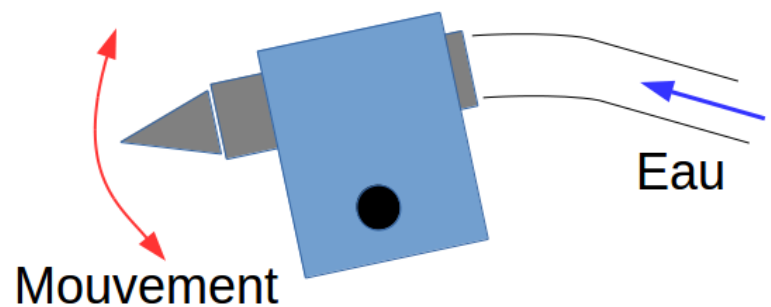


# Arduino Yun pour une commande moteur via wi-fi

Pilotage électrovanne et moteur pas-à-pas

*Pour un système de brumisation (pulvérisation fine d'eau), il était nécessaire de piloter deux éléments :*

- *Un robinet d'eau (électrovanne), temps d'ouverture/fermeture eau réglable à la seconde*
- *L'orientation de buse (moteur pas-à-pas)*



*Principe brumisation motorisée*

*Voici le schéma de principe de l'installation :*

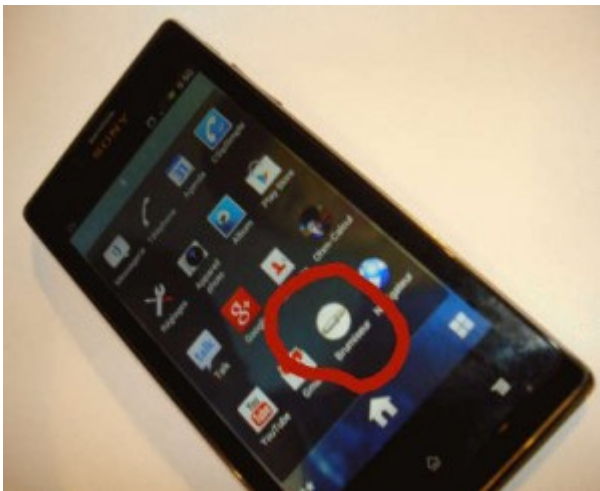
*Le contrôle de ceux paramètres doit être possible en tout temps pour l'utilisateur du système de brumisation, j'ai pensé une supervision par le smartphone était la meilleure option. En plus l'aspect "ludique", "geek" pouvait retenir des clients potentiels. Il n'y a plus qu'à faire ..... 😊*

## Transmission smartphone ⇔ Arduino

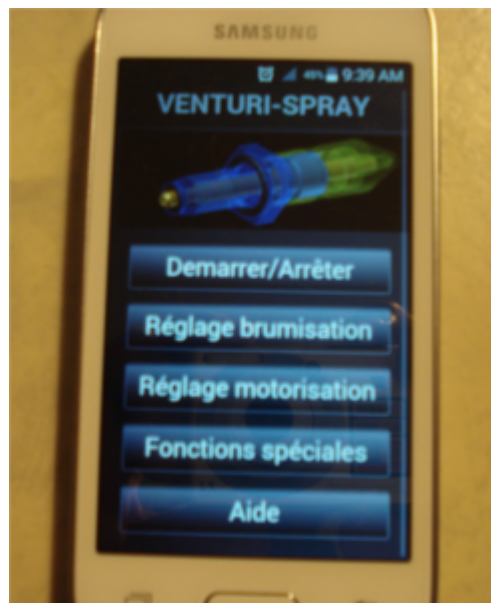
# Le Bluetooth

*Premier choix de transmission entre le smartphone et la commande : Bluetooth*

*Comme je n'avais jamais fait cela et après discussion avec des spécialistes (amis ayant un peu bricolé dans le domaine), j'ai programmé avec Cordova une interface pour smartphone. Bien sûr je n'ai réussi qu'avec Android et pour les PC, l'interface ne correspondait pas à grand-chose. Vous pouvez voir le résultat sur smartphone après de longues heures de déceptions .....*



