

Epreuve de projet en enseignement spécifique à la spécialité SIN

PROJET SERRURE SANS CONTACT



DOUTEAU Romain

Lycée François Rabelais - 45 Rue Rabelais - 85200 Fontenay-le-Comte

Baccalauréat technologique STI2D | Session 2015 - 2016

Sommaire

1– Présentation

- Pourquoi avoir choisi une serrure sans contact ?
- Principe de fonctionnement d'une serrure sans contact
- Produits existants

2– Analyse du besoin

- Diagramme bête à corne
- Diagramme pieuvre
- Tableau des contraintes
- Diagramme des exigences Sys MI

3– Solutions technologiques

- Solutions matérielles
- Solution matérielle choisie
- Conditions de fonctionnement du système
- Solutions logicielles choisies
- Architecture matérielle

4– Pilotage du système

- Tâches à réaliser
- Planification
- Câblage du système
- Programme contrôlant la gâche, les LEDS et le buzzer
- Programme permettant la réception de données par Bluetooth
- Programme permettant la réception de données par Ethernet
- Programme permettant l'envoi de données par Ethernet

5– Notice d'utilisation

- Installation du système
- Utilisation du système

6– Conclusion

- Aboutissement du projet
- Remerciements

1– Présentation

- Pourquoi avoir choisi une serrure sans contact ?

Au jour d'aujourd'hui, chaque salle du lycée François Rabelais s'ouvre avec une clé pouvant dans le cas des nouveaux bâtiments coûter jusqu'à 80 € l'unité. Nous nous sommes dit que limiter le nombre de clés permettrait de réaliser des économies. Bien que le lycée dispose de clés pass pouvant ouvrir plusieurs portes, les agents d'entretien ont besoin pour accéder à toutes les salles du lycée de beaucoup de clés pouvant ainsi créer des encombrements, ils doivent également chaque soir faire le tour du lycée afin de vérifier que chaque porte est fermée et verrouillée. Nous avons donc dans le cadre de ce projet proposé des solutions à tous ces problèmes.

Implanté sur chaque porte, notre projet permettrait une gestion complète de chaque salle de chaque bâtiment.

- Principe de fonctionnement d'une serrure sans contact

La contrainte première de notre projet est de dé(verrouiller) une porte à proximité ou à distance sans contact physique avec celle-ci. Sachant qu'aujourd'hui beaucoup de personnes possèdent un smartphone, nous avons trouvé judicieuse l'idée d'utiliser une application smartphone afin de contrôler les portes.



Pour rendre service aux agents, nous avons également pensé que pouvoir visualiser et contrôler l'état des portes à distance grâce à un ordinateur via une page internet leur rendraient un grand service.

- Produits existants

Il existe de nombreuses serrures connectées avec chacune leurs avantages, mais avec un inconvénient commun : leur prix. En effet pour acquérir cette technologie, il faut déboursier entre 150 € et 350 € (l'unité) en fonction des technologies utilisées, il faut à tout cela ajouter le prix de l'application permettant de l'utiliser et parfois un abonnement mensuel. C'est donc un produit domotique très coûteux dont beaucoup de personnes ne peuvent profiter. Nous avons donc décidé de reprendre ce projet afin de le rendre moins onéreux.



Serrure « august » : 230 €



Serrure « Genie Smart Lock » : 185 €

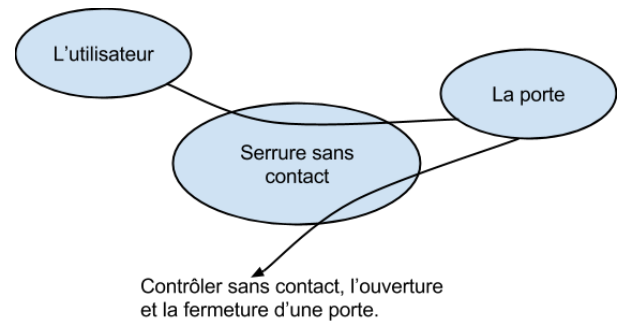


Serrure « Weiser » : 250 €

2– Analyse du besoin

- Diagramme bête à corne

La serrure sans contact rend service à son utilisateur et agit sur la porte en dé(verrouillant) celle-ci.



