|  |
| --- |
| **Projet Barrière** |
|  |
|  |
|  |
|  |
| **Delaune Rémi/LY Moussa** |
| **04/12/2017** |
|  |

Sommaire

[I. Elaboration du Projet 3](#_Toc500345752)

[1) Système existant : 3](#_Toc500345753)

[2) Analyse des besoins : 4](#_Toc500345754)

[3) Spécification : 5](#_Toc500345755)

[a. Diagramme des cas d'utilisation (Use Case): 5](#_Toc500345756)

[4) Conception : 5](#_Toc500345757)

[a. Fonctionnement du clavier matricielle: 6](#_Toc500345758)

[II. Réalisation et tests unitaires: 7](#_Toc500345759)

[1) Tests Unitaires: 7](#_Toc500345760)

[a. Fonctionnement de la barrière: 7](#_Toc500345761)

[b. Test des boucles : 7](#_Toc500345762)

[c. Lecture de la carte à puce : 8](#_Toc500345763)

[d. Fonctionnement de l'Afficheur : 9](#_Toc500345764)

[e. Lecture du clavier matricielle: 9](#_Toc500345765)

# Elaboration du Projet

Nous allons donc ici exprimer les besoins du système afin de mettre en place le suivi de sa réalisation. Le projet possédera donc un début (le besoin) et une fin (la réalisation).

## *Système existant :*

Nous avons à disposition une maquette :

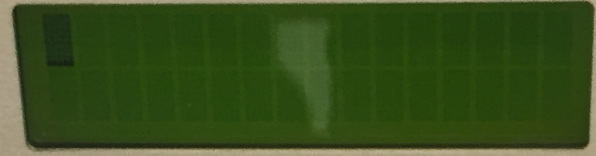


La maquette est constituée de plusieurs éléments:

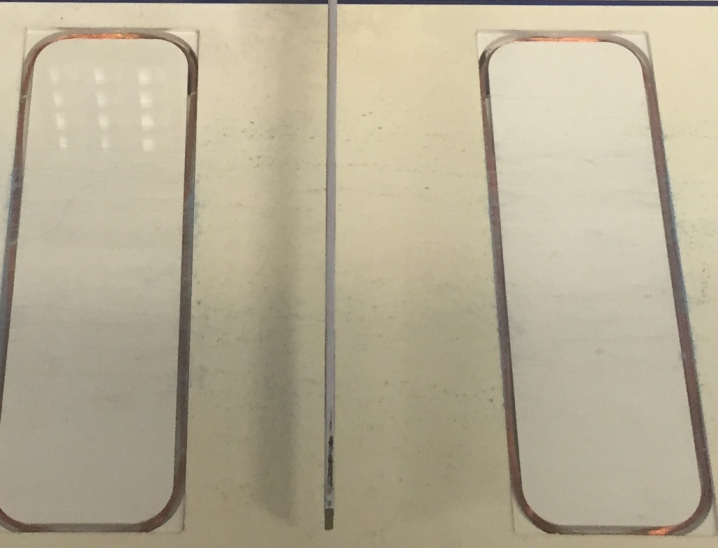
* Un clavier matricielle :



* Un afficheur :



* Les boucles:



## *Analyse des besoins :*

La barrière devra restreindre l'entrée au parking en laissant passer uniquement les usagers possédant un identifiant.



## *Spécification :*

La spécification des besoins permet de décrire comment va interagir la barrière avec l'environnement auquel il est exposé.

### *Diagramme des cas d'utilisation (Use Case):*



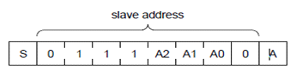
## *Conception :*

* Le programme sera écrit sous Arduino en C++ .
* Utilisation du BUS I2C afin de faire communiquer la carte Arduino et la maquette de la barrière du parking.
* Le programme sera envoyé vers une carte ARDUINO MEGA.
* La maquette de la barrière disposera d'un afficheur et d'un clavier.
* Soudure du câble faisant la liaison entre la carte Arduino et la Barrière .
* Test du fonctionnement du câble avec oscilloscope.

### *Fonctionnement du clavier matricielle:*

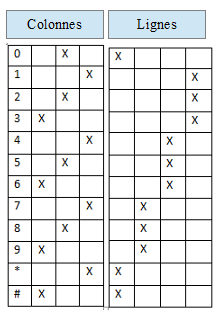
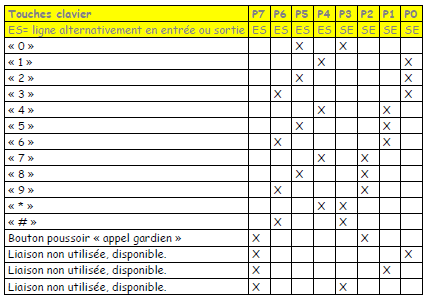
*Adresse du Clavier:*

L'adresse du clavier est déterminée de la façon suivante:



Les lignes A0 et A2 sont reliées au GND soit 0 logique. Alors que la ligne A1 est reliée à 5V soit 1 logique, cela nous donne l'adresse 0x22.

