

projet d'ingénierie

projet n°29

## **Babyfoot connecté**

# **RAPPORT FINAL**

**Membres du groupe :** COURTIAU Malou, FAROULT Paul, ZAKANI-FADILI Bilel,  
PIGA Caroline

**Tuteurs :** Jérémie Postel-Pellerin et Vincenzo Della Marca

2nd semestre année 2021-2022

# **TABLE DES MATIÈRES**

<b>INTRODUCTION : présentation du projet .....</b>	<b>3</b>
--	----------

\*\*\*

<b>I - Découverte .....</b>	<b>3</b>
-----------------------------	----------

I-1. Choix du projet et découverte avec nos tuteurs.....	3
I-2. Découverte du matériel .....	4
I-3. Prise en main du matériel.....	4

<b>II - Étude de faisabilité .....</b>	<b>5</b>
--	----------

II-1. Liste des composants .....	5
II-2. Fonctionnement des composants .....	5

<b>III - Recherche de solution .....</b>	<b>9</b>
--	----------

III-1. Établissement des règles après premiers tests .....	9
III-2. Premières mises en exécution .....	9

<b>IV - Conception .....</b>	<b>11</b>
------------------------------	-----------

<b>V - Conclusion : les perspectives .....</b>	<b>12</b>
--	-----------

V-1. Points forts et points faibles du projet .....	12
V-2. Difficultés rencontrées.....	13
V-3. Suites envisagées : possibilités d'amélioration.....	13

<b>ANNEXES.....</b>	<b>14</b>
---------------------	-----------

1 -prix et caractéristiques des composants.....	14
2 - Programme et Schéma : premier test.....	15
3 - Programme et Schéma : deuxième test.....	16
4 - Programme final (simplifié).....	17
5-Programme et Schéma final.....	18

## INTRODUCTION

Dans le cadre d'un projet d'ingénierie du second semestre, nous avons pour mission de choisir un projet : le nôtre consiste à améliorer un babyfoot classique à l'aide d'appareillages électroniques afin de le rendre "connecté" via des algorithmes de programmation. Le cœur de ce projet tourne autour d'un dispositif électronique bien précis : la carte arduino uno. Ce microcontrôleur très intéressant nous a permis de découvrir la microélectronique, d'étudier et de conceptualiser notre projet.

La carte Arduino Uno est la plus simple et la plus économique carte à microcontrôleur d'Arduino. Des connecteurs situés sur les bords extérieurs du circuit imprimé permettent d'afficher une série de modules complémentaires. Cette carte peut se programmer avec le logiciel Arduino disponible gratuitement en téléchargement.

Le microcontrôleur qui la compose (ATMega328) contient un bootloader qui permet de modifier le programme sans passer par un programmeur.

\*\*\*

## **I - Découverte**

### **I-1. Choix du projet et découverte avec nos tuteurs**

N'ayant pas reçu énormément d'informations concernant les projets, notre choix a été guidé par les affinités des membres du groupe. Que ce soit un goût pour le babyfoot ou un intérêt pour la microélectronique, notre choix s'est porté sur ce projet. Nous nous sommes donc décidés les premiers.

Ensuite, est venue la question de notre objectif dans ce projet qui nous paraissait plutôt flou. Nous avons donc décidé de prendre contact avec nos tuteurs le plus rapidement possible.

Cette rencontre nous a permis de mieux comprendre les enjeux du projet et les attentes à notre égard.

## **I-2. Découverte du matériel**

Lors de la découverte du projet avec nos tuteurs, ces derniers nous ont fourni du matériel afin de pouvoir réaliser notre dispositif pour le baby-foot connecté : deux cartes Arduino uno, des leds ,des résistances, des boutons poussoirs, des capteurs de force ,des capteurs ultrasons et d'un écran lcd.

Nos tuteurs nous ont prévenus, et nous avons très vite pu nous rendre compte que la communauté Arduino était très vaste et qu'il était assez simple d'apprendre à utiliser ce matériel vu la quantité de tutos présents sur YouTube ou encore tous les programmes déjà existants et disponibles sur des forums .

## **I-3. Prise en main du matériel**

Motivés par notre projet et la découverte de ce nouvel outil, nous nous lançons très vite dans la prise en main du matériel. Tout d'abord, on se familiarise avec des systèmes peu complexes afin de comprendre comment fonctionne l'Arduino, par exemple faire clignoter une led ou encore allumer une led lorsque l'on appuie sur un bouton .

Pour réaliser ces programmes basiques, on prend des programmes déjà existants dans lesquels on modifie quelque valeurs, par exemple pour que la led reste allumée plus ou moins longtemps .

Nous avons découvert par la suite qu'il existe un site internet nommé Tinkercad nous permettant de créer nos circuits et de tester nos programmes, ce site s'est avéré très pratique pour que chacun puisse avancer de son côté. Il possède deux gros avantages : le premier est que l'on peut partager notre montage et ces derniers deviennent coopératifs car dès que l'un d'entre nous modifie le programme et/ou le montage cela le modifie pour tout le monde ; le second est qu'il y a déjà des programmes de créés qui nous servent pour réaliser les nôtres.

