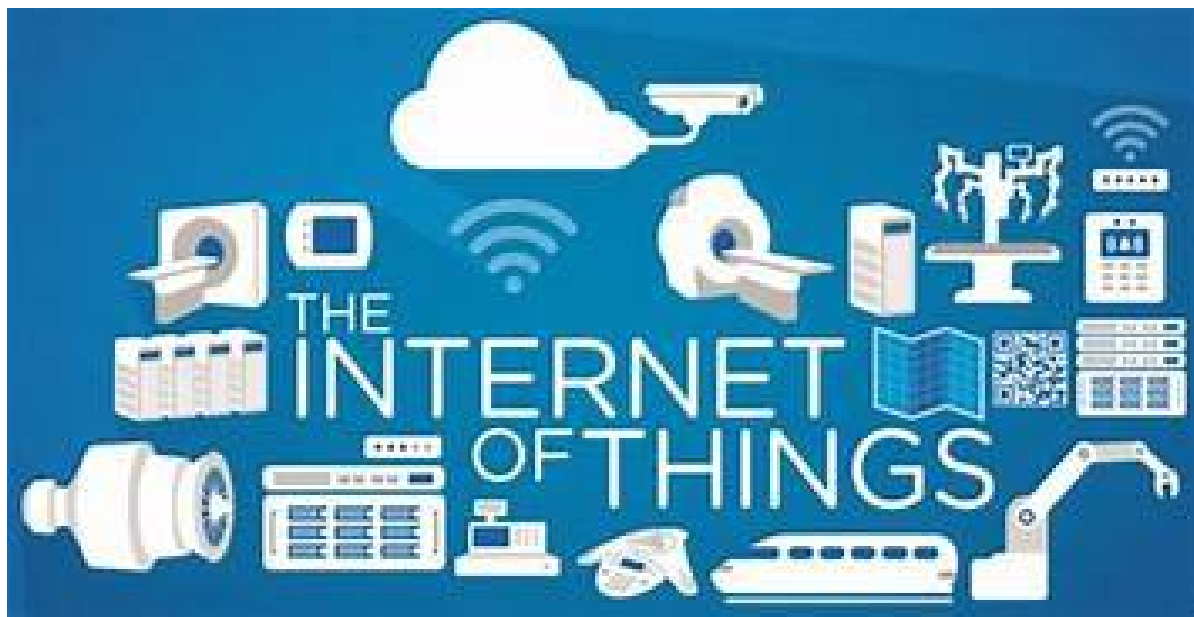


Internet Of Things

Hanae Seddiki
Nissrine Chaouqi

Rapport de projet : Maison fonctionnelle



Enseignant référent/ Chargé de TPs : Aomar Osmani, Hamidi Massinissa

Master 1 informatique semestre 2

SOMMAIRE

1. Introduction.....	3
2. Base de travail.....	3
2.1. Arduino.....	3
2.2. IDE Arduino.....	3
3. Fonctionnalités intégrés dans la maison.....	4
3.1. Ouverture du portail.....	4
3.2. Ouverture du garage avec un ServoMotor.....	6
3.3. Système de température/humidité.....	7
3.4 Lumière interagissant avec le Bluetooth.....	9
3.5 Ouverture de la porte d'entrée par carte RFID.....	11
3.6 Système d'alarme.....	13
4. Structure de la maison.....	15
5. Difficultés rencontrées.....	17
6. Manuel d'utilisation.....	18
7. Conclusion.....	18
8. Sources.....	18
9.Où trouver les composants	19

« Se réunir est un début, rester ensemble est un progrès, travailler ensemble est la réussite. », Henry Ford, 1922.

1. Introduction

Cette citation a pris tout son sens durant la production de ce projet.

Nous avons longtemps hésiter avant d'être sur de vouloir travailler sur le projet de la maison connectée. En effet, nous ne savions pas si on allait se pencher sur l'aspect « sécurité », « pratique », ou tout simplement « gadget superficiel » de la maison. Alors, chacune de notre coté avons fait des recherches. L'indécision ayant pris dessus, nous avons alors décidé d'allier toutes nos idées en un seul projet, à savoir, celui de la maison connectée qui comprendrait plusieurs fonctionnalités ; Une fonctionnalité dite de sécurité (code d'accès), une fonctionnalité portant sur le coté pratique de la maison (ouverture du portail ou du garage) et enfin une fonctionnalité « superficielle » permettant de contrôler les lumières de chez soi. Le but final a été de faire quelque chose que nous pourrions réutiliser dans la vie quotidienne.

La ligne directive de notre travail a porté sur le fait de commencer par faire des choses très simples et élémentaires dans le domaine de l'électronique puis de monter progressivement dans la complexité. Par ailleurs, nous avons mis l'appuie sur un point tout particulier qui est celui de l'esthétique.

Dans un premier le but a été de partir d'une base des plus simple possible, puis de compliquer les choses au fur et à mesure. Nous avons commandés chacune un kit Arduino mega 2560, pour pouvoir s'exercer et tester nos trouvailles.

2. Base de travail

2.1 ARDUINO :

Un Arduino est ce que l'on appelle un microcontrôleur c'est à dire un circuit intégré capable de recevoir de l'information comme un capteur de position ou encore un capteur de température. En fonction de ces informations ce circuit intégré va contrôler d'autres composants électroniques comme un moteur ou encore un écran LCD. Toutes ces interactions sont permises via une interfaces de programmation simplifiée.