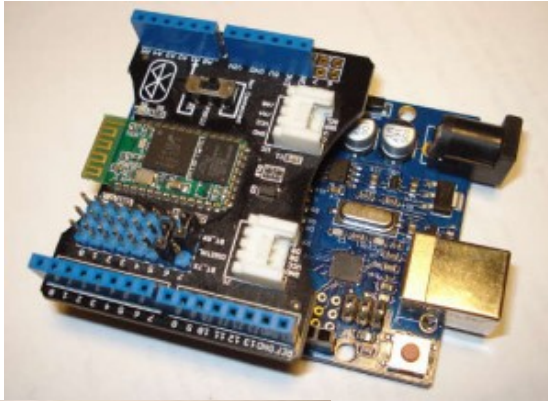
*et après "quelques heures" de plus :*



*Bref, la situation n'était pas brillante surtout que la liaison Bluetooth du côté de la commande ne donnait pas satisfaction. En effet pour réaliser cette commande, je suis parti avec ce que je connaissais, c'est-à-dire l'Arduino.*

*Donc, un Arduino et une carte de Bluetooth, quelques lignes de code et le tour est joué !! 🤪  
Je suis un optimisme irraisonné et irraisonnable !*

*premier essai*



*deuxième essai*

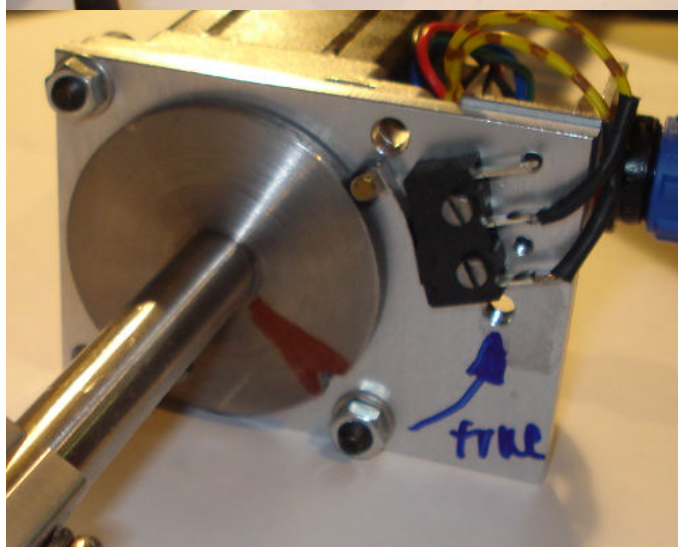
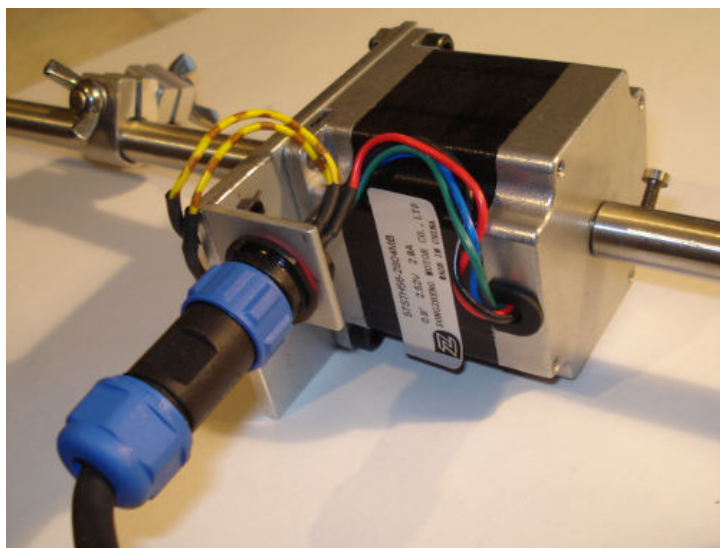


*J'ai donc pris une carte shield (photo carte shield bleue) et je n'ai jamais réussi à communiquer avec mon smartphone ..... sûr c'est la carte !! On recommence avec une carte shield RedBearLab et là, j'ai également des exemples de programmation, ça marche ! Pas trop de soucis pour communiquer. Tout ne marche pas super, mais dans l'ensemble c'est utilisable.*

## Arduino ⇒ Moteur et vanne

*Le pilotage du moteur pas à pas ne doit pas me poser beaucoup de problème, j'avais déjà fait quelques petites applications. Pour commencer, le principe de fonctionnement du moteur.*

- *Une origine (microswitch), au départ retrouver cette origine et ensuite mouvement de part et d'autre.*
- *Angle1 : amplitude du mouvement depuis l'origine (dans un sens)*
- *Angle2 : amplitude du mouvement depuis l'origine (dans l'autre sens)*



