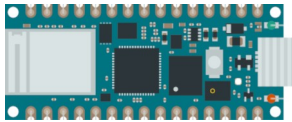


Ensemble gyroscopique de commande des projecteurs de virage.

Composition de l'ensemble :

La carte électronique Arduino à base de microcontrôleur MPU-6050 alimentée en 5 Volts qui fait tourner le programme qui gère la carte gyroscope-accéléromètre-thermomètre, c'est à dire

- l'acquisition des données des 3 gyroscopes (on n'en utilisera que 2),
- l'acquisition des données des 3 accéléromètres (inutilisées),
- l'acquisition de la température utile pour corriger les dérives des gyroscopes,
- la commande d'allumage de chacun des 2 projecteurs.



Carte Arduino Nano

Beaucoup de documentation, d'exemples de réalisations et d'explications sur le web, existe avec tous les modèles de

connecteur USB miniatures.

La carte électronique du gyroscope-accéléromètre-thermomètre dont les données sont lues par la carte Arduino selon le protocole I2C.

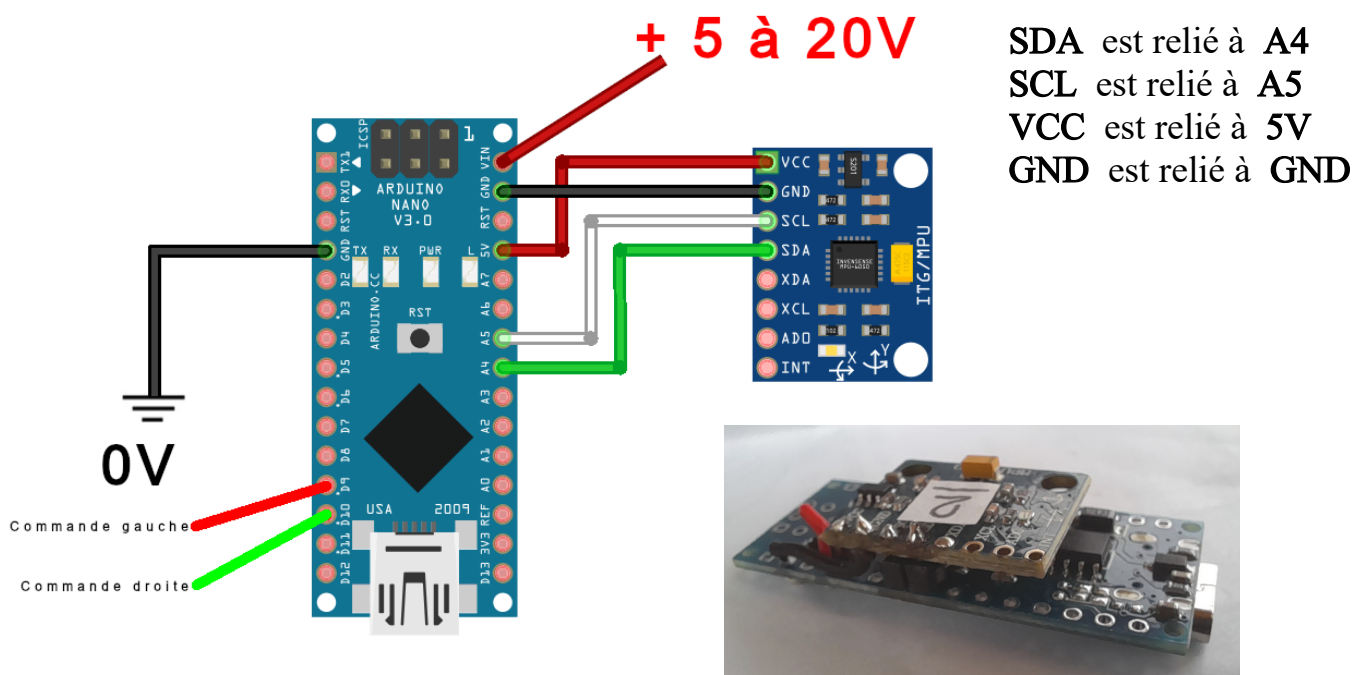


Module MPU 6050 GY-521

Gyroscope Accéléromètre 3 Axes + température.

