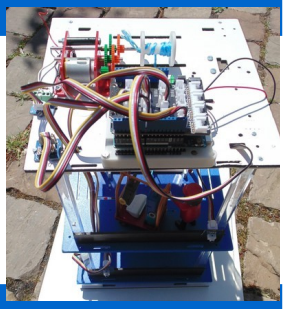
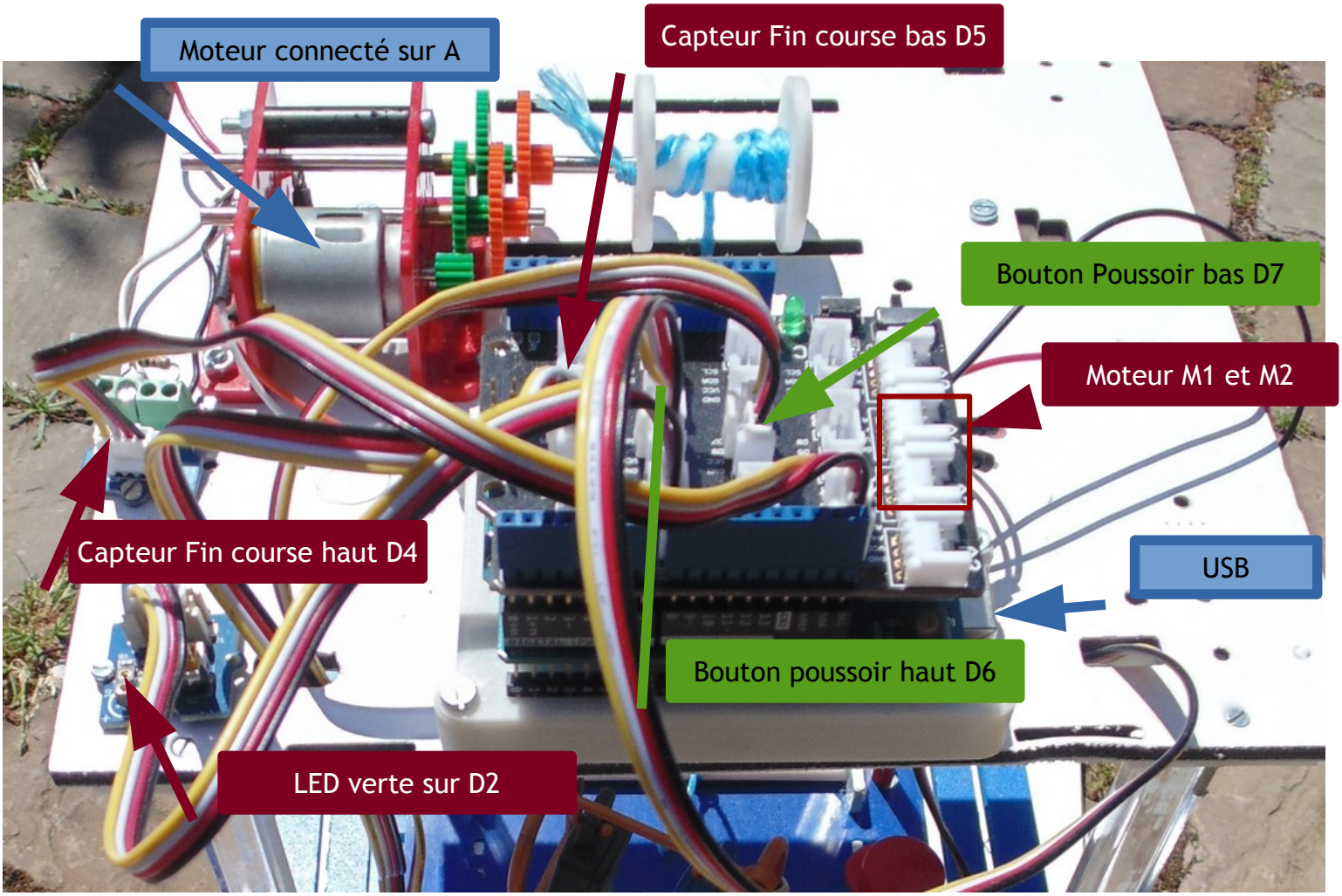
1 - LE MONTE CHARGE



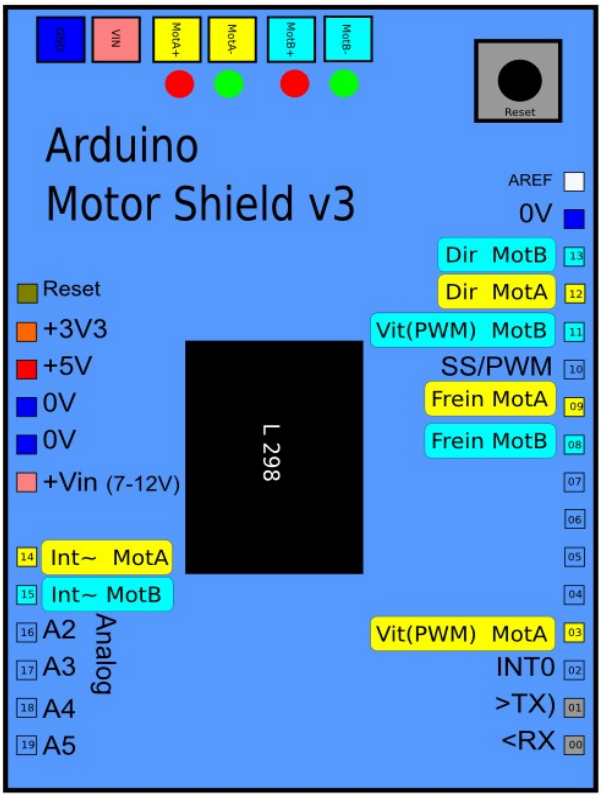
Le monte charge est réalisé sur une base d'un châssis classique. On utilise une carte ARDUINO UNO, avec un shield moteur v3 et un shield GROVE superposés sur la carte Arduino. On a utilisé la version 3.4.11 de mBlock.