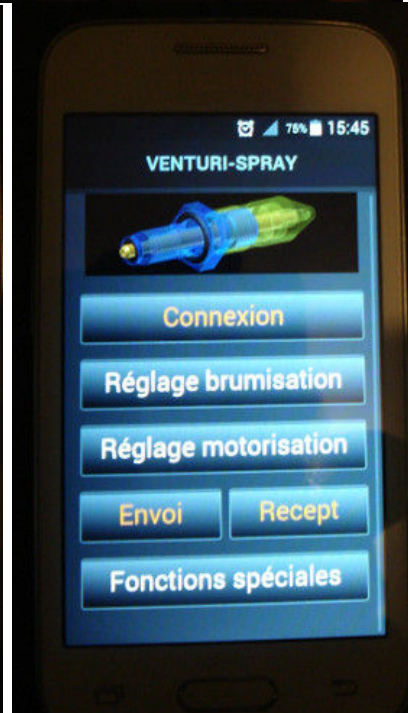
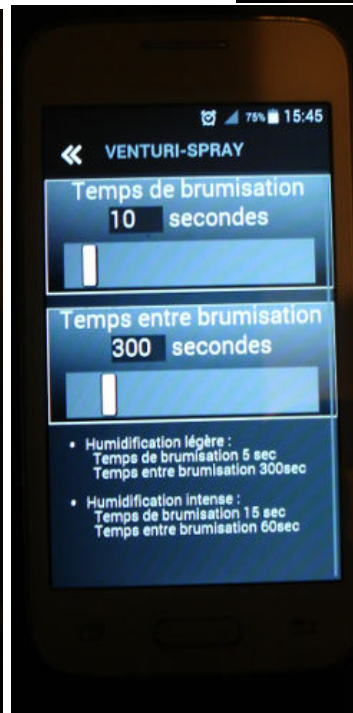
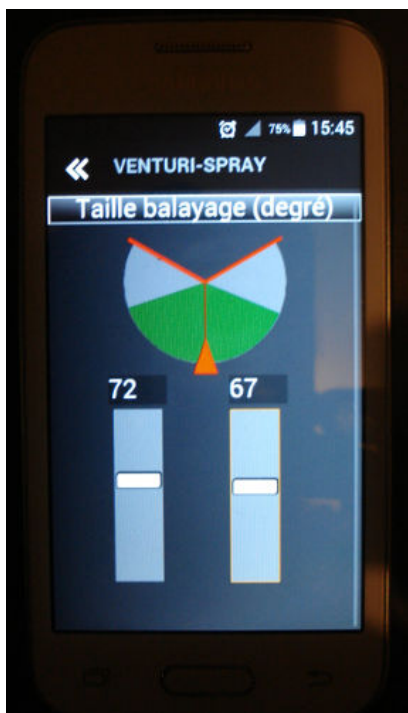
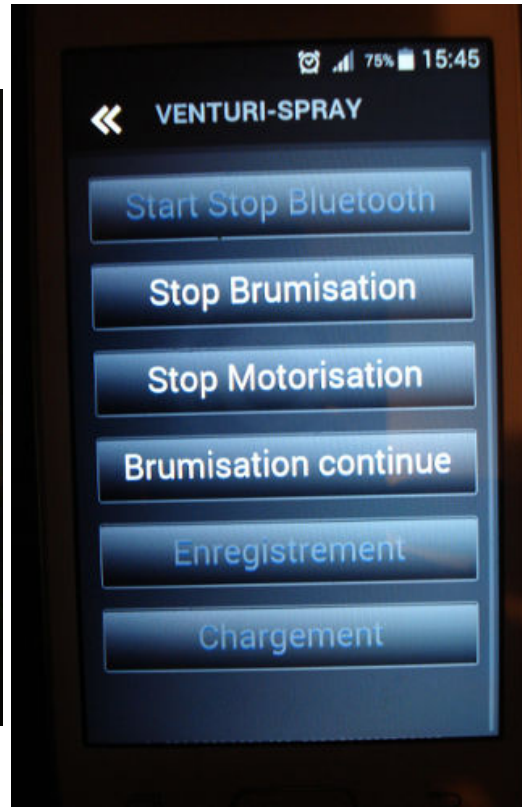
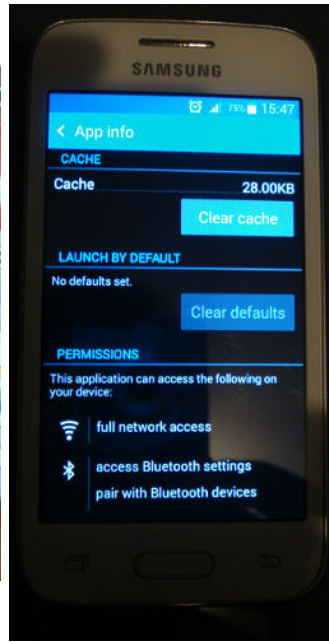
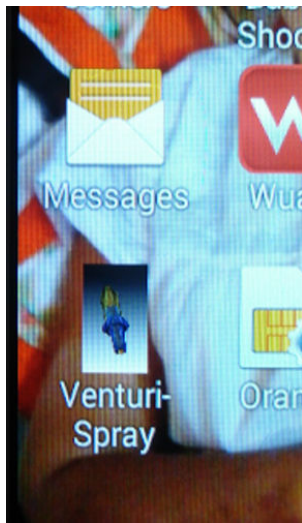
*Le seul problème est que l'on veut pouvoir en tout temps piloter le moteur (arrêt, changement d'angle, changement de vitesse), ce qui conduit à ne pas pouvoir utiliser la bibliothèque "stepper", mais intégré la gestion moteur au programme.*