Interconnexion des 2 cartes :

La carte gyro est alimentée par la sortie régulée 5V de l'Arduino



SDA est voisine de SCL, pareil pour A4 et A5, on peut donc solidariser les cartes en soudant les 2 picots correspondants et réaliser ainsi la liaison I2C, les 2 autres liaisons se font en soudant des fils isolés.

A ce stade, on peut commencer les essais.

1 - Utiliser la petite appli " Arduino IDE " gratuite et facile à trouver sur le web :

<https://www.arduino.cc/en/software>

vous pouvez apparemment l'utiliser en ligne ou le télécharger sur votre ordi.

En reliant le port USB de l'Arduino à celui de l'ordinateur, pas besoin d'autre alimentation tant que la carte est reliée à l'USB.

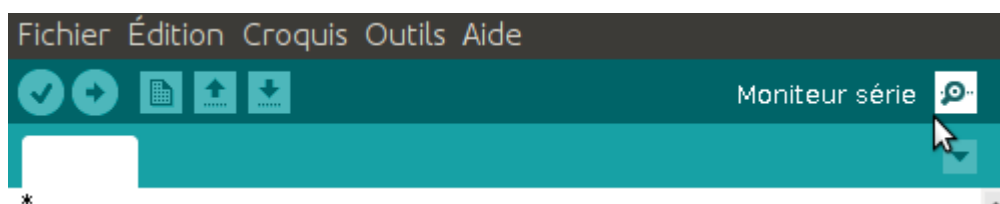
Vérifier qu'il fonctionne en ouvrant un modèle basique :

Fichier/Exemples/01.Basics/Blink par exemple.

Dans le menu " Fichier " cliquer téléverser et vérifier qu'une LED clignote.

2 - **Tous les gyroscopes n'ont pas tous exactement les mêmes caractéristiques**, elles varient de façon substantielle avec la température, mais on a de la chance car la puce contient aussi un thermomètre.

Il faudra utiliser cette fois le programme " Gyrotest,ino " que j'ai en grande partie récupéré sur le web, l'ouvrir et le téléverser dans l'Arduino pour vérifier la communication en ouvrant le moniteur série :



Exemple de ce qu'on voit :

Deg.C GyroX GyroY GyroZ

18,957 0,515 0,131 -0,3 si la température augmente vous verrez apparaître d'autres lignes.

3 - **Acquisition des paramètres de vos gyros :**

Munissez-votre Arduino + Gyro de son cordon USB, empaquetez-le dans un sac plastique, joignez-y une masse d'inertie thermique ou mettez- le dans un boîtier métallique, le tout bien emballé dans un emballage isotherme (à vous d'improviser) en laissant sortir l'extrémité du cordon et placez-le au congélateur un bon moment pour que la masse soit bien refroidie.

Quand c'est prêt ouvrez l'appli Arduino, branchez le cordon d 'Arduino sur l'ordi et assurez-vous que l'ensemble est bien calé car il ne doit pas bouger pendant le test, lancez immédiatement le moniteur série.

Vous verrez alors défiler les valeurs des 3 gyros à chaque 1/10 de degré C, laisser le défilement automatique.

L'idéal serait de partir d'environ -10°C et d'arriver à une température de 40°C pour voir large . . .

À vous de décider en fonction de votre usage ; vous pouvez sur la fin approcher une lampe à incandescence pour monter la température.

Enfin il faudra sélectionner et copier tout le listing en sélectionnant la dernière ligne puis Ctrl+A pour tout sélectionner et Ctrl+C pour copier et aller coller tout ça dans votre tableur

préférée et l'enregistrer (Ouf, Sauvé!). si les formules affichent #VALEUR ! vérifier s'il faut modifier le séparateur décimal (, ou .) pour que ça fonctionne.