- Tableau des contraintes

Classes	Flexibilités	Niveaux
F0	nulle	impératif
F1	faible	peu négociable
F2	moyenne	négociable
F3	forte	très négociable

Ce tableau indique les niveaux de flexibilité possible des différentes contraintes

Fonctions de services	Critères d'appréciation	Niveaux d'appréciation	Flexibilité
FP1 Contrôler sans contact, l'ouverture et la fermeture d'une porte grâce à un smartphone.	Piloter à proximité le système	Piloter l'ouverture et la fermeture	F0
	Visualiser à proximité l'état du système	Information visuelle et sonore de l'état du système (Verrouillé/Déverrouillé)	F3
	Choix de la porte à déverrouiller (choix de la salle)	Doit être présent	F0
	Piloter le système (carte + programme)	Doit être présent	F0
FP2 Contrôler sans contact, l'ouverture et la fermeture d'une porte grâce à une page internet	Piloter à distance le système	Piloter l'ouverture et la fermeture	F0
	Visualiser à distance l'état du système	Position de la porte (Ouvverte/Fermée)	F3
	Choix de la porte à déverrouiller (choix de la salle)	Doit être présent	F0
	Piloter le système (carte + programme)	Doit être présent	F0
FC1 Rendre le système autonome en énergie	Alimentation par batterie	Doit être présente	F0
	Tension d'alimentation du système	Tension: 5-12V DC	F1
	Autonomie du système	Autonomie: 24h	F3
FC2 Respecter un budget donné	Coût	maximum: 300€	F2
FC3 Ouvrir la porte avec un Smartphone	Distance pour détecter le Smartphone	minimum: 0 cm maximum: 5 cm	F2
FC4 Respecter les normes de sécurité	Ouverture mécanique (manuelle) depuis l'intérieur	Doit être présente	F0
FC5 S'adapter à la porte (au niveau de la serrure)	Dimensions	Hauteur: 2,9 cm Largeur: 1,2 cm	F1 F1
FC6 Pouvoir contrôler et (Dé) verrouiller serrure à distance	Contrôle à partir d'un poste relié à internet	Page internet hébergée par un serveur distant	F0
	Programmation langage html, PHP	Doit être présente	F0
FC7 Pouvoir commander le système à proximité	Programmation en langage Scratch	Doit être présente	F0
	Temps maximal de déverrouillage	+/- 15 secondes	F3
FC8 Créer un système de taille raisonnable	Dimension max (L x l x h)	30 cm * 30 cm * 10 cm	F3

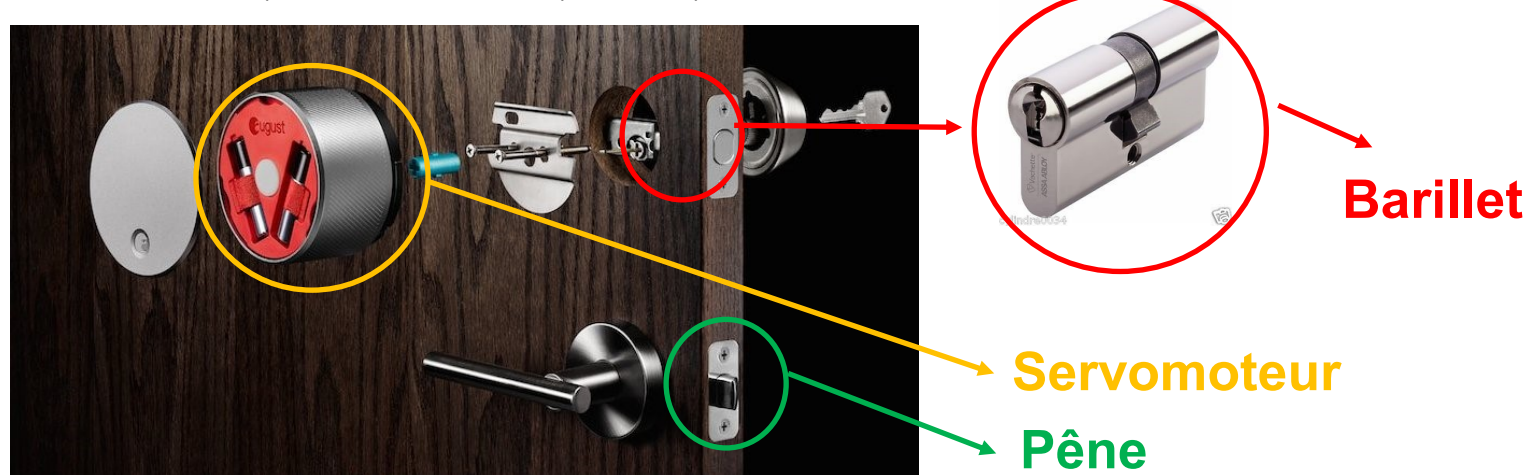
Les contraintes encadrées sont celles dont j'ai traité en particulier.

3– Solutions technologiques

- Solutions matérielles

Lors de nos recherches sur les solutions technologiques matérielles envisageable pour créer notre système nous avons trouvé deux possibilités :

- L'une avec un servomoteur qui viendrait actionner mécaniquement le barillet (partie permettant de fermer la porte à clé) de la serrure pour dé(verrouiller) la porte.