*Vous trouverez le programme général (bluetooth et gestion moteur) ici : [VenturiArduinoV3](#) et son organigramme [Organigramme.pdf](#)*

*Quelques vues du programme de pilotage*

*Je ne vous laisserais pas le programme réalisé avec Cordova, car vraiment c'est une galère, pour arriver à ce résultat (qui ne fonctionne que sous Android en plus !) et je ne le souhaite à personne. Voir plus loin, je laisse le code avec la transmission wi-fi, c'est mieux.*



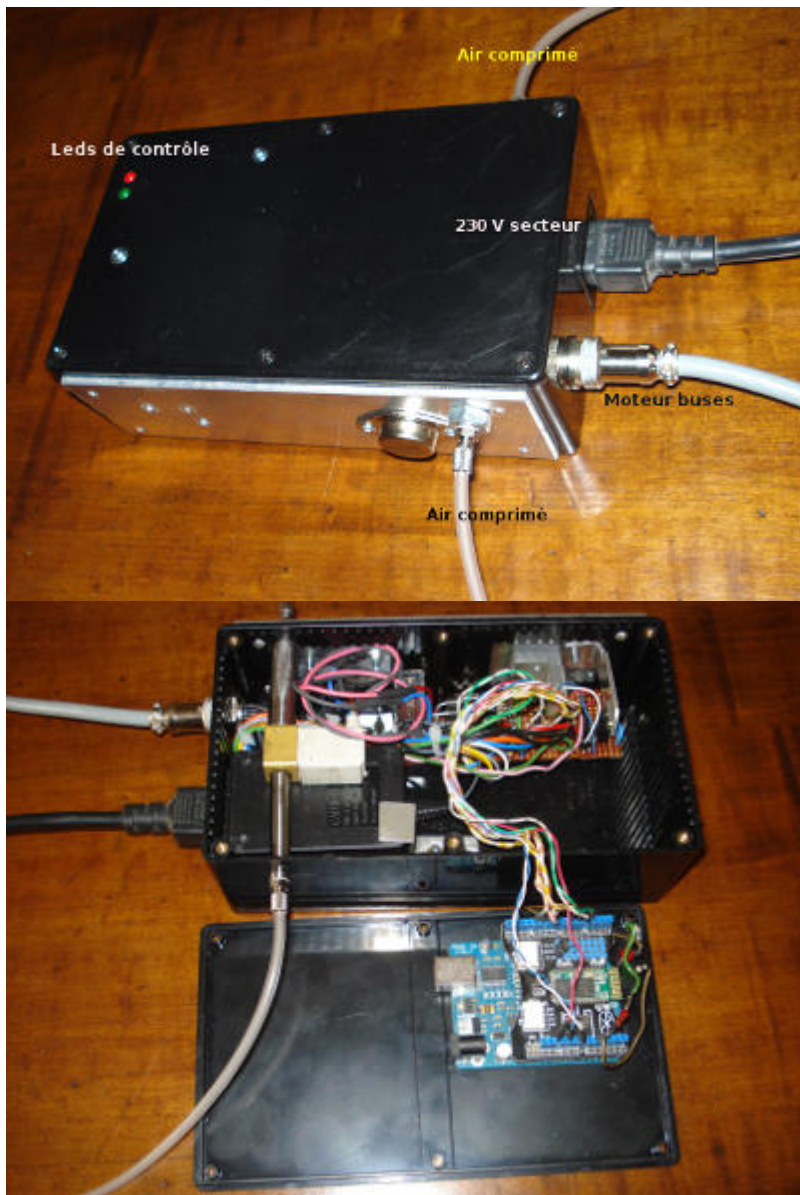


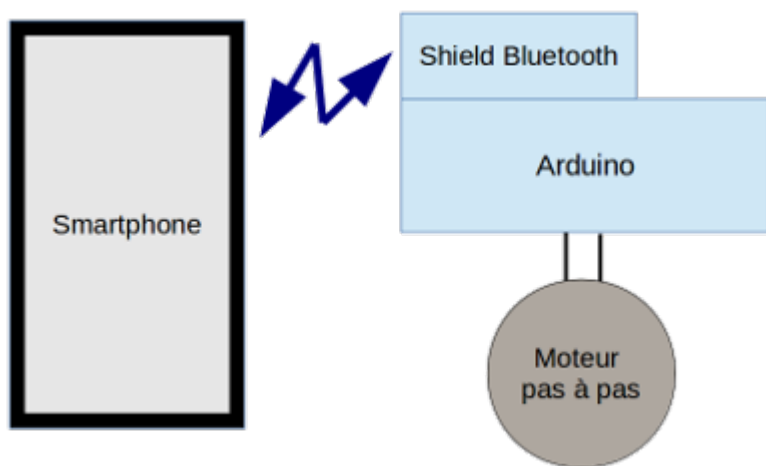
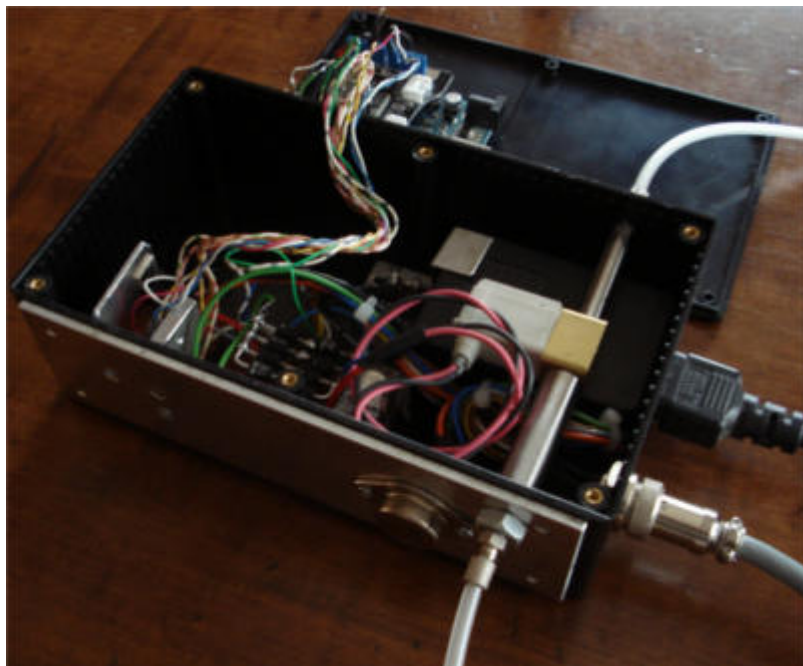
## L'électronique de commande

*Le boîtier de commande intègre :*

- un driver pour moteur pas-à-pas
- un relais avec une électrovanne (eau)

- un Arduino avec Bluetooth pour le dialogue.
- deux leds de contrôle





*Le moteur est piloté directement l'Arduino Yun sur le LM298N, après quelques essais sur divers moteurs, nous avons opté pour un moteur 2.5V pour 2.8A avec 0.9° de pas. C'est plus puissant que ce que j'avais prévu mais l'avantage principal est que le moteur sert de support aux bras de rotation, ce qui simplifie très nettement la mécanique et fin de compte le prix total n'est pas beaucoup plus important. Le seul problème restant étant le poids de l'ensemble que l'on doit contenir avec des barres en triangulation.*

