

Compte rendu projet électronique

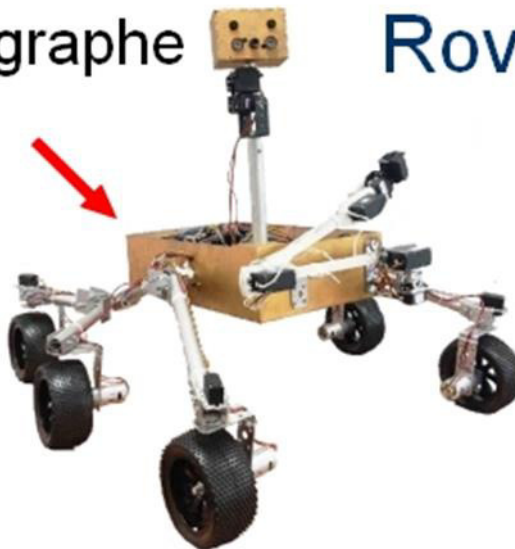
Les trois sismographes

31 mai 2018



Triple sismographe
RF

Rover R6P4



Sommaire

1. Introduction.....	3
2. Description du projet.....	3
3. Déroulement du projet.....	6
4. Objectifs réalisés.....	7
5. Problèmes rencontrés et solutions trouvées.....	7
6. Améliorations possibles.....	10
7. Ce qui nous a plus/moins plu.....	10
8. Conclusion.....	10

I) Introduction

Dans le cadre de notre deuxième année au Parcours des Ecoles d'Ingénieurs Polytech, nous avons à réaliser un projet Arduino au cours du 2nd semestre. Ce projet consiste à réaliser des systèmes complexes et communicants utilisant l'Arduino.

II) Description du projet

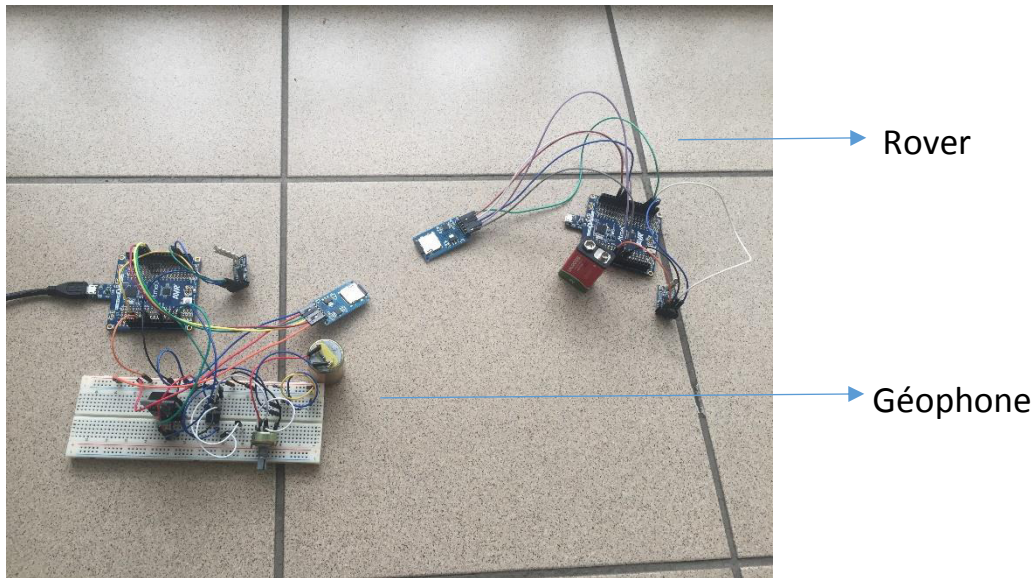
Nous avons choisi de faire un des 3 sismographes du Rover, projet qui a été réalisé en 2017. Ces sismographes doivent enregistrer les vibrations du sol et les envoyer au Rover sous forme d'un graphe. Ici, un premier module RF232 agira comme un émetteur et un deuxième module agira comme un récepteur (soit le Rover).

