POLYTECH LILLE

Université Lille1 Sciences et Technologies

2014

Projet IMA4 : Save the Rhino

Hautecoeur Mélanie , Violier Fabien

Polytech'Lille

15/04/2014

Sommaire

1) Présentation du projet

Notre projet, nommé “Save the Rhino” a été créé dans le but de maîtriser le braconnage des rhinocéros dans en Afrique du Sud, car leur nombre ne cesse de diminuer et la survie de l’espèce est sévèrement menacée. Il a été conçu en collaboration avec la Stellenbosch University d’Afrique du Sud.

Nous avons donc été chargé de concevoir un dispositif capable d’alerter rapidement un centre de secours lorsqu’un animal est abattu grâce à un SMS contenant ses coordonnées GPS, afin de pouvoir interpeler les chasseurs avant qu’ils ne s’enfuient.

Cet appareil se composera d’un capteur de vie situé dans l’oreille du rhinocéros, transmettant ces informations par émission radio à une balise accrochée à la jambe de l’animal. Cette balise sera constitué d’un GPS, ainsi que d’un émetteur GSM. La gestion des modules s’effectue de chaque côté par un arudino UNO :

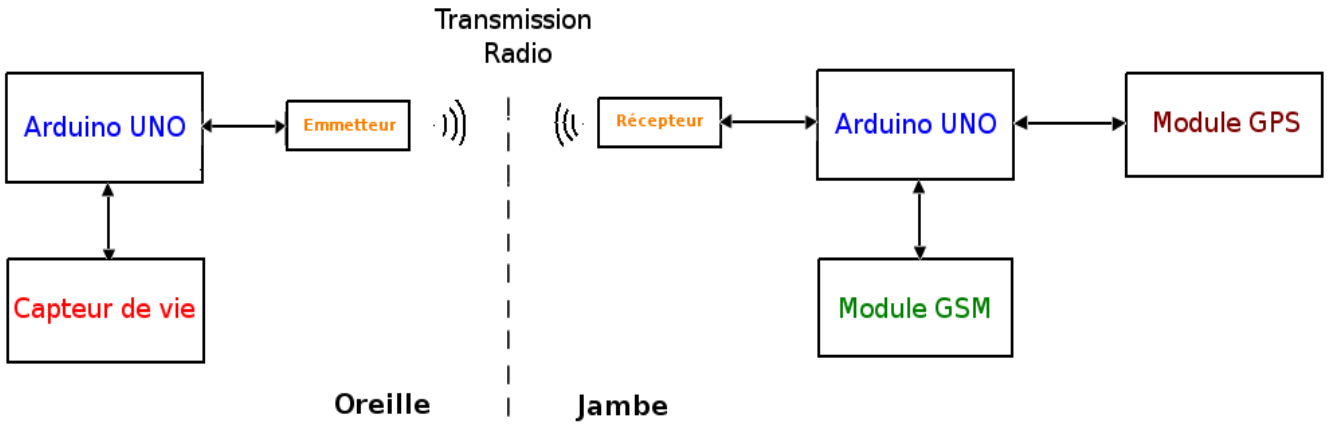


Figure 1 : Représentation des différents modules du systèmes

Notre tâche consistera donc à choisir, programmer et adapter chaque module du système afin qu’il puisse répondre aux critères demandés. Nous allons maintenant nous intéresser au cahier des charges défini par le sujet.

2) Cahier des charges

Afin de répondre au sujet, le système devra :

- Détecter la position GPS
- Envoyer cette position par émission GSM une fois par jour
- Evaluer l'état de l'animal
- Transmettre cet état par liaison radio
- Envoyer un message d'alerte si l'animal est détecter mort avec sa position
- Assurer une durée de vie de minimum deux ans
- Peser moins de 50kg
- Resister au poids de l'animal si celui-ci se couche dessus

3)Limites du système

Le système présente quelques faiblesses, pour lesquelles nous avons imaginés des solutions possibles.

La première limite que l'on observe concerne la transmssion radio. En effet, la communication peut être erronée, et un brouilleur peut être utilisé par les braconniers. Pour parer à cela, on peut utiliser un code de Hamming permettant la correction d'erreur du message reçu. On peut également penser à un SMS alertant le centre si le récepteur ne capte pas le message émis par le capteur pendant une certaien durée, ce qui permettrait de prévenir un brouilleur.