2- ANALYSE DE L'ENSEMBLE ARDUINO UNO + SHIELD MOTEUR + SHIELD GROVE

Les cartes sont superposées, l'une sur l'autre pour créer l'ensemble. Le shield moteur permet de piloter le moteur du monte charge. Pas besoin d'une alimentation supplémentaire. Le shield Grove permet de pouvoir connecter des accessoires sans penser à la résistance pull up ou pull down nécessaires sur les cartes Arduino. Ici on les utilisera pour les capteurs haut et bas qui ne sont pas en Grove. Une résistance de 10kOhms est nécessaire.



3 - Analyse du shield moteur



La carte shield moteur a plusieurs particularités dont il faut tenir compte.

Pour le moteur A, on utilisera les broches :

- 12 : si = 1 dans un sens, si = 0 dans l'autre

- 09 : si = 1 stoppe le moteur, si = 0 le moteur peut tourner

- 03 : on met une valeur 'analogique' entre 0 et 255 pour faire varier la vitesse du moteur. On peut l'utiliser pour décélérer la vitesse de la cabine par exemple quand il arrive presque à destination.

4 - Analyse du shield Grove

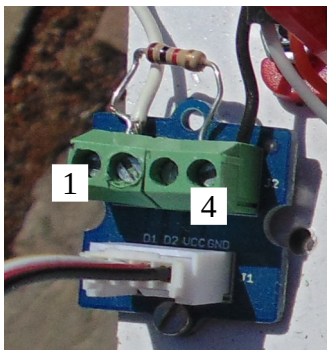
Le shield moteur permet de connecter des éléments sans s'occuper des résistances pull up ou pull down.

On va utiliser les broches suivantes dans notre cas :

- D2 = on connecte la led Verte, on pourra l'utiliser pour indiquer que la cabine monte ou descend.
- D4 = on connecte le capteur fin de course haut. Il faudra ajouter une résistance pull up car il n'y a pas de fin de course Grove.
- D5 = on connecte le capteur fin de course bas. Il faudra aussi ajouter une résistance pull up.
- D6 = on connecte le bouton poussoir haut en Grove.
- D7 = on connecte le bouton poussoir bas en Grove.

5 - Analyse des capteurs de fin de course

Le shield moteur permet de connecter des éléments sans s'occuper des résistances pull up ou pull down.



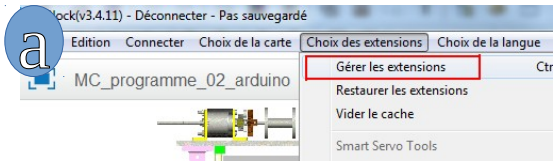
On utilise une résistance sur un connecteur Grove, ce qui permet de connecter différents éléments :

- Ssur la borne 1, une extrémité + une résistance
- sur la borne 2 rien
- sur la borne 3, l'autre borne de la résistance
- sur la borne 4, l'autre extrémité du capteur.

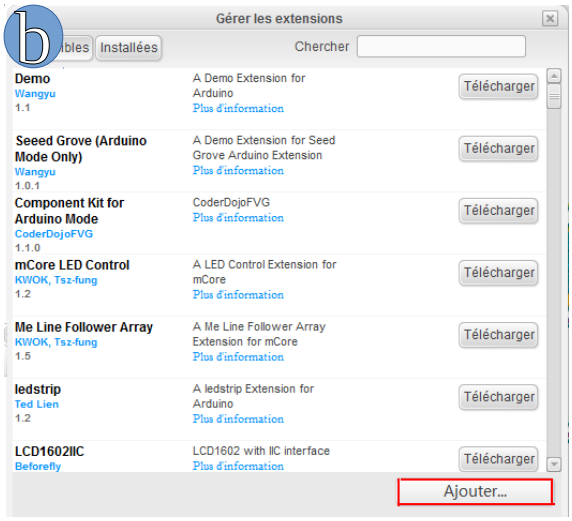
6 - Analyse de la connexion de la carte arduino

- Lorsque vous branchez la carte arduino il est préférable d'installer les pilotes de la carte arduino avec le logiciel arduino.
  https://www.arduino.cc/en/Main/Software
- Vous devez ensuite télécharger les extensions GROVE pour le logiciel mBlock 3.4.11
  https://www.technologieservices.fr/mBlock-extensions-ts-3-4-11-ress-175416.html
- Vous trouverez toutes les instructions nécessaires pour ajouter cette extension dans le logiciel mBlock.