4 - Calcul des coefficients de correction GyroXCorr, KTXCorr, KT2XCorr ici pour l'axe X :
Utiliser la feuille de calcul " GyroTestTemperature.ods " en collant les valeurs dans les bonnes colonnes.

Exemple pour le gyroscope X :

GyroXCorr corrige le décalage de 0 il suffit de noter la valeur de Gyrox pour 0°C et prendre la valeur de signe opposé.

KTXCorr corrige la pente générale du graphe, régler une valeur positive pour incliner le graphe vers le haut et inversement.

KT2XCorr corrige la courbure, régler une valeur positive pour courber le graphe (la concavité) vers le haut et inversement.

Procéder de même pour les deux autres gyros.

Ce travail fait, remplacer les valeurs des constantes GyroXCorr, KTXCorr, KT2XCorr, GyroYCorr, KTYCorr, KT2YCorr, GyroZCorr, KTZCorr, KT2ZCorr dans les déclarations au début du programme par les valeurs que vous avez trouvées puis vérifiez, enregistrez et téléverser le programme.

C'est presque terminé pour la programmation, le moniteur série vous renseignera sur le signe des valeurs issues des gyros :

En positionnant la carte Nano telle qu'elle sera à sa position définitive et en l'animant d'un léger mouvement de roulis ou de lacet vous déduirez quels sont les gyros qui correspondent à votre utilisation et le signe des valeurs.

Il faut que le roulis et le lacet soient positifs pour les virage à droite pour que les sorties 9 et 10 correspondent respectivement à l'allumage du phare gauche et du phare droit. une erreur sur le numéro de la sortie ne serait pas critique car on peut ensuite inverser les fils, en revanche comme le programme additionne les valeurs des deux gyros (le roulis permet d'anticiper sur le lacet) il est indispensable que les signaux de roulis et de lacet soient de même signe pour un virage donné. (on penche à Droite " roulis " pour un virage " lacet " à Droite et inversement).

Les lignes 106 et 107 de l'exemple sont à modifier en fonction de vos déductions :

Lacet = GyroX ou GyroY ou GyroZ ou -GyroX ou -GyroY ou -GyroZ, à choisir.

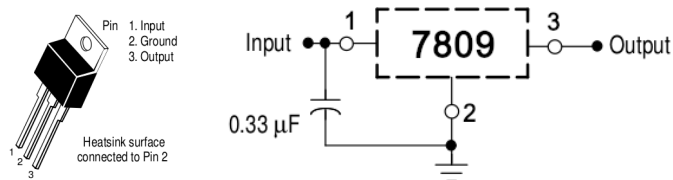
Roulis = à choisir de même.

et n'oubliez pas de vérifier, téléverser et enregistrer votre programme.