- L'autre avec une gâche électrique retenant le pêne de la porte (pièce de métal que la poignée fait entrer et sortir, dont l'extrémité est bloquée dans la gâche lorsque la porte est fermée) qui viendrait se bloquer ou se débloquer pour dé(verrouiller) la porte.



- Solution matérielle choisie

Pour notre système, nous avons choisi la gâche électrique car c'est la seule solution qui nous permettait de respecter les normes de sécurité régies par le code des ERP (Etablissements Recevant du Public) impose que "les bâtiments et les locaux où sont installés les ERP soient construits de manière à permettre l'évacuation rapide de la totalité des occupants en évitant la panique." il est donc primordial de pouvoir ouvrir la porte manuellement depuis l'intérieur de chaque salle sans problème. De plus, il suffit pour la déverrouiller de la mettre sous tension ce qui simplifie son contrôle.

Pour la carte électronique, nous avons choisi une carte « Arduino » car ce type de carte se programme en langage C/C++ qui sont deux langages de très haut niveau, faciles à apprendre et à comprendre. De plus, ce type de carte est très utilisée pour le prototypage de projets. Nous avons choisi le modèle « Arduino UNO » car c'est la carte de type Arduino la moins coûteuse et la mieux adaptée à notre système.

- Conditions de fonctionnement

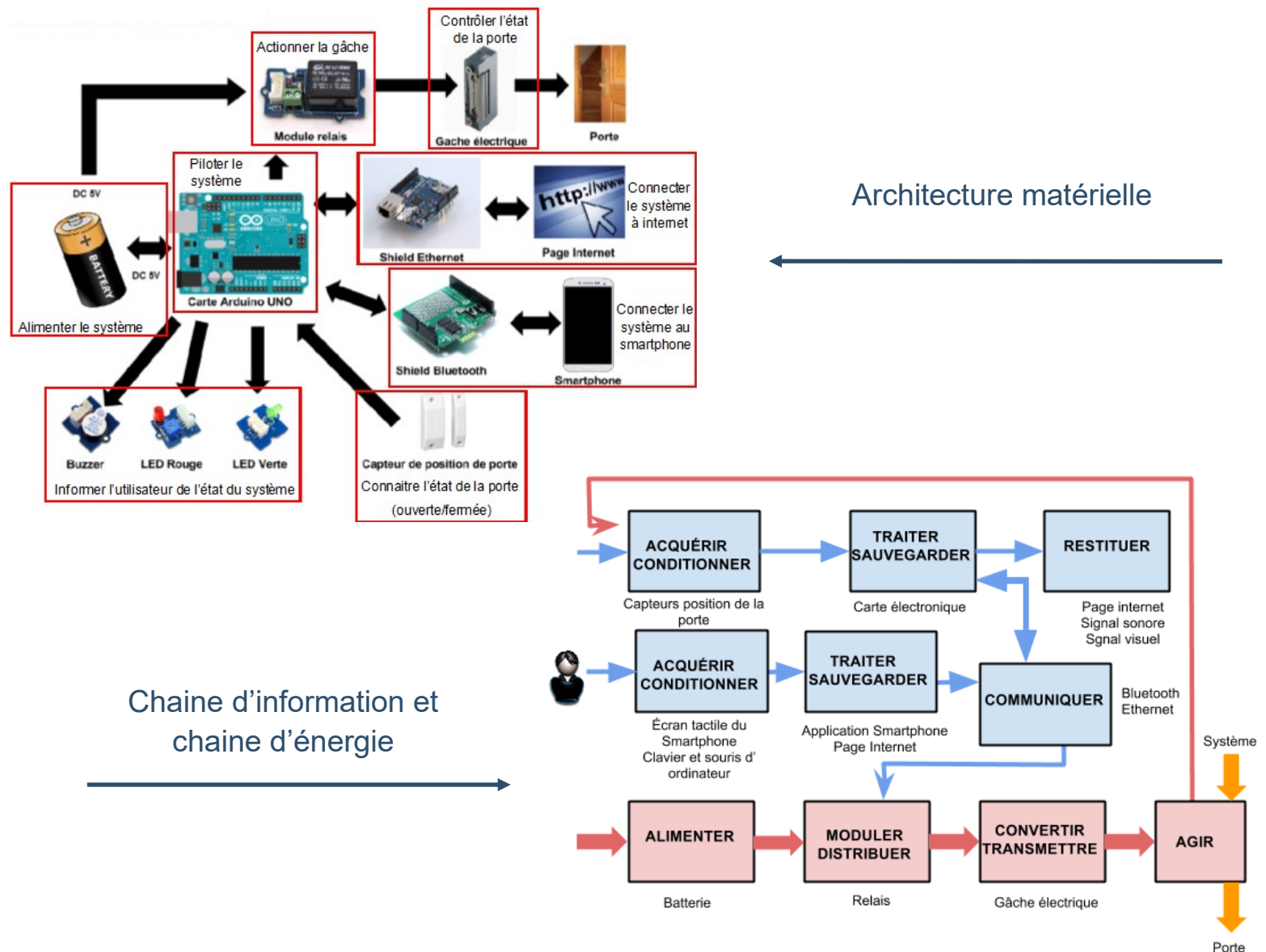
Le choix de la gâche électrique impose tout de même des contraintes pour que le système fonctionne :

- La poignée extérieure de la porte ne doit plus actionner le pêne.
- La porte ne doit pas être fermée à clé.