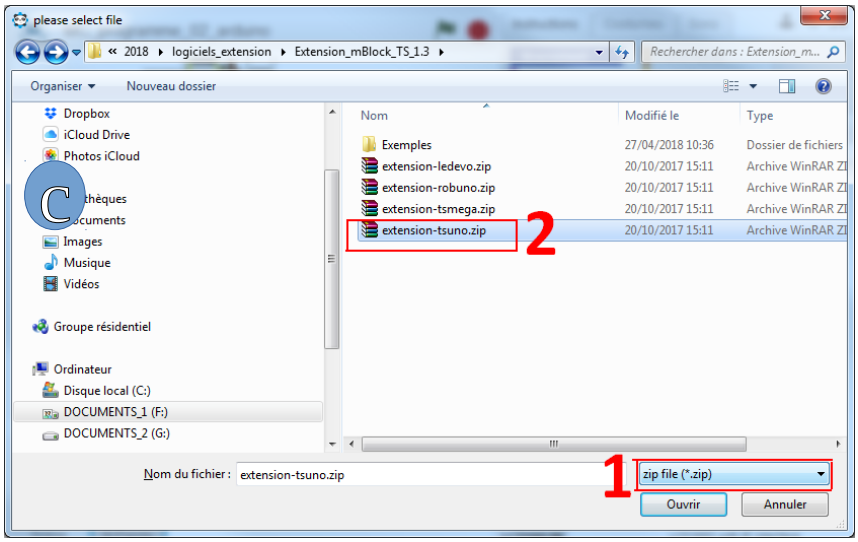
a



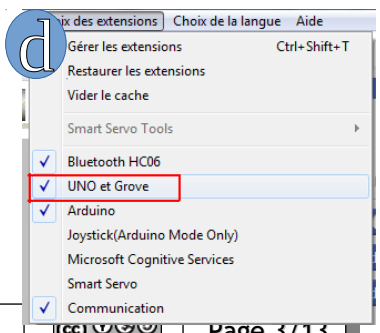
b



c



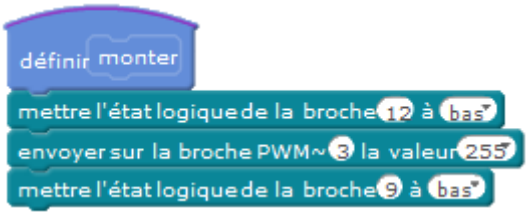
d



7 - Analyse du programme à réaliser pour piloter le monte charge en mode connecté

7.1 - Analyse du programme pour faire monter la cabine

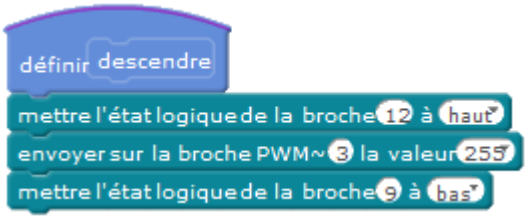
On va créer un bloc personnalisé pour chaque élément du programme.
Pour monter la cabine on réalise le programme suivant :



En fonction de la connexion de votre moteur peut être qu'il faudra inverser l'état de la broche 12.

7.2 - Analyse du programme pour faire descendre la cabine

Pour descendre la cabine on réalise le programme suivant :



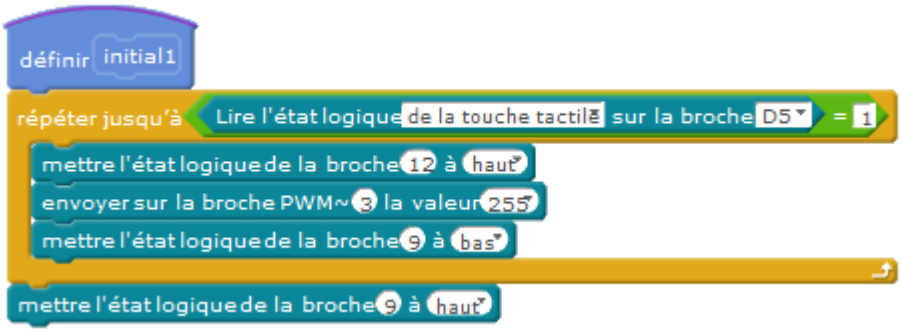
En fonction de la connexion de votre moteur peut être qu'il faudra inverser l'état de la broche 12.

7.3 - Analyse du programme pour arrêter le moteur

Pour arrêter le moteur il faut mettre l'état logique de la broche 9 à haut pour le moteur A :

7.4 - Analyse du programme pour initialiser le monte charge

Afin de pouvoir initialiser le monte charge au départ, on va descendre la cabine jusqu'à ce qu'elle arrive en bas. On le saura grâce au capteur connecté en D5 :