## II - Étude de faisabilité

### II-1. Liste des composants

- carte arduino uno
- Ecran LCD 2:16 (rétroéclairé)
- 2 Capteur à ultrasons
- 2 Capteurs de vibration
- 2 Leds
- 5 résistances
- Ensemble de composants pour Arduino

(voir ANNEXE 1 : prix et caractéristiques des composants)

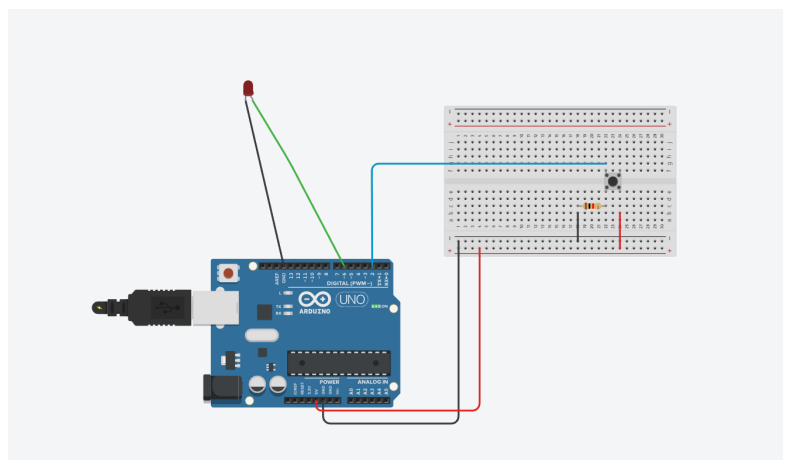
### II-2. Fonctionnement des composants

#### Carte arduino :

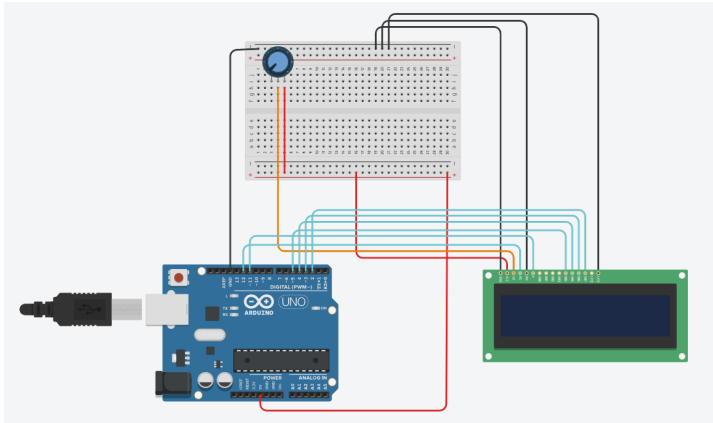
La carte arduino uno est en fait un microcontrôleur que l'on peut assimiler à un "mini-ordinateur". Ces microcontrôleurs sont en réalité partout dans votre quotidien et ont le même rôle qu'un ordinateur à plus petite échelle. Ce sont des systèmes embarqués que l'on peut retrouver dans le GPS d'une voiture, électroménager ou encore robotique. Ils disposent de pins où l'on va venir connecter nos composants. En effet, on va utiliser les pins extérieurs de la carte arduino pour envoyer de l'information grâce à un certain pin et pouvoir récupérer une information grâce à un autre pin.

Exemple :

Dans cet exemple où l'on met en œuvre l'allumage d'une LED avec un bouton poussoir, l'information du bouton est récupérée sur le pin n°2 et est renvoyée vers la LED grâce au pin n°6. Les pin GND (terre) et 5V sont des pins de références de voltage que nous utiliserons tout le long de ce projet. Il faut savoir que lorsque nous envoyons/recevons une information, celle-ci est de deux types : HIGH ou LOW (1 ou 0 en binaire ).



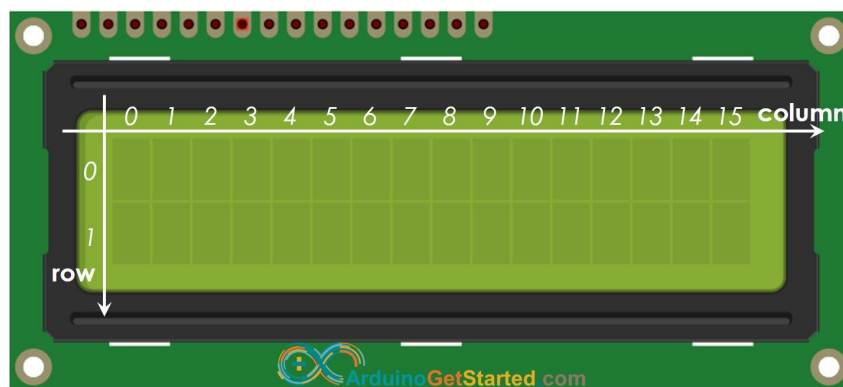
## Ecran LCD :



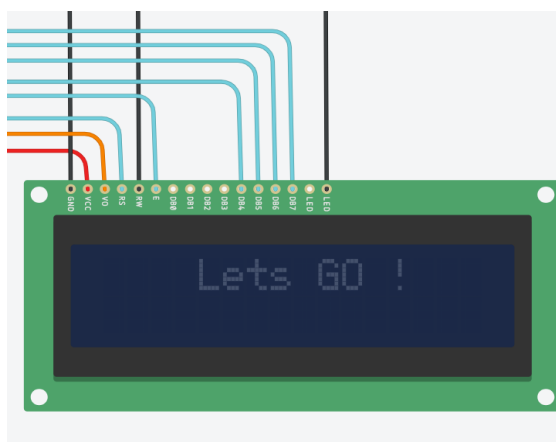
L'écran LCD est un écran assez classique qui va nous permettre d'afficher des informations grâce aux pins de la carte arduino.