- Solution logicielle choisie

Nous avons pour la connexion entre le système et le smartphone comparé la technologie NFC (Near Field Communication) avec la technologie Bluetooth qui sont toutes deux des technologies de liaisons sans fil, nous nous sommes rendu compte après des tests d'envoi et réception d'informations que la technologie NFC était moins fiable que la technologie Bluetooth. Nous avons pu observer qu'avec le NFC, il y avait des pertes d'informations ainsi que des erreurs de connexion. C'est pourquoi nous avons choisi d'utiliser la technologie Bluetooth.

Pour la connexion entre le système et internet, nous avons cette fois-ci comparé la technologie Wi-Fi avec la technologie Ethernet et c'est pour des raisons budgétaires que nous avons sélectionné la technologie Ethernet. En effet, pour une carte « Arduino UNO », le prix d'un shield Wi-Fi est d'environ 80 € contre seulement 23 € pour le shield Ethernet.



La carte Arduino ne délivre pas assez de courant (intensité trop faible) pour actionner la gâche électrique, il faut donc utiliser une alimentation externe. Dans ce cas, nous devons utiliser un relais qui fonctionne comme un interrupteur contrôlé électroniquement qui laissera passer ou non l'électricité de l'alimentation externe pour piloter la gâche.

Pour économiser l'énergie, nous avons choisi une gâche électrique qui se verrouille lorsqu'elle est hors tension, elle est néanmoins dotée d'un sélecteur mécanique de déverrouillage constant.

- Tâches à réaliser

Les différentes tâches que j'avais à réaliser afin de rendre concret ce projet sont:

- Créer un programme contrôlant les LEDS, le buzzer et la gâche.
- Créer un programme permettant de recevoir des informations par Bluetooth
- Créer un programme permettant de recevoir des informations par Ethernet
- Créer un programme permettant d'envoyer des informations par Ethernet

- Planification

Par soucis d'enregistrement, nous n'avons pu garder la première version de notre diagramme GANNT nous avons donc changer les années (2020 pour 2015 et 2021 pour 2016) car le logiciel ne nous permettait pas d'affecter des actions à des dates passées.

i	Nom	Durée	Début	Fin	Jan 18 - Jan 24 '21							Jan 25 - Jan 31 '21							Fév 1 - 1	
					L	M	M	J	V	S	D	L	M	M	J	V	S	D	L	M
1	☐ Commander une serrure sans contact	170j	08/09/2020	03/05/2021																
2	☐ Analyse	89j	08/09/2020	08/01/2021																
12	☐ Conception	39j	12/01/2021	05/03/2021																
13	☐ Pilotage de la serrure a distance	31j	12/01/2021	23/02/2021																
22	☐ Pilotage de la serrure a proximité	36j	12/01/2021	02/03/2021																
30	☐ Commande de la carte électronique	31j	12/01/2021	23/02/2021																
31	☐ Création d'un programme permettant la communication avec et la page internet	21j	26/01/2021	23/02/2021																
32	☑ Réalisation d'un programme afin d'envoyer les données sur la page internet	1j	02/02/2021	02/02/2021																
33	☑ Réalisation d'un programme afin de recevoir les données de la page internet	4j	26/01/2021	29/01/2021																
34	☑ Test de la communication avec la page internet	13j	05/02/2021	23/02/2021																
35	☐ Création d'un programme permettant la communication avec l'application smartphone	26j	19/01/2021	23/02/2021																
36	☑ Réalisation du programme afin de recevoir les données du smartphone	4j	19/01/2021	22/01/2021																
37	☑ Test de la communication avec l'application smartphone	13j	05/02/2021	23/02/2021																