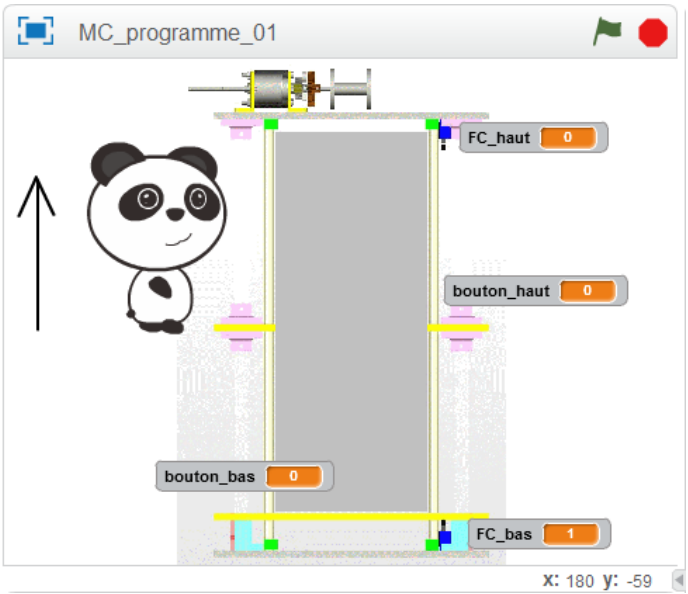


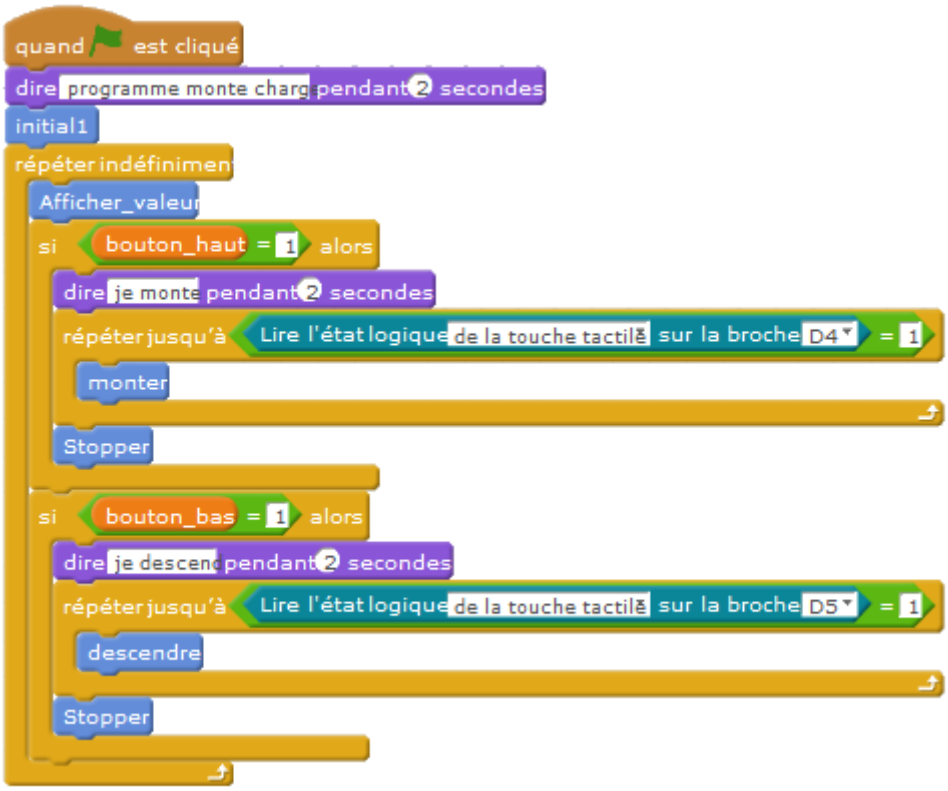
7.5 - Analyse du programme pour afficher les valeurs des capteurs

Ici, on va créer un bloc pour afficher les valeurs dans des variables :

7.6- Analyse du programme pour piloter le monte charge

On utilise une image pour créer le décor :





quand on démarre le programme
un texte s'affiche
initialisation (en bas)
répéter indéfiniment
on récupère les variables
si bouton haut appuyé
'je monte' s'affiche
répéter monter jusqu'à ce que la cabine soit en haut

Stopper le moteur

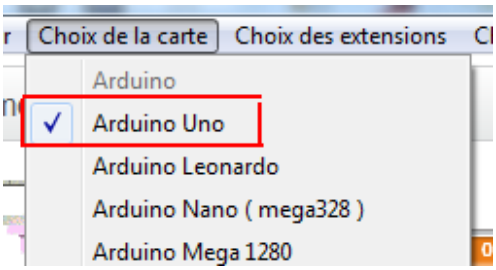
Si bouton bas appuyé
'je descend' s'affiche

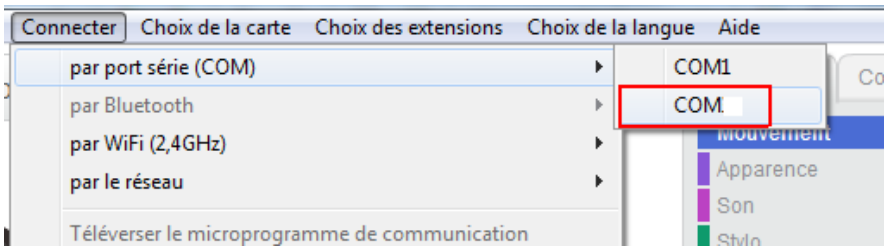
Répéter descendre jusqu'à ce que la cabine soit en bas

Stopper le moteur

7.7 - Analyse pour connecter la carte Arduino

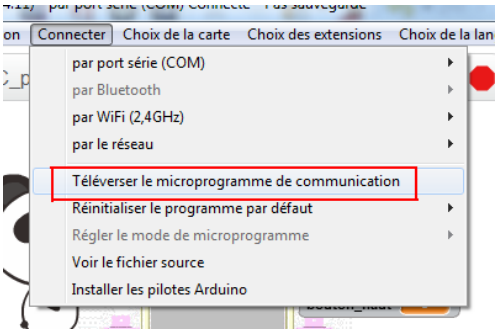
Connecter votre carte Arduino sur l'ordinateur, le pilote USB s'installe. Si ce n'est pas le cas, installer le logiciel Arduino avec les driver : https://www.arduino.cc/en/Main/Software





En général la carte se connecte sur le dernier port COM. Si cela n'est pas le cas, vérifier sur la gestion des périphériques de Windows.

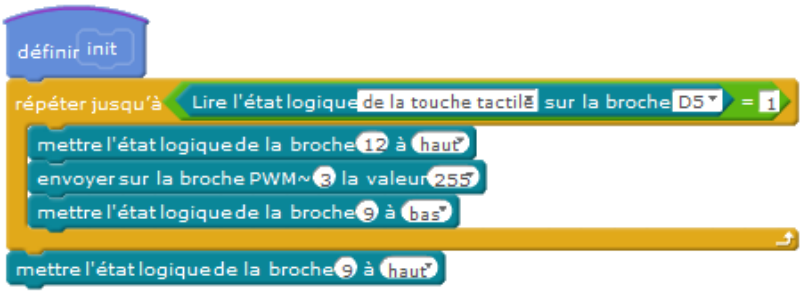
Puis :

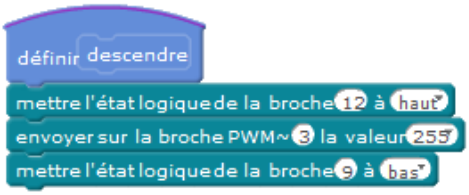


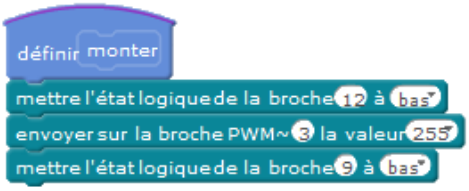
8 - Analyse pour programmer le monte charge en mode autonome

Dans le cas du programme précédent, il faut que l'ordinateur soit connecté sur la carte Arduino. Il est possible de faire un programme autonome qui permet d'utiliser le monte charge sans ordinateur. Par contre il est nécessaire dans votre programme de ne pas utiliser d'objet graphique rajouté (chandelier, personnage, etc) ni des éléments de l'ordinateur (clavier, etc).

On utilise les mêmes blocs (sous programme) que précédemment, sauf afficher les valeurs.



