# **Chapitre 2 : Présentation du matériels**

## Introduction :

La grande majorité des drones existants possèdent à peu prêt les mêmes Caractéristiques.

La première chose à comprendre est « De quoi est composé un drone », dans notre cas un quadricoptère.

## Matériaux nécessaires :

### ****Le châssis**** :

Le châssis ou (Frame en anglais). C’est la structure de notre drone, c’est sur lui que nous allons monter le reste des pièces. De lui dépend le comportement général de notre drone mais aussi son autonomie. Un châssis ultra léger nous permettra de gagner en autonomie. Des bras longs amélioreront la stabilité alors que des bras plus courts permettront des figures plus acrobatiques.

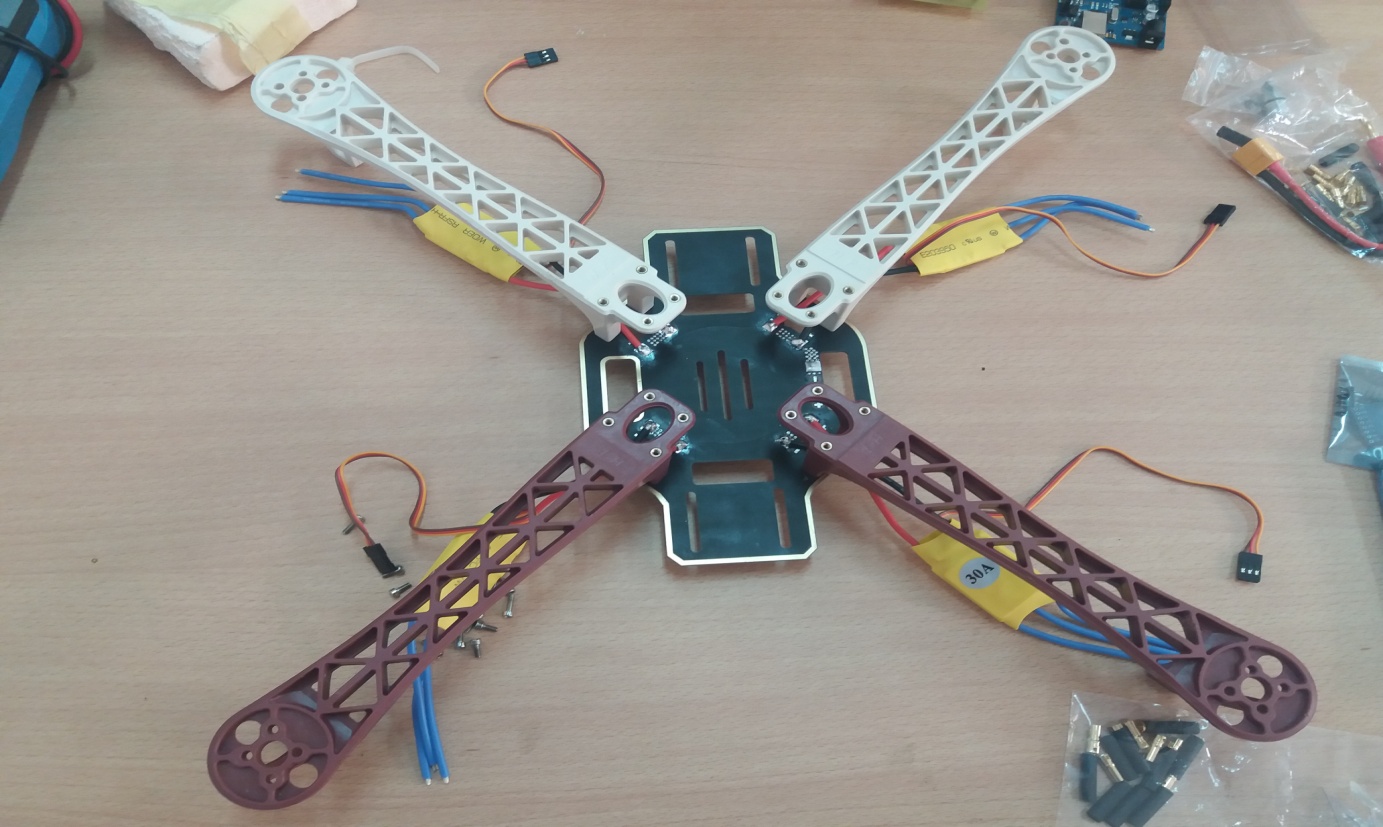
On en trouve principalement à base d’aluminium ou de fibre de carbone. Il existe des dizaines de formes différentes. Les châssis en aluminium sont faciles à fabriquer et bon marché.

