

## COMMENT FAIRE DE SON RASPBERRY PI UN INSTRUMENT DE MUSIQUE?

### I) Introduction :

A l'origine, nous avons comme projet de réaliser un programme pour calculer la vitesse d'un sprinteur lors d'une course. Mais nous avons rencontré quelques problèmes, notamment concernant la connexion de différents raspberry en réseaux. Nous sommes ensuite venus à la conclusion que le projet serait difficile à présenter et que les capteurs ne répondaient pas totalement à nos attentes (les capteurs ultrasons ne prenaient pas toujours le passage du « coureur » en compte). Nous nous sommes donc tournés vers un projet de conception d'un instrument de musique qui utiliserait toujours le capteur ultrason tout en s'inspirant du fonctionnement du theremine.

Les objectifs de notre projet sont donc :

- jouer des notes de musique à l'aide d'un capteur ultrason
- écouter ces notes « en direct »
- offrir à l'utilisateur la possibilité de réécouter le morceau en totalité

### II) Mode d'emploi :

- 1) Equipez votre Raspberry pi 3 (sous Raspbian) d'un shield. Branchez un capteur ultrason au port D4 et un bouton au port D3.
  - 2) Branchez le Raspberry pi à un cable ethernet, un clavier, une souris, des haut parleurs ainsi qu'un écran.
  - 3) Ouvrez le logiciel Sonic Pi( Sonic Pi v2.11.1) et charger le fichier « instrument.txt ».
  - 4) Lancez la lecture du fichier « instrument » depuis Sonic pi ( appuyer sur le bouton « Run » ) .
  - 5) Lancez le programme « instrument » depuis le terminal du raspberry via la commande « sudo ./instrument »
  - 6) Placez votre main au-dessus du capteur ultra-son pour générer du son.
  - 7) Jouez avec la distance pour créer des notes différentes.
  - 8) Appuyez sur le bouton une fois que vous avez fini de jouer.
  - 9) Réappuyez si vous voulez réécouter la musique.
-

### III) Moyens Matériels :

Pour le projet nous avons utilisé un raspberry pi, un shield, un cable ethernet, un capteur ultrason, un bouton, des hauts parleur, un clavier, une souris, un cable HDMI ainsi qu'un écran. Ces choix ont été surtout motivés par notre première idée de projet qui nécessitait d'utiliser ces mêmes capteurs à des distances trop grandes pour n'être reliés qu'à un seul raspberry (il fallait donc les mettre en réseau). Nous avons donc choisi de garder ces capteurs pour notre nouveau projet afin de garder notre méthode de travail et quelques éléments déjà codés. De plus, ces capteurs nous semblaient pertinents pour notre nouveau projet car nous avons pu utiliser le logiciel Sonic Pi, propre au raspberry, pour pouvoir jouer des notes de musique. Nous avons choisi un capteur ultrason car il nous paraît assez sensible et pratiques pour notre projet.

Tout au long de ce dernier, nous avons créer un répertoire Github qui nous servait de dépôt pour les codes.

### IV) Moyens Humains :

Nous avons toujours travaillé en duo. Lorsque nous ne savions pas comment faire une partie l'une de nous deux se détachait parfois pour faire quelques recherches et revenir plus tard avec un bout de code mais de manière général nous nous voyions toujours pour avancer sur le projet et avons effectué un travail égal. Lors des tests que nous faisons, l'une de nous testait faisait les tests tandis que l'autre s'occupait de lancer les tests, et permutations les rôles les fois d'après.

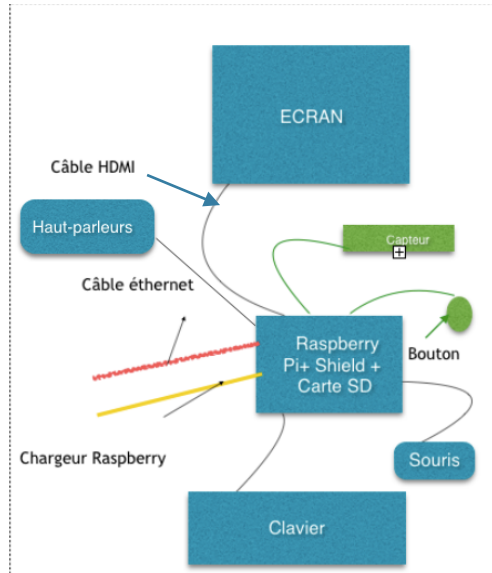
### V) Architecture :