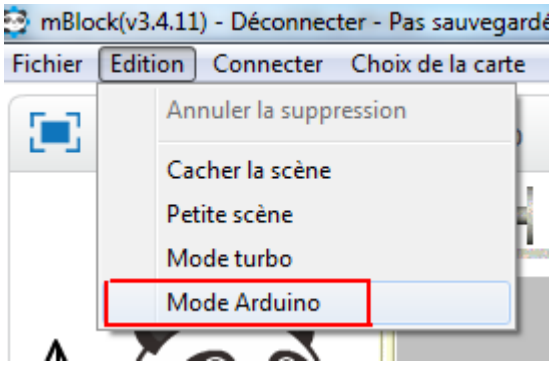


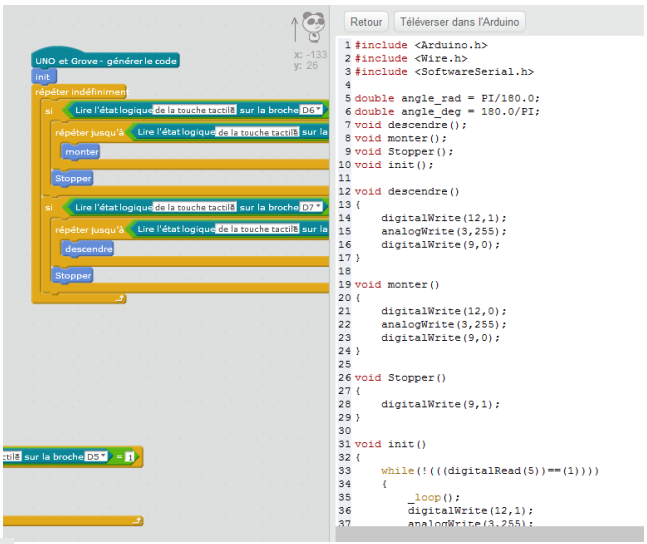


Il faut générer le code machine (compiler) qui sera ensuite téléversé dans la carte.

Attention à ne pas mettre de bloc nommé 'init', c'est incompatible avec le code Arduino. Il est déjà utilisé pour autre chose.

On va basculer dans le mode Arduino :





Votre maquette est maintenant autonome, vous pouvez juste connecter votre carte à une alimentation et votre maquette est opérationnelle.

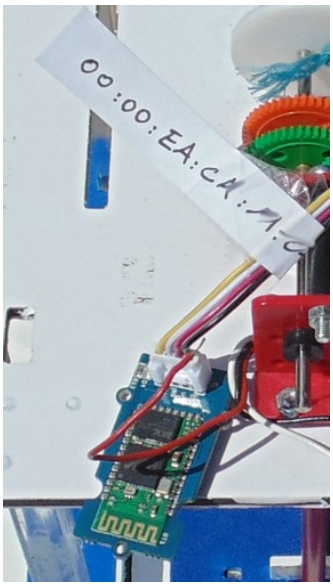
9 - Analyse pour piloter le monte charge avec une application android

On va utiliser AppInventor, une application de Google qui permet de créer graphiquement des applications pour une tablette Android.
Il y a 2 possibilités:

- soit avec un compte Google vous pouvez réaliser des applications http://appinventor.mit.edu/explore/ puis 'Create apps' !

- soit en téléchargeant la version portable de app inventor et la mettre en serveur autonome, donc plus besoin de compte google :
http://www.pedagogie.ac-nantes.fr/technologies-et-sciences-des-ingenieurs/documentation/didacticiels-tutoriels/appinventor-en-local-sur-votre-reseau-1025746.kjsp?RH=1160751856953

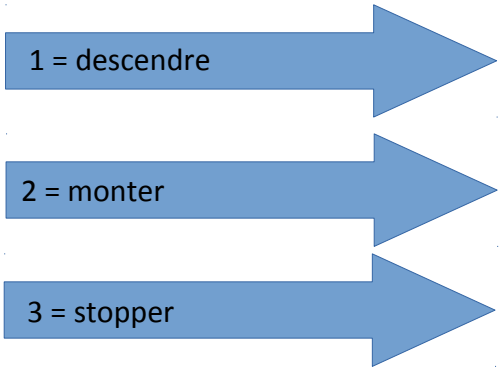
On doit créer un programme qui va permettre de communiquer entre l'application et la carte arduino.
Pour communiquer vous avez besoin d'un module bluetooth Grove que l'on va connecter en entrée D2.

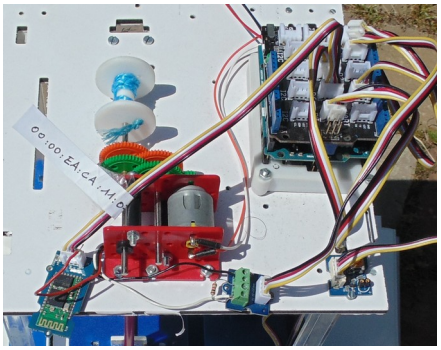


Ici, j'ai noté l'adresse mac du module bluetooth qui va apparaître sur votre tablette lors de la première connexion.
C'est utile de le faire surtout si vous avez plusieurs modules bluetooth.
En général le code pin du module est 1234, ou parfois 0000.
Ce code peut être changé dans le programme mBlock.