2.2 IDE ARDUINO

Nous avons principalement travailler sur l'IDE Arduino. Qu'est ce que c'est ?

C'est un environnement de développement(IDE) open source et gratuit qui est téléchargeable dans plusieurs versions via le site officiel d'Arduino. La version 1.8.5 et 1.6.5 sont les versions sur lesquelles nous avons travaillés. Cet IDE permet de communiquer avec la carte Arduino. Pour ce faire, il y a une partie qui permet d'éditer un programme, une autre partie qui permet de compiler ce programme. Une fois ces deux étapes finis, on a la possibilité de « téléverser » le programme dans la

mémoire de l'Arduino.

Le langage de programmation de cet IDE est le langage C et la structuration d'un programme Arduino comporte deux fonctions principales : - « Void SETUP » permet d'initialiser le microcontrôleur. C'est dans cette fonction qu'on va définir les numéros des ports sur lesquelles la LED est connectée ou encore définir le nombre de pas par minute d'un moteur pas à pas.

- « Void LOOP » : C'est la boucle infini, elle permet de lancer le programme.

3. Fonctionnalités intégrés dans la maison

Indication : La partie code n'est pas explicitement donné dans ce rapport. Nous avons pris le soin d'expliquer les codes qui nous ont permis faire ce projet. Pour obtenir les codes sources complets, veuillez vous référer à la partie « 8. Sources » du rapport.

3.1.Ouverture de portail

Matériel utilisé : Moteur pas à pas et son driver, infrarouge, télécommande, Arduino Nano, files.

Pour mettre en place un système d'ouverture de portail, on a besoin de deux composants principaux ; La télécommande (en interaction avec le système à infrarouge) et un moteur pas à pas (permettant l'ouverture du portail par rotation du moteur).

Explicatun de la partie code : Tout d'abord on a besoin de deux bibliothèques ; « Stepper.h » correspondant au moteur pas à pas et « IRremote » pour la télécommande. Vous pouvez vous référer au code source pour suivre les étapes.

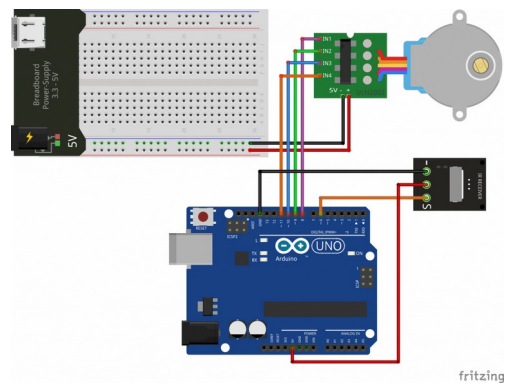
Dans un premier temps on définit une variable STEPS 45 qui permet de partitionner une rotation en 45 étapes. On définit également sur quel pin est connecté le récepteur infrarouge.

Il faut ensuite créer une instance : -du récepteur infrarouge avec "IRrecv irrecv(receiver); "
- de "decode_result" qui permettra de stocker le message reçu par le récepteur.

Pour l'initialisation, c'est à dire la fonction « Void SETUP », nous avons besoin de la commande « enableIRIn() ; » qui consiste à activer le module infrarouge.

En ce qui concerne la boucle infini (« Void LOOP »). Il faut détecter la présence d'un message sur la transmission. Si c'est le cas alors on a deux cas de figure ; Si on appuie sur le bouton 2 cela va enclencher l'ouverture du portail. Si on appuie sur le bouton 5 alors cela va enclencher la fermeture du portail. Ainsi, dès que le receptrer recoit un signal, il le traite. Si le récepteur infrarouge recoit un signal/message inconnu, alors rien ne se passe.

Branchements :



Alimentation : Bloc alimentation avec deux entrées. On a fait le choix de le régler à 5Volt.

Le portail vu de face :



On va le portail fermé et l'infrarouge situé au dessus du portail.

Le portail vu d'au dessus :



L'image ci-dessus est la photo du système d'ouverture du portail vue du dessus. Nous avons récupéré le système d'engrenage d'une unité centrale H.S (celui de l'ouverture du lecteur CD #recyclage).

3.2. Ouverture du garage avec un ServoMotor

Matériel utilisé: Arduino Nano, Servo motor, infrarouge, télécommande, files.

Explication de la paratie code : Nous avons besoin d'inclure la librairie « Servo.h » (propre au servo motor), « IRremote.h » pour le système infrarouge.

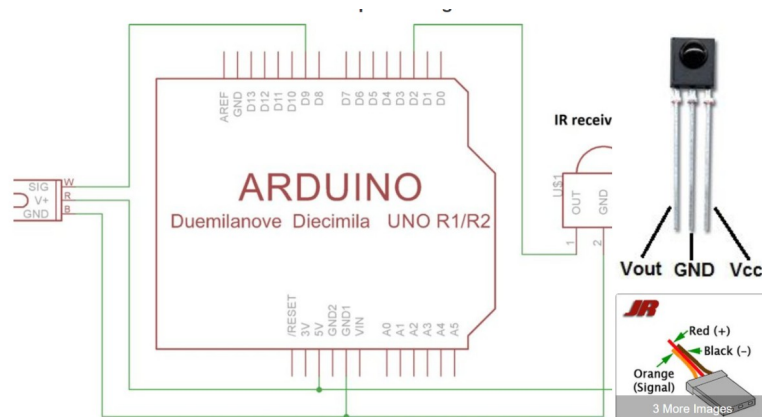
Premièrement il faut définir quel bouton correspondra à l'ouverture du garage et quel bouton pour la fermeture.

