

LE ZEPPELIN

COMPTE RENDU

par Jeremy Morel, Erwan Gaymard, Pablo Leibenguth



Sommaire:

1.Description

2.Les objectifs

3.Le matériel

4.Les schémas

5.Les difficultés et les solutions

6.Les perspectives

1. Description

Fr:

Notre projet est un zeppelin télécommandé par technologie arduino. La télécommande permet de le diriger à distance. Bouger le stick fait varier la valeur des potentiomètres à l'intérieur. Ces valeurs sont lues par une carte arduino reliée à la télécommande et sont transmises par ondes radios grâce à des modules RF.

Un transmetteur est branché à la carte arduino de la télécommande et un récepteur à une autre carte arduino placée dans la nacelle.

L'arduino de la nacelle transmet les tensions correspondantes aux trois moteurs dirigeant le zeppelin.

En:

Our project consists in a arduino technology remote zeppelin. The remote make it move from far away. The value of the potentiometers changes as the remote sticks are moving. These values are read by an arduino that we've wired to the remote and transmit by radio waves thanks to RF modules.

A transmitter is wired to the arduino card of the remote and a receiver to an arduino set in the gondola.

The gondola arduino transmits the voltage to the 3 motors which led the zeppelin.

2. Les objectifs

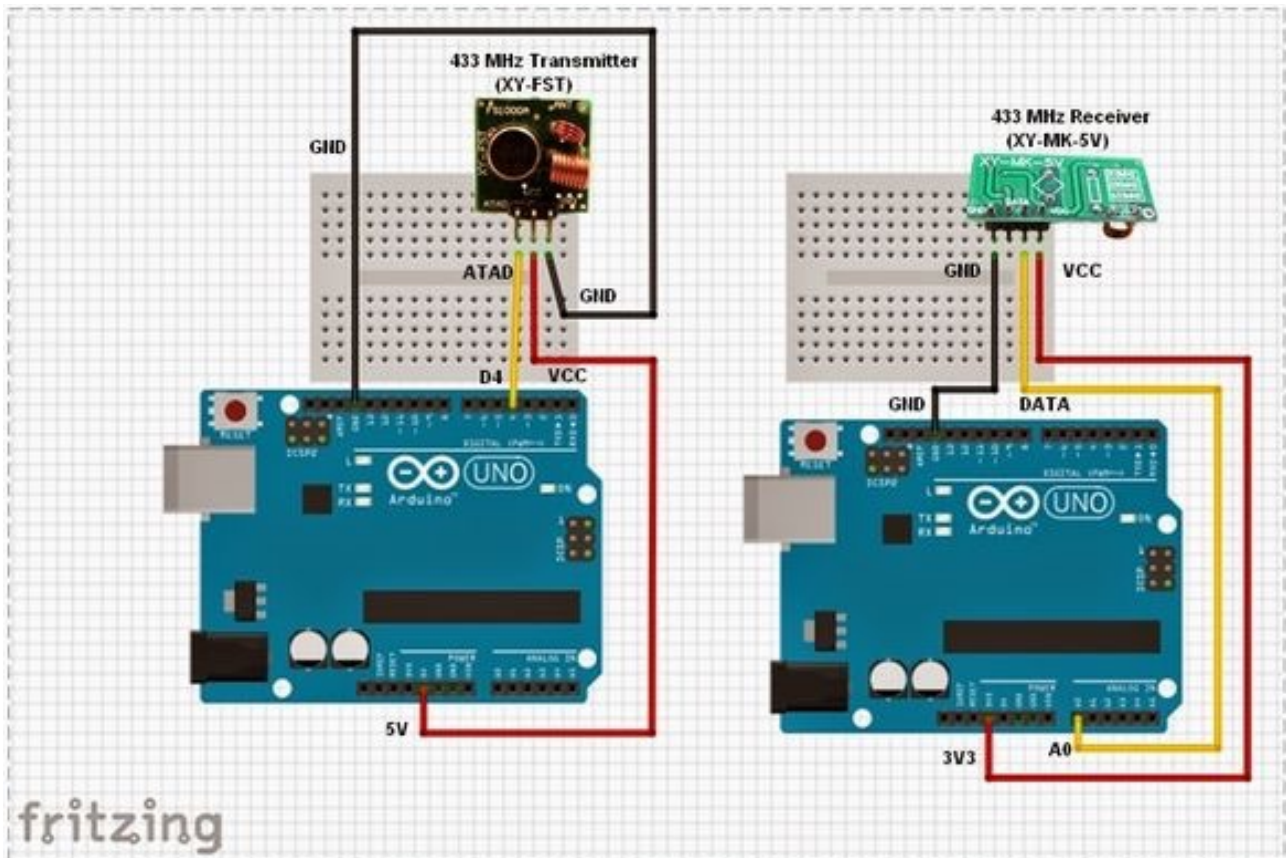
- Produire un code et un montage électronique nouveau
- Contrôler des moteurs à distance