Nous allons décider d'un **protocole** entre la tablette et la carte Arduino :

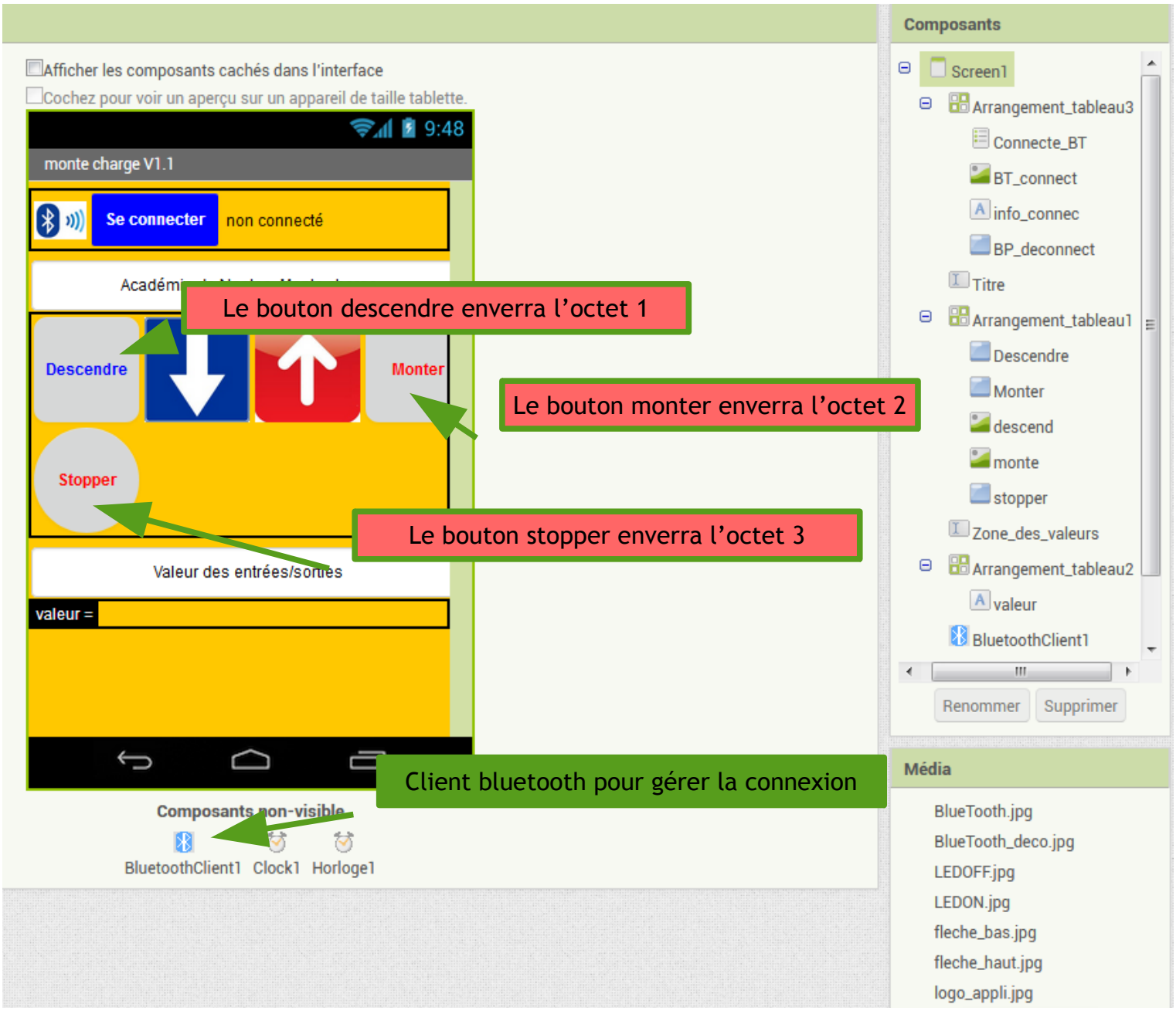
- on enverra le chiffre (octet) 1 pour descendre
- on enverra le chiffre (octet) 2 pour monter.
- on peut aussi décider également d'envoyer le chiffre 3 pour stopper le monte charge.





9.1 - Analyse pour créer le programme AppInventor : interface utilisateur

On va créer un programme sur AppInventor avec des boutons et des images.
On utilise la fonction bluetooth d' AppInventor.

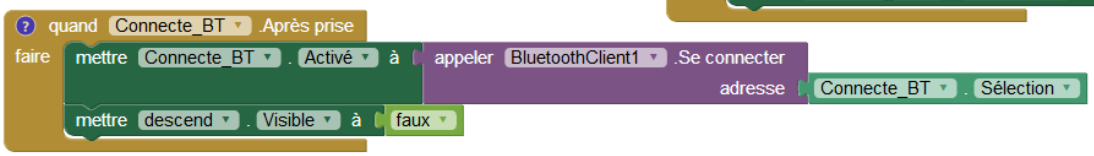


9.2 - Analyse pour créer le programme AppInventor : le programme

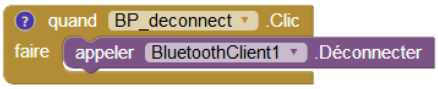
On va maintenant réaliser le programme pour gérer la connexion et envoyer les octets quand on appuie sur les boutons.

Pour gérer la connexion bluetooth, c'est toujours la même chose. :