Notre projet nécessite l'utilisation de l'application « Sonic Pi ». Nous l'utilisons afin de lire un fichier texte qui contient des nombres correspondant à des notes de musique lorsqu'elles sont lue par Sonic Pi. Lorsque « instrument » est lancé, on charge le fichier « instrument.txt » dans Sonic Pi, et on le « Run ». On lance alors le programme. Une fois celui ci le lancé, le capteur ultrason récupère la distance de l'obstacle le plus proche de lui jusqu'à ce qu'on appuie sur le bouton. Selon la valeur de cette donnée, une note sera copiée dans les fichiers appelés « musique\_live.txt » et « musique.txt ». La copie dans le premier fichier se fait avec écrasement de l'ancien contenu de ce dernier. Ainsi dans ce fichier, seule une seule note sera écrite. Contrairement à cette copie, celle qui s'effectue dans « musique.txt » n'écrase pas le contenu avant la copie. Aussi, ce dernier contient toutes les notes jouée par l'utilisateur depuis le lancement du programme. Pour mettre fin à sa compilation de musique, l'utilisateur doit appuyer sur le bouton. Ainsi, lorsqu'il aura fini de jouer son morceau, il devra appuyer sur le bouton dans les 10 secondes qui arrivent s'il veut écouter la totalité de la musique qu'il a composé. Alors le contenu de musique.txt sera copié dans musique\_live.txt et la lecture de ce dernier est effectué. Si par contre il n'a pas appuyé sur le bouton une deuxième fois, le fichier musique.txt est supprimé.