(Insérer une image du montage complet)

Matériel utilisé :

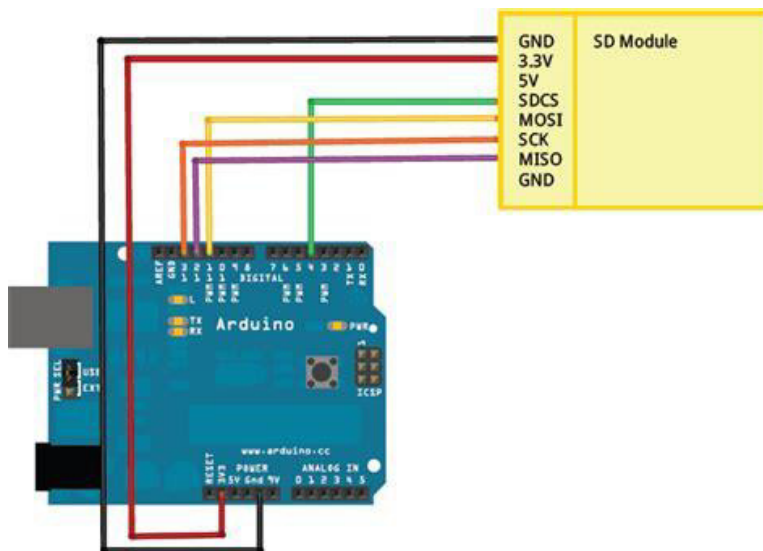
- Deux cartes Arduino nano
- Deux modules RS232
- Une horloge RTC 3231
- Un géophone
- Une résistance
- Un potentiomètre 5kΩ
- Une pile 9V
- Deux cartes SD
- Un AOP

Notre système est constitué de deux parties : celle du Rover (récepteur) et celle du géophone (émetteur).



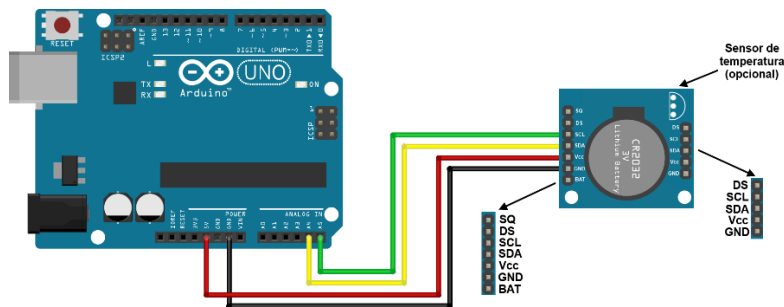
Présentation des circuits :

Le Rover est modélisé par l'ordinateur qui est connecté à un module RF232 et une carte SD par la carte Arduino.

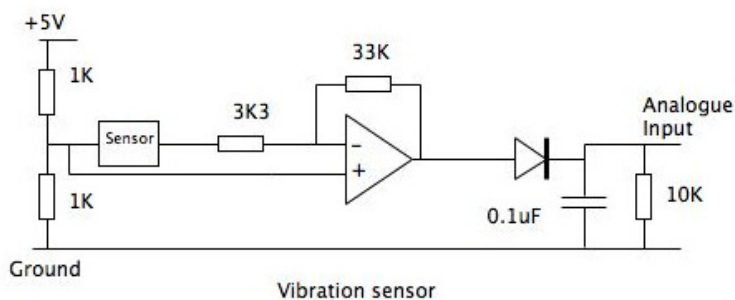


Branchement au module de la carte SD

Le géophone est constitué de : l'appareil de mesure (le géophone), une horloge, une carte SD et un module RF232.



Branchement du module de l'horloge



Circuit pour l'appareil de mesure

Présentation de la programmation :

Le Rover a un seul programme, celle de la réception des données du géophone. Il reçoit les données caractère par caractère, et reconstitue le nombre grâce au caractère de fin \$. On reçoit :

3

.1

4

\$

La fonction reconstitue le tout en : 3.14. Ensuite, il enregistre les nombres reconstitués.

Le géophone possède un programme avec plusieurs fonctions autour d'une seule : la fonction « déclenchement » tourne en permanence pour tester s'il y a un séisme. Dès qu'il y en a un, elle enregistre l'heure actuelle sur la carte SD et

enregistre des données du géophone sur la carte durant 1minute. Une fois l'enregistrement terminé, elle envoie les données précédemment enregistrées vers le Rover.

III) Déroulement du projet

Lors de la première séance nous avons fait des recherches sur les différents projets Arduino possibles. Après plusieurs heures d'hésitation nous avons finalement décidé de participer à la continuation du Rover en faisant la partie des sismographes.

Lors des deux séances suivantes nous avons réfléchi à comment aboutir à un sismographe avec l'Arduino, nous avons fait un plan de ce que nous devons faire et nous avons ensuite présenté notre sujet de projet à nos deux enseignants d'électronique M. Masson et M. Ferrero.