On trouve également des châssis en carbone bon marché sur les sites d’enchère.

Sur les quadricoptères, les châssis sont composés de 4 bras.

Les caractéristiques de l’ensemble des composants dépendront en partie du type de châssis choisi (type et taille, c’est-à-dire poids des hélices, des moteurs, de la batterie, des fixations…). Le châssis peut amener quelques restrictions quant au moteur ou aux hélices choisis, notamment concernant leur taille.

Nous choisissons le châssis F450.



Ce châssis F450 est très résistant aux chocs car il est conçu avec des pièces moulées dans des matériaux très résistants.

### ****Les moteurs**** :

C’est eux qui vont permettre à notre quadricoptère de voler. Là encore il existe des centaines de modèles plus ou moins adaptés à l’utilisation que nous allons en faire.

Le choix des moteurs ce fait en fonction de leur rendement soit le nombre de tours que peut faire le moteur en 1 minute et pour 1 volt. Cette valeur s’exprime en Kilo Volt noté KV. Plus le KV d’un moteur à grand, meilleur est sont rendement et donc plus économe en d’énergie. C’est pourquoi il est intimement lié au choix de la batterie.

Alors nous avons choisi des moteurs brushless 1000 KV.

Quatre moteurs brushless sont nécessaires qui doivent être commandés par des interfaces de

Puissance appelées « Contrôleurs ».



### Les contrôleurs :

Ce sont des cartes électroniques qui permettent de faire tourner les moteurs plus ou moins vite, à partir du courant délivré par la batterie. La commande vient du « contrôleur de vol ».

Il faut donc un ESC (Electronic Speed Control) pour chaque moteur.

On a choisis



### La Batterie et le chargeur :

Notre batterie de type Lipo (Lithium polymère Ion) c’est la technologie qui permet d'avoir une puissance suffisante pour une faible masse.

La batterie est la source d’énergie de notre drone. Il en existe des dizaines de modèles très différents et à tous les prix.

**Attention :**

Il faut un chargeur spécifique pour la recharger et son utilisation est encadrée par des règles très strictes sous peine de détérioration rapide voire d'accident !

Attention à la connectique ! Il faut souvent souder de nouvelles prises si celles fournies avec la batterie ne conviennent pas au drone ou au chargeur.



Nous avons choisis une batterie 5200 mAh pour plus longue durée.

Lipo 3s c'est-à-dire 3 cellules (1 cellule = 3,8v)



### ****Les hélices**** :

Elles doivent être adaptées à la taille de notre engin mais aussi aux moteurs que nous avons choisis. Les hélices sont présentées avec 2 valeurs : la 1ère exprime la longueur de l’hélice (8", 9", 10", etc. ...) et la 2ème valeur indique le "pas" de l'hélice (4 / 4,5 / 5 / etc. ...)

Nos hélices sont 10\*4,5.





### La radio commande :

Pour piloter le drone, il faut piloter une radio côté pilote et un récepteur dans le drone.

Les radios actuelles sont en 2,4GHz et elles doivent nécessairement être programmables pour être configurée avec le drone.