En termes de branchements, il lui en faut beaucoup, nous avons donc décidé de prendre les normes de branchements trouvés sur internet.

En revanche, le codage de l'écran est très simple, celui-ci est divisé en 2 lignes et 16 colonnes.



Notre code consiste donc à afficher du texte sur une ligne et colonne donnée :



```
1 // C++ code
2 //
3 #include <LiquidCrystal.h>
4
5 // initialise la librairie LiquidCrystal qui à chaque pin de l'écran quel pin
6 // de la carte arduino est associé
7 const int rs = 12, en = 11, d4 = 5, d5 = 4, d6 = 3, d7 = 2;
8 LiquidCrystal lcd(rs, en, d4, d5, d6, d7);
9
10 void setup() {
11   // Format de l'écran ( ici 16x2 )
12   lcd.begin(16, 2);
13 }
14
15 void loop() {
16   // On se place à la 4ème colonne et 0ème ligne (pour un texte centré)
17   // on rappelle que les indices des lignes et colonnes commencent à 0
18   lcd.setCursor(4, 0);
19   // on affiche le message
20   lcd.print("Lets GO !");
21 }
22 }
```

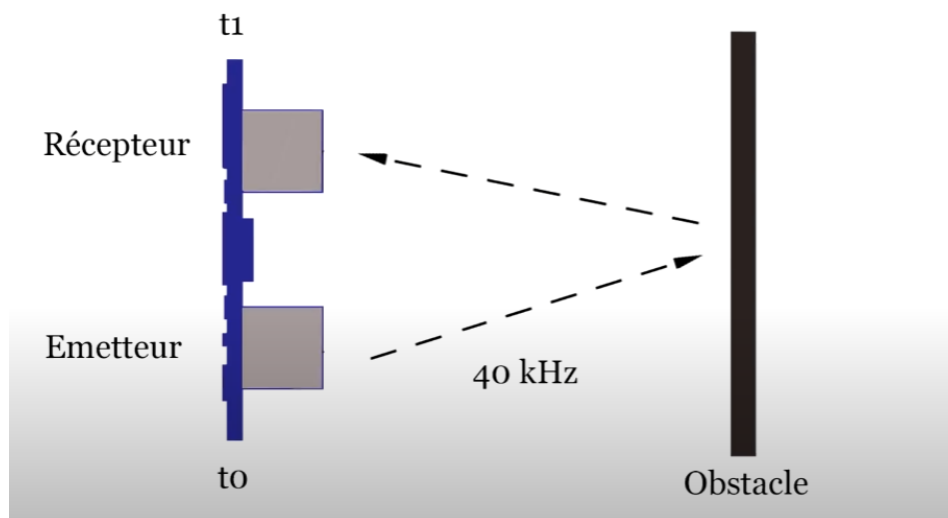
## Capteur à ultrasons :



Un capteur à ultrasons permet de mesurer des distances. Il existe différents types de capteurs permettant de le faire comme les capteurs laser, infrarouges ou encore ultrasons. Nous utiliserons ici le capteur HC-SR04 qui a une plage de mesures comprise entre 2 cm à 4m.

Les sortes de “yeux” que possèdent le capteur sont en réalité un émetteur et un récepteur (pin “Trigg” et “Echo”). Pour le fonctionnement, l’émetteur envoie une série d’impulsions sonores à 40 kHz (fréquence inaudible à l’oreille humaine) qui va être réfléchi après la rencontre d’un obstacle pour arriver au récepteur.

### PRINCIPE DE FONCTIONNEMENT



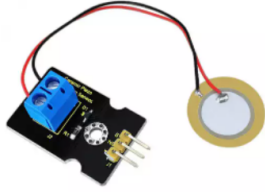
De plus, on connaît le temps d’émission et de réception de notre signal donc on en déduit facilement le temps de parcours du signal. Mais rappelons que la vitesse du son dans l’air est de environ  $340 \text{ m s}^{-1}$ , ce qui nous donne :

temps de parcours = (2. distance recherchée)/ vitesse du son

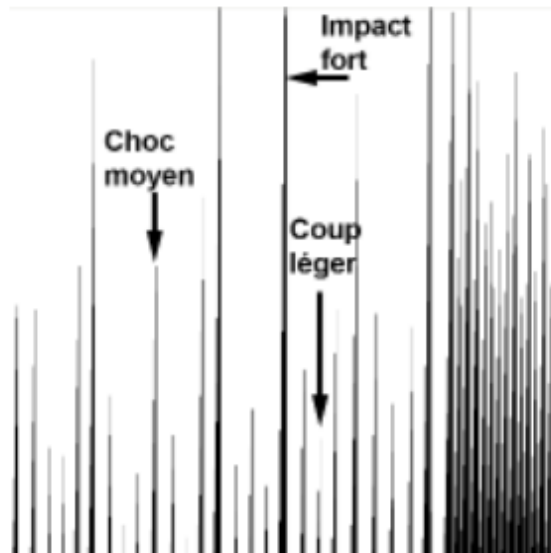
=> distance = (temps . vitesse)/2

Au niveau du montage et code, on va envoyer une information grâce à un pin de la carte arduino sur le pin “trigg” et ainsi le récupérer sur “Echo”.

