



IOT

PROJET ARDUINO

2023

DEFRANCE Baptiste
TASSONE Laura

Table des matières

INTRODUCTION	3
ORGANISATION.....	3
LA MAQUETTE	5
LE CAPTEUR DE DISTANCE PAR ULTRASONS :	10
LES LEDS :.....	12
BUZZER.....	13
TÉLÉCOMMANDE INFRAROUGE	14
SERVO MOTEUR.....	15
BADGE ET LECTEUR RFID.....	17
ECRAN LCD	19
LES INFORMATIONS :	21
STORYTELLING	22
MISE EN SITUATION RÉELLE	22
LES SHELLY.....	23
RACCOURCI	26
NFC	27
SHELLY CLOUD	28

Introduction

Dans ce projet, le but était de concevoir un portail électrique suivant plusieurs règles bien définies. Nous avons dû prendre en compte plusieurs manière d'aborder le sujet comme la sécurité physique, la sécurité d'intrusion ou encore la manière dont fonctionnera notre portail ainsi que les éléments utiles liés à celui-ci.

Nous avons modifié le projet pour l'améliorer à notre manière. Le but final est donc de faire un portail électrique avec une ouverture contrôlée avec contrôle d'accès : badge RFID et télécommande. Pour avertir les différentes actions de la porte, nous sommes partis sur des LED ainsi que des bruits d'avertissements. Nous aurons aussi toute la partie sécurité active. Pour finir, nous avons relié un vrai portail sur le réseau pour pouvoir le fermer et l'ouvrir à distance.

Organisation

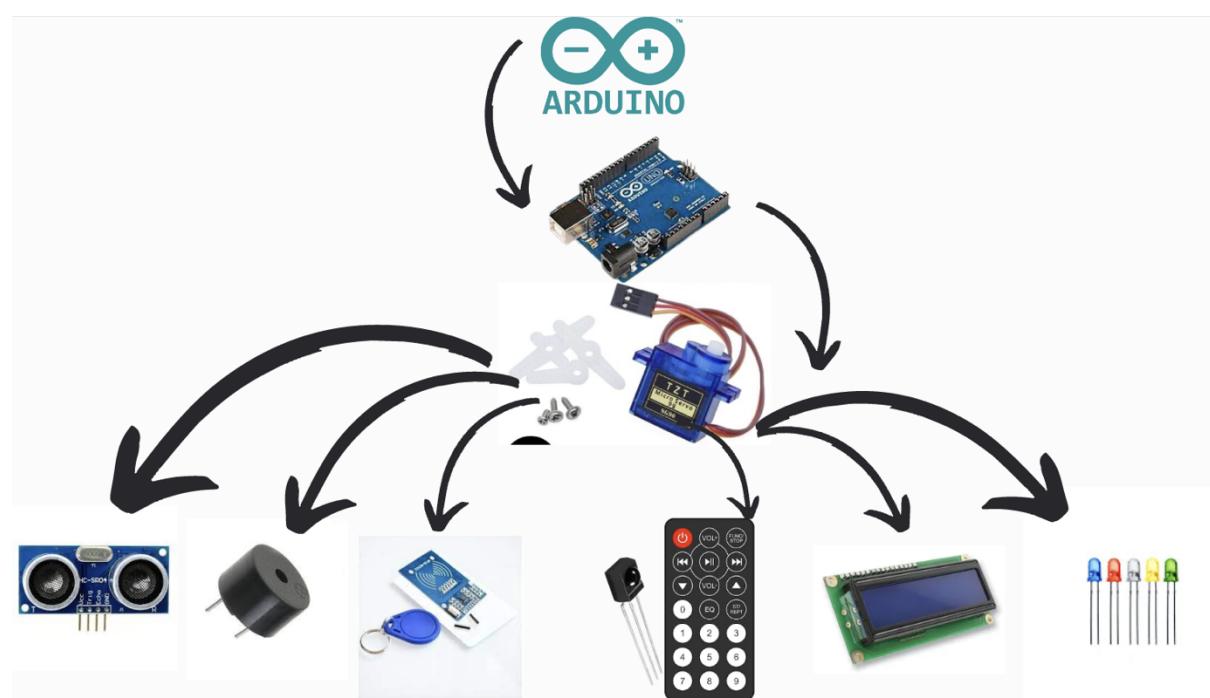
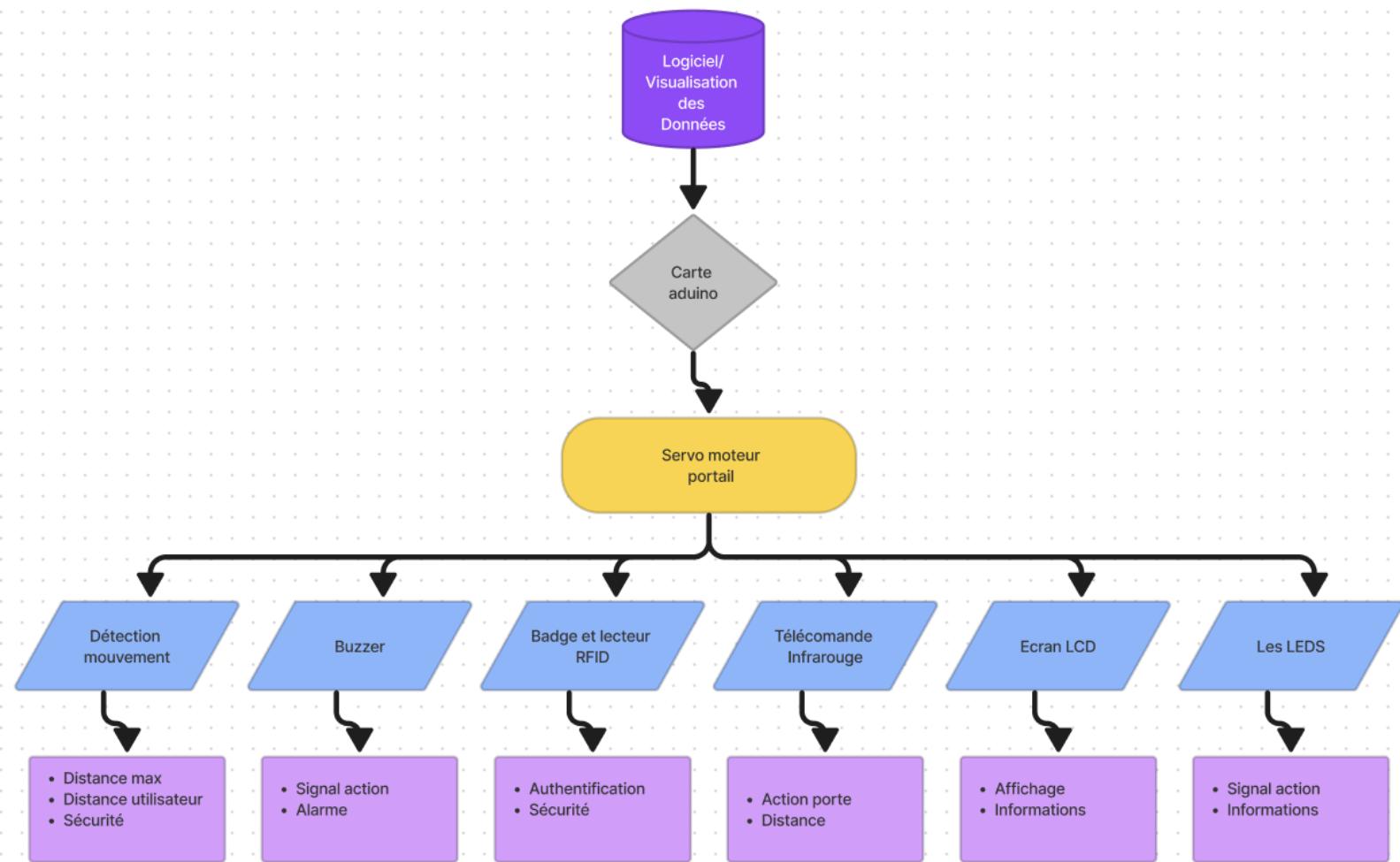
Pour mener à bien ce projet de groupe, nous avons dû nous organiser pour répartir les différentes tâches et avoir un projet structuré.

Nous avons utilisé Trello pour nous répartir les différentes tâches du projet. Cette plateforme permet de valider ou non des tâches en cours.

Pour la partie développement, nous utilisons git pour nous partager les différentes parties de code et donc ainsi pouvoir les tester sur nos deux ordinateurs.

Pour la partie maquette, nous avons utilisé le logiciel Tinkercad qui permet de faire des simulations de différents modules d'IOT.

Nous avons ensuite réalisé la maquette en réel grâce à une carte arduino, une bread board et plusieurs autres modules.