La seconde limite est l'alimentation du dispositif :

- Côté balise, il faut anticiper la décharge de la batterie. Pour cela, une détection de batterie faible entraînant un SMS peut être utilisé. On peut aussi équiper le système d'une batterie de secours dans le cas d'un dysfonctionnement de la batterie ou d'une impossibilité à changer rapidement la batterie principale.
- Côté capteur, l'alimentation n'est pas définie mais on peut imaginer un panneau solaire ou une batterie. Cependant cela ne représente pas une solution très pratique, car les rhinocéros se frappe régulièrement la tête, ce qui peut endomager un système volumineux.

Nous allons maintenant décrire les différents méthodes utilisées sur chaque module du système.

4)Méthodes utilisées

1)Transmission Radio

Pour établir la communication entre le capteur et la balise, on utilise une liaison à 433MHz. Il nous a été fourni 2 émetteur récepteur TR3000 de RFM, cependant nous avons choisi d'utiliser l'émetteur et le récepteur suivant pour notre développement, ceux-ci sont largement disponible sur le site de vente ebay.fr pour un faible prix.

Ces 2 modules utilisant la même technologie, les codes fournis devraient fonctionner pour les 2 modules.

La communication à 433MHz étant très chargée, le code fourni utilise un préambule sur 8bits, puis une data de 8bits. La transmission est assurée par un code de Manchester, et une redondance de la transmission. Chaque transmission est répétée 30 fois (Paramétrable) et la réception se fait sur la lecture de 10 données et une recherche du caractère le plus présent dans ces 10 données.

La liaison à 433MHz ne permettant pas un débit très élevé, nous avons choisi d'utiliser une transmission a 1200 bauds.

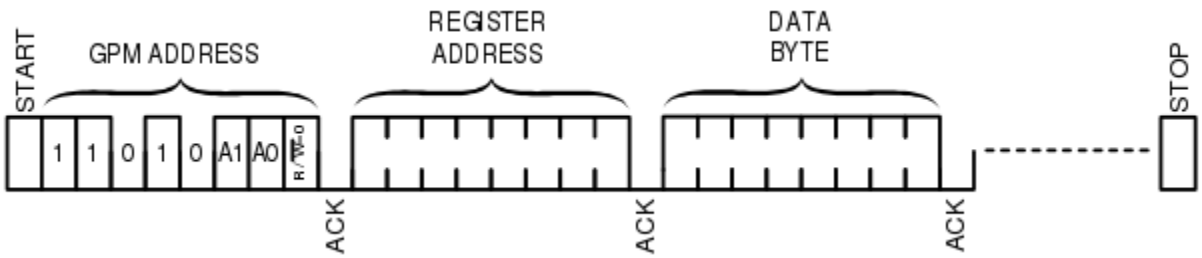
2)Module GPS

Le module GPS est le shield DS_GPM utilisable sur Arudino UNO, MEGA ou NANO. On l'appliquera donc ici sur notre Arduino UNO, et on communiquera avec celui-ci grâce au protocole I2C. Ce shield possède 112 registres contenant l'heure, la date, la latitude, la longitude, et la qualité de réception.

La documentation de l'ATMega 328p présente des morceaux de code aidant à la programmation en I2C, comme l'émission d'un Start, Stop ou l'envoi de données.

Le module DS_GPM répond aux trames de lecture et d'écriture suivantes :

Ecriture :



Lecture :

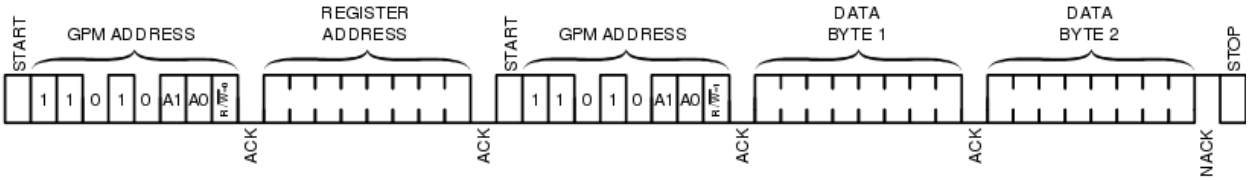


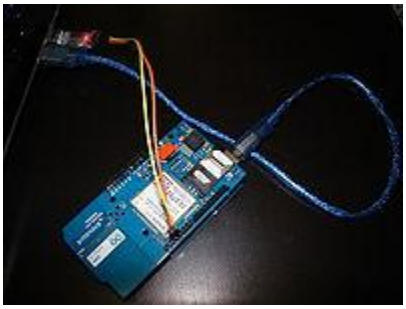
Figure 2 : Trame d'écriture et de lecture en I2c sur le shield DS_GPM

Les bits A1 et A0 correspondent à un jumper présent sur le shield. Grâce ces bits, on peut accéder à 4 esclaves différents. Le bit R/W permet à l'esclave de savoir si on se situe en mode lecture (1) ou écriture (0). On accède alors au registre défini par l'octet suivant.