## Détecteur de vibration :



Très simplement, un capteur de vibration piézoélectrique fonctionne par envoi et réception de signaux. Pour détecter les chocs, une perturbation va venir déformer le disque et ainsi envoyer un signal électrique dans les deux fils. Il capte les vibrations mécaniques qui se transmettent dans un matériau. Ce capteur fonctionne en analogique donc perçoit des différences entre les chocs forts et les coups faibles.



*Signaux envoyés par le capteur au cours du temps*



### **III - Recherche de solutions**

#### **III-1. Établissement des règles après les premiers tests**

Une fois le matériel pris en main, nous avons réfléchi ensemble à comment améliorer notre baby-foot sans pour autant dénaturer ce produit qui a traversé les époques et dans lequel se reconnaissent un large public de joueurs.

Nous avons donc décidé de faire un système permettant de compter les points automatiquement avec prise en compte des gamelles : la balle rentre dans les cages, tape le fond et ressort. D'un autre côté nous décidons de laisser de côté les demis et les pissettes car trop compliqué à mettre en place à notre échelle car pour cela il faudrait installer une caméra ou encore trouver le moyen de savoir avec quel joueur le but a été marqué .

Une fois notre objectif de projet établi, c'est le moment de se lancer dans la réalisation de notre projet en créant des programmes, d'abord simples, puis de plus en plus complexes se rapprochant de plus en plus de notre objectif final.

#### **III-2. Premières mises en exécution**

1er test :

Nous réalisons (sur Tinkercad) un montage avec un bouton et un écran où à chaque fois que l'on appuie sur le bouton le score s'incrémente de 1. Il se trouve que ce système permettait d'avoir le score d'un seul joueur et on ne pouvait pas remettre le score à 0, de plus le joueur, pour suivre son score, devait à chaque point faire une pause dans sa partie pour appuyer sur le bouton et mettre à jour son score .

(voir ANNEXE 2 : Programme et Schéma : premier test)

2ème test :

Nous décidons de nous concentrer sur le bon fonctionnement de notre programme mais dans un premier temps pour le jeu d'une seule équipe appelée équipe 1. Nous installons un capteur de force dans le fond de la gage ; un capteur à ultrason dans le tuyau par lequel la balle passe pour qu'on puisse la récupérer après un but ; et un écran afin de pouvoir afficher les informations (sa position reste à être déterminé afin d'assurer un confort optimal aux joueurs). Nous réalisons un programme lié à ce montage permettant d'afficher 2 type de score :

- Lorsque le capteur de force est activé et que le capteur à ultrasons s'active également ensuite (un intervalle de temps est prévu pour laisser le temps à la balle de descendre dans le tuyaux) : c'est le cas où un but est marqué, le score de l'équipe 1 augmente alors de 1 .
- Lorsque uniquement le capteur de force est activé, cela signifie qu'il est tapé par la balle mais que celle ci ne passe pas dans la tuyau, elle ressort ainsi des cages pour revenir sur le terrain : il s'agit d'une gamelle, le score de l'équipe 1 augmente de 1 et celui de l'équipe 2 diminue de 1, (ce système de score étant lié à la règle des gamelles).

Lorsque le score de notre équipe 1 atteint 10 (fin du match) le score est remis à 0 pour les 2 équipes.

Nous avons ensuite « dupliqué » ce programme afin qu'il soit valable pour les 2 équipes. Nous avons également rajouté un deuxième capteur de force et un deuxième capteur à ultrason dans les autres cages . Nous avons également modifié le programme pour qu'il puisse compter les gamelles et les but des 2 côtés et afin que le score se remette à 0 une fois que l'un des 2 joueurs atteint 10 points.

Pour finir nous avons pensé au fait de remettre le score à 0 manuellement dans le cas où les joueurs ne souhaitent pas aller en 10 ou voudraient simplement recommencer une partie n'importe quand. Pour cela nous avons tout simplement rajouté un bouton poussoir ainsi que quelques lignes à notre programme.

(voir ANNEXE 3 : Programme et Schéma : deuxième test)

## IV - Conception

À la suite de l'étude de la faisabilité des différents matériaux que nous pouvons utiliser dans ce projet ainsi que des règles à appliquer :

Nous avons donc décidé de placer un capteur à ultrason dans chaque tuyau de sortie de cage, en effet celui-ci permettra de détecter les gamelles (action où la balle rentre et sort de la cage sans pour autant entrer dans le conduit). Dans ce cas-là, l'équipe qui réalise la gamelle prend un score de +1 et l'autre équipe -1.

De plus, nous décidons de placer un capteur de vibration proche de chaque conduit dans chaque cage pour vérifier le passage de la balle dans celui-ci. Ainsi, si le capteur de vibration envoi une information, le score est de +1.