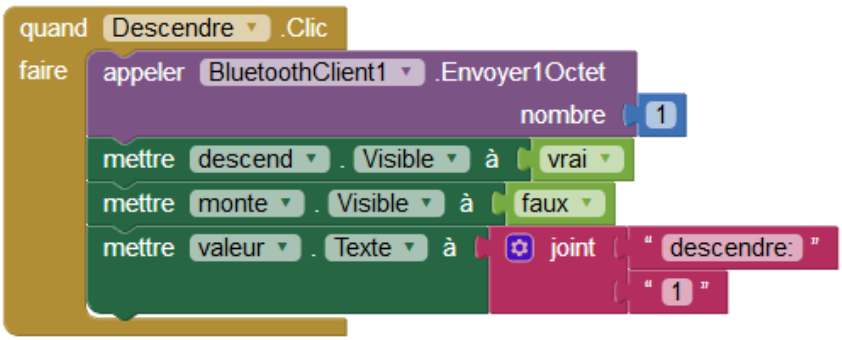
Ici, on initialise la connexion



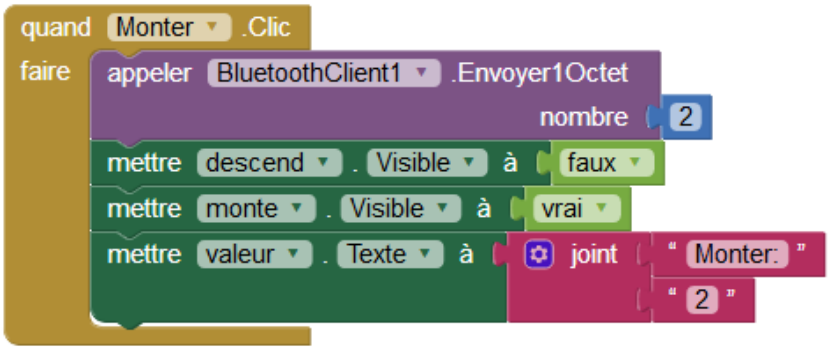
Quand on appuie sur le bouton *connecter,*
on lance la connexion,
on rend invisible le bouton descendre
puisque la cabine est en bas.



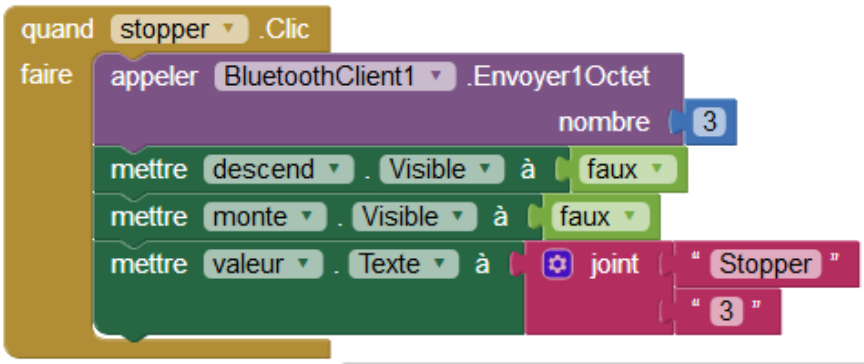
Quand on appuie sur le bouton *deconnec,*
on déconnecte le client bluetooth.



Quand on clique sur le bouton descendre :
- on envoie le nombre 1 par bluetooth
- on rend visible l'image descendre
- on cache l'image qui monte
- on affiche un texte sur l'écran



Quand on clique sur le bouton monter :
- on envoie le nombre 2 par bluetooth
- on rend visible l'image qui monte
- on cache l'image qui descend
- on affiche un texte sur l'écran

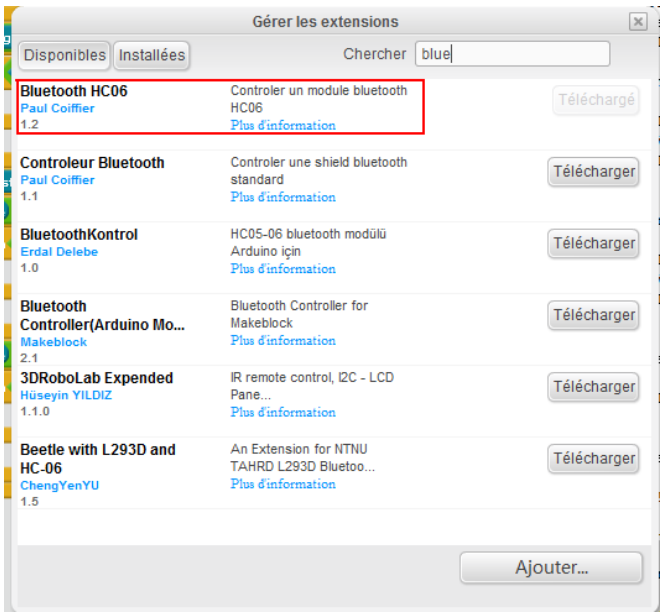


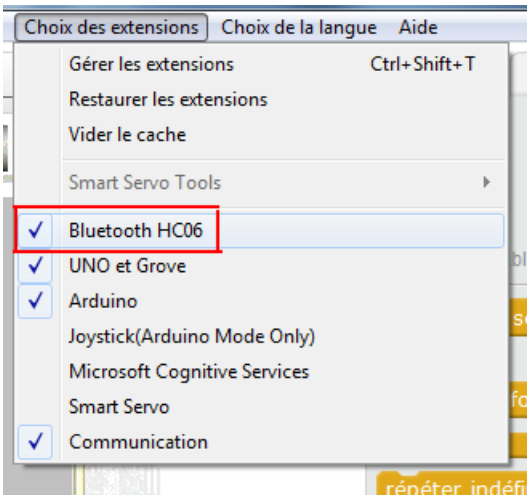
Quand on clique sur le bouton stopper :
- on envoie le nombre 3 par bluetooth
- on cache l'image qui monte
- on cache l'image qui descend
- on affiche un texte sur l'écran