Ensuite, nous devons déclarer un servo motor, un entier pour définir la rotation de l'angle et deux valeurs booléennes correspondant à l'état d'ouverture ou de fermeture. Ces valeurs booléennes ne peuvent pas être vraies simultanément. Enfin, il faut déclarer un récepteur infrarouge et une instance de « decode_results » capable de stocker le message reçu.

Dans la partie initialisation, il faut lancer le programme avec la commande « Serial.begin(9600) », attribuer un pin au servo motor (ici 9) et lancer l'infrarouge avec la commande « enableIRIn() ».

Dans la partie boucle infinie, il faut distinguer les signaux possible. Si le message reçu par le récepteur infrarouge est « plus » cela va enclencher l'ouverture, sinon si il s'agit du message « minus » alors cela va enclencher la fermeture.

Branchements :



;

Alimentation : Alimenté par l'arduino lui même .

Photo de l'ouverture du garage :



La photo est prise du dessus. Nous avons scotché le servo motor juste derrière la porte de garage pour permettre l'ouverture.



Nous avons ensuite placé l'infrarouge tout en haut de la maquette.

3.3. Système de température/humidité

Matériel utilisé : Arduino Nano, capteur humidité DTH11, écran LCD.

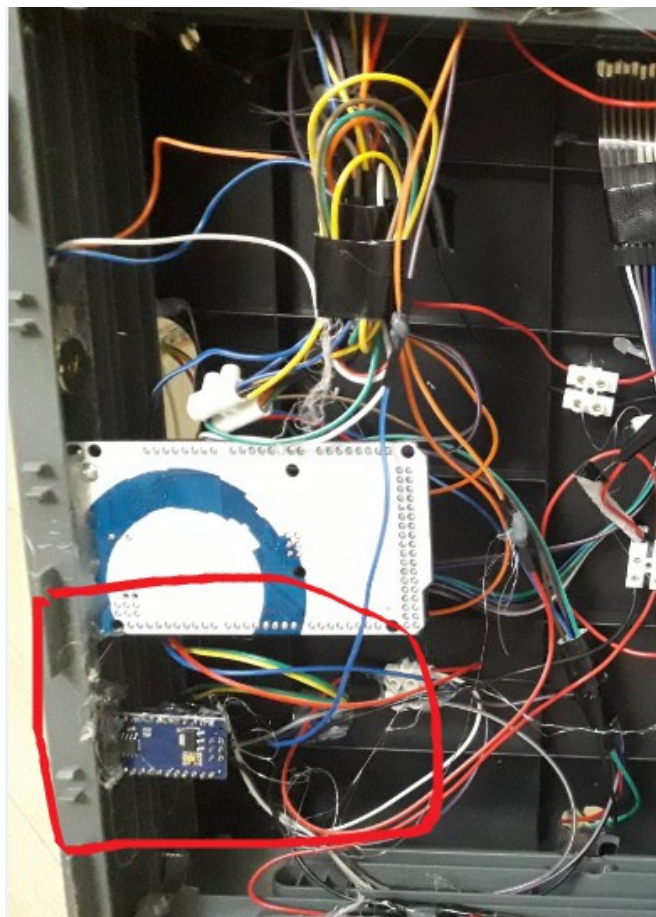
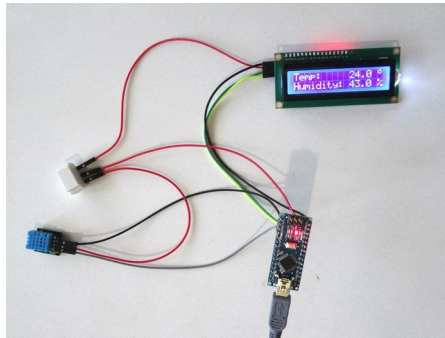
Explicatuin de la partie code : Tout d'abord, il faut utiliser les bibliothèques suivantes : « liquideCrystal.h » (écran LCD) et « dht11.h » pour le capteur.

Il faut commencer par déclarer l'écran lcd, le capteur dht et définir le numéro de pin du dht (ici 4).

Dans la partie « Void SETUP », il faut initialiser le lcd et le lancer.

Pour ce qui est de la boucle infini, il y a une partie de l'affichage sur le lcd qui est identique quelque soit la température ou l'humidité. Il s'agit du côté gauche de l'écran, les mots « Humidity » et « Temp » sont toujours affichés, par contre leurs indices changent et sont constamment recalculés. La fonction « humidity » calcule l'humidité et la fonction « temperature » calcule la température actuelle.

Branchements :



Voici comment se présente à la fin. L'écran LCV et le Capteur de DHT11 sont branchés cet Arduino Nano.

Alimentation : Bloc alimentation avec deux entrées. C'est le même qu'on a utilisé pour le système d'ouverture du portail.