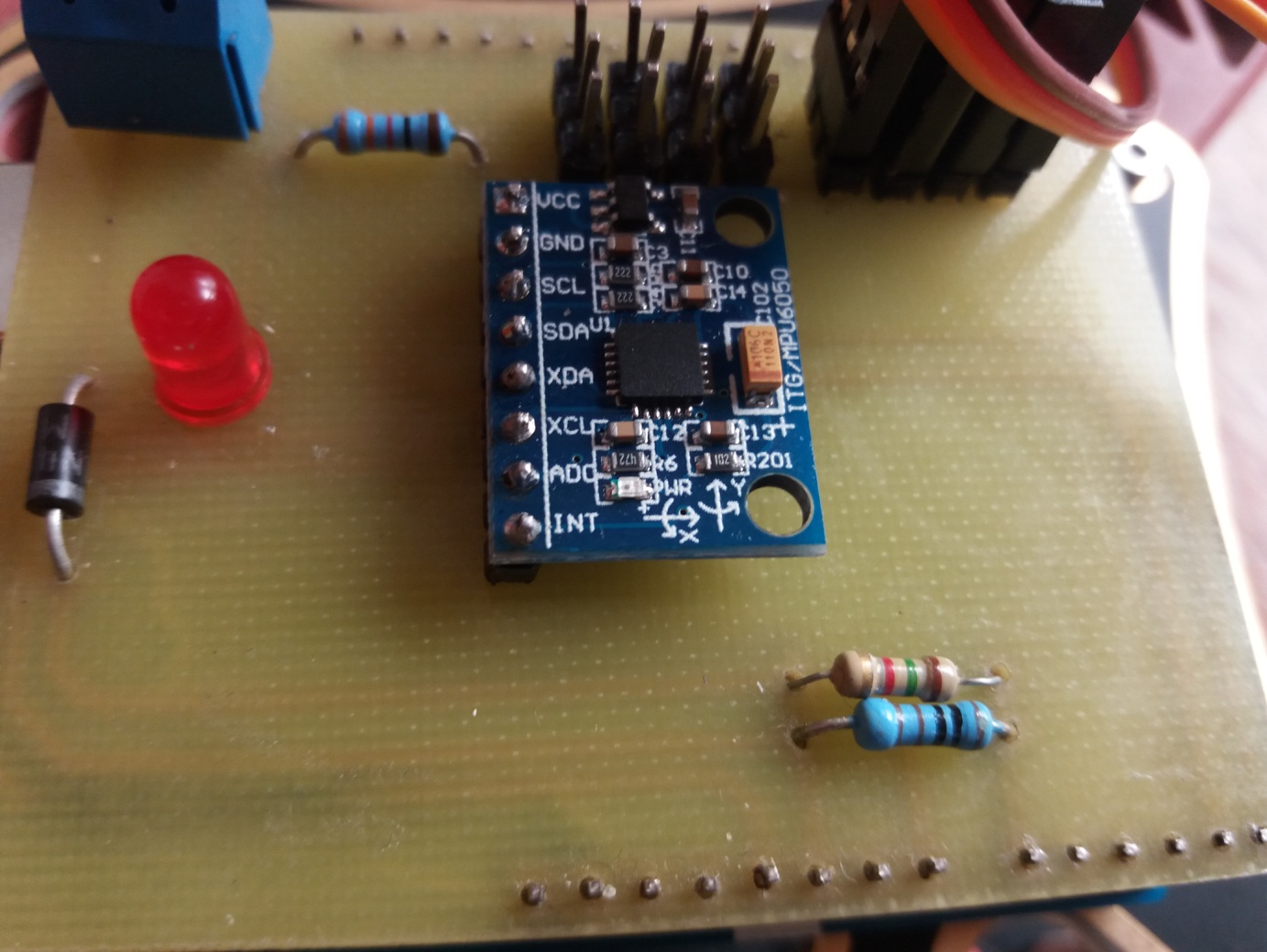
Cette radiocommande est polyvalente, elle peut- être utilisée pour d’autres engins(avion, hélicoptère, drone …)

### La carte contrôle de vole (gyroscope) :

C’est le cœur de votre drone. C’est lui qui va stabiliser l’engin mais aussi effectuer différentes tâches plus ou moins complexes comme suivre un plan de vol, atterrir automatiquement.

La carte de vol doit être alimentée par la batterie et on y branchera les ESC (reliés aux moteurs), puis pour la paramétrer il faudra un logiciel et un ordinateur.

Notez que certaines fonctions peuvent être paramétrées sans l’ordinateur, mais soit avec la radiocommande (avec certaines positions des manches) ou avec un Smartphone (il faut que la carte de vol soit connectée à un système Bluetooth).