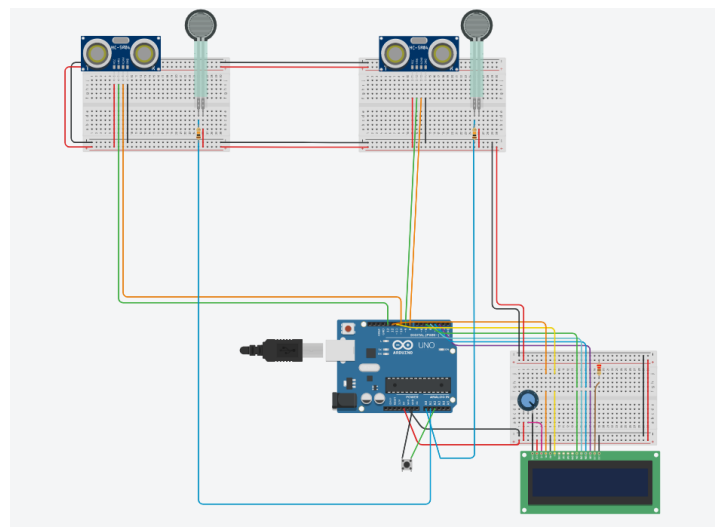
Enfin pour faciliter le jeu et permettre aux joueurs de profiter sans devoir garder en tête le score, nous avons mis un écran LCD qui affiche en temps réel le score des deux équipes. Il sera également équipé d'un bouton poussoir qui permettra la remise à 0 du score. Lors des tests de l'écran, on s'est rendu compte que lorsque nous faisions varier le potentiomètre, d'un moindre changement, l'écran ne fonctionnait plus. Nous gardons donc la valeur de base du potentiomètre.

Nous avons décidé sur la fin d'équiper le babyfoot connecté de LED de la couleur de chaque équipe car l'écran n'étant pas utilisable, nous voulions tout de même réaliser un montage fonctionnel avec les éléments à notre disposition. Malheureusement, si notre amélioration fonctionne sur Tinkercad, ce n'est pas le cas dans la vraie vie. En effet, on s'est rendu compte que les capteurs ultrasons étaient défectueux (nous les avons testés avec des programmes basiques de la bibliothèque arduino, et le moniteur série ne renvoyait que des points d'interrogations).

Lorsque l'on s'est posé la question d'autres améliorations, nous nous sommes rendu compte que la présence de l'écran LCD demandait énormément de pin sur la carte Arduino UNO.

Il n'y allait donc plus avoir assez de pin sur la carte pour d'autres éléments.

(voir ANNEXE 4 : Programme final (simplifié) et ANNEXE 5 : Programme et Schéma final)



## **V - Conclusion : les perspectives**

### **V-1. Points forts et points faibles du projet**

#### *Points forts :*

- Bas coûts : le projet que nous avons mené ne nécessite que du matériel électronique abordable (cf ANNEXE 1), le seul élément du rendu final qui coûterait cher serait alors le babyfoot, mais nous ne le comptons pas dans les dépenses car notre but étant de “l’améliorer” nous imaginons être déjà en sa possession.
- Ce projet nous a permis une initiation au Arduino. On rend la programmation intéressante en l’appliquant à un but qui nous anime, en l’occurrence : un babyfoot connecté.
- On a pu développer ce système concret et fonctionnel à partir d’un niveau 0. Nous nous sommes tous formés individuellement grâce à de nombreux tutoriels sur internet. Ensuite, on a su mettre en commun nos connaissances acquises pour améliorer au fur et à mesure notre modélisation et notre code par arborescence d'idées.
- L’outil Tinkercad nous a permis cela grâce à un fichier commun que chacun pouvait modifier à sa guise. Et nous n’avions pas besoin de plus de matériel qu’un ordinateur et une connexion internet pour avancer la théorie de notre projet.
- Nous avons décidé de prendre en compte les gamelles dans les règles de notre jeu ce qui ajoute un capteur et donc de la complexité. Mais le résultat numérique final est fonctionnel donc le défi est bien relevé.

#### *Points faibles :*

- Un problème que l’on n’a pas pu résoudre est l’impossibilité de réaliser notre projet concrètement avec le matériel mis à disposition. En effet, il manquait certains éléments comme des broches pour l’écran, un capteur de force ou même un babyfoot qui nous ont empêchés de tester notre système et ainsi éliminer des erreurs que nous n’aurions pas pu anticiper avec la version numérique.

## **V-2. difficultés rencontrés**

Tout au long de la mise en œuvre du projet, nous avons fait face à certains problèmes tels que des erreurs quand nous commençons à élaborer l'algorithme, exemple : bug d'affichage écran lors du passage des points négatifs en positifs. Ces difficultés n'ont été que temporaires et nous avons très vite appris à nous corriger afin de continuer à avancer. Une difficulté à laquelle nous n'avons pas pu faire face une fois notre programme fini, c'est le fait que l'on ne peut pas l'effectuer immédiatement, faute de matériel complet : il manque les broches de l'écran LCD nous permettant de réaliser le montage.

La principale difficulté qui a nuit à l'avancement de notre projet est le temps. En effet, le temps que nous pouvions accorder individuellement au projet s'est trouvé limité par celui accordé aux révisions et aux devoirs. De plus, n'habitant pas tous au même endroit et ne disposant pas de véhicule, il était compliqué de trouver un moyen de se retrouver à effectif complet pour avancer de manière optimale à la poursuite du projet..

## **V-3. suites envisagées :possibilités d'amélioration**

Si l'on cherche, on trouve toujours une manière d'améliorer quelque chose. C'est pourquoi, nous allons vous énoncer les idées d'amélioration possible pour notre babyfoot.

Nous avons pensé à ajouter une caméra qui filmerait le terrain du dessus et qui l'afficherait sur un écran. On pourrait passer au ralenti les buts marqués ou autres actions. Cependant ces améliorations nécessitent une formation plus poussée.

De manière encore plus poussée, on pourrait développer une interface mobile qui enregistrerait les informations d'un joueur, qui pourrait permettre de réserver un babyfoot pour une partie ou bien encore qui permettrait de choisir ses règles (gamelles ou non).

