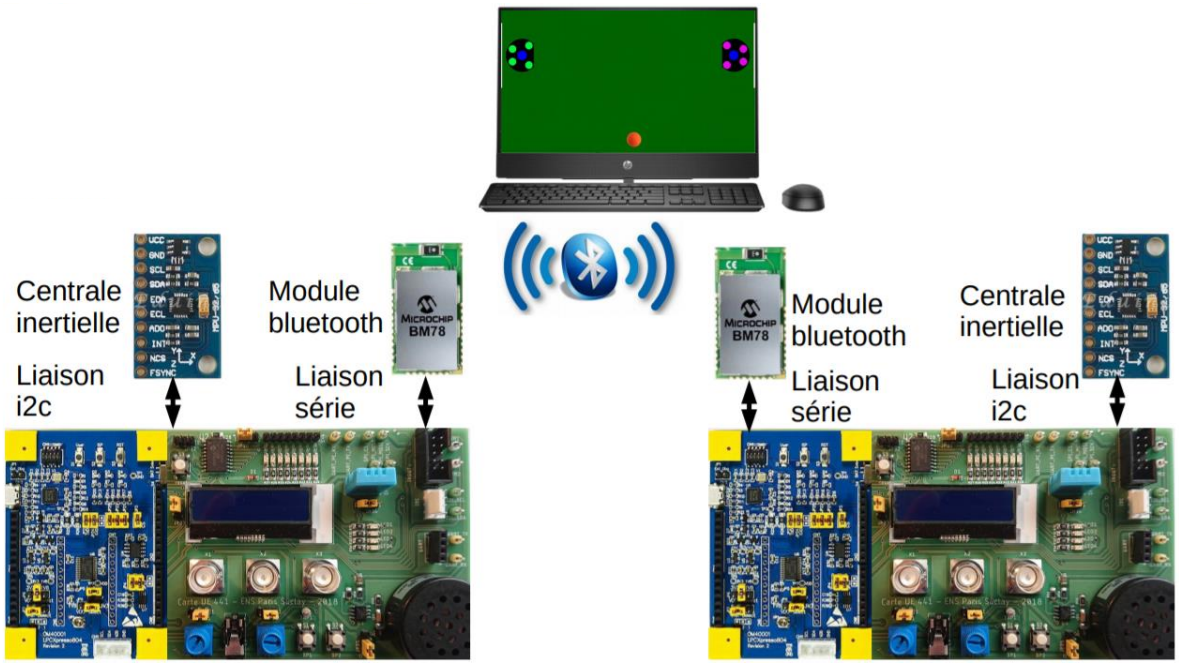
Dufourg Pierre-Alexis Mercredi 18 décembre 2019

Mini-Projet

Mallette de jeu Bluetooth



Sommaire :

Introduction :