Image du rendu final :



3.4. Lumière interagissant avec le Bluetooth

La gestion de la lumière est faite à travers un smartphone Android. Nous avons téléchargé une application Android sur le playstore qui se nomme « BlueCore Tech BT Control ». On peut se demander pourquoi n'avons nous pas fait le choix de créer nous même l'application. Dans un premier temps nous voulions d'abord essayer l'application pour voir le rendu, et par la suite, nous avons prévu de créer nous meme une application . Cependant, comme nous n'avions jamais fait ça auparavant, nous nous sommes rendu compte que nous perdions beaucoup de temps à chercher à faire une application mobile. Contraint par le temps, nous avons alors fait le choix de nous investir davantage sur le coté esthétique de la maquette.

Matériel utilisé : Arduino Mega 2560, bluetooth HC-06, breadbord, 8 relais, 8 résistances 220 Ohm.

Avant de décrire le code, il faut expliquer le fonctionnement des relais. Dans la gestion des différentes lumières chaque relais va jouer le rôle « d'interrupteur ». On aurait pu simplement utiliser la breadbord qui était à notre disposition. Cependant, nous avons un kit de 8 relais. Nous avons vu que sur plusieurs tutoriels proposés sur internet, les relais étaient utilisés pour les lumières, alors nous les avons inclus.

Explicatuin de la partie code : L'utilisation de bibliothèques spécifiques n'est pas necessaire dans ce programme. Il faut simplement définir les 8 relais mis à notre disposition comme étant des entiers..

A l'initialisation, on lance le programme de manière classique, c'est à dire avec la commande « Serial.begin(9600) ; ». Chaque relais est initialisé de la manière suivante : « pinsMode(R, OUTPUT) ; » ou R représente le relais et OUTPUT la sortie.

La boucle infinie met en place une succession de « if » correspondant au signal reçu pour chaque relais.

Informations importante : L'Arduino Mega alimente les relais et les relais alimentent les led. Si on ne mets pas de résistances alors cela va créé des court circuits. C'est pourquoi on a besoin de garder les résistance pour canaliser l'énergie émise par les relais et permettre une stabilité dans l'éclairage.

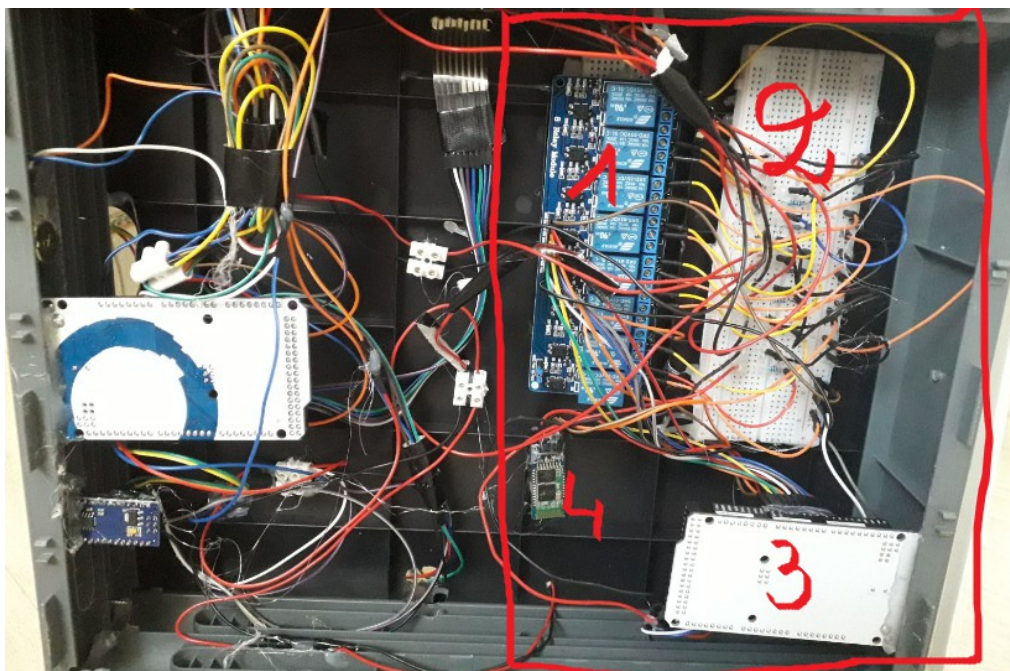
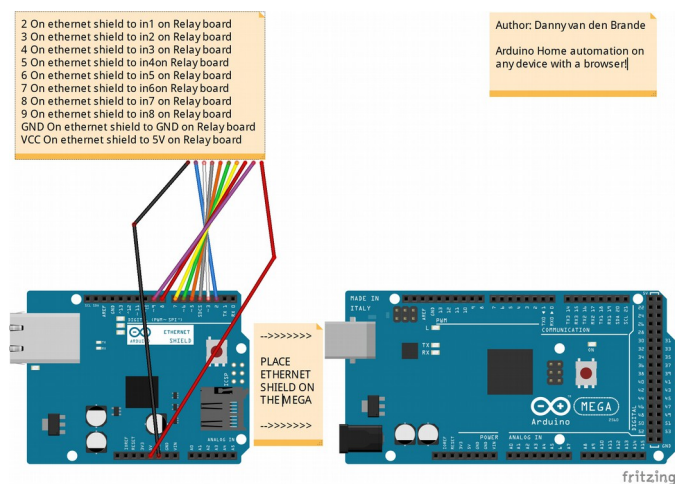
Le branchements suivie exactement celui-ci :