En ce qui concerne le projet en général et la manière dont il a été mené, étant donné que nous avons manqué de temps, peut-être qu'il serait bénéfique pour les potentiels projets à venir de posséder un créneau réservé à cet effet dans notre emploi du temps. Cela nous offrirait l'espace et le cadre de travail nécessaire à l'avancée du projet.

# ANNEXES

## 1 - prix et caractéristiques des composants

### carte arduino uno :



#### Carte Arduino UNO

Code article : 25950

La carte Arduino Uno est basée sur un ATmega328 cadencé à 16 MHz. Des connecteurs situés sur les bords extérieurs du circuit imprimé permettent d'enficher une série de modules complémentaires.

[Description complète](#)

Quantité :

✓ Quantité en stock : 1000+

Livraison à partir de 4,50€

16,25 € HT  
**19,50 € TTC**  
dont 0,02 € d'éco-part

[Ajouter au panier](#)

### Ecran LCD 2:16 (rétroéclairé) :



#### Afficheur LCD16x2

2 x 16 rétroéclairé bleu

Code article : 36672

Afficheur alphanumérique 2 x 16 caractères avec rétroéclairage bleu.

[Description complète](#)

Quantité :

✓ Quantité en stock : 107

Livraison à partir de 4,50€

5,58 € HT  
**6,70 € TTC**

[Ajouter au panier](#)

Afficheur alphanumérique 2 x 16 caractères à rétroéclairage bleu piloté par un driver HD44780.

Cet afficheur est livré avec un connecteur 16 broches mâles à souder permettant un accès simplifié aux E/S.

- Alimentation rétroéclairage: 5 Vcc
- Dimensions de l'afficheur: 80 x 36,5 x 11 mm
- Poids: 32 g

S

### 2 Capteur à ultrasons :



#### Module de détection US HC-SR04

Code article : 32454

Ce module permet d'évaluer les distances entre un objet mobile et les obstacles rencontrés.

[Description complète](#)

Quantité :

✓ Quantité en stock : 500+

Livraison à partir de 4,50€

3,25 € HT  
**3,90 € TTC**  
dont 0,02 € d'éco-part

[Ajouter au panier](#)

### 2 Capteurs de vibration :

#### CAPTEUR DE VIBRATIONS PIÉZOÉLECTRIQUE POUR ARDUINO

CompoZan, en collaboration avec Keyestudio, vous présente une gamme complète de modules, détecteurs et capteurs divers pour tous vos projets autour des cartes Arduino.

Sécurité incendie, surveillance, météorologie, applications médicales et bien d'autres encore sont possibles grâce aux capteurs en tous genres que nous vous proposons.

Ici, il est question d'un capteur de vibrations. Fonctionnant à l'aide d'une pastille en céramique piézoélectrique, il captera les différentes vibrations et vous restituera leur degré sous la forme d'une tension électrique exploitable pour en tirer une mesure.

SKU: 3322561308812

MARQUE: Arduino

4,45 €  
TTC


[AJOUTER AU PANIER](#)

✓ En Stock

♥ Wishlist  Compare

Garantie Paiement sécurisé

### Ensemble de composants pour Arduino :



#### Ensemble de composants pour Arduino

Code article : 35116

Contenu :

- breadboard

Ensemble de composants livré dans une boîte en plastique avec une plaque de montage rapide pour le prototypage des cartes Arduino.

[Description complète](#)

Quantité :

✓ Quantité en stock : 250+

Livraison à partir de 4,50€

13,25 € HT  
**15,90 € TTC**

Ensemble de composants livré dans une boîte en plastique avec une plaque de montage rapide pour le prototypage des cartes Arduino.

Contenu :

- une plaque de montage rapide 400 contacts
- un pack de 30 jumpers mâles/mâles
- un connecteur 40 broches
- 40 leds 5 mm (10 rouges, 10 jaunes, 10 verts et 10 bleues)
- 20 condensateurs chimiques (10 x 10 µF et 10 x 470 µF)
- 80 résistances (10 x 100 Ω, 10 x 220 Ω, 10 x 330 Ω, 10 x 1 kΩ, 10 x 4,7 kΩ, 10 x 10 kΩ, 10 x 47 kΩ et 10 x 100 kΩ)
- 8 boutons-poussoirs avec cabochons (2 rouges, 2 jaunes, 2 verts et 2 bleus)
- 40 condensateurs céramiques
- 2 buzzers (1 actif et 1 passif)
- 1 potentiomètre de 10 kΩ
- 2 leds RGB
- 1 boîte en plastique