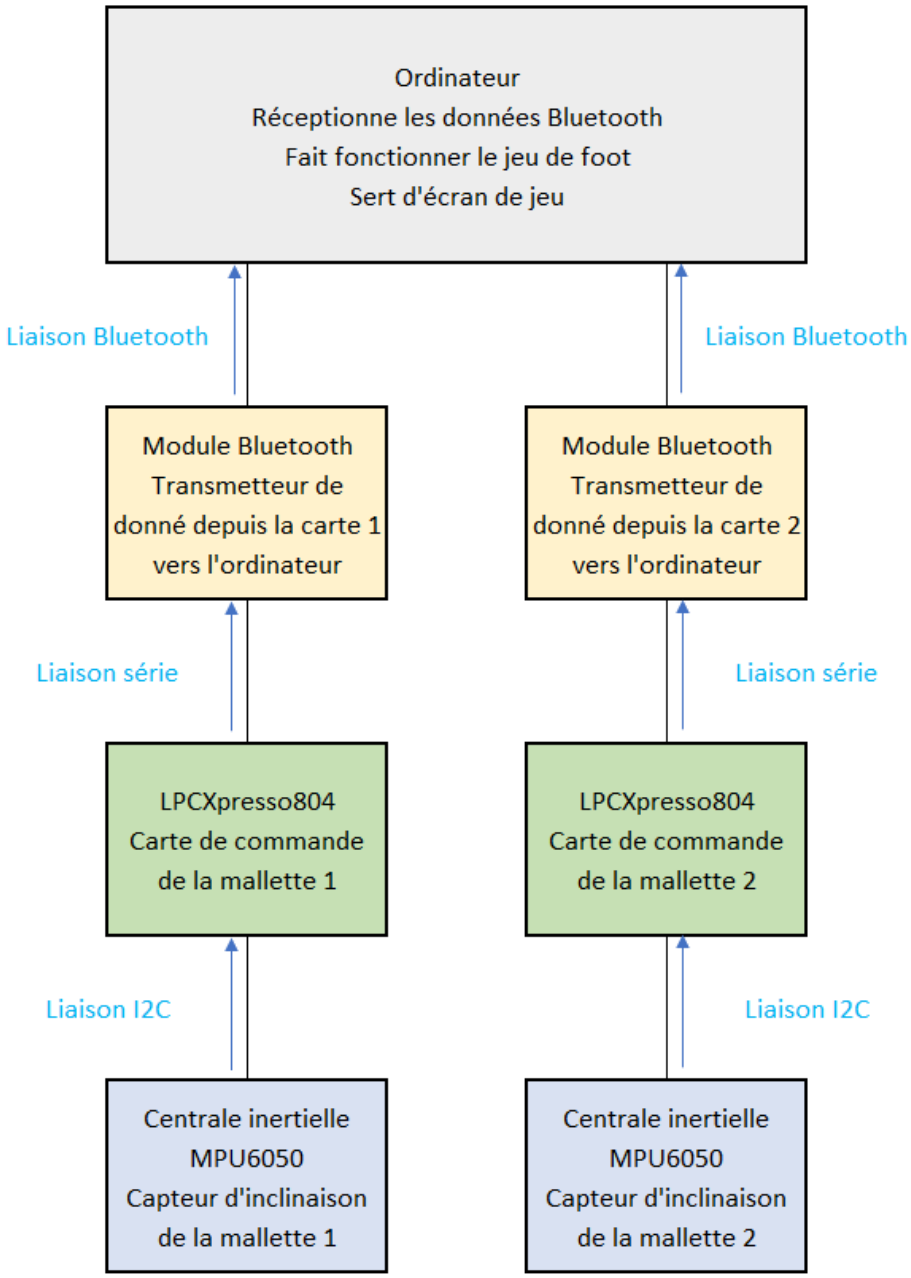
L'objectif de ce projet est de concevoir des mallettes de jeu Bluetooth fonctionnant avec le jeu de foot réalisé lors de la formation en langage C. Nous utiliserons la carte LPCXpresso804 manipulée depuis le début de l'année. L'environnement de développement MCUXpresso sera utilisé pour coder la carte.

Lors de ce TP, il va tout d'abord falloir découvrir le composant qui servira de capteur d'inclinaison de la carte : la centrale inertielle MPU6050. Son utilisation s'avère complexe en raison d'une mise en marche très lourde. En effet, il va falloir remettre à 0, calibrer et initialiser le capteur.

Ensuite il va falloir établir une liaison I2C entre le MPU6050 et la carte LPCXpresso804 pour être en mesure d'obtenir les informations délivrées par le capteur. Ces données devront être traité par la carte avant d'être envoyé à l'ordinateur grâce à une connexion Bluetooth. La carte LPCXpresso804 communiquera avec le module Bluetooth grâce à une liaison série.

Enfin, sur l'ordinateur, nous allons devoir adapter le code du jeu de foot afin qu'il puisse ouvrir les ports Bluetooth et recevoir les données envoyées par la carte. Les joueurs devront alors se déplacer en fonction de l'inclinaison de la mallette. Une deuxième télécommande sera ensuite ajoutée en suivant le même procédé.

Schéma synoptique :



Description détaillée des éléments spécifiques au projet :

Pour comprendre le fonctionnement global, nous allons décrire le cheminement du signal depuis sa création jusqu'à l'ordinateur. Nous ferons une description des composants et des méthodes de communication utilisés. Les difficultés rencontrées et leurs solutions éventuelles seront également présentées.

Point de départ, le MPU6050 :

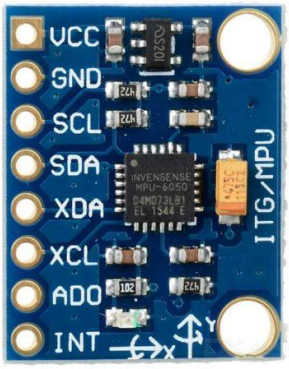


Image du MPU6050

Connexion nécessaire :
VCC connecté à l'alimentation 5V de la carte. Attention à ne pas le connecté à l'alimentation 3,3V comme indiqué sur la datasheet du composant. En effet, le MPU6050 nécessite 3.3V pour fonctionner, mais la carte bleue possède un adaptateur de tension qui fait passer tension de 5V à 3,3V.
GND à la masse de la carte.
SCL au SCL de la carte (protocole de communication I2C).
SDA au SDA de la carte (protocole de communication I2C).