Ce schéma représente la base de notre branchement. Tous les fils partant de la « Relaybord » sont branchés respectivement sur le branchement 2, 3, 4, 5, 6, 7, 8 et 9 de l'Arduino Mega 2560. La seule exception est bien entendu celle des branchement correspondant à la masse (GND et au VCC) qui sont respectivement branchement sur la breadbord. **Ensuite, a fallut brancher le bluetooth HC-06.** Voilà comment nous avons procédé : -GND du HC-06, nous l'avons branché au GND du Mega.

-VCC du HC-06 est branché sur 3v3 du Mega

-RX du HC-06 est branché sur le port 1 du Mega

-TX du HC-06 est branché sur le port 0 du Mega



La photo correspond aux branchements final que nous avons mis en place pour le système de lumière

1 : Les relais

2 : la breadBord sur laquelle on y trouve les résistances. Nous avons finalement opté pour des rubans à led et malgré ce choix il nous a fallu intégrer des résistances pour permettre une stabilité au niveau de la luminosité.

3 : Arduino Mega 2560 qui alimente les relais.

4 : Bluetooth HC-06.

Alimentation : Alimenté par le Mega 2560.

3.5. Ouverture de la porte d'entrée par carte RFID

Matériel utilisé : Carte RFID, Ecran LCD, Arduino UNO, buzzer.

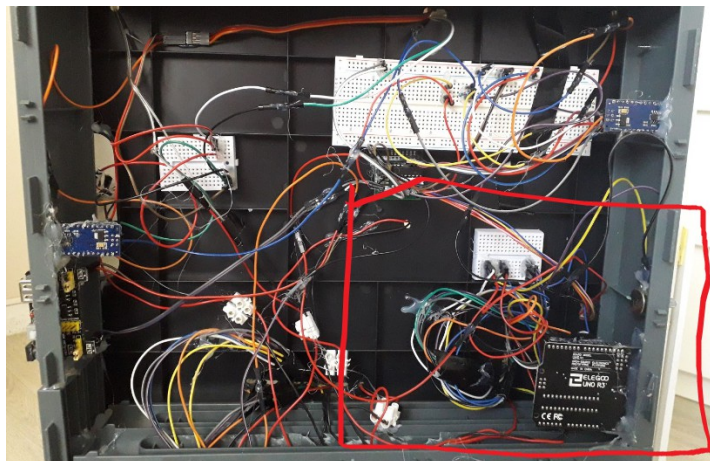
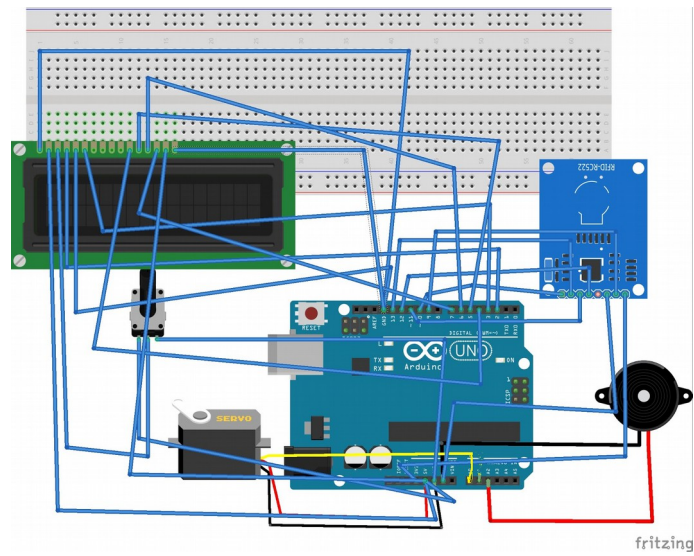
Explicatif de la partie code : On a besoin de la librairie « LiquidCrystal.h » (Ecran LCD) et « Servo.h » (ouverture de la porte), « MFRC522.h » (système RFID).

Premièrement, on définit ce dont on a besoin à savoir le numéro de pin de réception pour le système RFID. On crée une instance de du module RFID. Toute carte RFID possède un numéro correspondant à son identité. Ainsi, cela nous donne la possibilité de choisir les cartes qui seront autorisées ou non à ouvrir la porte d'entrée. C'est le système de « badge » que l'on retrouve souvent dans les résidences/propriétés privées. Nous devons ensuite définir la position du servo motor avec un entier (dans notre cas il commence à la position 0). Enfin, il faut également définir un écran LCD.

Deuxièmement, nous procédons à l'initialisation dans la fonction « void setup ». Il faut initialiser l'écran, le numéro de pin sur lequel le servo motor est connecté, le système RFID avec la fonction « PCD_Init() ».

Troisièmement, procédons à la fonction infinie (void loop). Cette dernière, va permettre d'afficher un message sur l'écran LCD dans l'attente du passage d'une carte RFID. Si la carte est correcte, c'est à dire que le numéro est reconnu cela implique l'ouverture de la porte. Par contre si le numéro est invalide alors la porte reste fermée un message s'affiche et une petite alarme sonne.

