# **Cas d’usage : Décibelmètre**

**Ce qui est fait** :

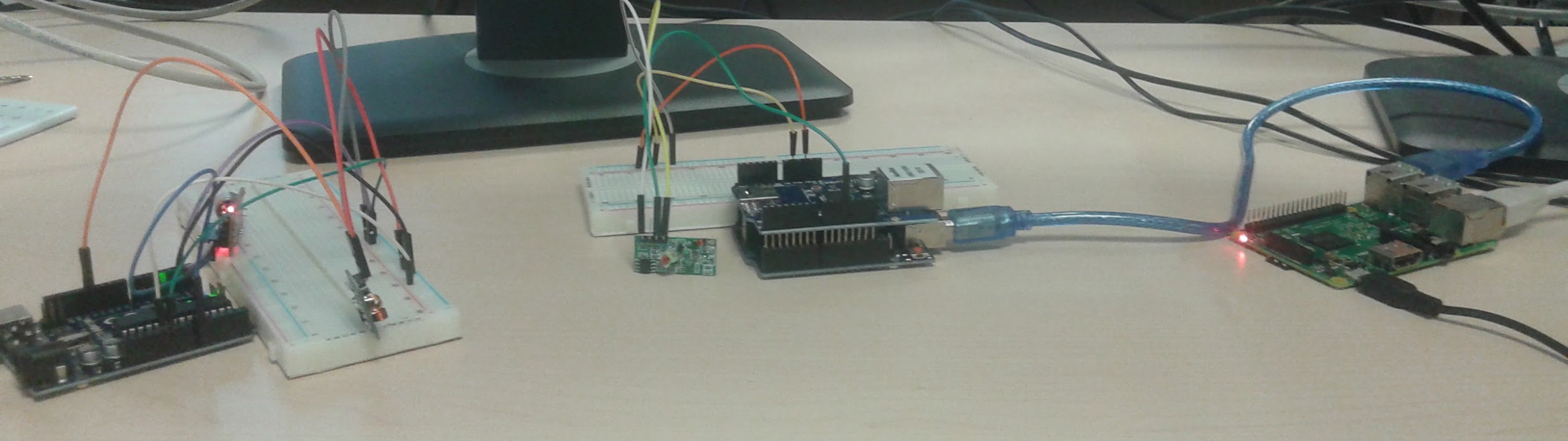
**Arduino 1** : Le capteur de son est dessus. Cet arduino comporte un module RF emetteur. Il envoie toutes les demi secondes si le nombre de décibels dépasse un certain “seuil” -> Celui qui est supérieur à 0 (reste à affiner).

**Arduino 2** : Arduino relié par USB au Raspberry. Module RF recepteur relié à l’arduino.

Cet arduino rècupère les données envoyées par Arduino 1.

**Raspberry** : Installation de raspian ok. ssh ok. Mise en place d’un script python qui lit ce que l’arduino reçoit par le biais de l’USB. Ce script envoie la donnée lue au broker qui est installé sur son système (Mosquitto).

Le broker a été mis en place pour tester notre système mais peut etre revu par une autre équipe sans problème.



Sketch arduino et python mis sur gitlab.

**Problèmes rencontrés** :

- Régler le potentiomètre du décibelmètre (étalonnage)

- Faire communiquer en RF un arduino et raspberry (sans passer par un 2ème arduino)

- Distance entre émetteur / récepteur en RF

**Reste à faire** :

- Régler potentiomètre du decibelmètre (étalonnage)

- Mise en place d’antennes sur le décibelmètre + tester la transmission à distance (plusieurs mètres

- Régler la fréquence d’envoi et à partir de quel seuil on envoi les données ?

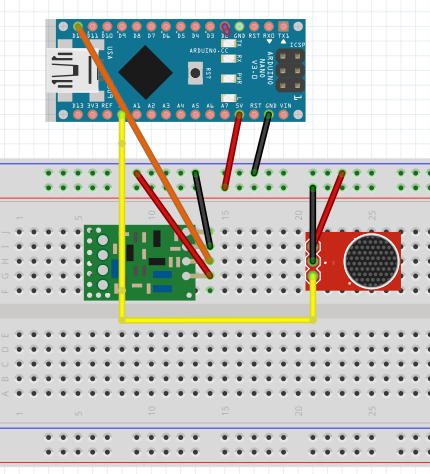
- Faire le montage avec arduino nano

- Prévoir des alimentations pour arduino

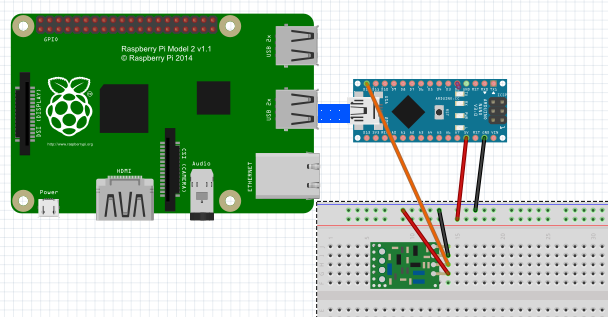
- Affiner le format de la donnée transférée au broker (timestamp + nb décibel)

- Pallier au problème de distance entre émetteur et récepteur RF (test alimentation)

**Montages :**



Arduino 1 (Capteur de son + emetteur RF 433hz)



Arduino 2 (recepteur RF 433hz)

Autre point : penser aux adaptateurs pour le XBee