Avant d'avoir nos géophones, nous avons fait nos premiers tests d'enregistrement de vibrations du sol avec un piezo lors de la 4^{ème} séance. Mais nous avons reçu les géophones la semaine d'après donc nous avons pu tester les programmes avec un des géophones. Nous sommes parvenues à visualiser les secousses mesurées avec le logiciel Serial Port Plotter mais les résultats n'étaient pas assez satisfaisants.

Ce n'est que lors de la 6^{ème} séance que nous avons réussi à faire un montage correct avec le géophone qui mesure les vibrations du sol avec une précision optimale. Nous avons aussi commencé à programmer la carte SD pour qu'elle écrive sur un fichier, qu'elle l'enregistre et qu'elle puisse le lire lorsque nous lui demandons. Tous ces programmes étaient fonctionnels.

Lors de la 7^{ème} séance nous nous sommes intéressées à la partie la plus importante de notre projet qui est la communication entre les deux cartes Arduino à travers une liaison RS232. Nous avons rencontré énormément de problèmes, ce n'est que deux semaines après que nous avons pu envoyer un message d'une carte Arduino à une autre.

Entre temps, nous avons amélioré notre programme d'enregistrement sur la carte SD et avons combiné le code de l'enregistrement des vibrations du sol par le géophone avec le code de la carte SD. Nous avons donc pu enregistrer les valeurs en Volt sur un fichier de la carte SD mais les valeurs ne s'affichaient pas

comme nous le souhaitions. Nous avons néanmoins pu résoudre ce problème (seulement lors de l'enregistrement sur la carte SD) en quelques jours.

Les séances suivantes ont été consacrées à la synchronisation de l'horloge RTC3231 et surtout à la communication entre les deux cartes Arduino et l'envoi des données de la carte SD 1 vers la carte SD2.

IV) Objectifs réalisés

Objectifs prévus :

- Réaliser 3 sismographes
- Enregistrer les vibrations du sol sur une carte SD
- Envoyer ces données à une autre carte Arduino pour les enregistrer sur une autre carte SD
- Visualiser un graphe à partir des données reçues
- Déduire l'épicentre à partir des 3 sismographes
- N'envoyer les données que lorsque le Rover les demande

Objectifs réalisés :

- Réaliser un sismographe (avec 1 géophone)
- Enregistrer les vibrations du sol sur une carte SD
- Envoyer ces données à une autre carte Arduino pour les enregistrer sur une autre carte SD
- Visualiser un graphe à partir des données reçues

Nous n'avons donc pas pu réaliser trois sismographes comme il avait été prévu de le faire au début du projet... En effet nous manquions énormément de temps car nous avons rencontré beaucoup de problèmes, notamment lors de l'envoi des données vers une 2^{ème} carte SD.

V) Problèmes rencontrés et les solutions trouvées

Lors de la réalisation de ce projet nous avons rencontré beaucoup de problèmes qui nous ont empêchées d'avancer au rythme prévu.

Ces problèmes rencontrés sont :

- Le géophone qui ne mesure pas de façon précise et qui affiche des valeurs absurdes.

Nous avons dû modifier le montage et ajouter un amplificateur au circuit pour que les valeurs enregistrées soient plus précises.

- La mise à l'heure de l'horloge RTC1302

Au début, nous voulions mettre les horloges à l'heure manuellement, c'est-à-dire en rentrant sur le programme l'heure que nous souhaitons. Or si nous mettons une horloge à l'heure de cette façon alors les autres horloges n'auront pas la même heure à cause du décalage dû à la compilation du programme et à son téléversement. Nous avons donc décidé de synchroniser les horloges (l'horloge plutôt) avec le PC, c'est-à-dire que l'horloge aura l'heure et la date exacte du PC avec lequel le code a été téléversé et ce jusqu'à ce que la batterie de l'horloge ne cesse de fonctionner. Mais un autre problème est survenu...

- La synchronisation de l'horloge RTC1302

Après plusieurs essais de codes, téléchargements de bibliothèque, nous avons compris que ce modèle ne pouvait pas se synchroniser avec le PC. Nous avons donc changé le RTC1302 en RTC3231 lui a une bibliothèque « officielle » sur Arduino, il était donc facile de trouver un code pour la synchroniser avec le PC.