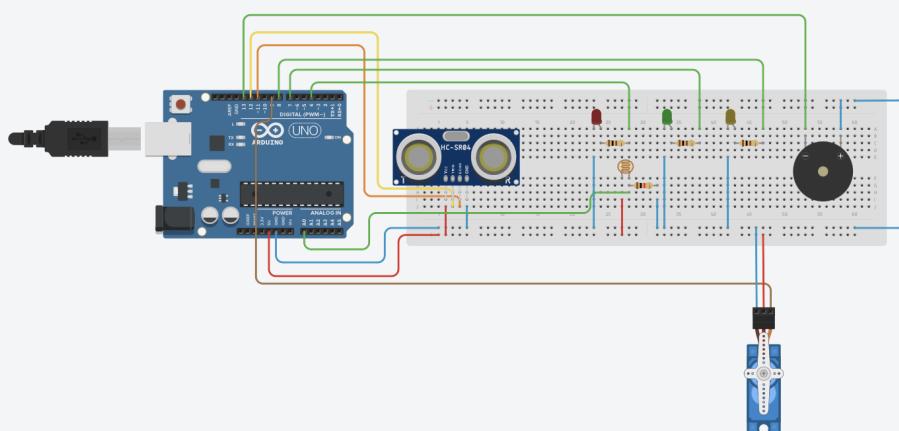


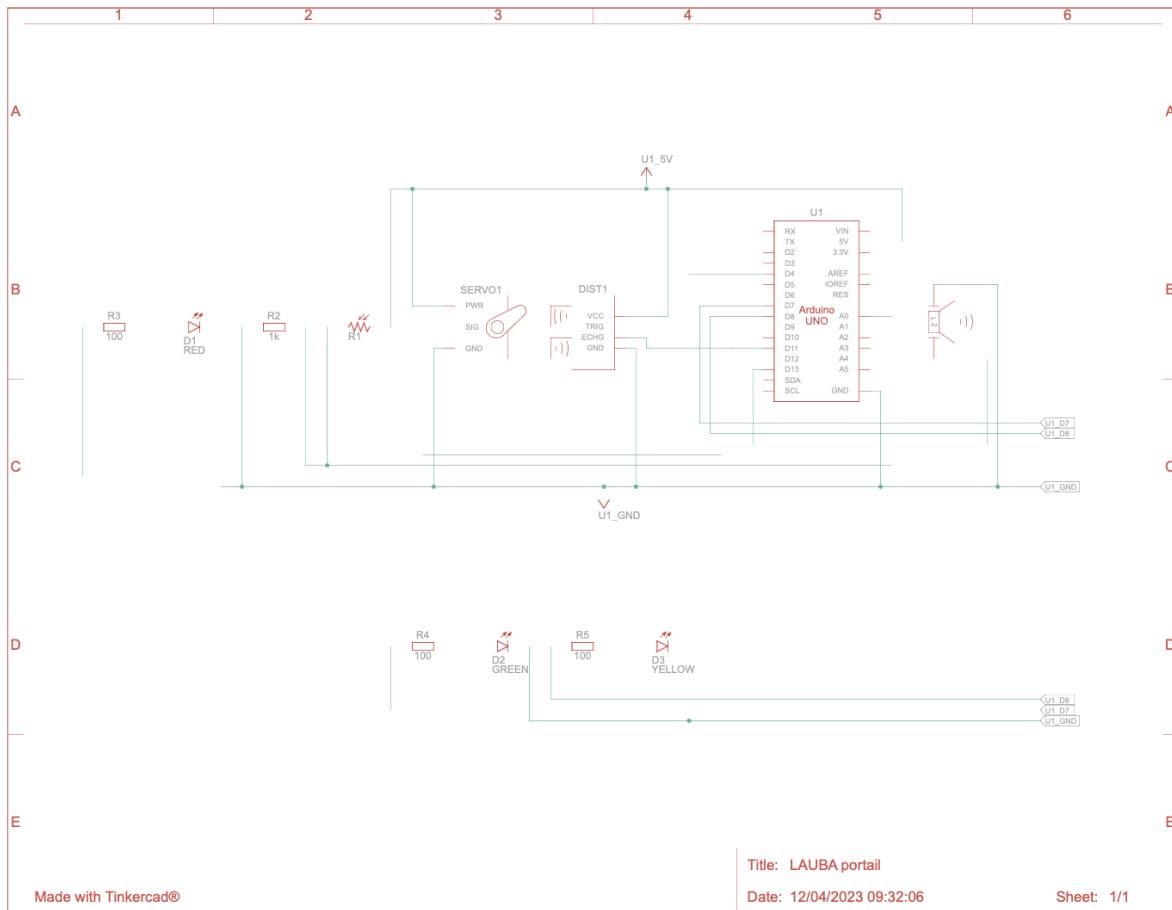
La maquette

Pour commencer le projet, nous avons d'abord créé un projet vide sur Tinkercad. Nous avons par la suite ajouté les différents composants que nous avons eu besoins pour notre projet :

- Carte arduino Uno
- Capteur de distance par ultrasons
- 3 leds
- Buzzer
- Servo moteur
- 3 résistances
- Des fils
- Un capteur infrarouge
- Une télécommande
- Un lecteur RFID
- 2 badges

Voici une capture d'écran de notre projet Tinkercad à la moitié de l'avancé :





Une fois les éléments principaux placés, nous commençons à coder le programme qui nous servira à commander et à faire fonctionner le portail.

Notion : Le codage en Arduino consiste à écrire des instructions en langage de programmation C/C++ pour contrôler le fonctionnement d'une carte électronique Arduino. Cette carte est équipée d'un microcontrôleur qui peut être programmé pour effectuer différentes tâches telles que la lecture de capteurs, la commande de moteurs ou la communication avec d'autres appareils. Le codage en Arduino implique généralement la définition de variables, l'écriture de fonctions et la gestion des entrées et sorties de la carte. Les programmes peuvent être téléchargés sur la carte via un câble USB, permettant ainsi de contrôler des projets électroniques de manière autonome.

Les fonctions `setup()` et `loop()` sont des fonctions principales dans la programmation Arduino. La fonction `setup()` est appelée une fois au démarrage du programme et sert à initialiser les différents paramètres et configurations nécessaires pour le bon fonctionnement du programme. Cela peut inclure la définition des broches d'entrée/sortie, l'initialisation de variables, la configuration de la vitesse de transmission de données, etc. La

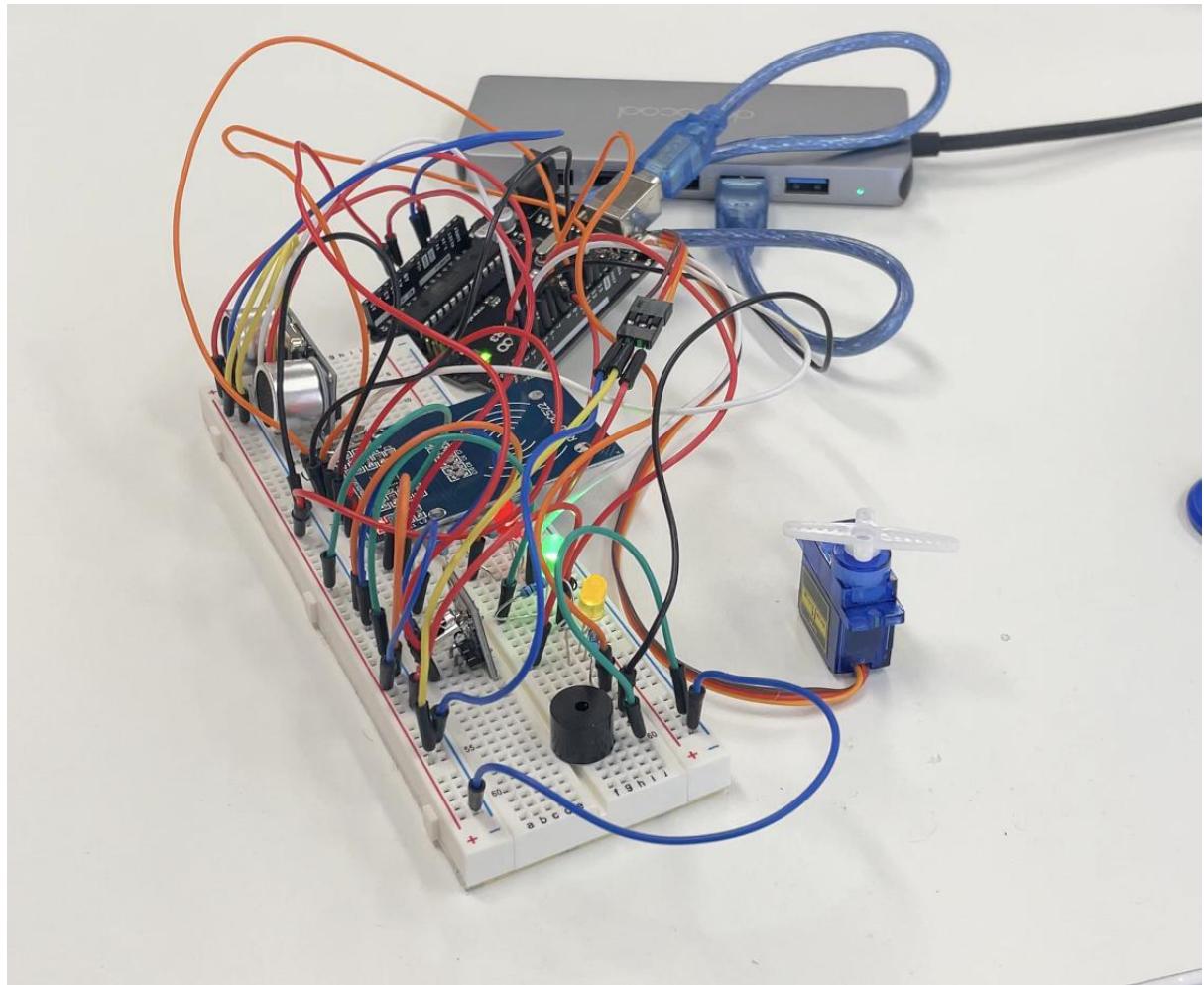
fonction `loop()` est ensuite appelée en boucle après l'exécution de la fonction `setup()`. Cette fonction contient les instructions qui seront exécutées en continu jusqu'à ce que la carte Arduino soit éteinte ou que le programme soit interrompu. La fonction `loop()` est l'endroit où les instructions de commande principale sont placées. Les instructions placées dans la fonction `loop()` sont exécutées en séquence, du haut vers le bas, jusqu'à ce que la fin de la fonction soit atteinte. En résumé, la fonction `setup()` est utilisée pour configurer et initialiser les paramètres de la carte Arduino au démarrage, tandis que la fonction `loop()` est utilisée pour exécuter les instructions en boucle qui contrôlent le comportement de la carte Arduino.