Et les autres broches ?
Une autre broche intéressante est la broche INT qui permet de récupérer une interruption sur certains évènements (non utilisé ici). La broche AD0 permet de configurer l'adresse du module : 68h (par défaut) si la broche est à la masse et 69h si elle est au 3.3V, ce qui permet de commander deux circuits (non utilisé ici). Les autres broches XDA et XCL peuvent être utilisées pour connecter des modules additionnels en I2C, un magnétomètre par exemple (non utilisé ici).

Pourquoi utiliser l'accéléromètre :
Seul l'accéléromètre nous sera utile pour ce projet. L'accéléromètre est un composant capable de détecter les accélérations suivant les trois axes x, y et z, l'axe z étant la verticale. Si la télécommande est tenue horizontalement, les axes x et y ne détectent que de légère vibration et du bruit. Si l'on penche la télécommande, la gravité se fait ressentir suivant x et/ou y. L'accéléromètre permet donc de détecter l'inclinaison de la mallette.

Fonctionnement de l'accéléromètre :
Un accéléromètre fonctionne sur le principe de l'effet piézoélectrique. Imaginons une boîte cuboïde avec une petite boule à l'intérieur, comme sur l'image ci-contre : Les parois de cette boîte sont réalisées avec des cristaux piézoélectriques. Chaque fois que l'on incline la boîte, la balle est obligée de se déplacer dans le sens de l'inclinaison en raison de la gravité. Le mur avec lequel la balle entre en collision crée de minuscules courants piézoélectriques. Il y a trois paires de murs opposés dans un cuboïde. Chaque paire correspond à un axe dans l'espace 3D : les axes x, y et z. En fonction du courant produit par les parois piézoélectriques, nous pouvons déterminer la direction de l'inclinaison et son amplitude.

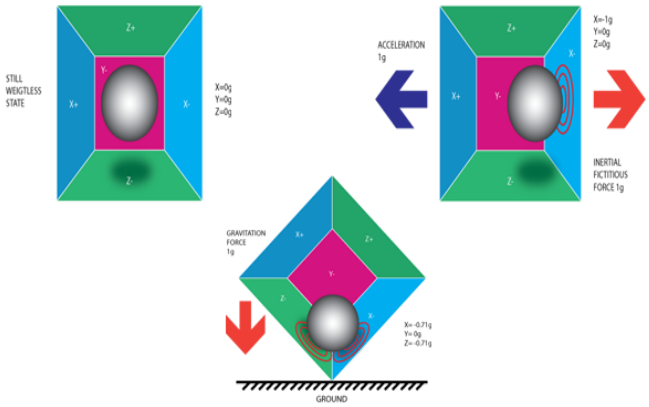


Image du principe de fonctionnement du MPU6050

Caractéristique de l'accéléromètre utilisé :
Accéléromètre Tri-Axis avec une pleine échelle programmable de ±2g, ±4g, ±8g et ±16 g.

Programmation :
Il faut remettre à 0, calibrer et initialiser le capteur avant de l'utiliser. Ensuite il suffit de lire les registres dans lesquels les accélérations sons stockés pour connaitre les accélérations suivant x, y et z.

Communication I2C :

Il faut ensuite être en mesure de transmettre les données du capteur à la carte. Pour cela, la datasheet du MPU6050 indique que la communication I2C s'impose.

Le protocole I2C :