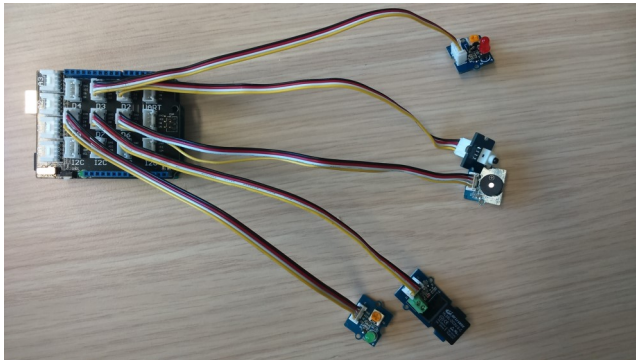
J'ai commencer créer les programmes contrôlant la gâche, les LEDS ainsi que le buzzer afin de pouvoir par la suite actionner ces différents composants en fonction des données reçues par Bluetooth ou Ethernet

	i	Nom	Durée	Début	Fin	11 - Jan 17 '21				
						M	M	J	V	
38	✓	☐ Création d'un programme contrôlant la gâche électrique	31j	12/01/2021	23/02/2021					
39	✓	☐ Réalisation du programme contrôlant la gâche	4j	12/01/2021	15/01/2021					
40	✓	☐ Test de la gâche avec un bouton	1j	15/01/2021	15/01/2021					
41	✓	☐ Test de la gâche à partir des données reçues de la page Internet	13j	05/02/2021	23/02/2021					
42	✓	☐ Test de la gâche à partir des données reçues de l'application sur le Smartphone	13j	05/02/2021	23/02/2021					
43	✓	☐ Création d'un programe contrôlant le buzzer	31j	12/01/2021	23/02/2021					
44	✓	☐ Réalisation d'un programme contrôlant le buzzer	4j	12/01/2021	15/01/2021					
45	✓	☐ Test du buzzer avec un bouton	1j	15/01/2021	15/01/2021					
46	✓	☐ Test du buzzer à partir de données reçues de l'application sur le Smartphone	13j	05/02/2021	23/02/2021					
47	✓	☐ Test du buzzer à partir des données reçues de la page Internet	13j	05/02/2021	23/02/2021					
48	✓	☐ Création d'un programme contrôlant les LEDS	31j	12/01/2021	23/02/2021					
49	✓	☐ Réalisation d'un programme contrôlant les LEDS	1j	12/01/2021	12/01/2021					
50	✓	☐ Test des LEDS avec un bouton	1j	15/01/2021	15/01/2021					
51	✓	☐ Test des LEDS à partir de données reçues de la page Internet	13j	05/02/2021	23/02/2021					
52	✓	☐ Test du LEDS à partir de données reçues de l'application sur le Smartphone	13j	05/02/2021	23/02/2021					

Ensuite, j'ai créer le programme permettant de recevoir et exploiter les données reçues par Bluetooth en provenance du smartphone puis celui pour recevoir et traiter les données reçues par Ethernet venant cette fois-ci de la page internet. J'ai continuer par réaliser le programme permettant d'envoyer la donnée du capteur de position de porte par Ethernet sur la page internet. Pour terminer, j'ai assembler tout ces programmes afin d'en avoir un seul complet pour rendre le système entièrement fonctionnel.

- Câblage du système

Le tableau ci-contre nous montre l'utilité des différentes broches de la carte « Arduino UNO ».



Broches	Utilité / Composant
0 - 1	Liaison série USB
2	Commande LED rouge
3	Commande LED verte
4 - 5	Liaison série Bluetooth
6 (analogique)	Commande Buzzer
7	Commande Relais (Commande gâche électrique)
10 ICSP 1 - 2 - 3	Liaison Ethernet

- Programme contrôlant la gâche, les LEDS et le buzzer

```
1  const int pinBuzzer=6;
2  const int pinLedRouge=3;
3  const int pinLedVerte=8;
4  const int pinGache=7;
5  const int pinBouton=2;
```

J'ai en premier lieu initialiser les constantes correspondantes aux broches des composants.

```
11  pinMode(pinBuzzer,OUTPUT);
12  pinMode(pinLedRouge,OUTPUT);
13  pinMode(pinLedVerte,OUTPUT);
14  pinMode(pinGache,OUTPUT);
15  pinMode(pinBouton,INPUT);
```

Puis j'ai déclaré l'état des composants (Entrées/ Sorties).

Ce programme sert à déverrouiller la gâche lorsqu'une variable est égale à une certaine valeur (ici si la variable *donnee* est égale à **HIGH**, soit 1 en binaire).

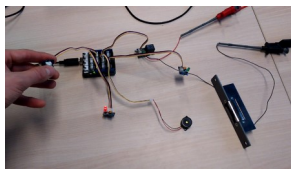
```
22  if(donnee==HIGH) {
23      digitalWrite(pinLedVerte,HIGH);
24      digitalWrite(pinLedRouge,LOW);
25      digitalWrite(pinGache,HIGH);
26      analogWrite(pinBuzzer,100);
27      delay(200);
28      analogWrite(pinBuzzer,0);
29      delay(5000);
30      analogWrite(pinBuzzer,100);
31      delay(200);
32      analogWrite(pinBuzzer,0);
33      delay(100);
34      analogWrite(pinBuzzer,100);
35      delay(200);
36      analogWrite(pinBuzzer,0);}
37
38  else{
39      digitalWrite(pinLedVerte,LOW);
40      digitalWrite(pinLedRouge,HIGH);
41      digitalWrite(pinGache,LOW);
42      analogWrite(pinBuzzer,0);}
```

Si la variable *donnee* est égale à **HIGH**, le programme allume la LED verte (l.23), éteint la LED rouge (l.24), déverrouille la gâche (l.25), puis allume le buzzer une fois durant 200 ms (l.26-28).