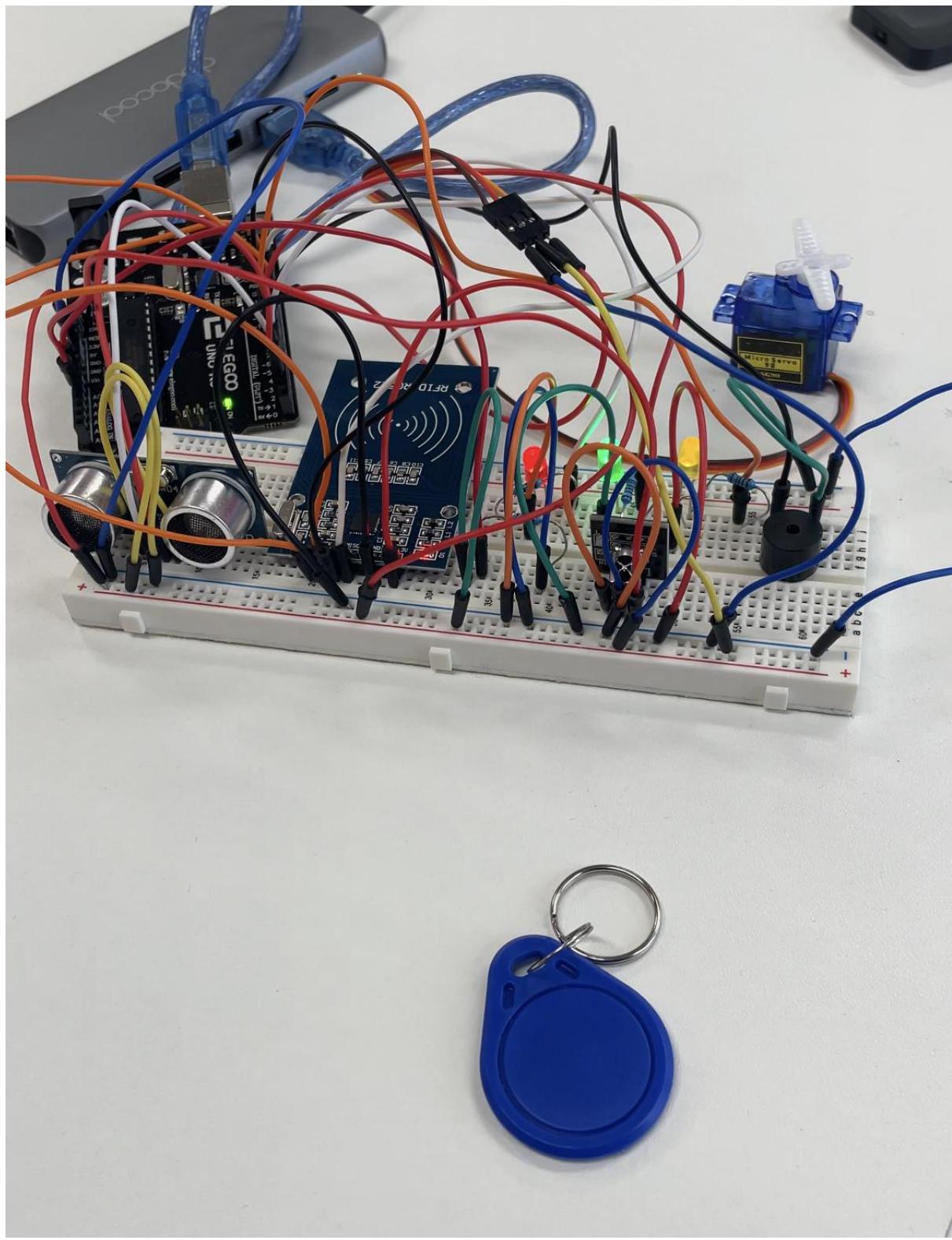
Voici une partie du code que nous avons fait pour faire fonctionner notre portail :

```
1 #include <Servo.h> // import du module Servo
2
3 Servo porte;
4
5
6 // Fonction mesurant la distance en cm au détecteur
7 long readUltrasonicDistance(int pinTrigger, int pinEcho)
8 {
9     pinMode(pinTrigger, OUTPUT); // Clear the trigger
10    digitalWrite(pinTrigger, LOW);
11    delayMicroseconds(2);
12    // Sets the trigger pin to HIGH state for 10 microseconds
13    digitalWrite(pinTrigger, HIGH);
14    delayMicroseconds(10);
15    digitalWrite(pinTrigger, LOW);
16    pinMode(pinEcho, INPUT);
17    // Reads the echo pin, and returns the sound wave travel time in
18    return 0.01723 * pulseIn(pinEcho, HIGH);
19 }
20
21
22 // Déclaration des variables utilisées
23 long distance;
24 int lum;
25 int compt=0;
26
27
28 // Définition des pins
29 int pinLED1 = 4;
30 int pinLED2 = 7;
31 int pinLED3 = 8;
32 int pinServo = 9;
33 int pinBuzzer = 13;
34 int pinTrigger = 12;
35 int pinEcho = 11;
```

Une fois notre maquette de portail fonctionnelle sur Tinkercad nous l'avons créé en physique avec une carte arduino et une breadboard.

Voici une photo de notre maquette :



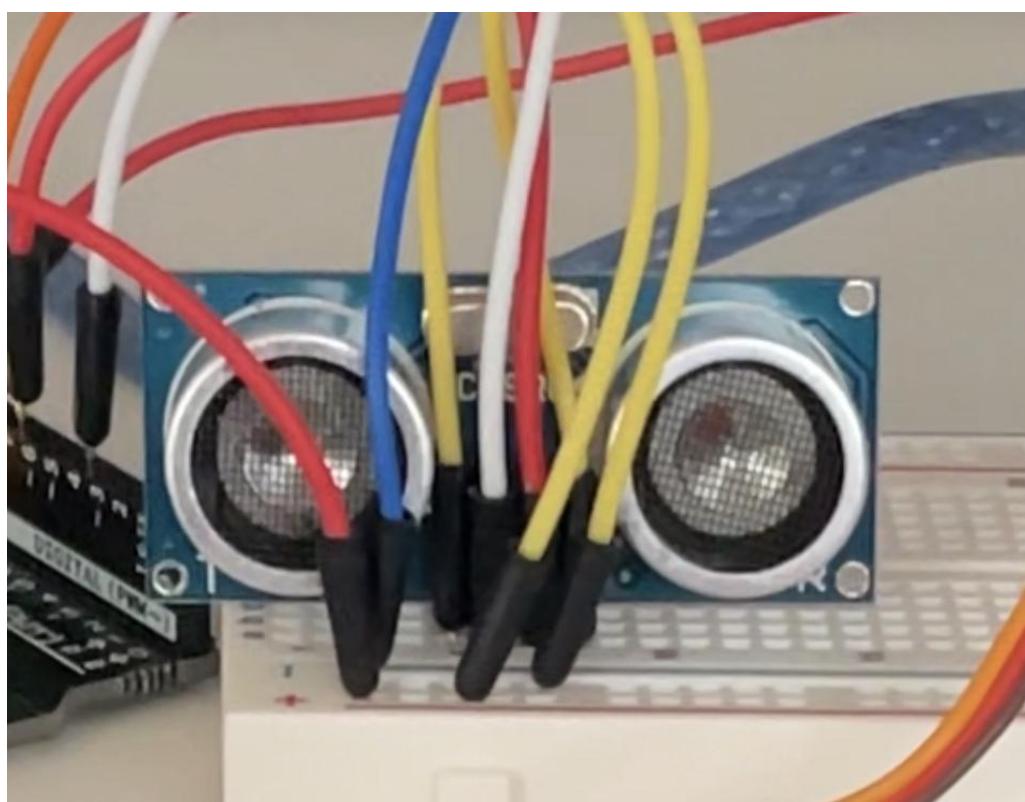


Le capteur de distance par ultrasons :

Pour pouvoir vérifier si quelqu'un se trouve devant le portail, nous avons dû mettre en place un capteur de distances par ultrasons.

