LOUBIERE Clément
AURARD Thaïs

Synthèse de projet

Introduction:

Il s'agit dans ce projet d'utiliser deux raspberry pi pour diffuser de la musique générée par des conditions extérieures. Un raspberry équipé d'un shield recueille les données de capteurs, et un autre, situé ailleurs, reçoit ces données et les interprète.
Cette musique pourra varier selon différents facteurs tels que la distance au capteur à laquelle passent les personnes présentes dans la salle et le volume sonore des conversations.
Les valeurs de volume sonore seront "traduites" et interprétées comme des notes par le logiciel Sonic Pi sur raspberry tandis que les valeurs de distance détermineront avec quel son (quel synthétiseur sur Sonic-Pi) les notes seront produites.

Mode d'emploi:

- Branchement des deux raspberry (alimentation et réseau par câble RJ45)
- Branchement des capteurs sur le raspberry 24, capteur son sur le port A0 et capteur ultrasons sur le port D4
- Branchement d'écouteurs ou d'enceintes sur le raspberry 18
- Connexion à la session utilisée sur chaque raspberry.
- Installation et lancement du logiciel Sonic-Pi sur le raspberry 18
- Chargement du fichier sonic.txt avec Sonic-Pi
- Exécution sur le raspberry 24 des exécutables server et ultraServer sur un numéro de port choisi, et des exécutables bashClient et bashUltraClient sur le raspberry 18

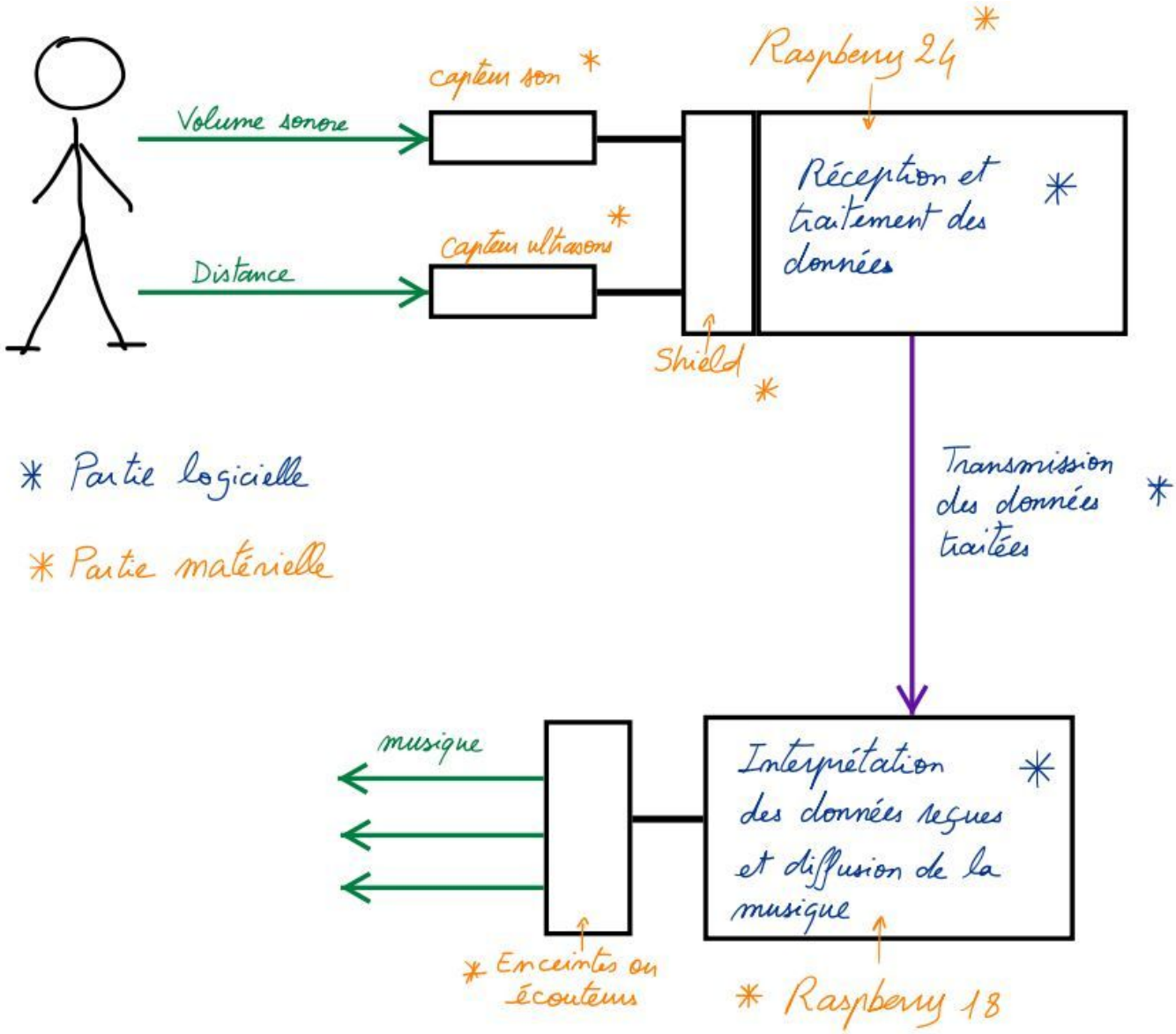
Moyens matériels:

- Deux Raspberry Pi 3 (un client et un serveur) et leurs alimentations
- Deux câbles RJ45 pour la connexion sur le réseau
- Un capteur de sons
- Un capteur d'ultrasons
- Une paire d'écouteurs ou des enceintes
- Un écran, un clavier et une souris (car le logiciel Sonic-Pi ne peut s'éxécuter sans écran)

Moyens humains:

Nous avons systématiquement travaillé ensemble sur le projet lors des heures d'autonomie.
Clément Loubière a apporté son expérience d'IUT en codant les sockets et en améliorant le code à tous les niveaux, notamment celui des drivers.
Thaïs Aurard a écrit le fichier Sonic lu par Sonic-Pi.
Les réflexions sur les buts et la mise en place du projet ont été menées en commun.

Architecture:



Code:

Dans le fichier grovepi_soundlib.c, la fonction recupValSon() récupère la valeur captée par le capteur de son et la fonction recupValUltraSon récupère la valeur captée par le capteur ultrasons.
Le fichier serverBis.c crée une socket en écoute sur le port correspondant à l'entier passé en paramètre. Lorsqu'un client se connecte, il récupère les données du capteur de son et les transmet divisées par 3 afin qu'elles correspondent aux valeurs lues par Sonic-pi. Le fichier serverBisUltra.c a la même fonction mais récupère les données du capteur d'ultrasons.
Le fichier clientBis.c crée une socket de connexion qui prend en paramètres une IP et un port et se connecte au serveur correspondant. Une fois connecté il reçoit des données et les stocke dans un fichier texte nommé données.txt correspondant aux données du capteur de son. Le fichier clientBisUltra.c a la même fonction mais stocke les données reçues dans un fichier nommé distance.txt correspondant aux données du capteur d'ultrasons.
Les fichiers bashClient et bashUltraClient sont des fichiers servant à exécuter clientBis.c et clientBisUltra.c en boucle afin d'avoir un résultat en temps réel. Il faudra les modifier selon sur quel port on veut se connecter.
Dans le fichier sonic.txt, grâce à la fonction quantity_to_sound(quantity), Sonic-pi n'interprètera pas le point et les caractères invisibles qui suivront dans la valeur passée en paramètre. La fonction recup() sert à ouvrir dans Sonic-pi le fichier texte passé en paramètre de File.open(fichier) et stocke chaque ligne dans une variable que l'on passe ensuite en paramètre à la fonction quantity_to_sound. La valeur obtenue est renvoyée par la fonction recup().
La fonction playing() évalue en boucle (grâce au loop) la valeur renvoyée par recup et définit un synthétiseur en fonction de cette valeur. Ce synthétiseur sera ensuite utilisé pour émettre la note correspondant à ce qui sera lu à chaque ligne dans le fichier données.txt en paramètre du File.open.
Dans recup() on lit dans le fichier distance.txt correspondant aux ultrasons et dans playing() on lit les données du fichier données.txt correspondant au volume sonore.

Directives de compilation:
gcc -Wall serverBis.c grovepi.c -o server
gcc -Wall serverBisUltra.c grovepi.c -o ultraServer
gcc -Wall clientBis.c -o client
gcc -Wall clientBisUltra.c -o ultraClient

Les sockets permettant la communication entre les deux Raspberry ont été codés en C.
Concernant les drivers, nous sommes partis de la correction de ceux faits en TP et nous les avons modifiés pour qu'ils correspondent à nos besoins.
Pour la partie interprétation d'un fichier texte par Sonic-pi, nous nous sommes aidés d'un code trouvé sur internet et qui permet de mener la même expérience avec en données des températures. Il a fallu modifier la fonction quantity_to_temp afin qu'elle corresponde aux

données de notre fichier texte et nous n'avons pas eu besoin de la fonction scale car nous réduisons déjà nos données au domaine de l'audible avant l'écriture sur le fichier texte.
https://github.com/stevelloyd/Learn-sonification-with-Sonic-Pi/blob/master/worksheet.md

Perspectives:

Ce projet n'a pas été pensé avec une visée commerciale; il s'agit plus d'une démarche artistique. Cela dit, un appareil qui produit une musique d'ambiance en fonction de l'ambiance pourrait trouver sa place sur le marché. Il faudrait alors plus de capteurs pour plus de précisions et plus d'options de diffusion et la possibilité pour l'utilisateur de régler lui même les paramètres et les seuils. Un handicap possible est que pour le moment, les deux Raspberry ne peuvent se trouver dans la même pièce à moins d'utiliser des écouteurs. En effet, le son produit par le Raspberry 18 ne doit pas venir perturber le capteur son du Raspberry 24. Un autre obstacle à la commercialisation pourrait être le fait que le son produit n'est pas obligatoirement mélodieux. Les notes émises le sont en fonction des valeurs reçues par les capteurs.
Il faudrait compter au moins 120€ pour les deux Raspberry, le Shield et les capteurs, en admettant que l'on pourrait brancher le dispositif à un écran, une souris, un clavier et des enceintes déjà existants chez l'utilisateur.