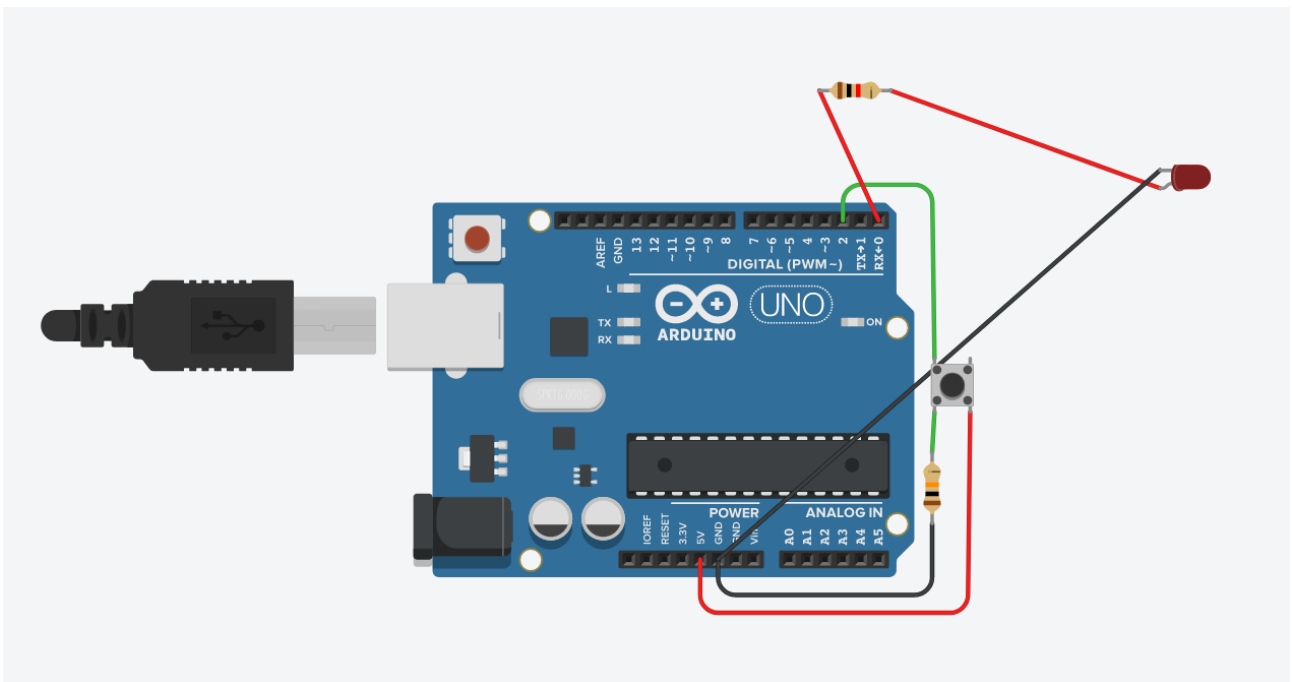
## 2 - Programme et Schéma : premier test

*On appuie sur un bouton, une lumière s'allume :*

```
int buttonState = 0;

void setup()
{
  pinMode(2, INPUT);
  pinMode(0, OUTPUT);
}

void loop()
{
  // read the state of the pushbutton value
  buttonState = digitalRead(2);
  // check if pushbutton is pressed.  if it is, the
  // buttonState is HIGH
  if (buttonState == HIGH) {
    // turn LED on
    digitalWrite(0, HIGH);
  } else {
    // turn LED off
    digitalWrite(0, LOW);
  }
  delay(10); // Delay a little bit to improve simulation performance
}
```



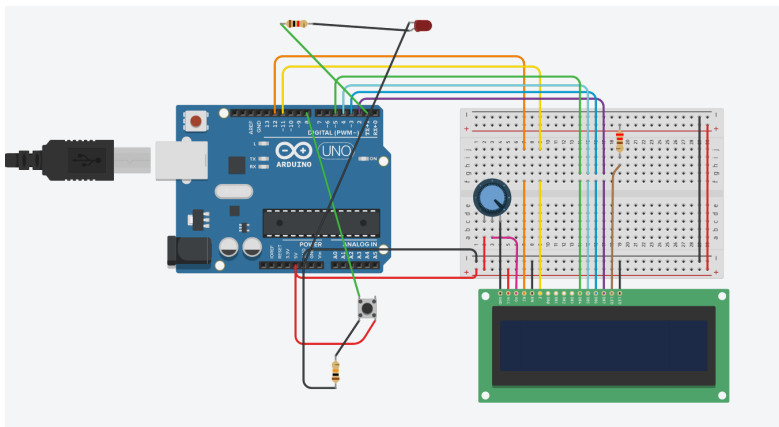
### 3 - Programme et Schéma : deuxième test

*On appuie sur un bouton, le score augmente de 1 et une led s'allume pendant une demi seconde :*

```
// include the library code:
#include <LiquidCrystal.h>

// initialize the library with the numbers of the interface pins
LiquidCrystal lcd(12, 11, 5, 4, 3, 2);
int buttonState = 0;
int a=0;
void setup(){
  // set up the LCD's number of columns and rows:
  lcd.begin(16, 2);
  // Print a message to the LCD.
  lcd.print("Score de Malou!");
  lcd.setCursor(0, 1);
  lcd.print(0);
  pinMode(0,INPUT);
  pinMode(1,OUTPUT);
}

void loop() {
  buttonState = digitalRead(8);
  if (buttonState==HIGH){
    digitalWrite(1,HIGH);
    delay(500);
    a=a+1;
  }else {
    digitalWrite(1,LOW);
  }
  lcd.setCursor(0,1);
  lcd.print(a);
  // delay a bit
  delay(50);
}
```





