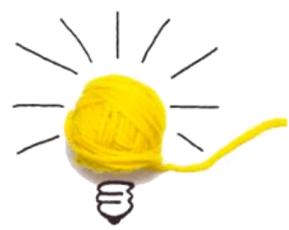


|  |
| --- |
| CPNV |
| Cahier des charges P1631 - Chevalier et Dragon V2 |
| P1704\_Manettes |

|  |
| --- |
| PICOTTE Alexandre & SANDOZ Pierre-Olivier  29/05/2018 |



# Manette des chevaliers

Pour cette manette, nous reprendrons celle développée pour le projet M1810 pour y ajouter encore des éléments pour répondre aux besoins du projet P1631 des chevaliers et dragon sur lequel travaillent HUMBERT Dominique et VALADO Dany.

Voici les contraintes de la manette :

1. Un **Arduino**
2. Un **Switch ON/OFF** qui active l’alimentation par batteries
3. Au moins **4 boutons** pour fonctions personnalisées
4. LED d’état ON/OFF
5. Des dimensions maximales de **200x100x60 mm** (joystick non-inclus)
6. **Ecran LCD** pour afficher :
   1. Infos de couplage Bluetooth
   2. Un menu permettant de sélectionner les différents modes de jeux grâce à un système de surbrillance avec demande de confirmation avant de lancer
   3. Bref descriptif du mode en surbrillance
   4. En jeu, informations sur l’état de la partie
      1. Points de vie
      2. Disponibilité de la capacité du bouclier
      3. Disponibilité de la capacité spéciale
      4. Temps restant de la partie (3 minutes)
   5. Mode pause
   6. Ecran de Game Over
7. Un joystick analogique
8. Communication sérielle en [Bluetooth 2.1 + EDR (Enhanced Data Rate)](https://www.generation-nt.com/bluetooth-2-1-edr-evolution-standard-fin-2007-actualite-23049.html) via le Bluetooth shield V2 de Seeed
9. Un **haut-parleur** sous la forme d’un simple buzzer pour remplir les fonctions des vibreurs initialement prévus
10. Le tout est monté sur **PCB (Printed Circuit Board)**
11. La structure est en PMMA et protège entièrement l’électronique

Le tout sera testé d’abord sur une plateforme de test et développement fixe sans structure mécanique.