Ensuite, le programme attend 5 s afin que l'utilisateur ait le temps d'ouvrir la porte (l.29) puis allume le buzzer deux fois durant 200 ms à intervalle de 100 ms pour prévenir du reverrouillage de la porte (l.30-36).

Si la variable *donnee* n'est pas égale à **HIGH**, le programme éteint la LED verte (l.39), allume la LED rouge (l.40), verrouille la gâche (l.41), puis éteint le buzzer (l.42).

Résultats :



Appui sur le bouton



Gâche déverrouillée



5 s



Gâche verrouillée

On remarque que le programme fonctionne car dès que l'on appuie sur le bouton, la gâche se déverrouille aussitôt et se reverrouille 5s plus tard. On remarque également que les LEDS et le buzzer fonctionnent correctement.

- Programme permettant la réception de données par Bluetooth

J'ai commencer par définir les broches utilisées par le shield Bluetooth pour envoyer (TX) et recevoir (RX) des informations (l.27-28) pour ensuite initialiser (l.31) et démarrer (l.49) cette liaison afin de l'utiliser.

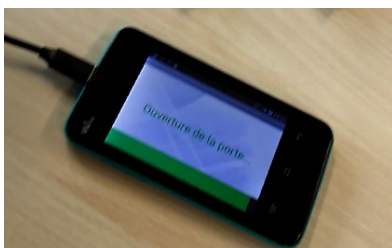
```
26 //Définition des broches nécessaires pour la communication bluetooth
27 #define BT_TX 4
28 #define BT_RX 5
29
30 //Initialisation de la liaison Bluetooth
31 SoftwareSerial BTSerial(BT_RX, BT_TX);
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47 //Démarrage de la liaison série bluetooth entre la carte Arduino et le smartphone
48 BTSerial.begin(9600);
49
```

Le programme vérifie si la liaison Bluetooth est disponible (en clair que le smartphone est connecté au shield) et récupère l'informations pour la stocker dans la variable `dataB` si c'est le cas.

```
112 if (BTSerial.available()) {
113     dataB = BTSerial.read();
114 }
```

Une fois les données reçues, il ne reste qu'à les traiter afin de rendre le système entièrement fonctionnel.

Résultats :



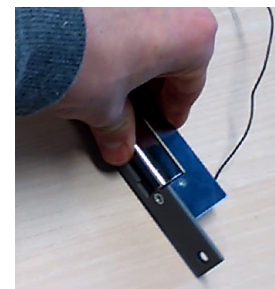
Déverrouillage par l'application



Gâche déverrouillée



5 s



Gâche verrouillée

On remarque que le programme ainsi que l'application smartphone fonctionnent car dès qu'on suis la procédure pour déverrouiller la gâche, la gâche se déverrouille dès que la barre de progression pas à pas de l'application est arrivée au bout puis se reverrouille 5s après.

- Programme permettant la réception de données par Ethernet

J'ai commencer par entrer l'adresse MAC du shield Ethernet (l.34) ainsi que l'adresse URL du serveur (l.37) avant d'initialiser la connexion Ethernet (l.40).

```
33 //Définition de l'adresse MAC du shield Ethernet
34 byte mac[] = {0x90, 0xA2, 0xDA, 0x10, 0x09, 0x99};
35
36 //Définition de l'adresse IP du serveur alwaysdata
37 char server[] = "texierelevessinrabelais.alwaysdata.net";
38
39 //Initialisation de la liaison Ethernet
40 EthernetClient client;
```

J'ai ensuite créer un programme qui test si la connexion avec le serveur est disponible et récupère les informations sur la fichier « data.txt » internet grâce à une requête HTTP (l.225-226) si c'est le cas.

```
221 //Récupération de la donnée de deverrouillage provenant du serveur
222 if (client.connect(server,80)) {
223     Serial.println("connexion OK");
224     //Faire une requette HTTP visant un fichier texte pour récupérer la donnée de deverrouillage sur le serveur
225     client.println("GET /data.txt HTTP/1.1");
226     client.println("Host: texierelevessinrabelais.alwaysdata.net");
227     Serial.println("Connexion avec le serveur etablie, dataE recuperee");
228     client.println();}
```

Une fois les données reçues, il ne reste une fois de plus qu'à les traiter (partie de programme ci-dessous) afin de rendre le système entièrement fonctionnel.