## 4 - Programme final (simplifié)

*Code format scratch*

On mesure distance 1 et distance 2 avec les 2 capteurs ultrasons

On mesure pression 1 et pression 2 avec capteurs forces

On affiche les scores bleu et rouge à l'écran avec comme valeur initiale 0

**Si** pression 1 > pression normal et distance 1 > 3 cm

alors score bleu = score bleu + 1

score rouge = score rouge - 1

led bleu s'allume

**Sinon si** distance 1 < 3 cm

alors score bleu = score bleu + 1

led bleu s'allume

**Sinon** led bleu s'éteint

**Si** pression 2 > pression normal et distance 2 > 3 cm

alors score rouge = score rouge + 1

score bleu = score bleu - 1

led rouge s'allume

**Sinon si** distance 2 < 3 cm

alors score rouge = score rouge + 1

led rouge s'allume

**Sinon** led rouge s'éteint

**Si** score bleu > 10 ou score rouge > 10

alors score bleu = 0

score rouge = 0

**Si** bouton pressé

alors score bleu = 0

score rouge = 0

## 5 - Programme et Schéma final

```
1 // include the library code:
2 #include <LiquidCrystal.h>
3
4 // initialize the library with the numbers of the interface pins
5 LiquidCrystal lcd(12, 11, 5, 4, 3, 2);
6
7 int capteur_force = 0; // FSR branché sur pin Analog 0
8 int lecture_FSR; // Lecture analogique de la tension du pont diviseur du capteur FSR
9 int pression;
10 int score=0;
11 int score2=0;
12 int capteur_force2 = 1;
13 int pression2;
14 int lecture_FSR2;
15 int ledr=6;
16 int ledb=7;
17
18 // paramètre bouton
19 const int btnPin = A2;
20 // variable bouton
21 int btnVal = 0;
22
23 void setup() {
24     // initialise le nombre de colonnes et de lignes de l'écran LCD
25     lcd.begin(16, 2);
26     // affiche un message sur l'écran LCD.
27     lcd.print("BLUE:      RED:");
28     Serial.begin(9600); // Envoi de message de débogage sur connexion série
29
30     //Initialisation du bouton
31     pinMode(btnPin, INPUT_PULLUP);
32
33     //Initialisation des leds
34     pinMode(ledr, OUTPUT);
35     pinMode(ledb, OUTPUT);
36     digitalWrite(ledr, LOW);
37     digitalWrite(ledb, LOW);
38
39
40 }
41
42
43 void loop() {
44
45     digitalWrite(ledr, LOW);
46     digitalWrite(ledb, LOW);
47     digitalWrite (9, LOW);
48
49
50     pinMode (9, OUTPUT);
51
52
53     delay (500);
54
55     // on envoie le signal
56
57     digitalWrite (9, HIGH);
58
59
60     delayMicroseconds (10);
61
62     digitalWrite (9, LOW);
63
64
65     // on reçoit le signal
66
67     pinMode(8, INPUT);
68
69
70     long temps = pulseIn(8, HIGH);
```

```

71
72
73 // on calcule la distance
74
75 float vitesse = 340.0;
76
77 long distance = (vitesse * temps)/1000;
78
79 distance = distance/2;
80
81
82 Serial.println (distance, DEC);
83
84
85
86
87 lecture_FSR = analogRead(capteur_force);
88 Serial.print("Capteur de Force = ");
89 Serial.println(lecture_FSR);
90
91
92
93 pression = map(lecture_FSR, 0, 1023, 0, 255);
94
95
96
97 if(pression > 0.5 and distance > 2000)
98 { score = score + 1;
99   score2= score2 -1;
100   digitalWrite(ledb,HIGH);
101   delay(500);
102 }
103
104 else if(distance < 2000)
105 {
106   score = score + 1;
107   digitalWrite(ledb,HIGH);
108   delay(500);
109 }
110
111 else
112 { digitalWrite(ledb,LOW);
113 }
114
115
116 // même code mais pour les capteurs de l'autre cage
117
118 digitalWrite (13, LOW);
119 pinMode (13, OUTPUT);
120 delay (500);
121 digitalWrite (13, HIGH);
122 delayMicroseconds (10);
123 digitalWrite (13, LOW);
124 pinMode(10, INPUT);
125 long temps2 = pulseIn(10, HIGH);
126 long distance2 = (vitesse * temps2)/1000;
127 distance2 = distance2/2;
128 Serial.println (distance2, DEC);
129 lecture_FSR2 = analogRead(capteur_force2);
130 Serial.print("Capteur de Force2 = ");
131 Serial.println(lecture_FSR2);
132 pression2 = map(lecture_FSR2, 0, 1023, 0, 255);
133
134
135
136 if(pression2 > 0.5 and distance2 > 2000)
137 { score2 = score2 + 1;
138   score= score -1;
139   digitalWrite(ledr,HIGH);
140   delay(500);
141 }

```

```

141
142 else if(distance2 < 2000)
143 {
144     score2 = score2 + 1;
145     digitalWrite(ledr,HIGH);
146     delay(500);
147 }
148     else
149     { digitalWrite(ledr,LOW);
150     }
151
152
153 // réinitialise les scores quand la partie est finie
154
155 if (score > 10)
156 {
157     score=0 ;
158     score2=0;
159     lcd.setCursor(1,2);
160     lcd.print("      ");
161     lcd.setCursor(12,2);
162     lcd.print("      ");
163 }
164 if (score2 > 10)
165 {score=0;
166     score2=0;
167     lcd.setCursor(1,2);
168     lcd.print("      ");
169     lcd.setCursor(12,2);
170     lcd.print("      ");
171 }
172
173 // permet d'éviter les erreurs d'affichage lors du passage négatif à positif
174 lcd.setCursor(1,2);
175 lcd.print("      ");

```