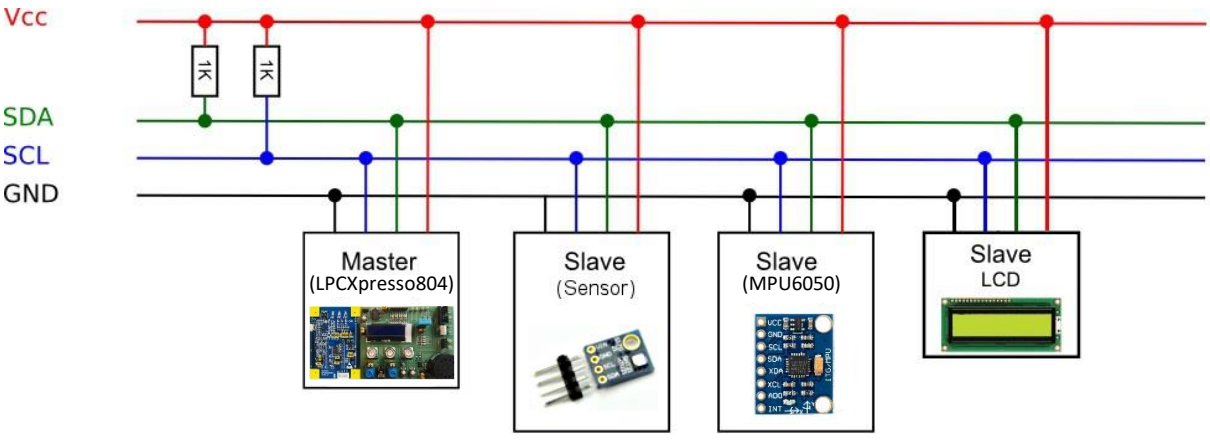
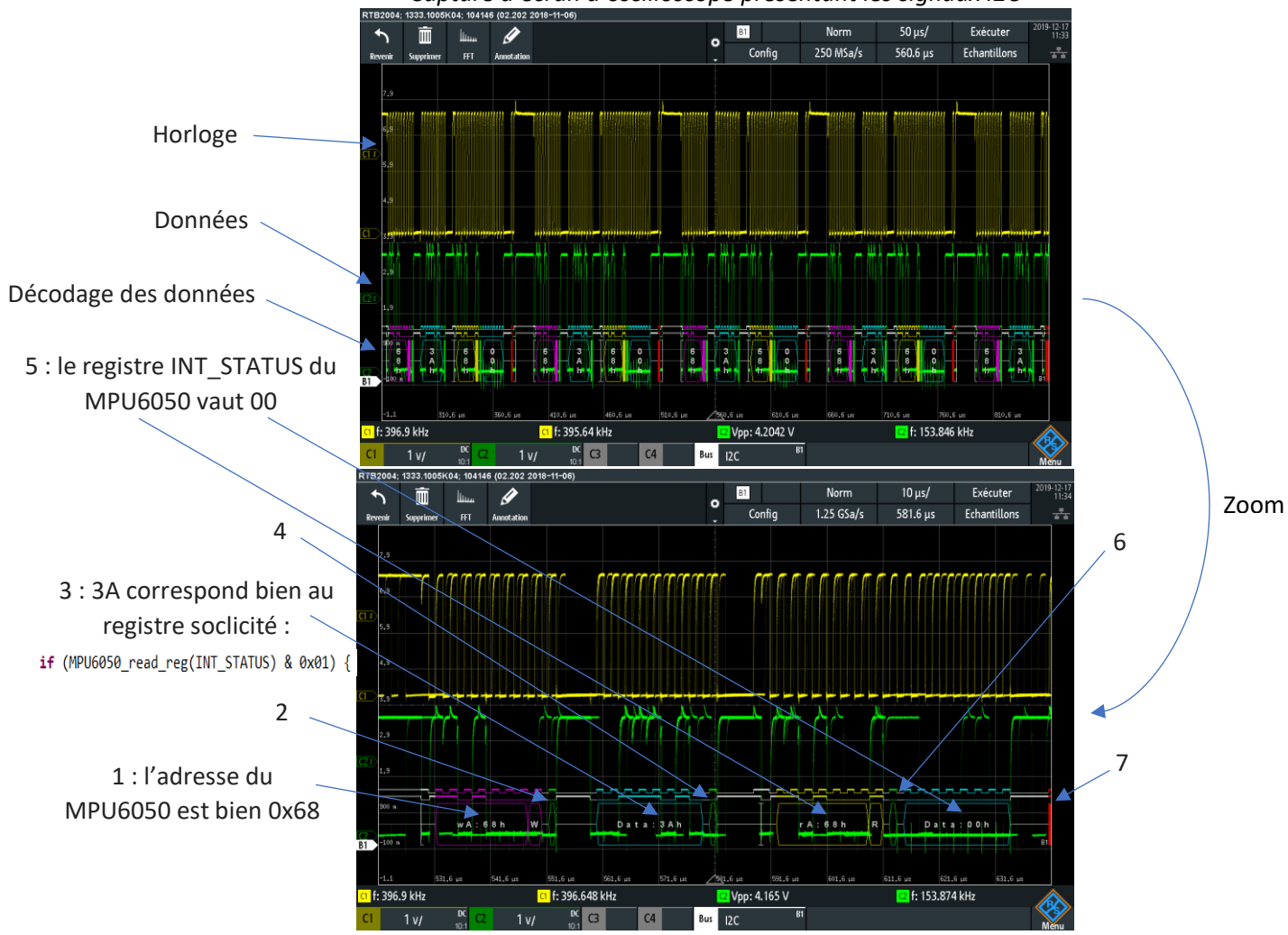