Schéma des branchements que nous avons utilisé :



Ceci est l'Arduino uno qui gère le système d'ouverture de porte par RFID.





3.6. Système d'alarme

Matériel utilisé : Ecran LCD, Arduino, Potentiomètre, clavier 4x4, capteur à ultrasons, buzzer.

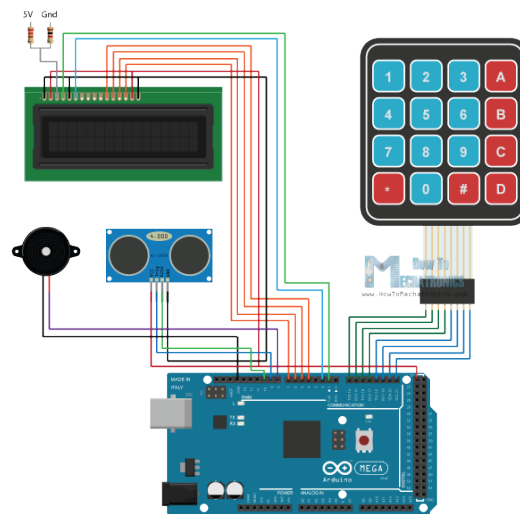
Explicatuin de la partie code : Tout d'abord, les bibliothèques à inclure sont « LiquidCrystal.h » (écran LCD) et « Keypad.h » pour le clavier 4x4.

Premièrement, il faut définir un numéro pour le buzzer (qui fait office d'alarme forte), on déclare, un mot de passe de type « String ». Ensuite, il faut déclarer différents booléens qui nous serviront comme observateurs d'états. Par exemple, il y a « isActivated » ou encore « enteredPassword ». Nous devons également, déclarer un tableau à deux dimensions permettant de définir le clavier 4x4.

Deuxièmement, on initialise ce dont on a besoin à savoir l'écran, et les triggers (émetteur et récepteur) pour l'alarme « pinMode(trigin, OUTPUT) » et « pinMode(echoPin, INPUT) ».

Troisièmement, la boucle infinie. Cette fonction est plus longue dans les autres fonctionnalités. La première phase vise à l'activation de l'alarme. Ensuite, il faut régler la distance de surveillance pour le capteur ultrasons. Une fois ces deux éléments fait. On peut passer à l'affichage et à ce qui sera proposé sur l'écran directement à l'utilisateur. Choix A : activer l'alarme, choix B changement du mot de passe par l'utilisateur. Ainsi, on donne la possibilité à l'utilisateur de changer de mot de passe directement sans passer par le programme. Bien évidemment au départ il y a un code par défaut qui est 1234*. Ainsi, l'utilisateur peut changer le code au sein du programme, sinon il doit sélectionner « B » puis entrer le mot de passe courant, puis entrer le nouveau mot de passe.

Branchements :



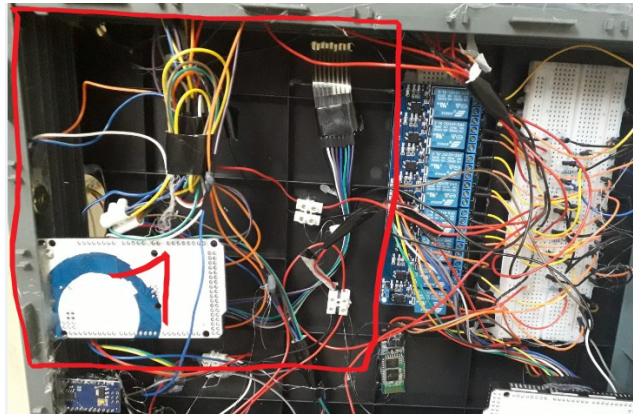
Mise en place :



Ici on voit le clavier 4x4, l'écran LCD et le buzzer.



En face se trouve le capteur. L'arduino Mega 2560 gère le système de sécurité.



C'est la photo de la mise en place finale.