### La carte arduino (uno) :

La carte Arduino Uno est une carte à microcontrôleur basée sur l'ATmega328. Elle dispose:

* De 14 broches numériques d'entrées/sorties (dont 6 peuvent être utilisées en sorties PWM (largeur d'impulsion modulée).
* De 6 entrées analogiques (qui peuvent également être utilisées en broches entrées/sorties numériques).
* D’un quartz 16Mhz.
* D’une connexion USB.
* D’un connecteur d'alimentation jack.
* D’un connecteur ICSP (programmation "in-circuit").
* Et d'un bouton de réinitialisation (reset).

Elle contient tout ce qui est nécessaire pour le fonctionnement du microcontrôleur. Pour pouvoir l'utiliser, il suffit de la connecter à un ordinateur à l'aide d'un câble USB (ou de l'alimenter avec un adaptateur secteur ou une pile, mais ceci n'est pas indispensable, l'alimentation étant fournie par le port USB).

### Système des caractéristiques :

|  |  |
| --- | --- |
| Microcontrôleur | ATmega328 |
| Tension de fonctionnement | 5V |
| Tension d'alimentation (recommandée) | 7-12V |
| Tension d'alimentation (limites) | 6-20V |
| Broches E/S numériques | 14 (dont 6 disposent d'une sortie PWM) |
| Broches d'entrées analogiques | 6 (utilisables en broches E/S numériques) |
| Intensité maxi disponible par broche E/S (5V) | 40 mA (ATTENTION : 200mA cumulé pour  l'ensemble des broches E/S) |
| Intensité maxi disponible pour la sortie 3.3V | 50 mA |
| Intensité maxi disponible pour la sortie 5V | Fonction de l'alimentation utilisée - 500 mA  max si port USB utilisé seul |
| Mémoire Programme Flash | 32 KB (ATmega328) dont **0.5 KB** sont utilisés  par le bootloader |
| Mémoire SRAM (mémoire volatile) | 2 KB (ATmega328) |
| Mémoire EEPROM (mémoire non volatile) | 1 KB (ATmega328) |
| Vitesse d'horloge | 16 MHz |

### Brochage de la carte UNO :



### Alimentation :

La carte Arduino Uno peut-être alimentée soit via la connexion USB (qui fournit 5V jusqu'à 500mA) ou à l'aide d'une alimentation externe. La source d'alimentation est sélectionnée automatiquement par la carte.

L'alimentation externe (non-USB) peut être soit un adaptateur secteur (pouvant fournir typiquement de 3V à 12V sous 500mA) ou des piles (ou accus). L'adaptateur secteur peut être connecté en branchant une prise 2.1mm positif au centre dans le connecteur jack de la carte. Les fils en provenance d'un bloc de piles ou d'accus peuvent être insérés dans les connecteurs des broches de la carte appelées Gnd (masse ou 0V) et Vin (Tension positive en entrée) du connecteur d'alimentation.

La carte peut fonctionner avec une alimentation externe de 6 à 20 volts. Cependant, si la carte est alimentée avec moins de 7V, la broche 5V pourrait fournir moins de 5V et la carte pourrait être instable. Si on utilise plus de 12V, le régulateur de tension de la carte pourrait chauffer et endommager la carte. Aussi, la plage idéale recommandée pour alimenter la carte Uno est entre 7V et 12V.

Les broches d'alimentation sont les suivantes :

* VIN : C’est la tension d'entrée positive lorsque la carte est utilisée avec une source de tension externe (à distinguer du 5V de la connexion USB ou autre source 5V régulée). La carte peut être alimentée à l'aide de cette broche, ou, si l'alimentation est fournie par le jack d'alimentation, accéder à la tension d'alimentation sur cette broche.
* 5V :C’est la tension régulée utilisée pour faire fonctionner le microcontrôleur et les autres composants de la carte. Le 5V régulé fourni par cette broche peut donc provenir soit de la tension d'alimentation VIN via le régulateur de la carte, ou bien de la connexion USB (qui fournit du 5V régulé) ou de tout autre source d'alimentation régulée.
* 3V3 : C’est une alimentation de 3.3V fournie par le circuit intégré FTDI (circuit intégré faisant l'adaptation du signal entre le port USB de l’ordinateur et le port série de l'ATmega) de la carte. L'intensité maximale disponible sur cette broche est de 50mA
* GND : Broche de masse (ou 0V).