## Problèmes !!

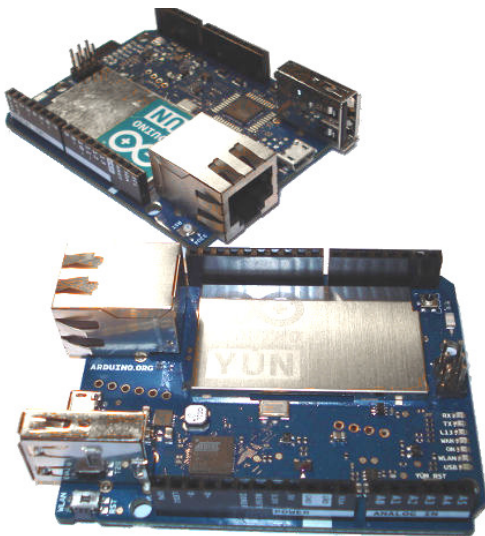
*Voilà tout semble fonctionner mais en fait nous avons un problème car la commande de moteur pas à pas avec réception (dialogue) Bluetooth demande trop de calcul à l'Arduino Uno. Résultat : la commande du moteur pas-à-pas fonctionne très bien seule et la*

*transmission Bluetooth, seule aussi parfaitement. Mais les deux simultanément pas possible ! De plus ce système présentait deux autres problèmes pas très visible mais néanmoins handicapant. Premièrement la programmation du logiciel du smartphone était lourde et difficile et ce programme ne fonctionnait que sur les dernières générations de smartphones Android. Deuxièmement la communication Bluetooth permet une distance d'au maximum 10m en ligne directe et il est difficile d'envisager une commande à distance via le réseau internet.*

*Nous avons changé radicalement de dialogue avec le smartphone.*

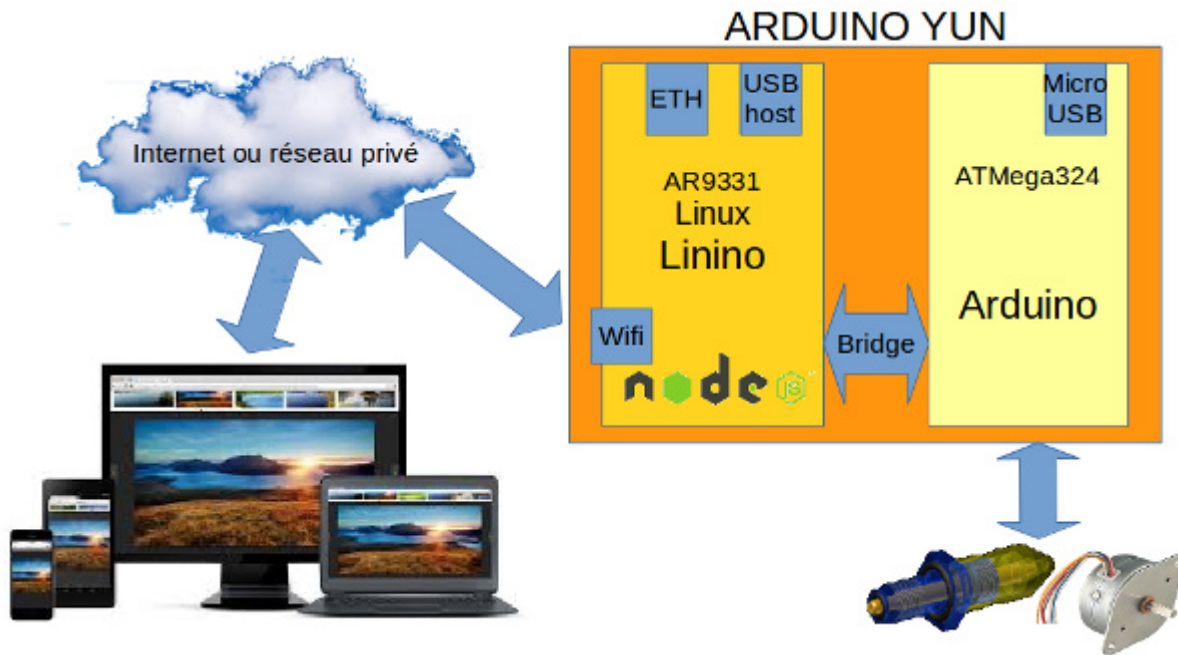
## Wi-Fi et Arduino Yun

*Fort de ces constats, j'ai opté pour une carte de commande plus puissante : l'Arduino Yun et je suis passé au Wi-Fi. Le principe de communication avec cette carte de commande sera le suivant :*