Un capteur de distances par ultrasons fonctionne en utilisant les propriétés des ondes sonores pour mesurer la distance entre le capteur et un objet. Le capteur envoie un signal sonore à haute fréquence (ultrasons) qui se propage dans l'air et rebondit sur l'objet. Le capteur mesure le temps que prend le signal pour se rendre à l'objet et revenir, puis calcule la distance en fonction du temps de vol. Plus précisément, le capteur envoie une série de signaux sonores à une fréquence élevée, généralement autour de 40 kHz, en utilisant un émetteur ultrasonique. Le signal se propage dans l'air en formant une onde sonore qui se déplace à une vitesse constante. Lorsque l'onde sonore rencontre un objet, elle est réfléchie et retourne au capteur. Le capteur utilise ensuite un récepteur ultrasonique pour détecter l'onde sonore réfléchie et mesure le temps écoulé entre l'émission du signal et la réception de l'onde sonore réfléchie.

Voici le capteur à ultrasons :



Les fonctionnalités de ce capteur :

Pour notre projet, ce capteur nous a été utile pour effectuer plusieurs actions.

La fonctionnalité principale est de détecter si un mouvement se manifeste devant le capteur. Si la distance entre l'objet et le capteur est inférieur à 30 cm, alors nous pouvons lancer la fonction qui ouvrira le portail.

```
int seuilDistance = 30;
```

```
if (distance < seuilDistance) {  
    actionporte();  
}
```

La deuxième fonctionnalité est que si un mouvement est détecté lors de l'ouverture ou de la fermeture du portail, celui-ci se bloque pour éviter de causer des dégâts (sur une personne ou sur du matériel).

Les LEDS :

Pour se rendre compte des différentes actions qui seront effectuées, nous avons installé des leds qui permettent à un utilisateur de savoir ce qu'est en train de faire son portail.

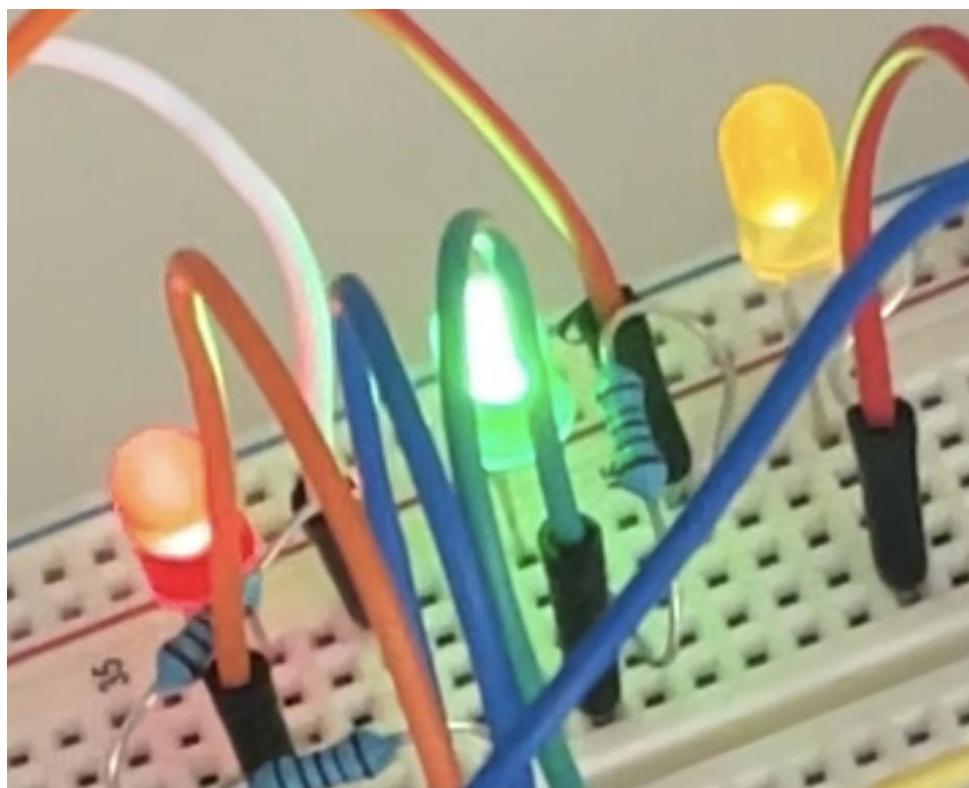
3 différentes actions des leds sont possibles :

Leds éteintes : Lorsque les leds sont éteintes c'est que le portail est fermé.

Leds allumés : Lorsque les leds sont allumées, c'est que le portail est en position ouverte.

Leds clignotantes : Lorsque les leds clignotent, c'est que le portail est en train de s'ouvrir ou de se fermer.

Voici nos leds en action :



Buzzer

En plus des leds qui guide l'utilisateur à connaître les différentes actions menées par le portail, le buzzer créer un son et permet à l'utilisateur de comprendre plusieurs actions :

Buzzer sonne une fois : Lorsque le buzzer sonne une fois, c'est que le portail va commencer à s'ouvrir ou à se fermer.

Buzzer sonne à intervalle régulier : lorsque le buzzer sonne à un intervalle régulier, c'est qu'une intrusion est en cours et déclenche l'alarme sonore grâce au buzzer.

Voici une image de notre buzzer :



Télécommande infrarouge

Pour ouvrir notre portail, nous utilisons une télécommande infrarouge.

La télécommande infrarouge en Arduino est un système qui permet de contrôler les appareils électroniques à distance en utilisant une télécommande infrarouge.

Cette télécommande envoie des signaux infrarouges qui sont détectés et interprétés par un récepteur infrarouge connecté à une carte Arduino.

La carte Arduino peut ensuite exécuter des instructions en fonction des signaux reçus, permettant ainsi de contrôler divers appareils électroniques.

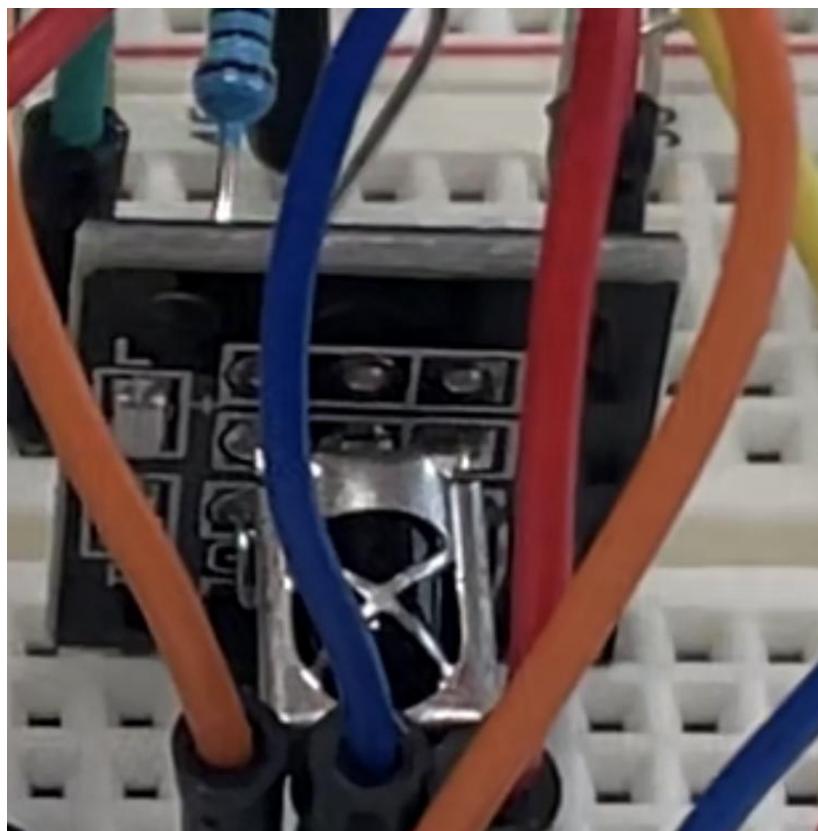
Pour utiliser une télécommande infrarouge avec une carte Arduino, il faut connecter un module récepteur infrarouge à la carte.

Le module récepteur infrarouge est composé d'un phototransistor qui détecte les signaux infrarouges émis par la télécommande et d'un circuit intégré qui décode ces signaux pour en extraire les informations de commande.

Une fois que le module récepteur infrarouge est connecté à la carte Arduino, il est possible de programmer la carte pour exécuter différentes tâches en fonction des signaux reçus.

Par exemple, en appuyant sur un bouton de la télécommande infrarouge, il est possible de contrôler la vitesse d'un moteur, la luminosité d'une LED ou d'exécuter toute autre commande que l'on souhaite programmer.

Voici une photo de notre récepteur :



Servo Moteur

Un servo-moteur est un dispositif de contrôle de mouvement qui peut être utilisé avec une carte Arduino pour contrôler l'ouverture et la fermeture d'un portail.