Cette partie de programme permet de traiter l'information reçue précédemment de la page internet afin d'extraire la valeur nécessaire au déverrouillage de la gâche.

```
149 //Recuperation caractères par caractères de toutes les donnees contenues dans le fichier texte
150 while (client.available()) {
151     caractere = client.read();
152     txt = txt + caractere;
153     //Détermination de la position de la donnée que l'on souhaite récupérer
154     positiondata = txt.indexOf("DATA");
155     debutdata = positiondata+5;
156     findata = debutdata+1;
157     data = txt.substring(debutdata, findata);
158     //Conversion de la chaine de caractères contenant la donnée en entrée numérique
159     dataE = data.toInt();}
```

On commence par récupérer les données du fichier « data.txt » caractères par caractères que l'on stocke dans une chaine de caractère « txt » (l.152) puis on détermine la position du premier caractère du mot « DATA » sur le fichier « data.txt » (l.154). Ensuite on détermine la position du premier et du dernier caractère de la chaine de caractère « txt » que l'on souhaite extraire (l.155-156) puis on les stocke dans une nouvelle chaine de caractère « data » contenant uniquement la donnée que l'on souhaite utiliser (l.157). Enfin, on convertit la chaine de caractère « data » en variable numérique « dataE » (l.159). Cette variable « dataE » correspond à la donnée de déverrouillage de la gâche par internet.

Résultats :

Nous pouvons voir sur les résultats du moniteur série que l'on arrive bien à recevoir les différentes valeurs en provenance de la page internet (1 à gauche et 0 à droite).

```
connexion...
connexion OK
Connexion avec le serveur etablie, dataBtn envoyee : 0
connexion...
connexion OK
Connexion avec le serveur etablie, dataE recuperee : 1
```

```
connexion...
connexion OK
Connexion avec le serveur etablie, dataBtn envoyee : 0
connexion...
connexion OK
Connexion avec le serveur etablie, dataE recuperee : 0
```

- Programme permettant l'envoi de données par Ethernet

Ce programme commence par affecter à la variable *dataCaptPos* la valeur du capteur de position de porte (I.238) et envoie celle-ci à l'adresse du serveur grâce à une nouvelle requête HTTP (I.248-251)

```
237 //Lecture de l'etat du capteur de position
238 dataCaptPos = digitalRead(pinCapteurPosition);

244 //Envoie de la donnée du capteur de position sur le serveur
245 if (client.connect(server,80)) {
246   Serial.println("connexion OK");
247   //Faire une requette HTTP visant un script php pour envoyer la donnee du capteur de position sur le serveur
248   client.print("GET /posporte.php?DATA=");
249   client.print(dataCaptPos);
250   client.println(" HTTP/1.1");
251   client.println("Host: texierelevessinrabelais.alwaysdata.net");
252   Serial.print("Connexion avec le serveur etablie, dataCaptPos envoyee : ");
253   Serial.println(dataCaptPos);
254   client.println();}
```

Une fois les données envoyées sur le serveur, il est possible de visualiser à distance l'état de la porte

Résultats :

Nous pouvons voir sur les résultats du moniteur série que l'on arrive bien à envoyer les différentes valeurs du capteur de position de porte sur la page internet (1 à gauche et 0 à droite).

```
connexion...
connexion OK
Connexion avec le serveur etablie, dataBtn envoyee : 0
connexion...
connexion OK
Connexion avec le serveur etablie, dataE recuperee : 1
```

```
connexion...
connexion OK
Connexion avec le serveur etablie, dataBtn envoyee : 1
connexion...
connexion OK
Connexion avec le serveur etablie, dataE recuperee : 0
```

5– Notice d'utilisation

- Installation du système

Pour installer le système sur votre porte :

1– Retirez la gâche présente sur le montant de votre porte.



2– Creusez la partie du montant retenant la pêne.



3– Remplacez la gâche par une gâche électrique et installez le capteur de position sur la porte pour qu'il soit enclenché lorsque la porte est fermée.



4– Branchez la gâche électrique au relais et le capteur de position directement sur la carte « Arduino » du système et fixez le système sur la porte.



- Utilisation du système

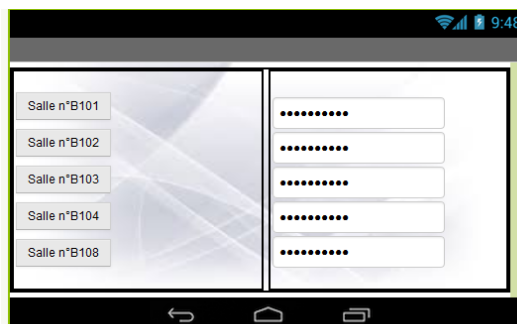
Avec l'application smartphone par Bluetooth :

1– Lancez l'application sur votre smartphone puis patientez, la page ci-dessous va s'afficher.