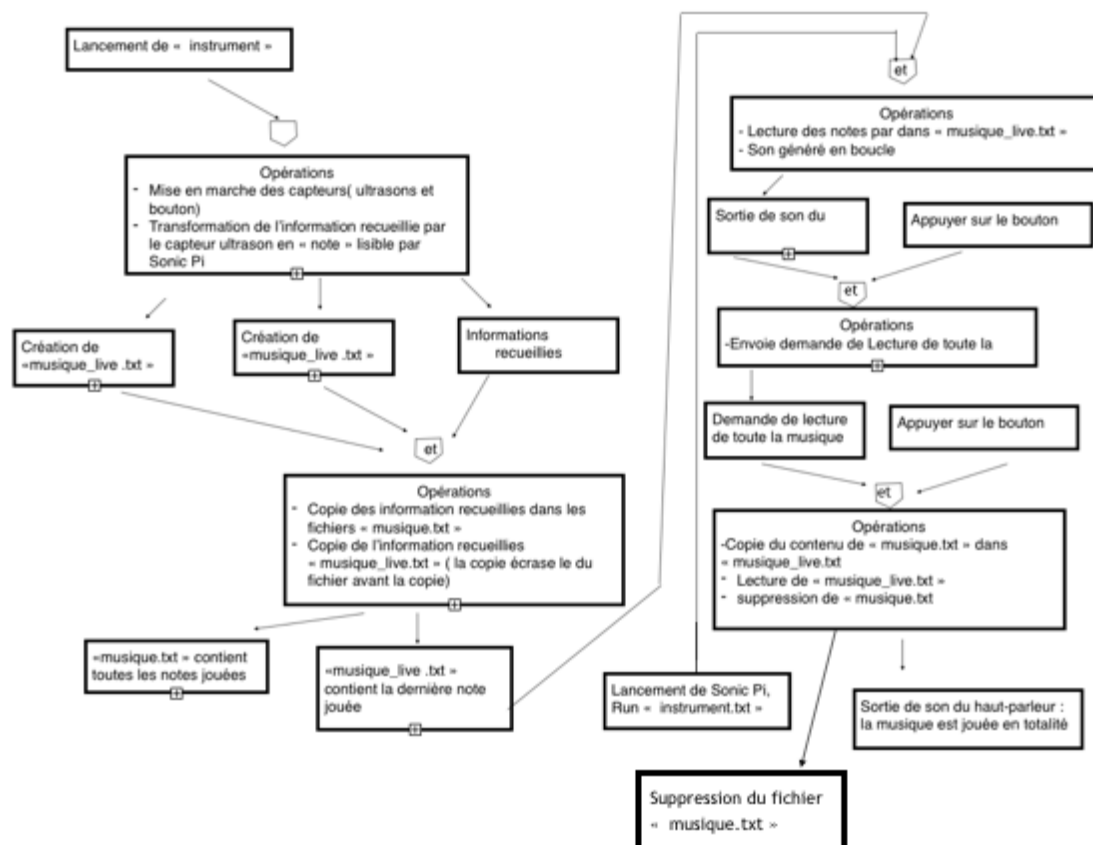
---

## COMMENT FAIRE DE SON RASPBERRY PI UN INSTRUMENT DE MUSIQUE?

Ci dessous figurent des schémas de l'architecture de notre projet:



Document 1 : Schéma du matériel utilisé pour ce projet



Document 2 : Schéma des opérations effectuée lors du lancement du programme

### VI)Code :

Dans « instrument.txt » nous avons utilisées les fonctions et procédures suivantes :

- note\_from\_text(T): retourne les caractères avant un « . » sur une ligne
- to\_i : transforme un caractère en entier

Dans « instrument.c » nous avons utilisé les fonctions et procédures suivantes :

- digitalRead(Int) : pour lire la valeur du bouton ( pour faire le test si on avait ou non appuyé sur le bouton)
- write\_block( char, char, char, char) : fonction définie dans le fichier grovepi.c
- pi\_sleep(Int) : fonction définie dans grovepi.c, permet de faire une pause dans l'exécution des instructions dans le main
- read\_byte(void), read\_block(void) : fonctions définies dans grovepi.c
- fopen(), fprintf(), fclose() : pour ouvrir( ou créer si le fichier n'existe pas), écrire et fermer un fichier( en l'occurrence ici cela ces fonctions sont utilisée pour le fichier texte lu dans Sonic Pi)
- printf() : affichage à l 'écran
- system(« ») : permet d'exécuter des commandes depuis le fichier instrument.c, dans notre programme il s'agissait de supprimer un fichier.

Sources :

<http://sonic-pi.net>

<https://github.com/stevelloyd/Learn-sonification-with-Sonic-Pi/blob/master/worksheet.md>

---

### VII) Perspectives :

Nous avons fait le choix en cours de route de recommencer un nouveau projet car nous n'étions plus convaincues par le premier, nous en étions au point mort. Nous avons rencontré multitudes de problèmes techniques et organisationnel (travailler sur plusieurs raspberrys/ capteurs etc ...). Evidemment cela a réduit le temps que nous avons consacré au projet. Nous avons donc fourni beaucoup de notre temps à la réalisation de ce projet dans les délais. Il est actuellement opérationnel cependant nous pensons que l'amélioration de quelques éléments pourrait le rendre meilleur. Notamment offrir le choix à l'utilisateur de changer la sonorité, lancer le programme de manière automatique, permettre à l'utilisateur de lancer le programme par le son de sa voix, permettre à l'utilisateur de télécharger sa musique, ...

Quant à sa commercialisation, ce produit pourrait convenir à des personnes en apprentissage de musique ou encore des personnes qui ne jouent pas d'un instrument mais qui veulent faire de la musique d'une manière « facile » et automatisée. Ce produit ciblerait donc les amateurs de musique, ce qui représente un vaste public. Par ailleurs le marché des instruments de musique subit une baisse importante en France ces dernières années. De moins en moins de personnes achètent des instruments de musique. Dans l'ère actuelle du numérique, ce produit pourrait se vendre à plusieurs exemplaires. Son prix pourrait s'élever à une soixantaine d'euros ( en incluant le raspberry ), ce qui serait un peu trop onéreux au vu de son utilité secondaire. C'est pourquoi pour sa commercialisation nous pensons que ce serait plus rentable de vendre le programme en lui-même et par conséquent l'adapter à une autre application de lecture de musique qui serait présente sur le l'ordinateur de l'acheteur.

Malgré tout nous ne regrettons pas notre changement de sujet car nous avons réalisé plus de choses, à force de travail acharné, en moins de deux semaines qu'en un mois et notre programme est opérationnel et ludique.

---