Un servo-moteur est un type de moteur qui peut être contrôlé avec précision pour effectuer des mouvements précis sur une plage limitée d'angles.

Pour utiliser un servo-moteur avec une carte Arduino pour contrôler un portail, il est nécessaire de connecter le servo-moteur à la carte Arduino à l'aide de fils électriques.

Le servo-moteur aura généralement trois fils : un pour l'alimentation, un pour la terre, et un pour le signal.

La plupart des servo-moteurs ont un angle de rotation limité, souvent de 180 degrés, bien que certains modèles aient un angle de rotation continu.

Pour contrôler la position du servo-moteur, la carte Arduino envoie des signaux de commande au servo-moteur via le fil de signal.

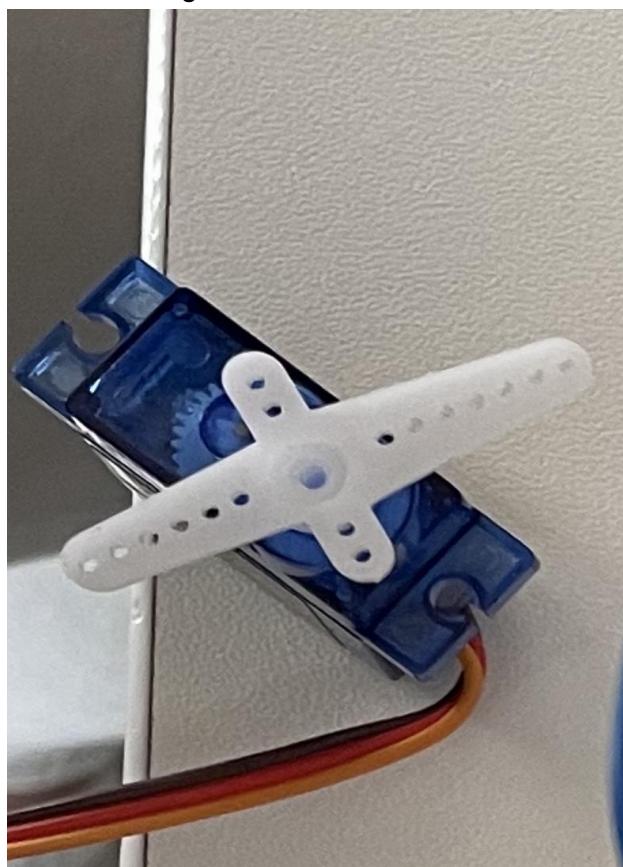
La commande est généralement envoyée sous forme de signal PWM (modulation de largeur d'impulsion) qui indique la position souhaitée du servo-moteur.

Pour utiliser un servo-moteur pour contrôler un portail, il est possible de le fixer à une partie mobile du portail, comme une charnière ou une poignée, de sorte que le mouvement du servo-moteur déplace la partie mobile du portail.

La position du servo-moteur peut être contrôlée via un programme Arduino, qui peut être conçu pour ouvrir et fermer le portail en fonction des signaux de commande reçus.

Il est important de noter que le choix d'un servo-moteur approprié dépend de la taille et du poids du portail ainsi que de la force nécessaire pour le déplacer.

Voici une image de notre servo moteur :



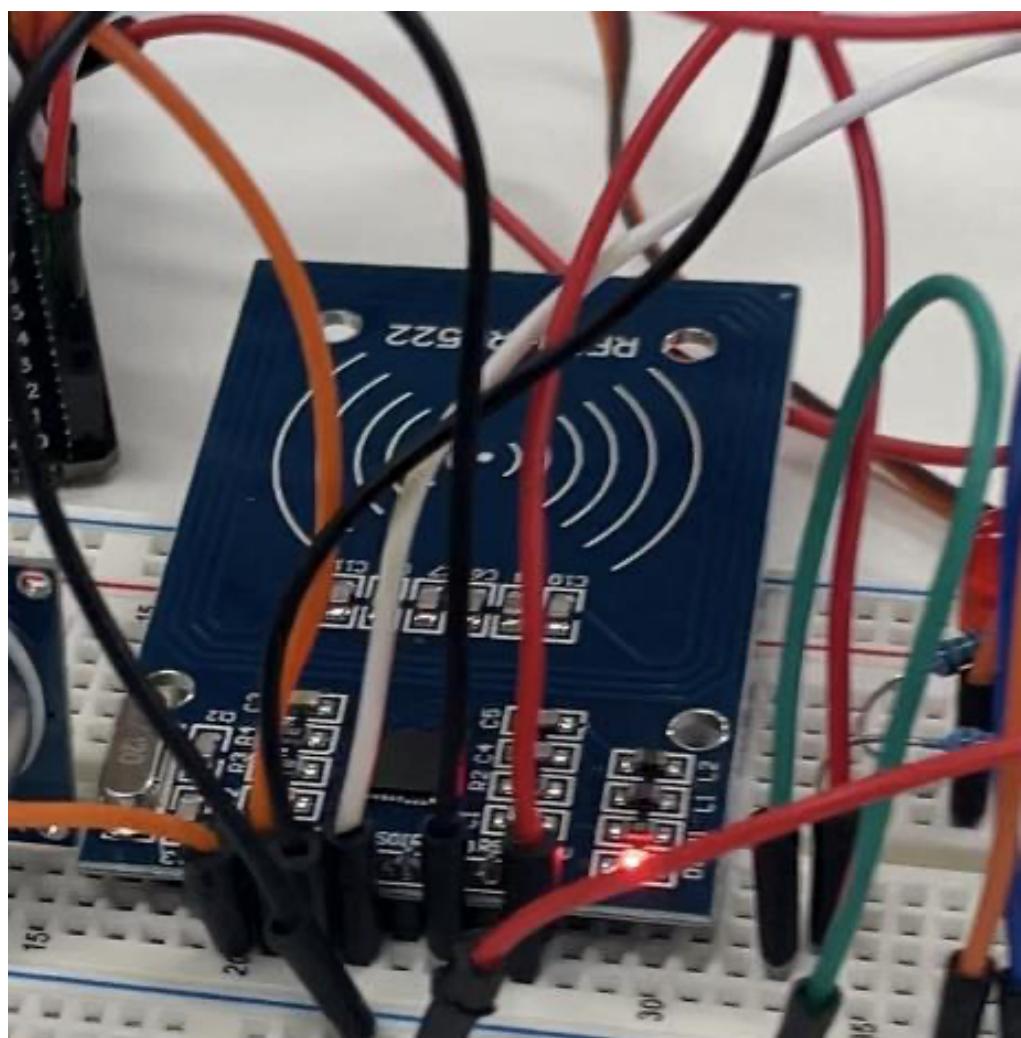
Badge et lecteur RFID

Un badge RFID (Radio Frequency Identification) est une carte ou un tag qui contient une puce et une antenne qui permettent de stocker et de transmettre des informations via des ondes radio. Un lecteur RFID est un dispositif qui peut lire les informations stockées sur un badge RFID en utilisant des ondes radio. En utilisant une carte Arduino et un lecteur RFID, il est possible de créer un système de contrôle d'accès qui utilise des badges RFID pour autoriser ou refuser l'accès à une zone sécurisée.

Dans notre cas, le badge RFID doit être présenté au lecteur. Un badge RFID dispose d'un UID unique. Si l'UID du badge correspond à celui précisé dans notre code, alors le portail s'ouvre. A l'inverse, si le badge n'a pas un bon UID alors au bout de 3 essais, l'alarme se déclenche.

Voici notre lecteur et notre badge :





Écran LCD

Un écran LCD 1602 est un type d'affichage à cristaux liquides qui peut être utilisé avec une carte Arduino pour afficher du texte et des symboles.

Il est appelé "1602" car il peut afficher 16 caractères par ligne et 2 lignes de texte. Pour utiliser un écran LCD 1602 avec une carte Arduino, il est nécessaire de le connecter à la carte à l'aide de fils électriques.

L'écran LCD 1602 aura généralement 16 broches qui doivent être connectées à des broches spécifiques de la carte Arduino.

Ces broches sont généralement désignées comme des broches de données, des broches d'alimentation et des broches de commande.

Une fois que l'écran LCD 1602 est connecté à la carte Arduino, il est possible d'afficher du texte et des symboles sur l'écran en envoyant des commandes à l'écran via la carte Arduino.