Image du principe de fonctionnement de la liaison I2C

I2C est un protocole de communication crée pour standardiser l'échange de données entre différents circuits intégrés d'un même système. En pratique, le bus I2C est constitué de deux câbles, un pour les données, nommé SDA (Serial Data) et l'autre faisant office d'horloge pour déterminer la fréquence de la communication, nommé SCL (Serial Clock).
Dans la mesure où tous les périphériques sont connectés sur le même bus, comment donc faire pour qu'un seul des capteurs ne communique ses données en même temps et éviter une saturation de la ligne ? Ceci est garanti par le protocole à proprement parler.
Tout d'abord, il n'y a qu'un maître, la carte LPCXpresso804 et une multitude d'esclaves, nos périphériques, chacun identifié par une adresse unique. Seul le maître peut initier une communication. Le maître envoie tout d'abord l'adresse du périphérique dont il désire recevoir les données (1 sur la photo). L'esclave envoie un premier signal de confirmation pour signifier qu'il a bien reçu la demande (2 sur la photo). Puis le maître envoie l'adresse d'un registre interne du périphérique (3 sur la photo). Pour un accéléromètre, on a trois registres stockant respectivement les données de l'accélération selon x, y et z. Un deuxième signal de confirmation est envoyé par le périphérique (4 sur la photo). Enfin c'est le périphérique qui émet cette fois le message, en transférant la valeur du registre qui a été sollicité (5 sur la photo). Il termine avec une dernière confirmation (6 sur la photo), après quoi le maître envoie un signal spécifique pour mettre fin à la communication (7 sur la photo).

Nous avons relevé à l'oscilloscope les signaux issus de la communication I2C :

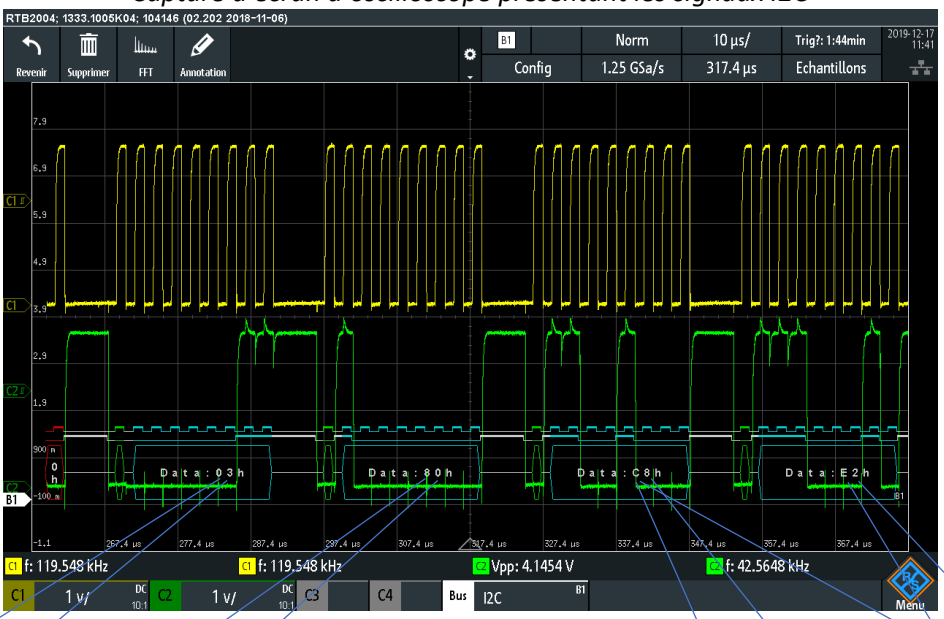
Capture d'écran d'oscilloscope présentant les signaux I2C