*Correspondance des touches:*

# Réalisation et tests unitaires:

Mémoire des composants I2C :

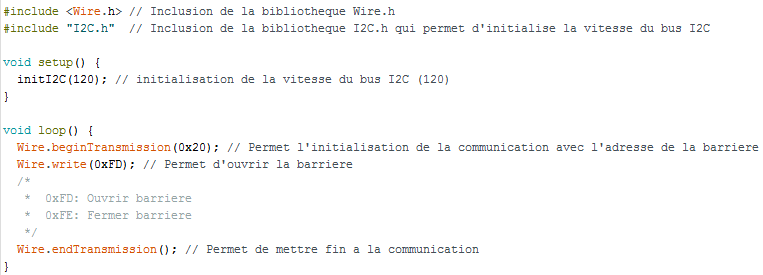
Les adresses sont codés sur 7 bits

## *Tests Unitaires:*

### *Fonctionnement de la barrière:*

Ce test unitaire permet l'initialisation et de tester la communication entre la carte et la barrière dont le moyen de communication est le bus I2C à l'adresse 0X20.

*Le programme est le suivant:*

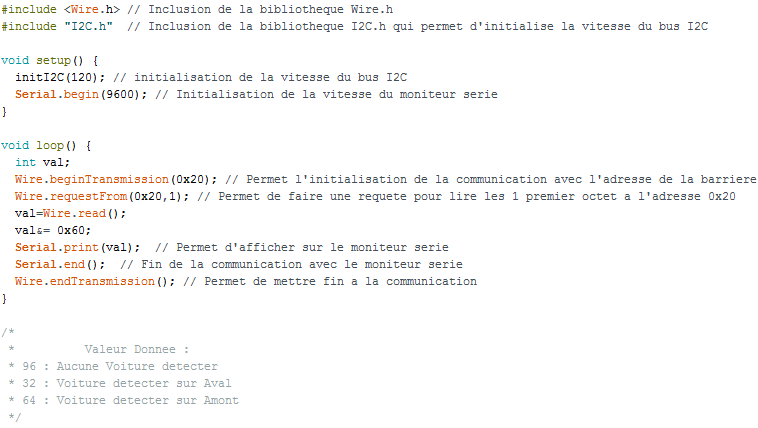


### *Test des boucles :*

Ce test unitaire permet de tester la détection du véhicule par la boucle d'entrée (Amont) et par la boucle de sortie (Aval), l'adresse des boucles est 0x20.

On déclare alors une valeur ou l'on affecte un masque afin d'avoir une valeur précise, on affiche ensuite le résultat sur le moniteur série qui m'affichera un résultat selon la boucle utilisée.

*Le programme est le suivant:*



### *Lecture de la carte à puce :*

Ce test unitaire permet de détecter si la carte à puce est insérée, l'adresse de la carte à puce est 0x21.

Comme pour la lecture des boucles, on déclare une variable à laquelle on affecte un masque qui permettra d'afficher une valeur précise sur le moniteur série.

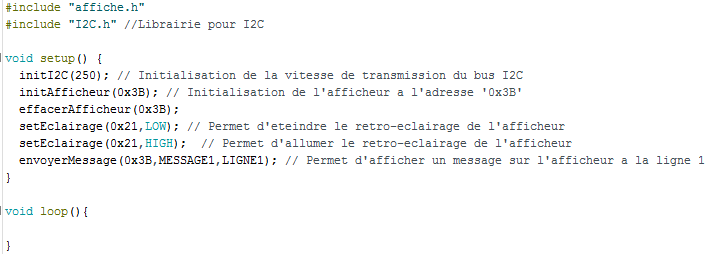
*Le programme est le suivant:*



### *Fonctionnement de l'Afficheur :*

Ce test permet de tester l'afficheur de la maquette qui est à l'adresse 0x3B.

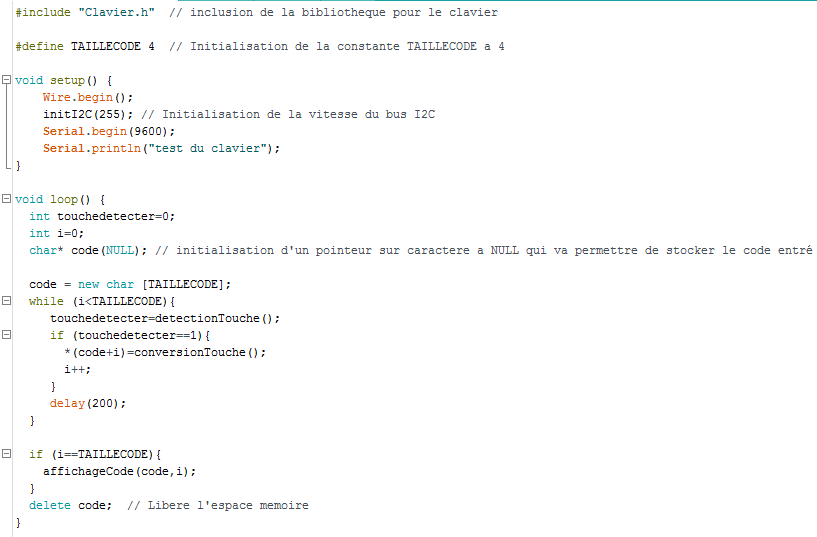
*Le programme est le suivant:*

**

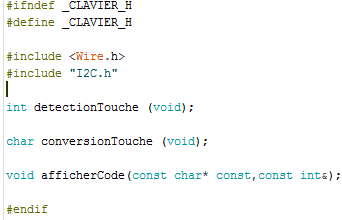
### *Lecture du clavier matricielle:*

Ce test unitaire permet de lire le code écrit sur le clavier par l'utilisateur, l'adresse du clavier est 0x22.

*Le programme principale est le suivant:*



*Le fichier "Clavier.h" est le suivant:*



Le fichier "Clavier.ino" sera mis en Annexe n°1.