Il est possible d'accéder à plusieurs registres qui se suivent, car le GPS renvoie un octet tant que l'on ne renvoie pas un NACK. On procédera donc ainsi pour récupérer tous les registres qui concernent la position GPS, soient 19 registres. 9 registres concernent la latitude et 10 la longitude:

- La latitude est décrite par 2 registres des degrés (un pour les dizaines et un pour les unités), 6 registres des minutes (dizaines, unités, dizièmes, centièmes, millièmes et dizaine de millièmes), et un pour la direction, Nord ou Sud).
- La longitude est décrite de la même façon avec un registre supplémentaire pour les degrés (centaines), et la direction indique alors Est ou Ouest.

Une fois ces données récupérées, on les transmet par émission GSM.



3) Module GSM

Le shield GSM est le shield GSM officiel Arduino.

Ce shield permet d'utiliser toutes les fonctionnalités du réseau GSM :

- Appel
- Envoi et réception de SMS
- Internet

Il utilise la communication série UART pour communiquer avec l'Arduino. Pour établir cette communication, il utilise un certain protocole, les commandes AT.

On peut trouver la liste des commandes disponibles à cette adresse :

http://arduino.cc/en/uploads/Main/Quectel_M10_AT_commands.pdf

Chaque commande doit être validée par l'envoi du caractère <CR> (retour chariot).

Ainsi pour tester la communication avec le shield, on peut envoyer la commande :

émission:
AT\r

réception :
AT \n\r
OK\n\r

Afin de se familiariser avec les commandes AT, et la configuration du shield, il est conseillé d'utiliser un dongle usb, permettant de communiquer directement avec le shield sans passer par l'arduino. Ainsi, il est plus facile de récupérer les informations retournées par le shield, comme les erreurs.

En effet, chaque commande peut renvoyer des erreurs en cas de problème.

On peut trouver une liste décrivant l'erreur en fonction de son numéro ici :

http://www.smssolutions.net/tutorials/gsm/gsmerrorcodes/

Il est nécessaire d'activer le shield pour communiquer avec celui-ci. Pour cela, il suffit d'appuyer sur le bouton power pendant 2s, ou d'activer le Pin de control situé sur le PIN 7 de l'Arduino.

Ainsi après une mise en veille de l'arduino, il faut rallumer le shield GSM, il faut donc remettre le PIN 7 à l'état 1.

5) Problèmes rencontrés

1)Transmission Radio

Le principal problème rencontré pour établir la liaison radio fut le bruit de la communication à 433MHz. En effet, la simple communication utilisant le port série avait un taux de réussite de 25% au maximum. De plus, le port série étant déjà utilisé par le module GSM, il n'était pas question de l'utiliser à cette fin.

Il a donc fallu mettre en place un protocole de communication sur un des ports digitaux. Nous avons donc créé un algorithme permettant d'intégrer un préambule à notre message et d'appliquer le codage de Manchester.

On observe une parfaite transmission lors de l'utilisation de câble pour transmettre nos données. Malgré cela, la transmission radio ne se fait pas correctement et certaines données se voient modifiées. Ainsi, lors de l'envoi du caractère 'a', on reçoit '}'. Seuls certains caractères ne sont pas modifiés.

Ainsi nous avons choisi le caractère 'm' et 'j' pour l'envoi de l'état mort ou vivant du rhinocéros. Ces 2 caractères présentent 3 bits de différence, ce qui assure une certaine sécurité à la transmission du message.

2)Module GPS

Le module GPS a posé quelques problèmes. Tout d'abord, la communication I2C était nouvelle, il a donc d'abord fallu apprendre à s'en servir. La programmation de ce dernier n'a pas été évidente. Premièrement, le shield répondait qu'il ne recevait pas les données transmises. Ce problème était dû à un désordre des ordres donnés, qui doivent respecter les trames demandées. Il n'a pas fallu longtemps pour trouver le problème.