5 - Alimentation

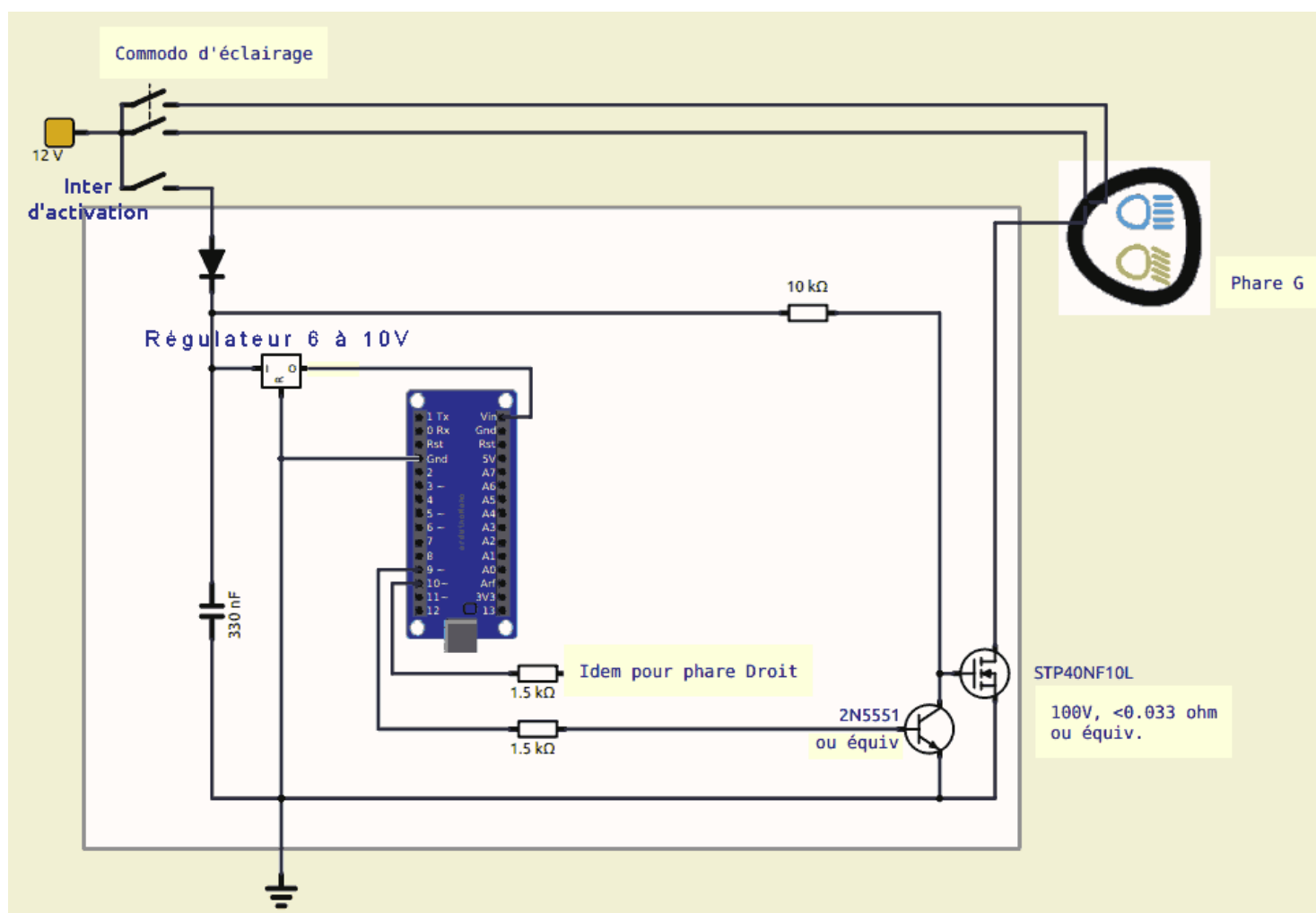
La carte Arduino (qui alimente la carte Gyro) peut être alimentée dans la plage 6V à 20V.

- Comme le dispositif doit être alimenté sur la batterie du véhicule, je conseille de passer par un régulateur donnant une tension intermédiaire de 8V à 10V car si un jour vous avez un problème sur votre circuit électrique le dispositif sera protégé. Un mauvais contact batterie ou un défaut du régulateur de charge peut conduire à des tensions (transitoires ou pas) élevées.



Utilisation d'un régulateur genre LM7809, existe chez de nombreux fabricants, le visser sur le boîtier métallique si possible bien qu'il ne chauffe pas vu la consommation du circuit,