- Les valeurs lors de l'envoi du fichier avec les valeurs enregistrées par le géophone vers une autre carte SD

Les valeurs envoyées n'étaient pas du tout sous la même forme à laquelle elles ont été enregistrées sur la 1^{ère} carte SD. Lors de l'envoi, elles sont enregistrées comme étant des caractères donc les valeurs sont décomposées en plusieurs valeurs à un chiffre, il n'y a qu'un seul caractère/chiffre par ligne. Cela était très embêtant puisque nous avons besoin des valeurs exactes pour réaliser le graphe. Nous avons essayé de créer un tableau avec 4 caractères (pour 3.70 par exemple, le « . » étant le 2^{ème} caractère du tableau) à chaque fois mais lors de l'envoi certaines valeurs n'étaient pas envoyés.

Nous avons donc utilisé une méthode où nous rajoutions un signe « \$ » à la fin de chaque valeur lors de l'enregistrement (donc sur la 1^{ère} carte SD où les

valeurs sont écrites correctement) et nous utilisons ce signe \$ sur notre programme de réception, pour qu'il lise sur une ligne tous les caractères avant qu'il y ait un signe « \$ », cette ligne correspondra alors à une valeur.

- La vitesse d'envoi du fichier avec les valeurs enregistrées par le géophone vers une autre carte SD

C'est le plus gros problème que nous avons rencontré. Nous le rencontrons toujours d'ailleurs. Lorsque nous utilisons les programmes qui vont envoyer/lire les données de façon à ce que les caractères forment une chaîne de caractères (voir problème ci-dessus), l'envoi va se faire très lentement, de l'ordre de la seconde pour chaque valeur. Nous devons donc jouer sur le délai avant l'envoi de chaque caractère, mais si le délai est trop petit alors les valeurs envoyées seront incorrectes (des chiffres seront « mangés » ce qui résultera à des valeurs absurdes) et si le délai est trop grand alors la vitesse d'envoi est infernalement lente. Nous avons donc choisi le temps de délai minimum pour que les valeurs soient envoyées correctement sauf que la vitesse d'envoi reste lente. Nous avons donc résolu notre problème à moitié.

- L'enregistrement des valeurs du géophones sur la 1^{ère} carte SD

Ce problème est survenu lorsque nous avons combiné notre code pour l'émetteur... En effet le code qui envoie à vitesse lente doit être dans le même programme que l'enregistrement des valeurs, or cela perturbe aussi la vitesse de l'enregistrement et nous ne comprenons toujours pas pourquoi puisque l'envoi se fait après la boucle de l'enregistrement. Il ne devrait pas y avoir de lien.

Nous avons donc rencontré beaucoup de problèmes qui nous ont empêché de réaliser exactement ce que nous voulions, notamment pour la partie envoi puisque l'envoi ET l'enregistrement se font très lentement il est presque impossible d'obtenir un graphe, à part si on attend quelques minutes mais ce n'est alors pas efficace...

VI) Améliorations possibles

- Apparence : mettre les boards Arduino dans des petits boîtiers, masquer les fils...
- Envoyer les données au Rover que lorsqu'il en envoie le signal

VII) Ce qui nous a plus/moins plu

- Nous avons beaucoup apprécié le fait que nous étions celles qui devaient réfléchir à notre sujet projet, à sa conception et à sa réalisation. Nous sommes bien sûr loin d'être des ingénieurs mais cela nous a permis d'avoir un petit aperçu sur ce que c'était d'être responsables d'un projet de groupe. Nous avons aussi appris à se répartir le travail en fonction des aptitudes de chacune mais aussi du temps imposé. Il ne faut pas oublier aussi la satisfaction que nous avons lorsqu'un code marchait, ça nous rendait fières de nous et nous regagnions confiance en nous.
- Par contre, les difficultés rencontrées au cours de ces 5 mois nous ont moins plu. Surtout lorsque nous ne savions pas d'où provenaient ces problèmes/erreurs de code et que nous ne pouvions pas les régler. Il y a aussi le stress de savoir si nous parviendrons à finir à temps, si notre projet aboutira à quelque chose...

VIII) Conclusion

Ces 5 mois passés à réaliser notre projet Arduino étaient très enrichissants. Nous avons appris beaucoup de choses dont des concepts électroniques que nous n'aurions sûrement pas vu en cours dû à leur complexité. Bien que nous n'ayons pas réussi à bien gérer notre temps nous sommes quand même contentes du résultat et nous espérons utiliser cette expérience pour nous améliorer au niveau de l'organisation.

BERTAINE Charlène
M'HAMDI Imen (G2)