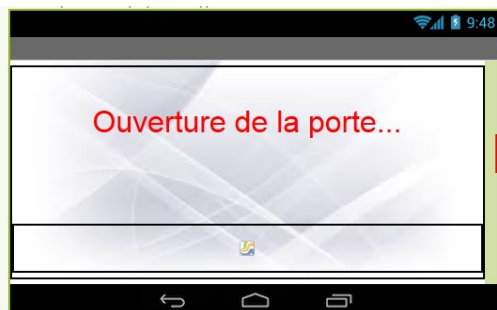


2– Pour quitter l'application, appuyez sur « Close » en bas à droite de l'écran.

– Pour déverrouiller une salle, appuyez sur « Choix de la salle » puis patientez, la page ci-dessous va s'afficher.

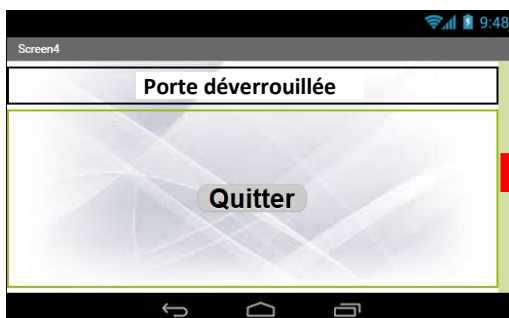


3– Appuyez sur la salle à déverrouiller puis tapez le mot de passe correspondant à celle-ci (si le mot de passe tapé est correct, la case de saisie du mot de passe deviendra verte et la page ci-dessous s'affichera, sinon, elle deviendra rouge et vous devrez retaper le mot de passe).



4– La gâche électrique va se déverrouiller, une barre de progression pas à pas s'affichera afin de suivre l'évolution du déverrouillage de la porte.

5– Lorsque la barre arrive au bout, la porte se déverrouille et la page suivante s'affiche.



6– Appuyez sur « Quitter » pour quitter l'application.

Avec la page Internet par Ethernet :

1– Ouvrez la page internet sur votre ordinateur (<http://texierelevesinrabelais.alwaysdata.net/systemeporte.html>).



2– Appuyez sur le bouton à coté de la salle de votre choix verrouiller ou déverrouiller celle-ci.

P.S : Le système ne dispose par d'un reverrouillage automatique par Ethernet, il faut donc une fois la salle déverrouillée la reverrouiller manuellement en cliquant sur « Verrouiller ».

6– Conclusion

- Aboutissement du projet

Nous avons réaliser à la fin de la conception du système des tests de fonctionnement qui nous ont confirmé que le système est fonctionnel, toutes les contraintes imposées par le cahier des charges sont validées le programme principal fonctionne correctement, bien que nous avons parfois des problèmes de connexion dus à la configuration du serveur du lycée. Ces problèmes auraient pu être facilement corrigés en hébergeant la page internet ainsi que tout les autres fichiers qui permettent de faire fonctionner notre système sur le serveur local du lycée. Cela permettrait d'éviter tout les conflits dus aux systèmes de filtrages du serveur en ce qui concerne les connexions à internet depuis l'intérieur du lycée; cela gagnerais également du temps lors de la réception et de l'envoi de données par Ethernet. De plus, il serait possible d'améliorer le système en ajoutant une fonction de reverrouillage automatique de la gâche électrique lorsqu'elle est déverrouiller via la page internet.

En ce qui concerne le coût de notre projet, il nous reviens au total à 98,24 € (sans la batterie). Il est donc moins couteux que toute les autres serrures connectées disponible sur le marché.

- Développement durable

Notre système est peu couteux d'une part car il évite d'acheter des clés pour tout le monde et d'autres part car notre serrure sans contact est moins cher que toutes les autres disponibles sur le marche. C'est donc un bon point en ce qui concerne les enjeux économiques. De plus grâce à notre système l'utilisateur n'est plus obligé d'avoir ses clés en permanence avec lui ce qui permet d'éviter les pertes ainsi que les encombrements. Notre système contribue à la partie concernant les enjeux sociaux du développement durable.

En revanche, le cycle de vie du système à plus d'impact sur l'environnement que celui des clés d'une part car il consomme de l'énergie électrique et d'autres part car l'extraction des matières premières, la fabrication ainsi que la valorisation des composants qui composent notre système consomme beaucoup d'énergie contrairement aux clés qui ne sont que de simples morceaux de métal.

- Remerciements

Je remercie M. ANTONY notre professeur de SIN pour nous avoir suivi et guidé dans nos démarches tout au long de cette année ainsi que les autres membres de mon groupe sans qui notre projet n'aurait probablement pas vu le jour.