### Mémoire :

L'ATmega 328 a 32Ko de mémoire FLASH pour stocker le programme (dont 0.5Ko également utilisés par le **bootloader**\*). L'ATmega 328 a également 2ko de mémoire SRAM (volatile) et 1Ko d'EEPROM (non volatile - mémoire qui peut être lue à l'aide de la librairie EEPROM).

\*Le bootloader est un programme pré-saisi dans l'ATméga qui permet la communication entre l'ATmega et le logiciel Arduino via le port USB, notamment lors de chaque programmation de la carte.

### Entrées et sorties numériques :

Chacune des 14 broches numériques de la carte UNO (numérotées des 0 à 13) peut être utilisée soit comme une entrée numérique, soit comme une sortie numérique, en utilisant les instructions pinMode(), digitalWrite()et digitalRead() du langage Arduino. Ces broches fonctionnent en 5V. Chaque broche peut fournir ou recevoir un maximum de 40mA d'intensité et dispose d'une résistance interne de "rappel au plus" (pull-up) (déconnectée par défaut) de 20-50 KOhms. Cette résistance interne s'active sur une broche en entrée à l'aide de l'instruction digitalWrite(broche, HIGH).

De plus, certaines broches ont des fonctions spécialisées :

* **Communication série**: broches 0 (RX) et 1 (TX). Utilisées pour recevoir (RX) et transmettre (TX) les données sérient de niveau TTL. Ces broches sont connectées aux broches correspondantes du circuit intégré ATmega8U2 programmé en convertisseur USB-vers-série de la carte, composant qui assure l'interface entre les niveaux TTL et le port USB de l'ordinateur.
* **Interruptions externes**: broches 2 et 3. Ces broches peuvent être configurées pour déclencher une interruption sur une valeur basse, sur un front montant ou descendant, ou sur un changement de valeur. Voir l'instruction attachInterrupt ()pour plus de détails.
* **Impulsion PWM (largeur d'impulsion modulée)**: Broches 3, 5, 6, 9, 10, et 11. Elles fournissent une impulsion PWM 8-bits à l'aide de l'instruction analogWrite ().
* **SPI (Interface Série Périphérique)**: broches 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK). Ces broches supportent la communication SPI (Interface Série Périphérique) disponible avec la librairie pour communication SPI. Les broches SPI sont également connectées sur le connecteur ICSP qui est mécaniquement compatible avec les cartes Mega.
* **I2C**: broches 4 (SDA) et 5 (SCL). Elles supportent les communications de protocole I2C (ou interface TWI (Two Wire Interface - Interface "2 fils"), disponible en utilisant la librairie Wire/I2C (ou TWI - Two-Wire interface - interface "2 fils").
* **LED**: broche 13. Il y a une LED incluse dans la carte connectée à la broche 13. Lorsque la broche est au niveau HAUT, la LED est allumée, lorsque la broche est au niveau BAS, la LED est éteinte.

### Broches analogiques :

La carte Uno dispose de 6 entrées analogiques (numérotées de 0 à 5), chacune pouvant fournir une mesure d'une résolution de 10 bits (i.e sur 1024 niveaux soit de 0 à 1023) à l'aide de la très utile fonction *analogRead()* du langage Arduino. Par défaut, ces broches mesurent entre le 0V (valeur 0) et le 5V (valeur 1023), mais il est possible de modifier la référence supérieure de la plage de mesure en utilisant la broche AREF et l'instruction *analogReference()* du langage Arduino.

Remarque : les broches analogiques peuvent être utilisées en tant que broches numériques : elles sont numérotées en tant que broches numériques de 14 à 19.