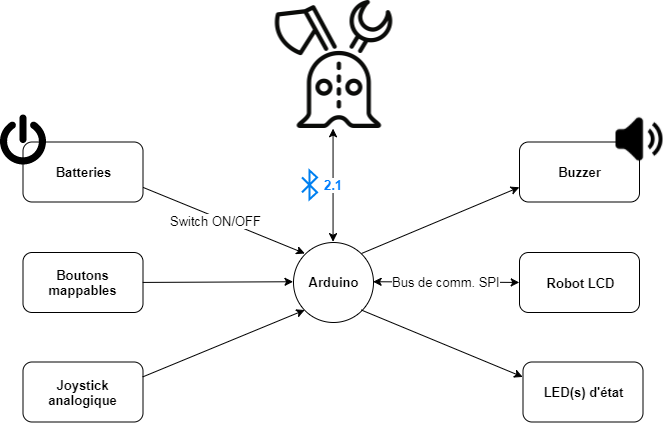


Figure 1 - Schéma de fonctionnement

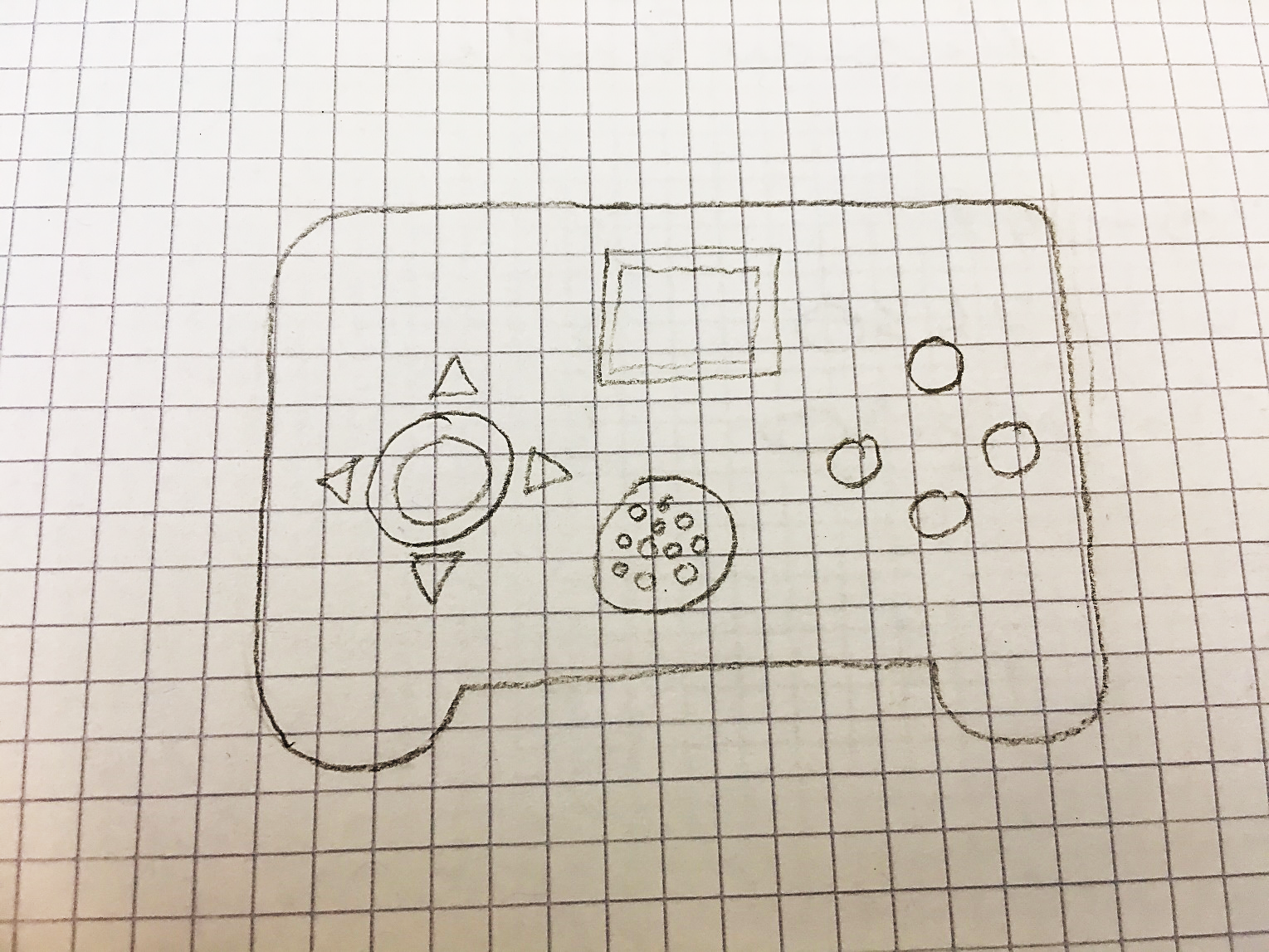


Figure 2 - Visuel en vue du dessus

## Envoi de données

Nous prévoyons d’envoyer des paquets de données en Serial via les modules Bluetooth grâce à la [librairie PJON](https://github.com/gioblu/PJON/tree/master/documentation) et la stratégie de communication [ThroughSerial](https://github.com/gioblu/PJON/tree/master/src/strategies/ThroughSerial). Cette dernière est bien documentée et possède des méthodes de gestion des erreurs et des fonctions de retour d’informations sur la transmission des données.

Nous utiliserons la méthode de [Data transmission](https://github.com/gioblu/PJON/blob/master/documentation/data-transmission.md).

Ainsi HUMBERT Dominique et VALADO Dany n’auront qu’à interpréter ce que nous leur envoyons grâce à la méthode de [Data reception](https://github.com/gioblu/PJON/blob/master/documentation/data-reception.md) clairement expliquée.

Par exemple pour envoyer un message simple, cela prendra cette forme :

// Send to device id 10 the string "Hi!"

bus.send\_packet(10, "Hi!", 3);

Cela est similaire à la méthode développée lors du projet P1631 de 2017 avec le packet ID, le message, et la longueur du message envoyé. Elle est cependant mieux documentée et optimisée.

## Échéances

04/05/2018 Une Breadborad avec :

* 1 joystick
* 1 bouton
* 2 Arduino qui communiquent via les Bluetooth Shields selon le protocole de communication défini précédemment

15/06/2018 Manettes fonctionnelles

19/06/2018 Manettes fonctionnelles et montées