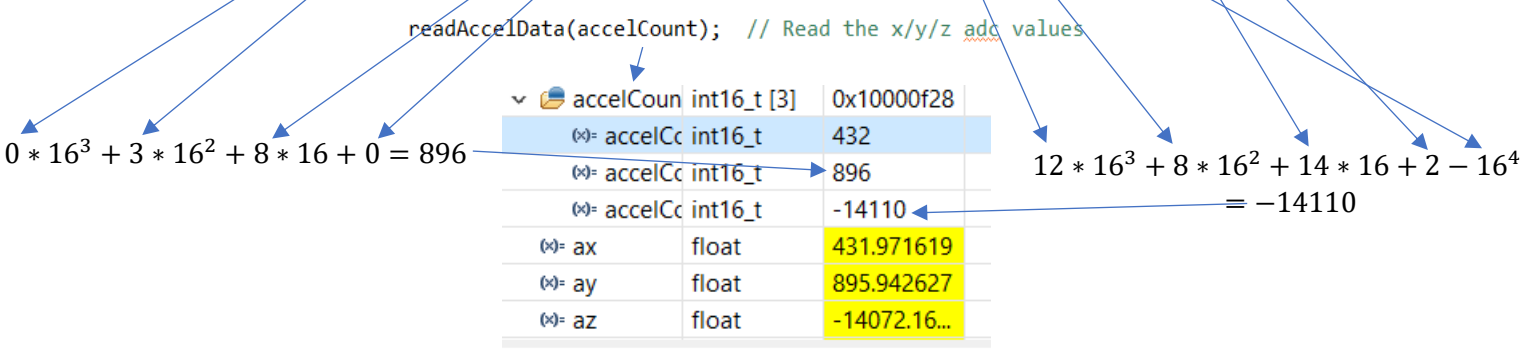


Capture d'écran d'oscilloscope présentant les signaux I2C

Avec l'exemple ci-dessus, on constate que la liaison I2C est correctement établit. Cependant on ne sait toujours pas ce qui est lut dans les registres associés aux accélérations suivant x, y et z (ce qui nous intéresse pour le projet). Avec un point d'arrêt placé après la lecture d'un des accélérations suivant x, y et z, on peut observer :

Capture d'écran d'oscilloscope présentant les signaux I2C





On constate que la communication I2C fonctionne et transmet des valeur d'accélération dans l'ordre attendu : suivant x (non visible ici), puis y et enfin z.

Remarque : il est important de noté que des résistances de tirages ont dû être ajouté sur la carte LPCXpresso804 afin de permettre le bon fonctionnement de l'I2C :

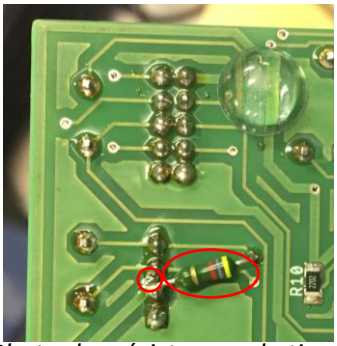


Photo des résistances de tirages

Sans ces dernières, la communication I2C n'était pas fonctionnelle.

La carte LPCXpresso804 :

Le signal issu du capteur est désormais réceptionné par la carte grâce à la communication I2C. Ce signal va être rapidement traité : si l'inclinaison de la carte est nulle ou légère, nous imposerons la valeur de l'accélération à 0 afin que les joueurs ne se déplacent pas en permanence (même avec la mallette horizontale) à cause du bruit et des légères vibrations. Si l'inclinaison est suffisamment élevée, le signal transmis sera du « tout ou rien » pour être compatible avec le jeu de foot d'origine qui fonctionnait en « tout ou rien ».

La liaison série :

La carte transmet une chaine de 4 caractères au module Bluetooth. Le premier indique l'inclinaison suivant l'axe x. S'il vaut 1, le joueur se déplacera vers le haut. S'il vaut 2, le joueur se déplacera vers le bas. S'il vaut 0, le joueur ne se déplacera pas verticalement. Le second indique l'inclinaison suivant l'axe y. S'il vaut 1, le joueur se déplacera vers la droite. S'il vaut 2, le joueur se déplacera vers la gauche. S'il vaut 0, le joueur ne se déplacera pas horizontalement. Les deux caractères suivants permettent d'effectuer le retour chariot et le retour à la ligne. Ils servent à réduire légèrement le retard temporel entre l'inclinaison de la télécommande et le déplacement d'un joueur provoqué par la connexion Bluetooth des deux télécommandes à la fois.

Fonctionnement :
Les dispositifs de transmission série à bas débit ou à courte distance peuvent fonctionner comme des bus électroniques, avec plusieurs composants connectés sur la même ligne. Le protocole de communication détermine quel composant peut émettre ; les autres reçoivent tous l'information et l'exploitent si elle leur est destinée. La distance ne peut dépasser le quart de celle que le signal peut parcourir dans la durée la plus courte entre deux états successifs.
Les dispositifs sans fil fonctionnent de la même manière avec des rayonnements infrarouges, ou, plus souvent, des ondes radioélectriques à très haute fréquence.