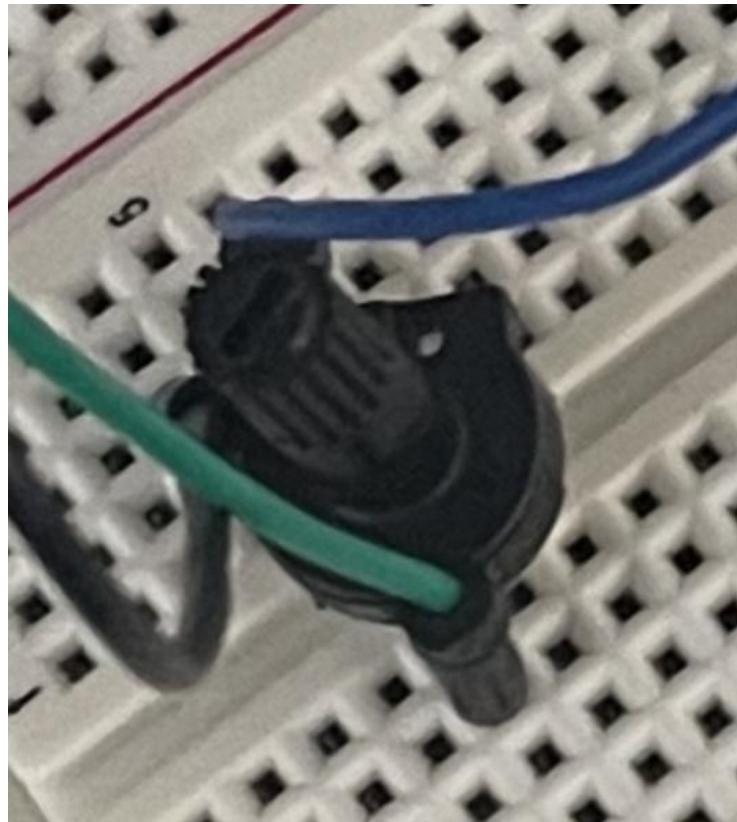
Le programme Arduino peut être conçu pour envoyer des commandes spécifiques à l'écran LCD 1602, comme l'affichage d'un texte ou d'un symbole spécifique à une position spécifique sur l'écran.

Il est également possible de contrôler l'éclairage de l'écran LCD 1602 en utilisant une commande spécifique.

Le contraste de l'écran peut également être ajusté en modifiant la tension de la broche de contrôle de contraste de l'écran.

En somme, un écran LCD 1602 en Arduino est une solution pratique pour afficher des informations en temps réel, notamment pour les projets nécessitant une interface utilisateur ou une interface de contrôle.

Voici une image du tensiomètre qui règle le contraste et l'écran LCD :



Les informations :

Pour informer l'utilisateur des différentes actions qui seront effectuées, il est possible de consulter les activités en direct de la carte arduino qui gère toutes les actions.

En effet, que ce soit les actions effectuées sur la porte, les distances de sécurité, les distances des objets détectés, la lumière des leds et toutes les informations liées à chaque module.

Nous avons aussi créé un historique qui enregistre les ouvertures / fermetures du portail ainsi que l'heure à laquelle l'action s'est effectuée.

```
Un badge est détecté
L'UID du tag est:
85 C8 51 83
Distance détectée : 9 cm
85 C8 51 83 Code érroné
Un badge est détecté
L'UID du tag est:
85 C8 51 83
Distance détectée : 8 cm
85 C8 51 83 Code érroné
Un badge est détecté
L'UID du tag est:
15 14 13 D3
Distance détectée : 269 cm
15 14 13 D3 Ouverture de la porte
Aucun objet détecté
-----
Un badge est détecté
L'UID du tag est:
15 14 13 D3
Distance détectée : 12 cm
15 14 13 D3 Ouverture de la porte
Buzzer !
Ouverture de la porte
Lumière : 27
Fermeture de la porte et extinction des LED
```

```
Un badge est détecté
L'UID du tag est:
85 C8 51 83
Distance détectée : 270 cm
85 C8 51 83 Code érroné
Un badge est détecté
L'UID du tag est:
85 C8 51 83
Distance détectée : 270 cm
85 C8 51 83 Code érroné
Un badge est détecté
L'UID du tag est:
85 C8 51 83
Distance détectée : 268 cm
85 C8 51 83 Alarme!
Alarme!
Alarme!
Alarme!
Alarme!
Alarme!
Alarme!
```

Storytelling

Le projet étant très intéressant pour nous, nous avons décidé de ne pas s'arrêter là.

Pour aller plus loin, nous avons voulu faire le projet en taille réelle.

En plus de notre maquette fonctionnelle, nous avons eu l'idée de faire en réel ce projet.

Sachant que le père de Baptiste habite dans une maison, nous lui avons donc proposé de lui automatiser les actions de son portail.

La proposition lui a tout de suite beaucoup plu.

Pour compléter le projet, ce petit "challenge" était l'opportunité idéale.

Nous avons donc pu s'entraîner à taille réelle avec du matériel adapté pour un portail grâce à de nombreuses heures passées à rechercher comment nous allions faire.

Finalement, nous sommes heureux d'avoir réussi à finir ce projet.

Sur ce projet, nous avons beaucoup appris sur différentes technologies que nous n'aurions peut-être jamais découvertes.

Nous avons pu explorer de nouvelles perspectives et découvrir des domaines que nous ne connaissions pas auparavant. Cela nous a permis d'enrichir notre expérience et d'élargir notre champ de compétences en IOT.

Mise en situation réelle

Pour cela nous avons regardé les technologies qui pourraient être intéressantes pour faire ce projet.

Nous en avons conclu que les "Shelly" étaient les meilleurs modules pour automatiser toutes ces actions en réseau.

Les Shelly

Le module Shelly est un appareil d'Internet des objets (IoT) fabriqué par la société bulgare Allterco Robotics.

Il est conçu pour être installé derrière un interrupteur mural existant pour contrôler l'éclairage, les volets roulants, les ventilateurs et d'autres appareils électriques à partir d'un smartphone ou d'une tablette.

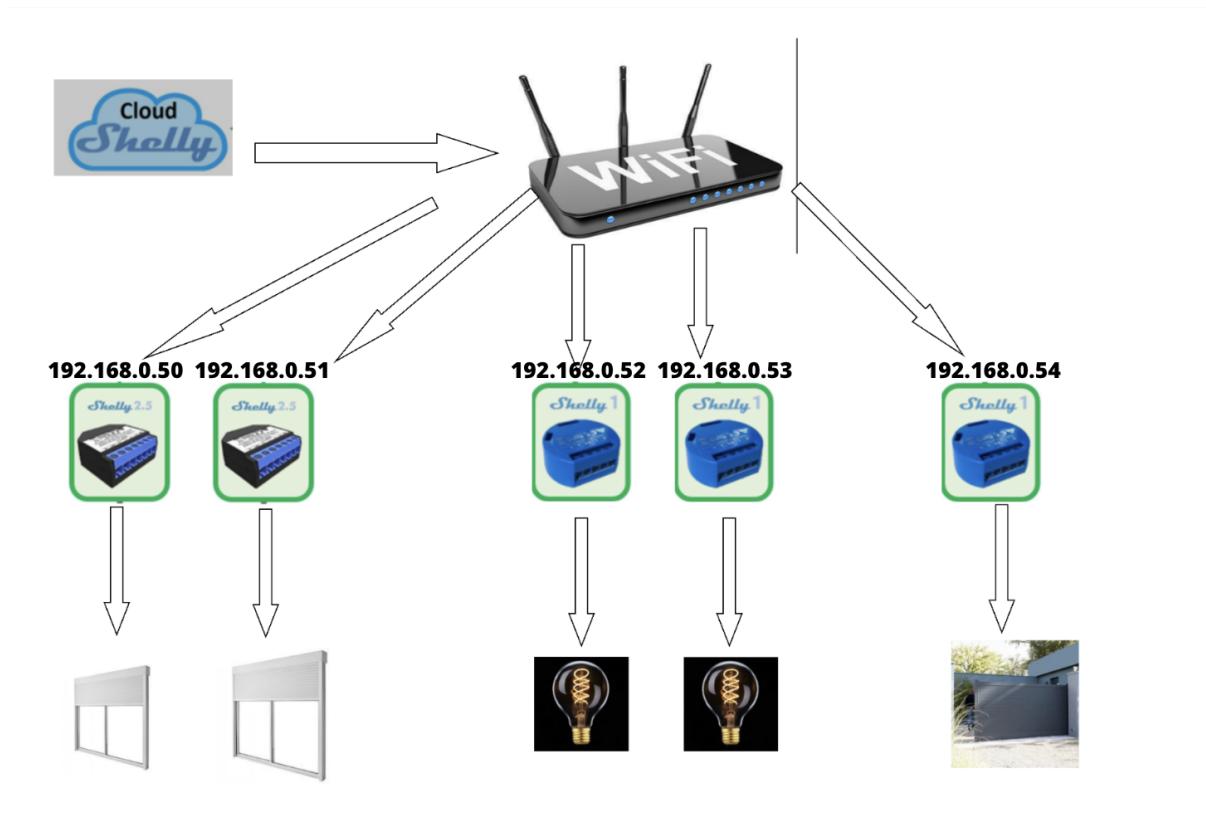
Le module Shelly se connecte au réseau Wi-Fi domestique et peut être contrôlé à l'aide de l'application mobile Shelly ou via des plates-formes de maison intelligente telles que Amazon Alexa ou Google Assistant.

Il est également compatible avec les protocoles MQTT et CoAP, ce qui permet une intégration facile avec d'autres systèmes domotiques.