### Autres broches :

Il y a deux autres broches disponibles sur la carte :

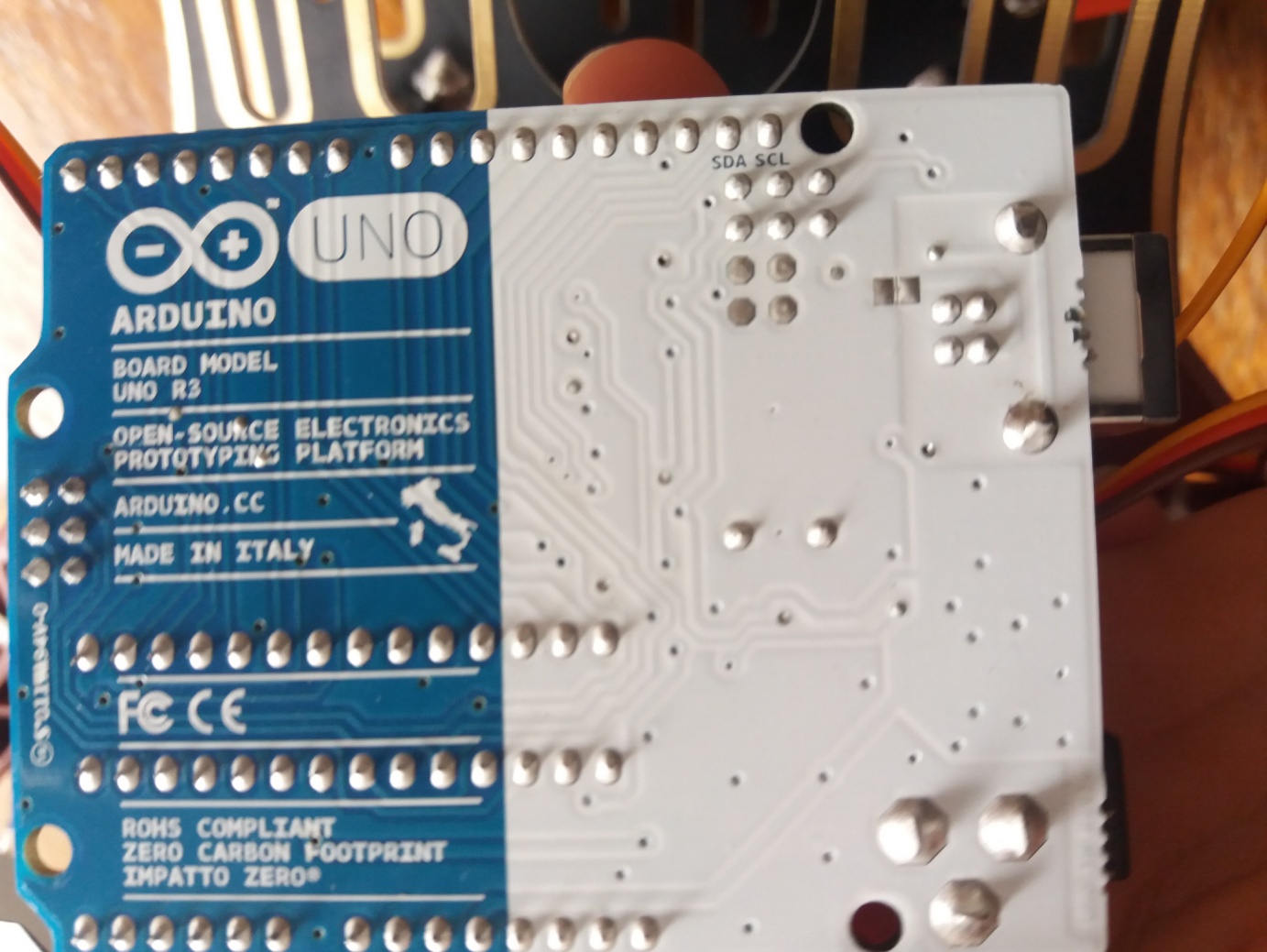
**AREF** : tension de référence pour les entrées analogiques (si différent du 5V).

**Reset** : mettre cette broche au niveau BAS entraîne la réinitialisation c'est-à-dire le redémarrage du microcontrôleur. Typiquement, cette broche est utilisée pour ajouter un bouton de réinitialisation sur le circuit qui bloque celui présent sur la carte.

