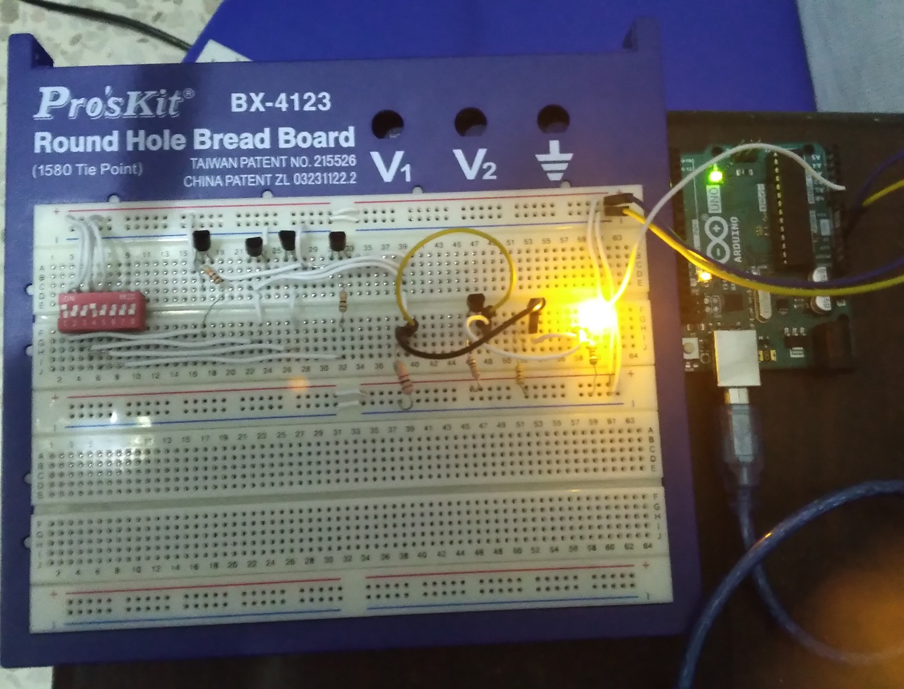
**Voilà le shéma Tinkercad et les vraies images de la maquette du circuit électronique de la maquette demandé pour ce livrable :**

1. **Le schéma Tinkercad du circuit électronique :**

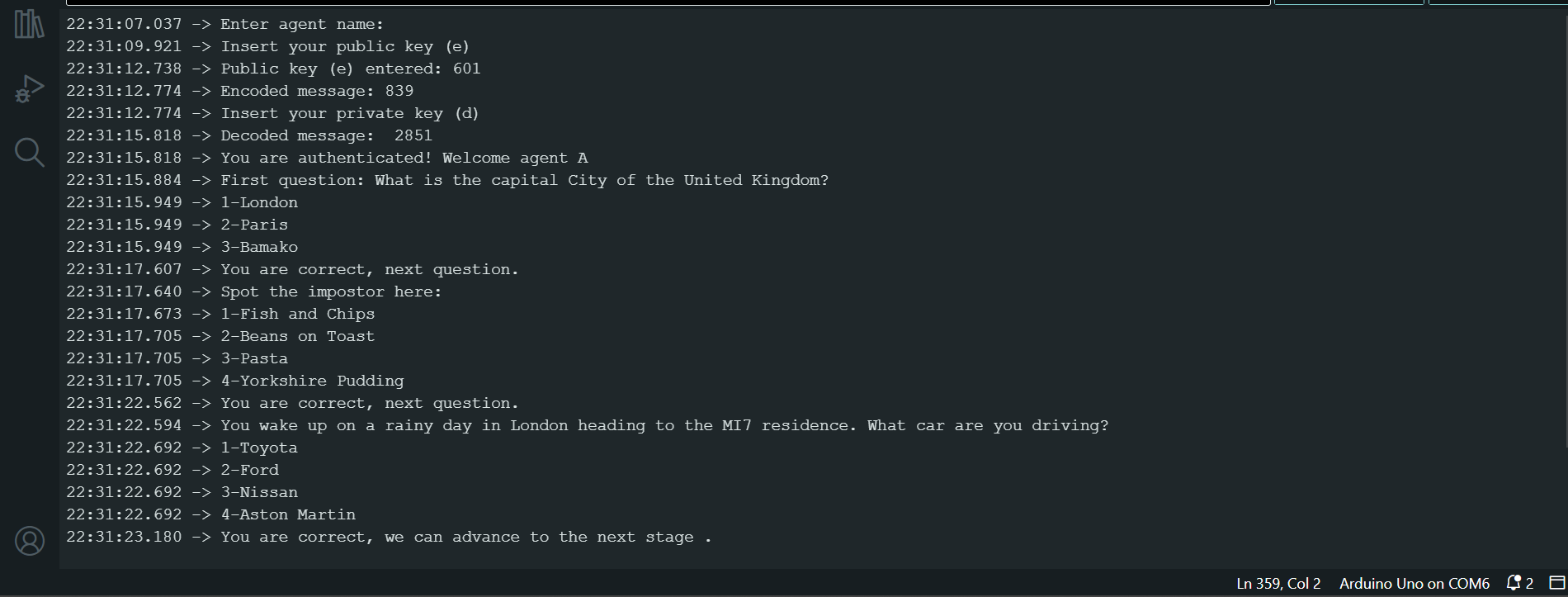


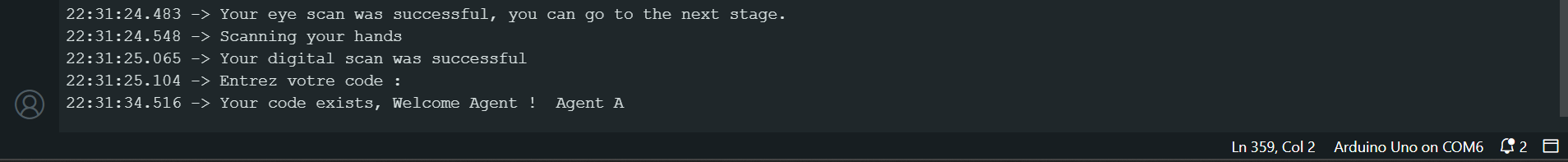
1. **Les vraies images de la maquette du circuit électronique :**

****

****

**Et bien sûr nous vous présentons également la simulation à la fin pour que vous soyez sur tout marche correctement :**

****

****

**Merci pour votre attention à notre travail, nous espérons avoir tout bien fait et dans les normes.**