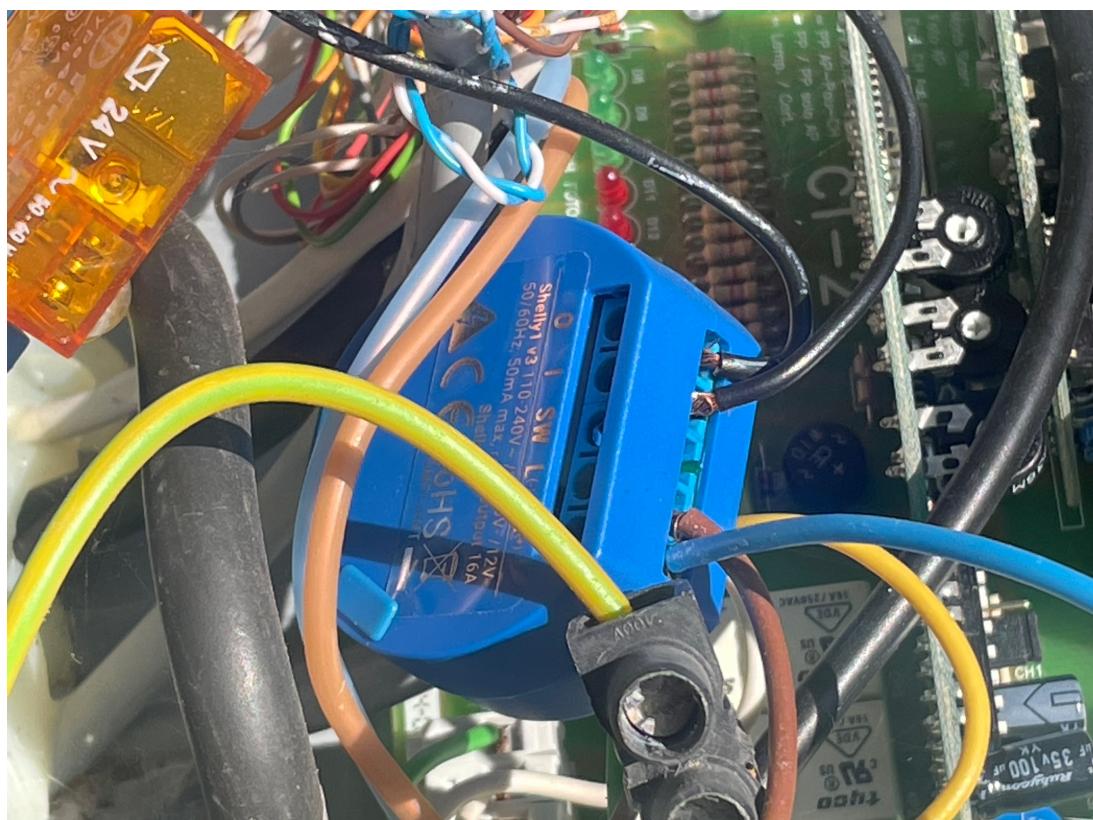
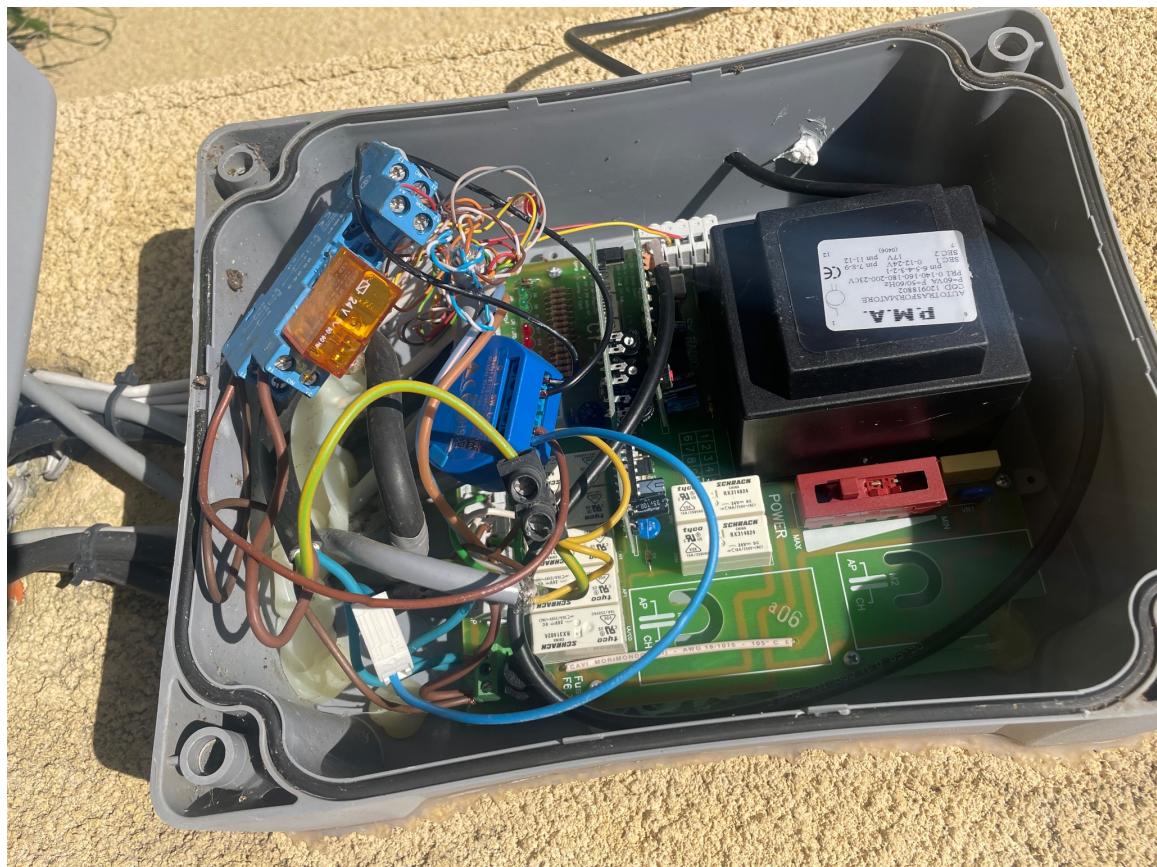
Le module Shelly est disponible dans plusieurs variantes pour répondre à différents besoins, notamment le Shelly 1 pour les interrupteurs simples, le Shelly 2.5 pour les interrupteurs doubles et le Shelly Dimmer pour les lumières à intensité variable.

Nous avons décidé d'installer un Shelly 1 permettant de gérer le contact sec directement relié au portail, au système d'alimentation et à la carte qui gère l'automatisation.

Voici à quoi ressemble l'infrastructure du projet avec les Shelly :



Voici une photo du Shelly sur notre portail :

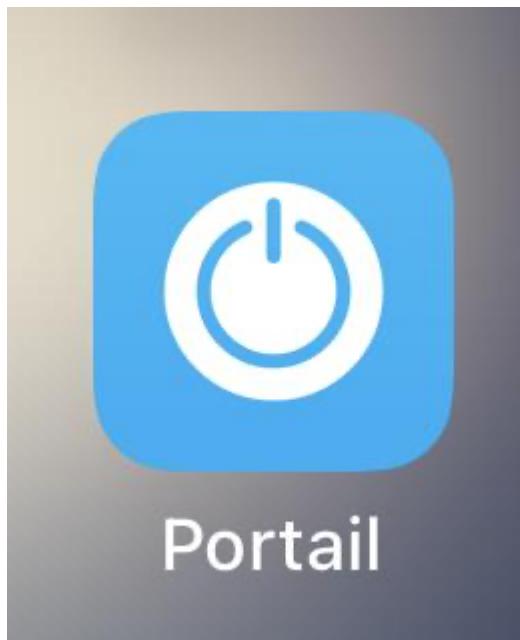


Le shelly est reconnu par la box et lui attribue une adresse IP. On peut donc en tapant l'ip privé tomber sur la page de notre module.

Nous allons voir comment actionner de différentes manières l'ouverture et la fermeture du portail.

Raccourci

Nous allons faire en sorte d'ouvrir notre portail via un raccourci sur un Iphone.
Nous allons pour cela créer un raccourci :



Par la suite, nous allons créer une requête http qui ira en direction du shelly grâce à la redirection de port.

Nous avons redirectionné le port 20000 de la box vers l'ip privé de notre shelly.

Nous avons mis dans la requête le fait d'effectuer un contact sec pour actionner l'ouverture du portail :



En effectuant cette requête, nous demandons au Shelly associé au portail de faire un contact sec et donc d'ouvrir le portail.

NFC

Pour ouvrir notre portail, nous avons aussi fait en sorte de pouvoir l'ouvrir via un badge NFC. Pour cela, nous avons enregistré la requête http sur la puce NFC. Comme nos Iphone comporte un lecteur NFC nous avons pu activer la puce NFC avec l'Iphone :



Vous retrouverez la vidéo complète du lecteur NFC dans notre montage vidéo.

Shelly cloud

Shelly Cloud en domotique fait référence à l'utilisation des dispositifs de domotique connectés Shelly, développés par la société Allterco Robotics, qui peuvent être contrôlés et gérés à distance via la plateforme cloud Shelly Cloud.