### Communication :

Le logiciel Arduino inclut une fenêtre terminal série (ou moniteur série) sur l'ordinateur et qui permet d'envoyer des textes simples depuis et vers la carte Arduino. Les LEDs RX et TX sur la carte clignotent lorsque les données sont transmises via le circuit intégré USB-vers-série et la connexion USB vers l'ordinateur (mais pas pour les communications série sur les broches 0 et 1).

Une librairie Série Logiciellepermet également la communication série sur n'importe quelle broche numérique de la carte UNO.



### Interface zigbee (module xbee) :

Le ZigBee est un standard de communication sans fil à bas coût pour échanger des données issues d’équipements sans fil simples et de faible consommation dans le milieu industriel. Avec la convergence de l’informatique, de l’électronique et des télécommunications, il est ainsi possible de mettre en place des réseaux de capteurs sans fil dans un contexte domotique ou de contrôle industriel. Il existe un marché que ZigBee permet de combler : surveillance de locaux pour la détection de départ de feu, surveillance de bâtiments contre les intrusions, Gestion Technique de Bâtiment (GTB), aide à la personne, assistance aux personnes…

Ce standard est promu par l’alliance ZigBee qui regroupe un ensemble d’industriels travaillant sur l’élaboration de spécifications afin de pouvoir développer des applications sans fil bon marché, de faible consommation et sécurisées.

### Brochage :

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Broche |  | Nom |  | Direction |  | Description |
| 1 |  | VCC |  | - |  | Alimentation |
| 2 |  | DOUT |  | Out |  | Sortie UART |
| 3 |  | DIN/CONFIG |  | In |  | Entrée UART |
| 4 |  | DO8 |  | Out |  | Sortie digitale 8 |
| 5 |  | RESET |  | - |  | Reset (au moins 200 ns) |
| 6 |  | PWM0/RSSI |  | Out |  | Sortie PWM0/Indication puissance Rx |
| 7 | PWM1 | | Out | | Sortie PWM1 | |
| 8 | Réservé | | - | | - | |
| 9 | DTR\*/SLEEP\_RQ/DI8 | | In | | Contrôle Sleep/Entrée digitale 8 | |
| 10 | GND | | - | | Ground | |
| 11 | AD4/DIO4 | | Inout | | Entrée analogique 4 ou E/S digitale 4 | |
| 12 | CTS/DIO7 | | Inout | | Clear To Send/ E/S digitale 7 | |
| 13 | ON/SLEEP | | Out | | Indicateur état | |
| 14 | VREF | | - | | Tension de référence pour conversion | |
| 15 | Associate/AD5/DIO5 | | Inout | | Indication association/Entrée analogique 5 ou E/S digitale 5 | |
| 16 | RTS/AD6/DIO6 | | Inout | | Reday To Send/Entrée analogique 6 ou E/S digitale 6 | |
| 17 | AD3/DIO3 | | Inout | | Entrée analogique 3 ou E/S digitale 3 | |
| 18 | AD2/DIO2 | | Inout | | Entrée analogique 2 ou E/S digitale 2 | |
| 19 | AD1/DIO1 | | Inout | | Entrée analogique 1 ou E/S digitale 1 | |
| 20 | AD0/DIO0 | | Inout | | Entrée analogique 0 ou E/S digitale 0 | |

Au minimum, le module XBee nécessite de câbler les broches d’alimentation (VCC et GND) et les signaux DIN et DOUT pour respectivement les données entrantes et sortantes de l’interface UART. Les autres broches utilisées servent à configurer le module…

### Communication avec le module Xbee :

Le module XBee peut être connecté à n’importe processeur possédant une interface série UART comme indiqué sur cette figure :