9.3 - Analyse pour créer le programme mBlock pour piloter le monte charge par BT

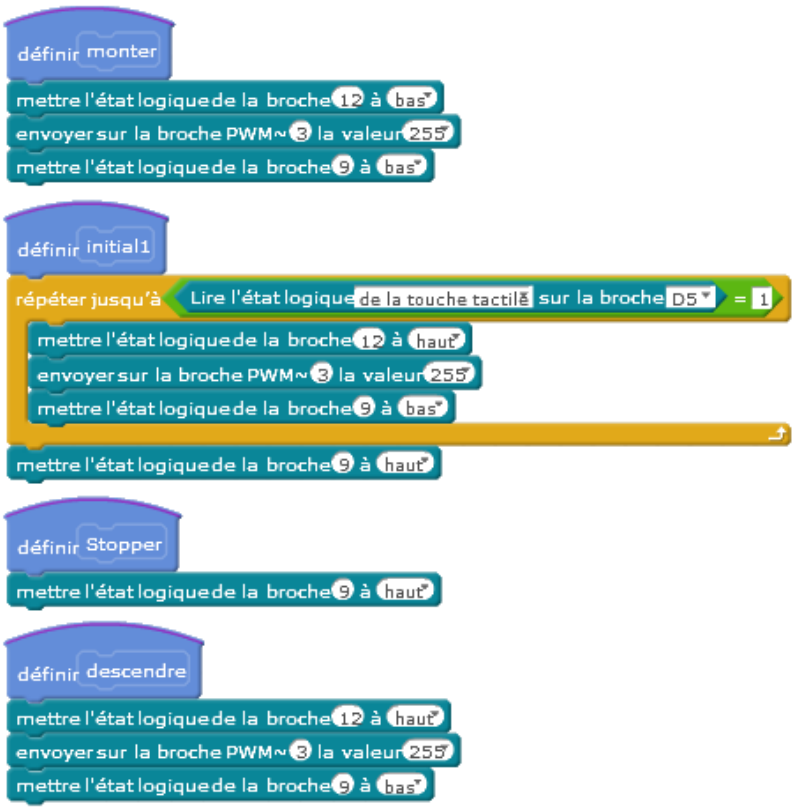
On garde la même structure que le programme précédent qui pilotait la maquette en autonome.
On va ajouter une extension pour gérer le module bluetooth.

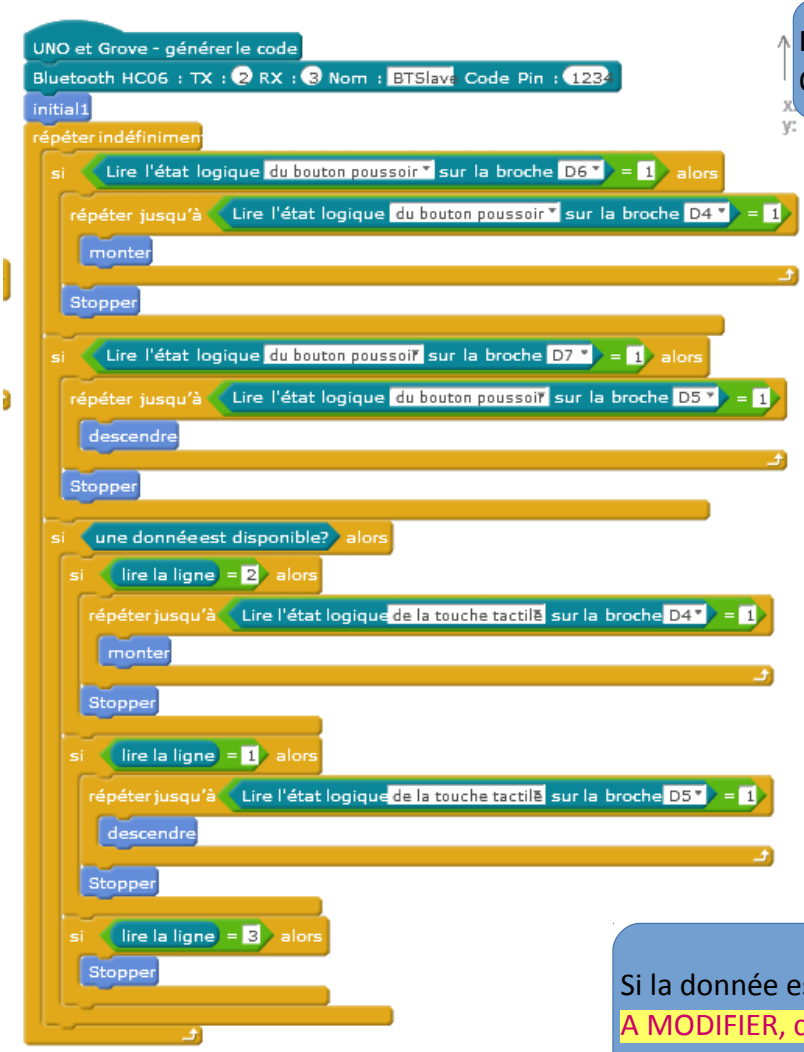
Dans les extensions disponibles, on va ajouter l'extension bluetooth HC06 :





On garde les mêmes blocs que précédemment :





Ici on active le module bluetooth connecté en D2.
On peut changer son nom et le code pin.

Si une donnée arrive sur le module bluetooth alors on va lire les lignes arrivées.

Si la donnée est = 2 alors on va monter jusqu'à ce que la cabine soit en haut.

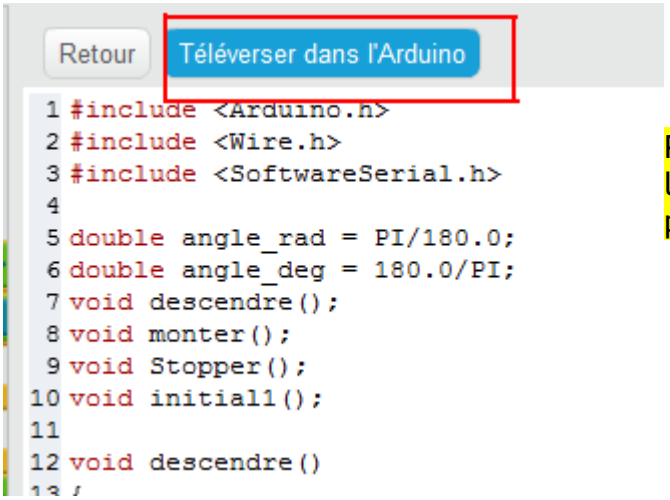
Puis on stoppe.

Si la donnée est = 1 alors on va descendre jusqu'à ce que la cabine soit en bas.

Puis on stoppe.

Si la donnée est = 3 alors on stoppe
A MODIFIER, car précédemment il y a des boucles.

Pour finir, il faut téléverser le programme en mode Arduino :



C'est parti, patience mais votre programme fonctionne.

PS : la fonction Stopper doit être retravaillée car le programme est en boucle et l'information ne peut être traitée à ce moment là.