Arduino Yun





*Comme l'Arduino Yun est suffisamment puissant pour servir de serveur web, j'ai changé d'architecture de transmission de commande. A chaque demande http (recherche web) parvenant au serveur web intégré à l'Arduino Yun, node.js pour les spécialistes, il est retourné une page web qui est semblable au logiciel de pilotage précédent.*

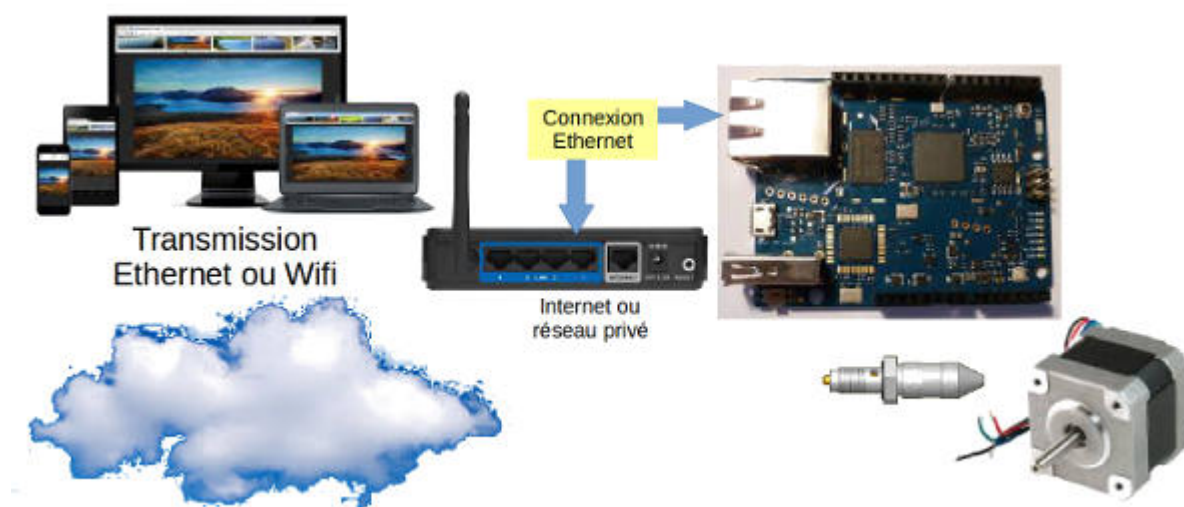
*Visuellement il n'y a pas de grande différence avec la version Bluetooth, fonctionnellement c'est notablement différent. Dans un cas c'est un logiciel spécifique qui ouvre une liaison Bluetooth avec la commande qui contrôle le dialogue. Dans la nouvelle approche c'est un composant d'un réseau Wi-Fi qui répond à une sollicitation extérieure (demande de page web).*

*Cette carte, l'Arduino Yun, est soit intégrée dans un réseau local, soit elle crée son propre réseau Wi-Fi. Le choix du réseau dépend du client. Dans le cas de l'intégration de la commande de brumisation à un réseau local il est possible de faire de la commande à distance par le réseau internet via une ouverture dans le routeur local. Enfin autre avantage, ce système (serveur web) permet de piloter la brumisation depuis n'importe quel système doté d'un navigateur web (PC, tablette, smartphone, Android, IOS, Windows, Linux, etc,,,) ceci en simultané ou unitaire. J'ai choisi le dialogue avec les websockets bien que cela ne soit pas nécessaire, mais pour explorer cette voie.*

## Problème non prévu (comme toujours !!!!!)

*Le Yun fonctionne très bien en serveur Web, mais il se pose de grosses difficultés s'il y a nécessité de "faire reset" sur le système. Par exemple une coupure de courant, un redémarrage etc. En fait j'utilise l'Arduino Yun avec une micro-carte pour étendre sa mémoire (node.js prend de la place) et ceci "influence" sur le reset du Wi-Fi : impossible de faire un reset propre du Wi-Fi.*

*Après une "guerre", j'ai abandonné d'utiliser l'Arduino Yun comme point d'accès Wi-Fi. J'utilise un routeur Wi-Fi qui centralise les connections. Le schéma final est le suivant :*



Vous trouverez les fichiers du code ici : [App Venturi-Spray](#)