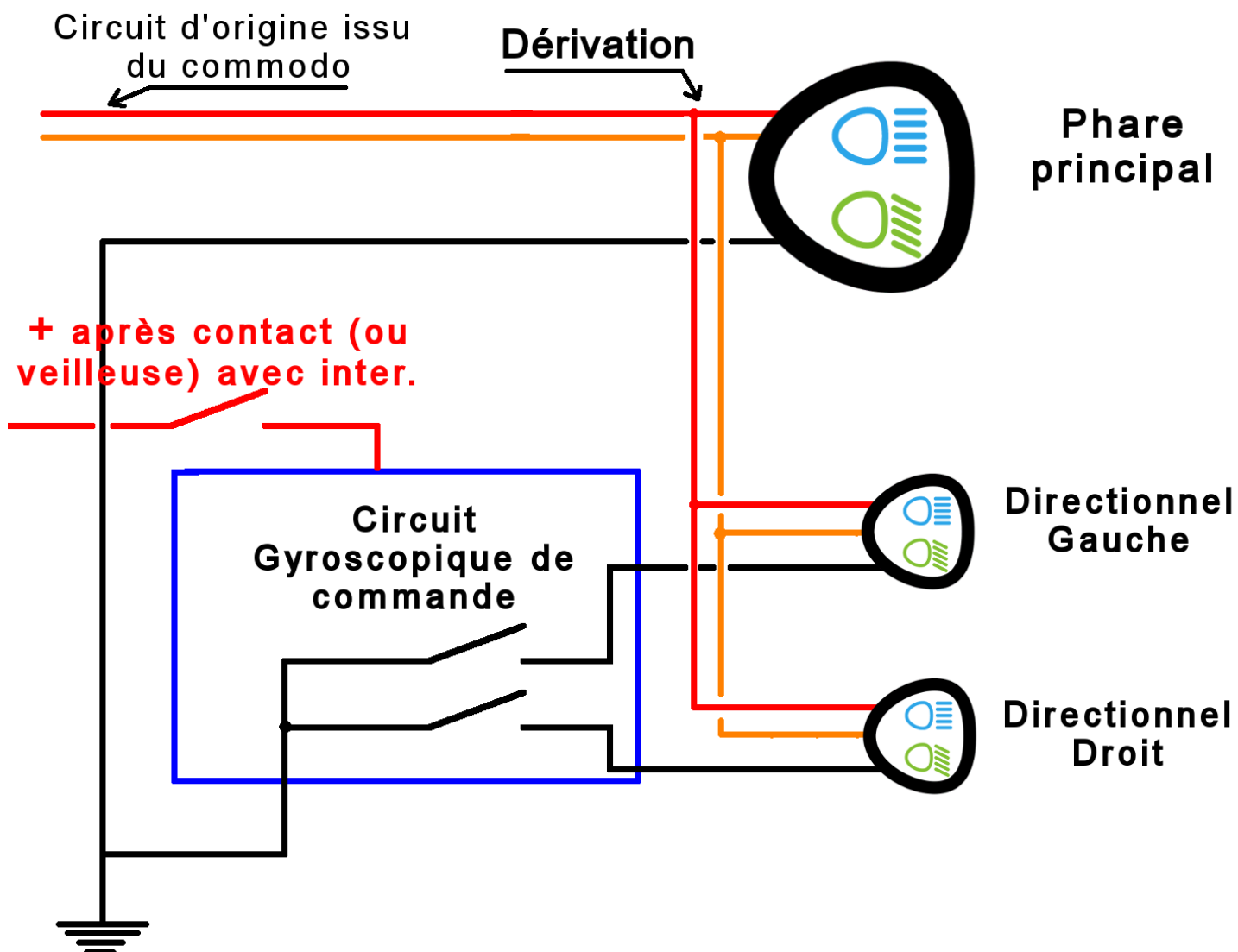
6- Interfaces de commande des lampes



Les composants ne sont pas très critiques (récup.), choisir un MOSFET à très faible résistance si pas de dissipateur, perso MOSFET vissé sur un boîtier alu pas de souci, une capacité de 0,33 μ F est préconisée pour le régulateur, une diode évitera les dégâts en cas d'erreur de branchement.

Prévoyez d'emblée un accès facile au connecteur USB pour pouvoir une fois installé adapter éventuellement le programme ou modifier la sensibilité.

7 - Câblage sur la moto :



Vous noterez que la commande s'effectue par le retour à la masse pour bénéficier du commodo côté positif, les phares additionnels devront donc comporter 3 fils dont aucun relié au boîtier,

J'ai probablement oublié des indications, ou laissé des erreurs ou des imprécisions, n'hésitez pas à me le signaler, je compléterai au fur et à mesure.