4. Structure de la maison

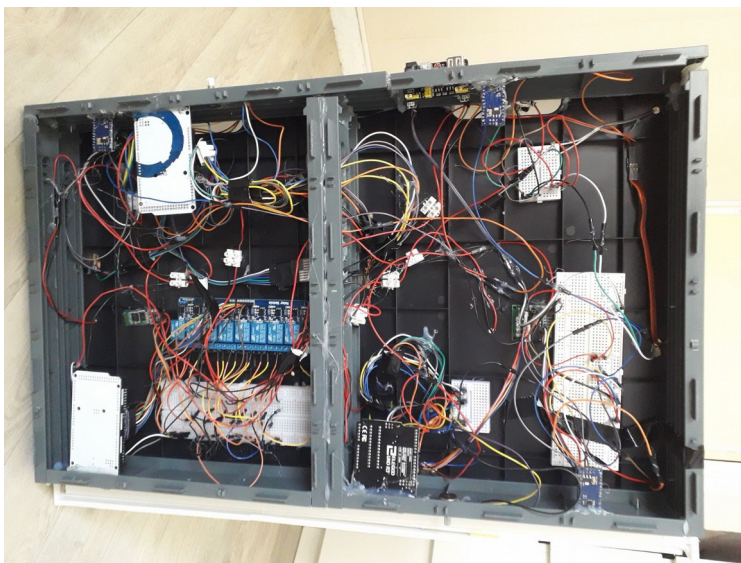
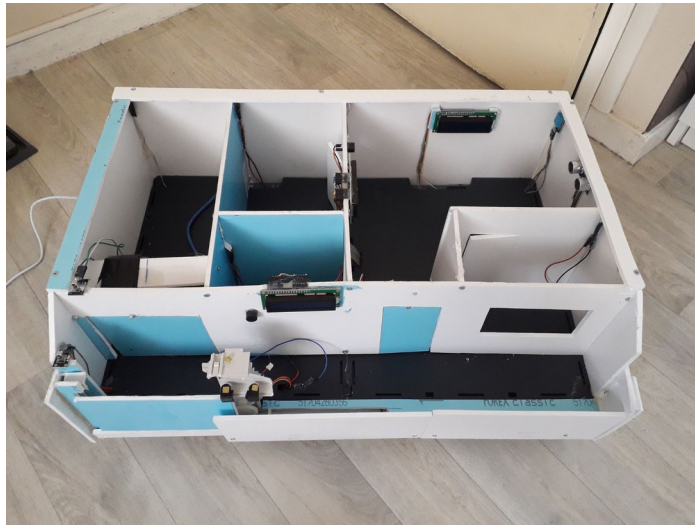
La construction de la maison s'est faite progressivement. Au départ de ce projet, nous pensions faire une maison en cartons mais cela n'a pas été fructueux car ce n'est pas très résistant. Alors nous nous sommes rendu à Castorama pour demander conseil à un spécialiste. Nos critères de recherches étaient , un matériau solide, maniable mais également léger et facile à découper. Un vendeur nous a donc orienté vers des plaque de « forex classic ». Ce sont des panneaux en PVC mais très légers, pour un prix accessible (8euros le panneau de 60x40cm) .

Le côté esthétique qui est un critère important de notre projet, nous a forcé à nous pencher sur la question de savoir « comment faire pour cacher tous les câbles, etc. ? ». Toujours à Castorama, nous avons exposé notre problème à un vendeur, ce dernier nous a fortement conseillé d'acheter des « dominos ». Les dominos permettent de regrouper plusieurs branchements à la fois. Il s'agit uniquement des branchements le GND et le + . Ensuite, nous avons pensé à faire une base de maison dans laquelle on regrouperai tous les branchements. Pour ce faire, nous avons recyclés des clusters de rangements. Si vous ne voyez pas de quoi il s'agit, alors plus simplement, il s'agit des rangements pour feuilles. C'est à la fois solide et léger.

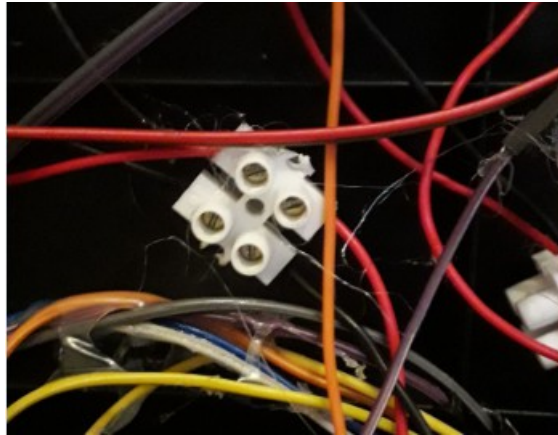
Ouverture du portail : Pour simuler une ouverture de portail, il fallait penser à un système de routage. Avant de choisir l'option « achat », nous avons tenté plusieurs choses jusqu'à trouver une alternative. En effet, nous avons trouvé une unité centrale HS mais l'engrenage pour le système d'ouverture et de fermeture du lecteur cd est intact. Alors on a démonté, juste la partie voulu. Puis à l'aide d'un pistolet à colle chauffante on a mis en place le « Portail » .

Ouverture de la porte et du garage : Nous avons collés les servo moteurs au dos des portes.

Base de la maison : Nous avons choisi de protéger les composants en faisant une base. Pour ce faire, nous avons utilisés des clusters de rangements que nous avons collés.



Pour faciliter les branchements et éviter « le faux contact » nous avons ajouté des dominos :



5. Difficultés rencontrées

Matériel : Plusieurs fois nous ne comprenions pas d'où venait le problème. Il y a le facteur qualité du produit, mais également sensibilité des branchements. Les fils se débranchent facilement c'est pourquoi nous avons investi dans les « dominos ». La mauvaise alimentation de certains pins de mauvaise qualité qui se décrochent de leur support. Cela coupe l'alimentation et le programme ne peut pas s'exécuter.

Le module RFID ne fonctionne pas avec l'Arduino Mega bien que tout soit correct. C'est pour cela que nous avons fait interagir l'Arduino UNO avec le système RFID.

Nous aurions aimé regrouper toutes les alimentations en une seule mais manque de moyens et en raison du matériel de basse qualité il a fallu « jongler » entre les différents modules que nous avons installés. Par ailleurs, nous voulions mettre en place un système de « Maître/Esclave » entre l'Arduino Mega 2560 et les Arduino Nano, mais cela n'a pas fonctionné. C'est pourquoi nous avons tout séparé.