Voici comment l'information est transmise sur la ligne :

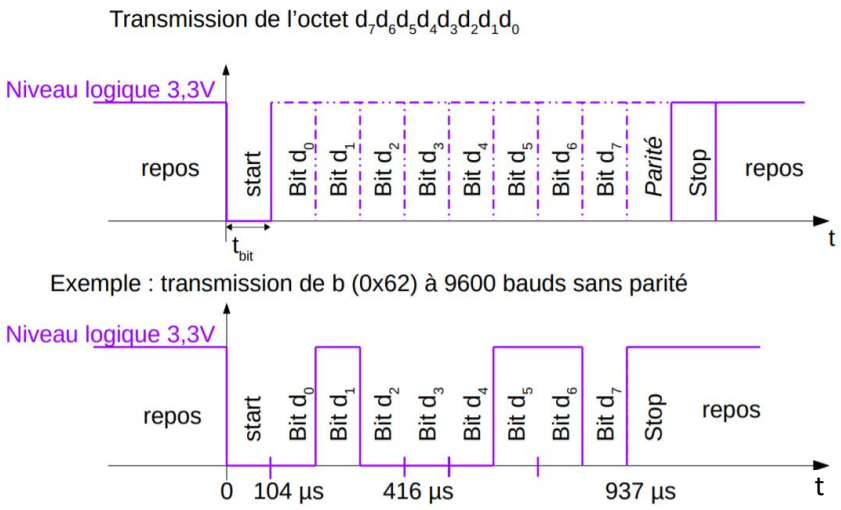


Image du principe de fonctionnement de la liaison série

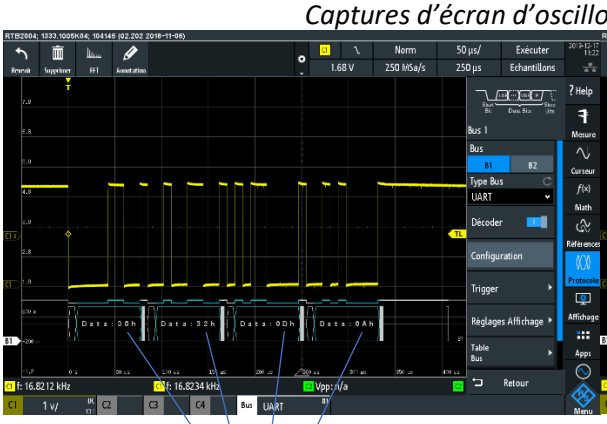
Regardons ce que nous avons avec l'oscilloscope :

Captures d'écran d'oscilloscope des signaux envoyé

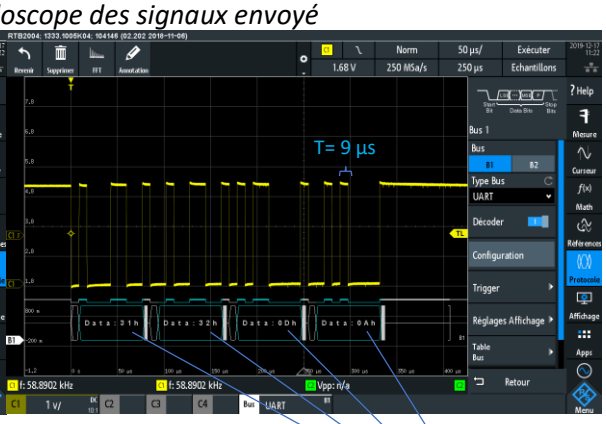
Inclinaison « nul » : envoie de « 00\r\n »

Inclinaison à droite (suivant y) : envoie de « 01\r\n »

Captures d'écran d'oscilloscope des signaux envoyé



Inclinaison à gauche (suivant y) : envoie de « 02\r\n »



Inclinaison en avant (suivant x) et à gauche (suivant y) : envoie de « 12\r\n »

La liaison série est fonctionnelle. La communication fonctionne bien à 115200 bits par secondes :

$\frac{1}{T} = \frac{1}{9\times10^{6}} \approx 115200$ bits/s

Remarque : on a fait attention à envoyer 0 1 ou 2 avec la carte et non -1, 0 et 1 car le nombre de caractère envoyé varierait avec le signe « - » compliquant le code de réception.

Le module Bluetooth :

Image du module Bluetooth

Bluetooth est une technologie de réseau personnel sans fil. Contrairement à la technologie de liaison infrarouge, les appareils Bluetooth ne nécessitent pas d'une ligne de vue directe pour communiquer, ce qui rend plus souple son utilisation et permet notamment une communication d'une pièce à une autre, sur de petits espaces. L'objectif du Bluetooth est de permettre de transmettre des données entre des équipements possédant un circuit radio de faible coût, sur un rayon de l'ordre d'une dizaine de mètres à un peu moins d'une centaine de mètres et avec une faible consommation électrique. Le Bluetooth permet d'obtenir des débits de l'ordre de 1 Mbps.