Lorsque le GPS a pu communiquer de façon normale, tous les registres demandés contenaient 0. Après de nombreux tests et des vérifications du code, l'observation de l'ensemble des registres montrait que deux d'entre eux n'étaient pas nuls : celui contenant les milliers d'années indiquant 2, ce que l'on suppose être par défaut, et celui contenant l'état du GPS indiquant 1, ce qui signifie que le GPS ne capte pas. La lumière STATUS étant statique, nous en arrivions a la même conclusion. Nous avons tout de même vérifié que nous envoyions les bonnes trames sur un analyseur logique :

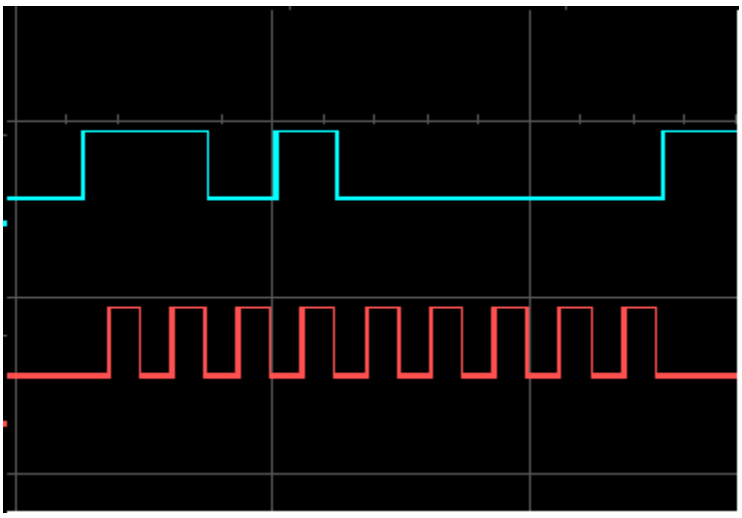


Figure 3 :Emission de l'adresse 0xD0 par l'Arduino puis réponse d'un ACK par le shield

La détection de données GPS ne nécessite pas de configuration initiale, des recherches s'imposaient donc. Des rapports sur le GPS DS_GPM indique que celui-ci peut nécessiter un temps non négligeable (environ 1h) afin de détecter des satellites s'il n'a pas été alimenté depuis longtemps.

Nous avons alors laissé le GPS alimenté pendant plusieurs heures, ce qui n'a pas été concluant. La lumière ne clignote toujours pas, et le GPS ne recoit toujours aucune donnée.

3)Module GSM

Nous avons rencontré deux problèmes pour la configuration shield GSM.

Le premier fut de choisir comment communiquer avec le shield sachant que la communication se fait sur les pins digitaux 2 et 3. Ainsi plusieurs solutions ont été proposé, établir un 2e port série sur les pins digitaux correspondant ou relier le port série au pin de communication du shield.

Après avoir exploré les 2 voies, comme nous n'utilisions pas la liaison série pour noter projet, nous avons choisi de relier la liaison série au shield GSM.

Le second problème a été communiquer avec le shield. En effet, après avoir longuement étudier le code Arduino fourni par le fabricant et d'autre code fourni par des développeurs, la communication ne se faisait toujours pas. Pour régler ce problème, nous avons utilisé un dongle USB permettant de récupérer les broches Rx et Tx du port USB, et donc de communiquer via un terminal de communication série tel que "minicom". et d'observer le comportement du code arduino sur le 2ème port série créé par celui ci sur les pins digitaux 2 et 3.

4)Programme principal

Un des problèmes rencontrés lors de la mise en place du programme principal, a été de pouvoir rallumer le shield GSM après le réveil de l'Arduino sur interruption watchdog. En effet, la mise en veille de l'arduino éteint le shield GSM, il faut donc pouvoir le rallumer sans utiliser le bouton physique. Ce problème a rapidement pu être réglé grâce à l'utilisation du bit de controle du shield qui permet de pouvoir rallumer le shield par sa mise à 1.

Un autre problème fut de d'appliquer un time out de 16s sur la réception radio. 16s car c'est le temps maximal de d'émission de l'état du rhinocéros. En effet, l'émetteur situé sur le capteur envoie les donnée toutes les 8s et mets un certains temps pour envoyer les 30 données. Ce problème n'a pu être réglé à temps.

6)Conclusion

Pendant ce projet, nous avons donc dû faire fasse à beaucoup de difficultés qui nous ont freinés. Nous n'avons pas pu gérer l'ensemble des modules simultanément, ni répondre aux problématiques du système dans sa globalité. Cependant, nous avons pu faire fonctionner chaque partie séparément, en prévoyant de respecter ces problèmatiques (faible consommation énergétique, réveil de l'appareil). Même si le système n'est pas fonctionnel, tous les outils sont donc opérationnels pour la suite du projet.

Sur le plan personnel, cette expérience nous a apporté beaucoup, autant sur le plan des connaissances que de la gestion de projet. Nous avons appris de nouveaux protocoles, languages, etc … Mais surtout nous avons pu réaliser que le déroulement d'un projet peut facilement être ralenti par des difficultés successives. Nous avons pu apprendre comment faire face à ces situations afin de les gérer le plus rapidement possible. Ce projet aura donc été pour nous très bénéfique.