3. Le matériel

- Un ballon gonflable
- 3 moteurs
- Une télécommande
- 2 modules RF: un transmetteur et un récepteur
- 2 cartes arduino nano

4. Schéma explicatif

Ci-dessous le schéma des modules RF. Nous avons testé la manette en connectant des diodes au récepteur et les modules ont communiqué en ondes radios.



Le but était de mettre l'arduino avec le récepteur dans la nacelle et de lui faire affecter des tensions aux moteurs.

5. Explication du code

Le code se décompose en deux parties : le code du transmetteur et le code du récepteur. On utilise la bibliothèque Virtual Wire qui permet la communication entre les modules RF. Pour faire fonctionner Virtual Wire, on l'importe :

```
#include <VirtualWire.h>
```

puis on doit exécuter trois fonctions dans le *setup()* :

```
vw_set_ptt_inverted(true) :
```

```
vw_set_tx_pin(position) : set la position du transmetteur.
```

```
vw_set_rx_pin(position) : set la position du récepteur.
```

```
vw_setup(4000) : vitesse de transmission des données.
```

Les valeurs des potentiomètres sont lues grâce à *analogRead()* puis converties en trois valeurs, une par moteur.

Les valeurs converties sont transmises par le transmetteur avec les commandes :

vw_send(values, sizeof(values)) : *values* est un tableau de byte dans lequel on ajoute les valeurs à transmettre.

vw_wait_tx() : attend que les données soient envoyées avant de passer à la suite.

Les valeurs sont reçues par le récepteur grâce aux méthodes :

if(vw_get_message(buf, buflen)) : attend de recevoir des informations pour passer le *if*. *buf* est la variable qui contient les valeurs reçues et *buflen* est sa longueur.

Les valeurs sont ensuite utilisées comme argument des *analogWrite(buf[1])* en utilisant le PWM de l'arduino.

6. Les problèmes rencontrés et solutions proposées

- Le zeppelin a mis du temps à arriver : nous n'avons réellement pu commencer à travailler que le 15 mars.
- On a eu des problèmes lors des premiers essais de l'utilisation de la bibliothèque Virtual Wire pour allumer plus au moins des diodes en fonctions de potentiomètres. Certaines diodes répondaient plutôt bien aux variations du potentiomètre, mais lorsque l'on faisait varier plusieurs potentiomètres, la carte Arduino ne semblait plus répondre. Après des nombreuses recherches sur Internet et avec l'aide des professeurs, nous avons conclu que le PWM rentrait en conflit avec Virtual Wire. En effet ils utilisent tous deux l'horloge de la carte Arduino. On pensait avoir résolu le problème en choisissant des entrées spécifiques, mais parfois, la carte du récepteur continue de freezer.
- À chaque fois que l'on retravaillait sur notre projet, il fallait re-téléverser les programmes car cela ne marchait jamais correctement du premier coup. Donc si on avait placé l'arduino avec le récepteur dans la nacelle aurait-elle fonctionné et appliquée son programme ?
- La télécommande et la nacelle comportent des éléments inconnus et incompris qu'on a fait le choix de ne pas démonter pour éviter que rien ne marche plus. Il y avait donc des composants inconnus dans notre projet

arduino. C'est en quelque sorte l'inconvénient de commencer avec un projet déjà tout fait qu'il faut démonter: que garder et comment l'utiliser ?

7.Perspectives

Notre projet pourra continuer à être travaillé par d'autres groupes, dans le but de faire de la pub à Polytech. Peut être un jour aura-t-il toutes les fonctions que l'on avait espéré initialement: une certaine autonomie, une plateforme d'atterrissage, une caméra...