Les dispositifs Shelly comprennent une gamme de produits tels que des interrupteurs, des prises électriques, des capteurs de température et d'humidité, des détecteurs de mouvement, des modules de contrôle d'éclairage et des thermostats.

Ces dispositifs peuvent être installés dans une maison ou un bâtiment pour automatiser et contrôler diverses fonctions telles que l'éclairage, le chauffage, la climatisation, les stores, etc.

Shelly Cloud permet aux utilisateurs de contrôler leurs dispositifs Shelly à distance via une application mobile ou un navigateur web. Ils peuvent également créer des scènes pour automatiser des tâches en fonction de différents déclencheurs tels que l'heure de la journée, la température ou la présence de personnes.

En utilisant Shelly Cloud en domotique, les utilisateurs peuvent optimiser l'efficacité énergétique de leur maison ou de leur bâtiment, améliorer leur confort et leur sécurité, et économiser de l'argent sur les coûts énergétiques.

Les Shelly sont reconnus par le Cloud grâce au réseau et à un identifiant spécifique.

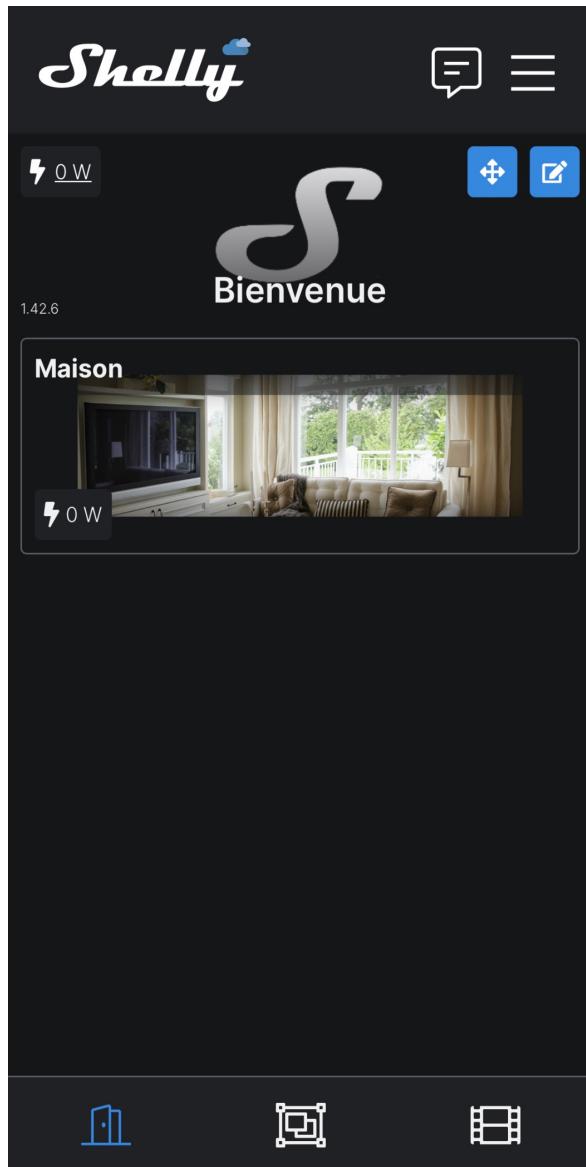
Une fois que vous avez ajouté vos appareils Shelly à votre compte Shelly Cloud, ceux-ci communiqueront avec le serveur Shelly Cloud via Internet.

Les appareils Shelly envoient des messages à intervalles réguliers pour informer le serveur Shelly Cloud de leur état actuel (allumé ou éteint, par exemple).

Le serveur Shelly Cloud peut alors utiliser ces informations pour afficher l'état de vos appareils dans l'application Shelly Cloud et vous permettre de les contrôler à distance.

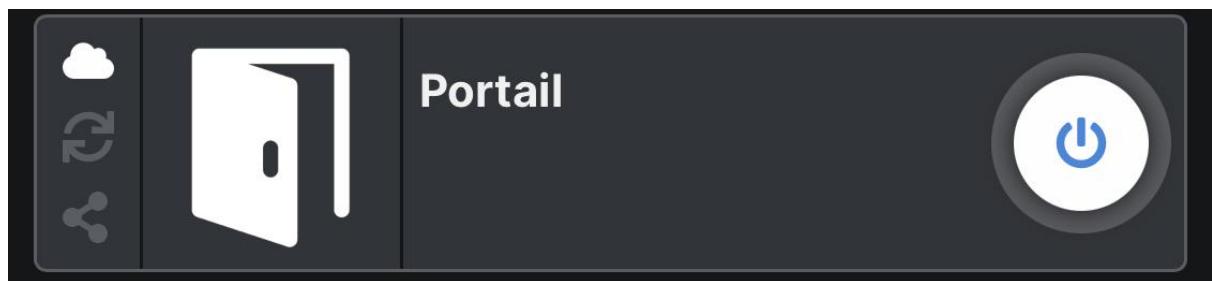
En résumé, Shelly Cloud reconnaît les appareils Shelly de votre réseau local en les ajoutant à votre compte Shelly Cloud via l'application mobile Shelly, puis en communiquant avec eux via Internet pour connaître leur état actuel.

Voici à quoi ressemble l'interface Shelly Cloud :

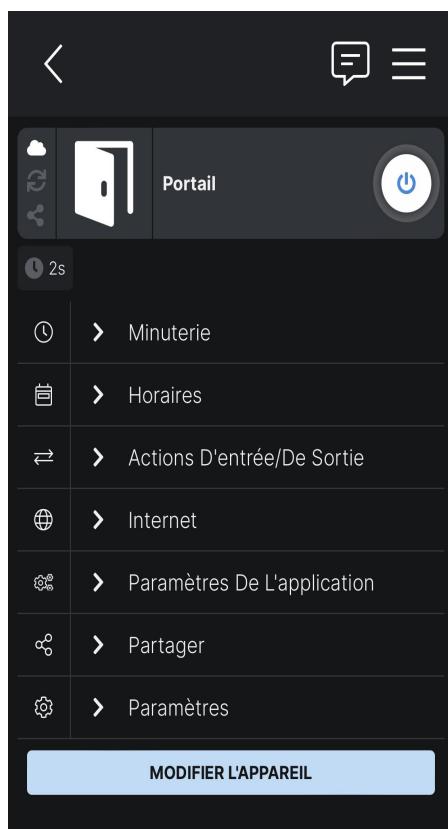


Une fois le Shelly de notre portail détecté, nous pouvons renommer l'adresse IP par le nom que l'on veut. Dans notre cas nous mettons le nom de portail :

Comme nous utilisons un Shelly 1, le bouton associé est par défaut un bouton ON / OFF



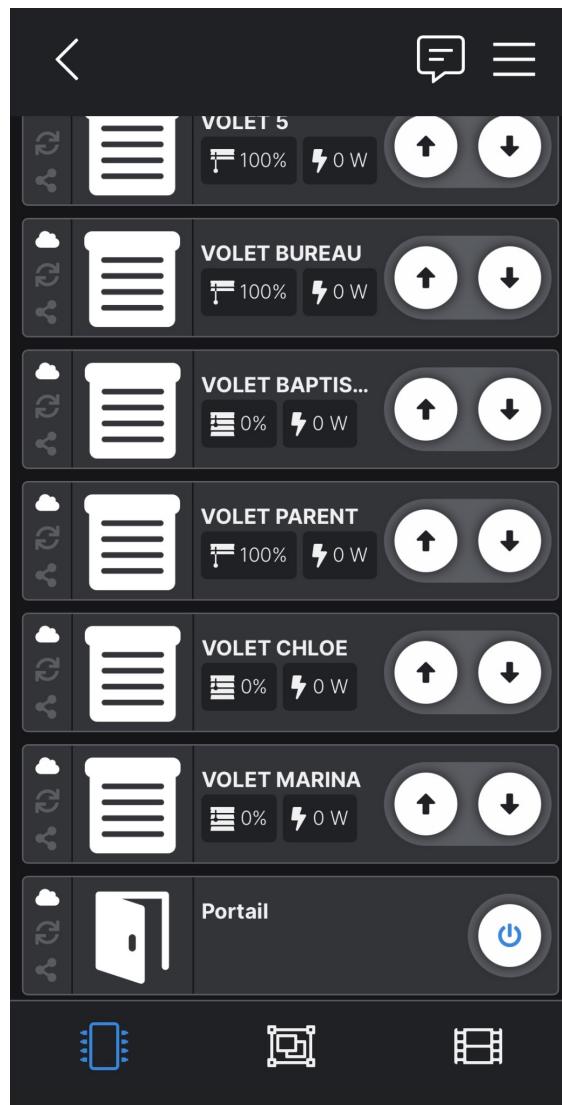
Quand l'on rentre dans notre portail nous pouvons configurer pleins d'options :



Maintenant, en appuyant sur le bouton ON / OFF, nous pouvons ouvrir ou fermer notre portail via Shelly Cloud.

Comme il nous restait des Shelly 1 et 2.5, nous avons décidé de les intégrer dans la maison pour effectuer différentes actions.

Nous avons réussi à monter et baisser les volets de la maison grâce au Shelly 2.5 :





IOT

