

Fabrication d'un drone

Cela fait des années que nous réalisons des petits projets de robots, en grande majorité à l'aide de cartes arduino. Les ados de la Maison de Quartier La Touche demandent depuis longtemps qu'on se lance dans la fabrication d'un drone. La réglementation paraît lourde dans le cadre d'un vol en extérieur, en particulier sur l'espace public. Nous décidons donc de penser le projet pour un vol en intérieur (salle des fêtes, gymnase, etc...) Nous ne souhaitons pas fabriquer un drone pilotable avec une télécommande : nous voulons relever le défi de réaliser un drone capable de se déplacer tout seul, en évitant les obstacles, en faisant demi tour s'il arrive contre un mur, etc...

 Difficulté **Difficile**

 Durée **30 jour(s)**

 Catégories **Électronique, Machines & Outils, Jeux & Loisirs, Robotique**

 Coût **200 EUR (€)**

Sommaire

Introduction

Étape 1 - Découverte des moteurs et des ESC

Étape 2 - Piloter la carte arduino avec une application mobile sur android

Étape 3 - Mise en place de la carte arduino micro

Notes et références

Commentaires

Introduction

Matériel et Outils :

- Un châssis
- 4 moteurs, 4 ESC et 4 hélices
- Une batterie
- Une carte arduino micro
- 4 capteurs de distance infrarouge
- un gyroscope
- ~~un baromètre~~ deux sonars HC-SR04. Après tests, le baromètre n'est pas assez précis pour un vol en intérieur
- un capteur bluetooth

Explications :

- Le Realacc 220 est un châssis de 22cm (22cm de diagonale, c'est la distance entre les 2 centres d'hélices). Nous avons cherché un compromis entre un châssis assez gros pour embarquer tous nos capteurs électroniques, tout en essayant de ne pas être trop volumineux, étant donné qu'il est destiné à voler en intérieur.
- Nous avons acheté un kit complet, connu pour être adapté à un châssis de 22cm. Si l'on veut aller dans le détail, il faut choisir les moteurs en fonction du poids de notre drone et de la vitesse maximale que l'on souhaite. Puis choisir des ESC avec une puissance suffisante pour les moteurs. Puis choisir les hélices adaptées. Notre kit rassemble des moteurs Racerstar 2205, des ESC 30A et des hélices 5 pouces. On verra bien ce que ça donne.
- La batterie Li-po que nous avons choisie est une 3S 70C 1500mAh.
- La carte de vol sera une carte arduino, que nous allons programmer. Nous pensons partir sur une carte arduino micro pour qu'elle soit la plus légère possible.
- Les 4 capteurs infrarouges seront des Sharp GP2Y0A21YK0F, qui sont capables de détecter les obstacles entre 10 et 80 cm.
- Le gyroscope est un MPU6050.
- ~~Le baromètre est un BMP180.~~ Un sonar orienté vers le haut pour ne pas se cogner au plafond, un autre orienté vers le bas pour gérer la hauteur du drone.
- Le capteur bluetooth est un HC05

Étape 1 - Découverte des moteurs et des ESC

On prend cette page comme point de départ : <http://arduino.blaisepascal.fr/controler-un-moteur-brushless/>

Et ça ne fonctionne pas. En testant avec un autre lien on y arrive : <https://www.firediy.fr/article/calibrer-ses-esc-avec-un-arduino-drone-ch-3>

L'erreur du premier lien, c'est qu'il manque des informations dans la ligne `esc.attach(12)`; il fallait mettre `esc.attach(12, 1000, 2000)`; Pourquoi ?

Par ailleurs il est important de comprendre la séquence de calibration des ESC. Pour calibrer un ESC, il faut mettre plein gaz dès l'allumage, attendre les bips aigus réguliers, et relâcher les gaz à ce moment précis.

Dans notre code arduino, nous avons mis `esc.write(180)`; pendant quelques secondes, puis `esc.write(0)`; 180 représente la valeur maximale (plein gaz) et 0 la valeur minimale. Le passage à 0 ne se faisait pas au bon moment, et le calibrage ne se faisait pas. Le deuxième lien (firediy) permet d'envoyer la valeur minimale à un moment précis. C'est grâce à ce code que nous avons réussi à calibrer notre ESC. Ensuite nous utilisons le 1er lien (blaisepascal) pour envoyer une valeur précise à l'ESC. Tout se passe bien, on arrive à faire tourner plus ou moins vite notre moteur. Ouf !

Étape 2 - Piloter la carte arduino avec une application mobile sur android

Pour l'application mobile, on utilise app inventor : <http://appinventor.mit.edu/>

Pour communiquer avec la carte arduino, on passe par une carte bluetooth HC-05

Il existe plein de vidéos qui expliquent comment faire la connexion entre le smartphone et la carte bluetooth : <https://www.qwant.com/?q=app%20inventor%20bluetooth%20arduino&t=videos>

Au niveau des problèmes rencontrés, nous nous sommes cassés les dents sur :

- les pins à brancher pour les liaisons RX et TX via la bibliothèque SoftwareSerial. Beaucoup d'exemples utilisent les pins 2 et 3 de la carte arduino. Mais comme nous utilisons une carte arduino micro, ces pins ne sont pas compatibles. (*Not all pins on the Leonardo and Micro support change interrupts, so only the following can be used for RX: 8, 9, 10, 11, 14 (MISO), 15 (SCK), 16 (MOSI).*) Il va donc falloir changer notre schéma !
 - dans le code arduino, attention au baudrate de la liaison par la bibliothèque SoftwareSerial ! Beaucoup d'exemples montrent un baudrate à 57600 ou 115200. Pour nous, ça n'a fonctionné que avec 9600.
 - la valeur envoyée à la carte arduino est un nombre. Nous aurions voulu envoyer du texte mais il semble que ça ralentisse énormément la vitesse d'exécution du programme. Nous allons donc échanger uniquement des nombres pour garder une vitesse d'exécution optimale.
-

Étape 3 - Mise en place de la carte arduino micro

Jusqu'ici nous faisons nos tests avec une carte arduino Uno. En adaptant nos branchements avec la carte arduino micro, il semble impossible de piloter le moteur branché sur le port 12. Nous modifions donc le branchement du moteur arrière gauche pour le mettre sur le port 7. Le port 7 devait servir plus tard pour brancher un sonar, il faudra voir si on peut le mettre sur le 12.

Notes et références

1. <https://diydrones.com/profiles/blogs/arduimu-quadcopter-part-iii> : un exemple de drone avec des capteurs de distance. La vidéo dans le parking montre bien un exemple de ce qu'on veut réaliser
2. http://www.electrionoobs.com/eng_robotica_tut6_2.php : on va pas y couper, il va falloir se plonger dans les boucles PID, pour corriger les consignes (inclinaison, hauteur) avec les mesures effectives.
3. <https://www.firediy.fr/article/calibrer-le-capteur-mpu6050-avec-un-arduino-drone-ch-5> : utilisation du gyroscope
4. <https://www.firediy.fr/article/utiliser-sa-radiocommande-avec-un-arduino-drone-ch-6> : code Arduino : `attachInterrupt` et registres d'interruption
5. <http://mitappsinventor.blogspot.com/2015/05/rgb-slider-color-selector-rgb-led.html> appli Android pour contrôler le drone : un bon point de départ avec cet exemple de l'utilisation d'un slider pour contrôler une LED.
6. <https://www.carnetdumaker.net/articles/mesurer-une-distance-avec-un-capteur-ultrason-hc-sr04-et-une-carte-arduino-genuino/> débiter avec les capteurs de distance ultrasons