Nous avons voulu utiliser qu'un infrarouge pour à la fois le garage et le portail, mais il y a eu des interférences, ce n'est pas stable. C'est pourquoi, il a fallu partitionner en utilisant 2 infrarouges. Cela s'explique par la puissance du moteur du garage. Il génère un champ magnétique qui crée une interférence. Il aurait fallu, utiliser un moteur plus petit. Manque de moyens...

Domino : Nous avons beaucoup trop de branchement sur le GND et d'autre sur le +. En allant à Castorama, nous avons demandé l'avis à un vendeur, et il nous a conseillé d'acheter un domino. C'est un branchement permettant de regrouper les GND ensemble et les + ensemble. Problème résolu !

Problème de compatibilité de la version de l'IDE Arduino 1.8.5 avec l'Arduino Mega pour la mise en place du système d'Alarme. Il a fallu utiliser la version 1.6.5.

Temps : Le temps a été une contrainte, car nous aurions aimé aller plus loin dans le projet et le rendre plus personnel.

6. Manuel d'utilisation

Portail : Presser le bouton 2 de la télécommande en direction de l'infrarouge pour l'ouverture. Le bouton 8 pour la fermeture.

Garage : Presser le bouton 1 pour l'ouverture et le bouton 4 pour la fermeture.

Porte d'entrée : Passé le badge blanc en dessous de l'écran LCD pour que la porte s'ouvre. Si vous passé n'importe quel autre badge rien ne se passe.

Système d'alarme : Appuyer sur A pour activer l'alarme, un compte à rebours de 6 secondes commence. Un fois l'alarme activée, si vous passer la main au niveau de la pièce surveillée alors le buzzer commence à sonner. Le seul moyen permettant de désactiver l'alarme est d'entrer le bon code. Par défaut, il s'agit de 1234*.

Gestion des lumières de la maquette : Il faut au préalable télécharger l'application Android « BlueCore Tech BT Control ». Une fois l'application télécharger, il faut activer le bluetooth du téléphone, ensuite sélectionner le bouton « Select Device » qui détectera le bluetooth HC-06 de la maquette. Sélectionner le bluetooth. A vous de jouer, vous pouvez allumer ou éteindre selon vos envies.

7. Conclusion

Ce projet a été une totale découverte. La notion d'électronique qui nous paraissait lointaine et fort peu compréhensible est dorénavant plus claire et moins effrayante. Il est vrai que nous en apprenons toujours un peu plus sur nous même. D'une certaine façon, le fait d'avoir été « livré à soi même » sur une terre inconnue, cela nous a forcé à mettre les bouchés doubles pour parvenir à mener un projet d'un bout à l'autre.

Ce projet nous a-t-il servi ? Nous servira-t'il plutard ? La réponse est oui. Maintenant que nous pensons connaître (je dis bien connaître et non pas maîtriser) les notions de bases de l'électronique, la suite logique serait d'allier ces nouvelles compétences avec la spécialité que nous souhaitons faire. Etant donné que nous envisageons de poursuivre nos études dans le domaine du décisionnel, nous avons pensé qu'il serait intéressant de faire du deep learning avec les modèles Arduinos. Par exemple étudier toutes le fois où nous allumons la lumière etc.

8. Sources

L'idée générale de la maison et la vidéo qui nous a servie de support est celui-ci :

https://www.youtube.com/watch?v=dRCnccv_dVE&t=10s

Système d'alarme avec le clavier à code :

<https://howtomechatronics.com/projects/arduino-security-alarm-system-project/>

Garage, servomotor

<http://www.instructables.com/id/Controlling-servo-motor-using-IR-remote-control/>

Ouverture du portail:

<https://www.brainy-bits.com/control-stepper-motor-with-arduino/>

Système de température :

<http://www.instructables.com/id/Arduino-Nano-DHT11-Temperature-an-Humidity-I2C-2-X/>

Système RFID :

<http://arduinoinnovation.blogspot.fr/2017/05/rfid-home-security-using-rc522.html>

9. Où trouver les composants

NOM	FOURNISSEUR	QUANTITE	PRIX EN EUROS
Pack complet avec Arduino Mega 2560	Amazon	1	56
Arduino Uno	Amazon	1	10
Arduino Nano	Amazon		11
Servo Motor	Boutique LetmeKnow / Site internet LetmeKnow	3	3,50
Moteur Stepper/ Moteur pas à pas	Amazon	1	4,50
Ecran LCD	Amazon	1	2,50
Capteur DHT11 (température)	Amazon	1	Présent dans le pack complet
Pavé numérique 4x4	Amazon	1	Présent dans le pack complet
Capteurs ultrasons	Amazon	2	Présents dans le pack complet
Infrarouge	Amazon	3	Présent dans le pack complet
Buzzer	Amazon	2	Présent dans le pack complet
Bluetooth	Amazon	1	9
Rubans à LED/ LED	Amazon	20	Présent dans le pack complet
Files électriques	Amazon	120	7
Fer à Souder	Amazon	1	16,50
Plaques PVC Florex classic	Castorama	4	34
Domino	Castorama	11	4,50

Cable USB pour l'alimentation	Amazon		7 € (les 4)
-------------------------------	--------	--	---------------