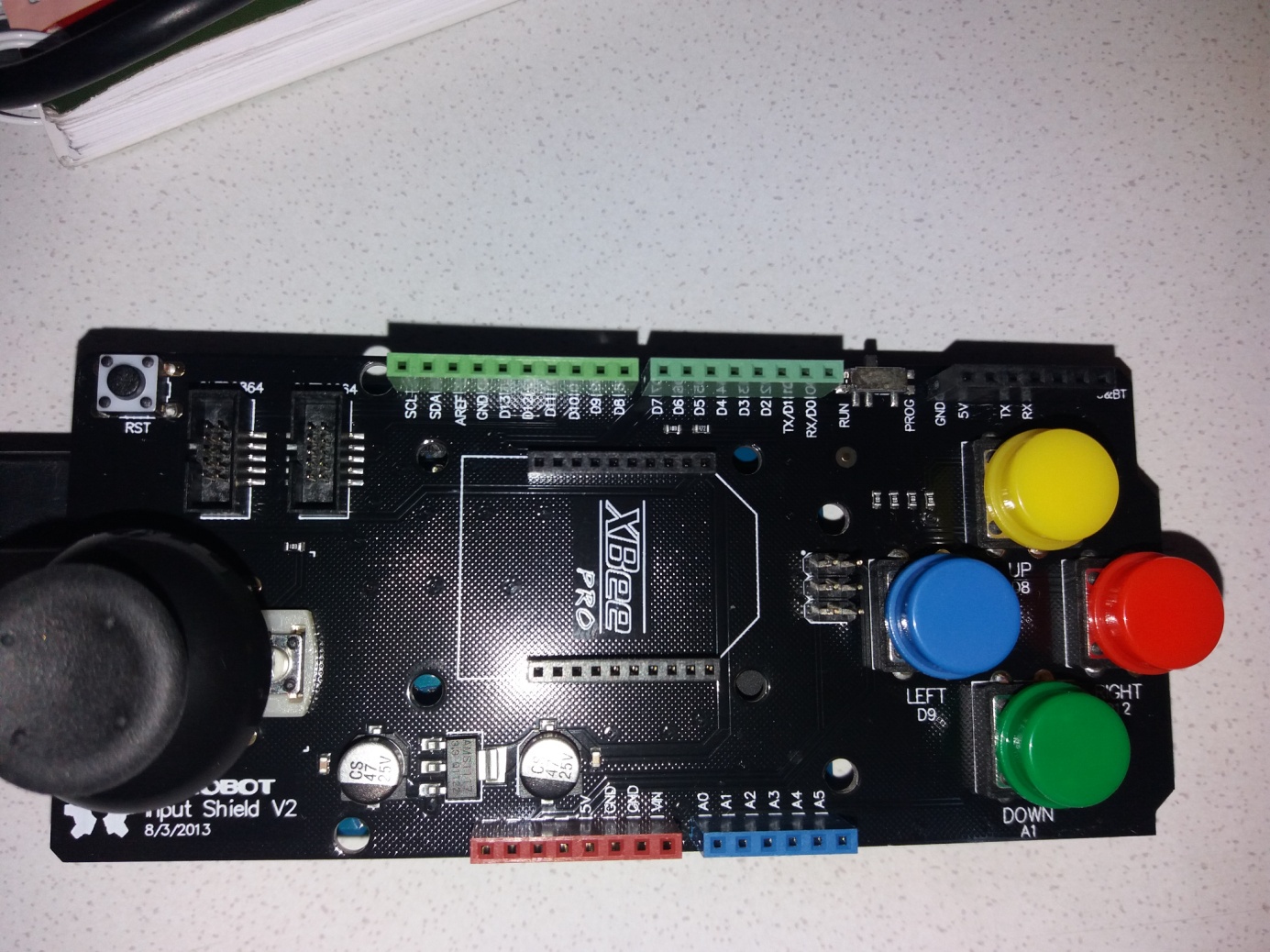


Le processeur envoie ses données (caractères) sur le port DI en mode série asynchrone. Chaque caractère est composé d’un bit de start (niveau logique 0), suivi de 8 bits de données avec le bit de poids faible en premier et enfin un bit de stop.

La figure qui suit présente le mode de fonctionnement interne du module XBee.



Il existe un buffer à l’émission (DI buffer) et un buffer en réception (DO buffer). Un contrôle de flux matériel peut être mis en place en utilisant le traditionnel couple CTS/RTS pour gérer le flux à l’émission et à la réception des données suivant le taux de remplissage des buffers (comme dans un circuit UART classique).

-

Dans notre cas on va utiliser une manette XBee pour commander notre drone.

## Conclusion :

Ce sont tous les matériaux que nous avons utilisé dans notre projet.

Le drone est facilement réparable (maintenance facile) mais les pièces sont défilement interchangeables avec un délai de livraison rapide.