L'ordinateur :

Une fois les données Bluetooth envoyées, il a fallu que l'ordinateur les récupère. Pour ce faire, il faut commencer par connaitre les ports Bluetooth associé à la carte/module Bluetooth. Pour cela, il faut connecter en Bluetooth l'ordinateur et le module Bluetooth puis aller dans la gestion des périphériques de l'ordinateur pour trouver le port Bluetooth associé. Ensuite, un bout de code permet d'ouvrir les ports en question et d'extraire les données émises par la carte.

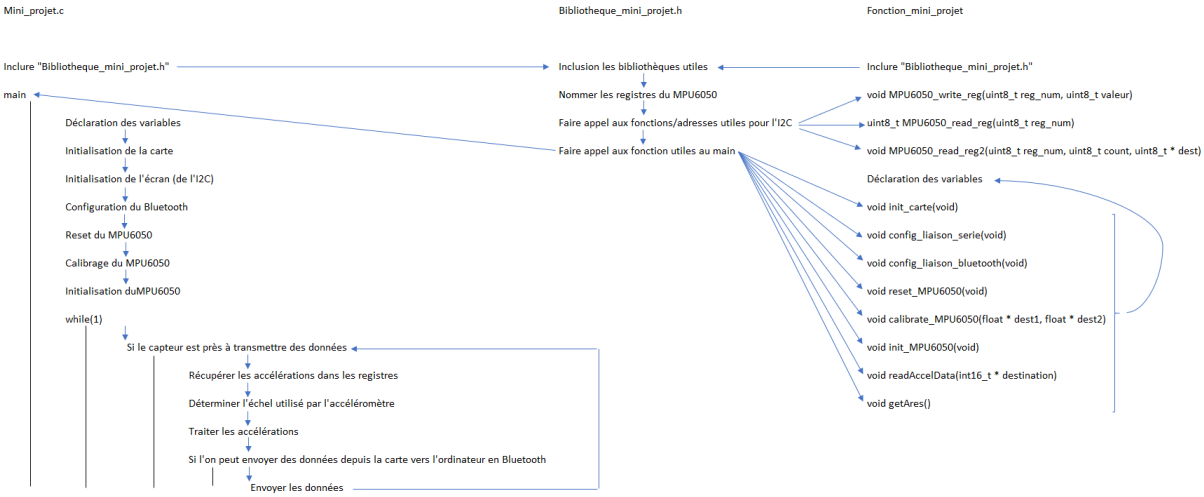
Remarque 1 : il est impossible d'ouvrir 2 terminales associés au même port en même temps.

Remarque 2 : le bout de code permettant d'ouvrir les ports le plus rependu sur internet ne permet d'ouvrir que les ports allant de 1 à 9. Il a fallu le modifier pour le rendre fonctionnel pour tous les ports.

L'extraction de donnée et son traitement se fait sans difficulté. Il faut adapter le code du jeu de foot pour qu'au lieu d'être sensible au clavier il devienne sensible aux inclinaisons de la mallette.

Le jeu fonctionne très bien avec une seule télécommande. L'ajout d'une deuxième télécommande provoque un retard temporel de l'ordre de la seconde entre l'inclinaison de la mallette et le déplacement d'un joueur. Pour remédier à ce retard, nous avons effectué un retour chariot et un retour à la ligne des données transmise par la carte. Malgré cela, le retard reste supérieur à la demi-seconde. Il n'est malheureusement pas possible d'y remédié car l'une des limites du Bluetooth est la difficulté qu'il a à traiter les données provenant de plusieurs appareils en même temps. Nous avons donc ralenti la vitesse du jeu en incluant un temps d'attente dans la boucle. Ainsi, la balle et les joueurs vont moins vite ce qui permet aux personnes manipulant les mallettes de mieux s'adapter à ce retard.

Représentation graphique de l’algorithme :



Conclusion :

Le projet fut achevé avec succès. Tous les objectifs ont été atteint. Les joueurs 1 et 2 se déplacent en fonction de l'inclinaison de leur télécommande respective grâce à une connexion Bluetooth. Nous avons eu l'occasion de revoir plusieurs notions tels que la liaison série et la liaison I2C et d'en découvrir de nouvelles tel que l'utilisation du MPU6050 et l'ouverture de ports série sur l'ordinateur. La seule limite ayant un impact sur le fonctionnement du jeu est la perte de performance de la connexion Bluetooth quand plus d'un composant est connecté. Ce problème aurait pu être résolu en utilisant une liaison série filaire, mais l'énoncé du projet imposait l'utilisation de la connexion Bluetooth.