

Rapport de séance: Projet industriel 19

Drone de cartographie

Séance du 8 / 02 / 2019: Récupération matériel/distribution des tâches

Rédacteur: Vincent

Erwann:

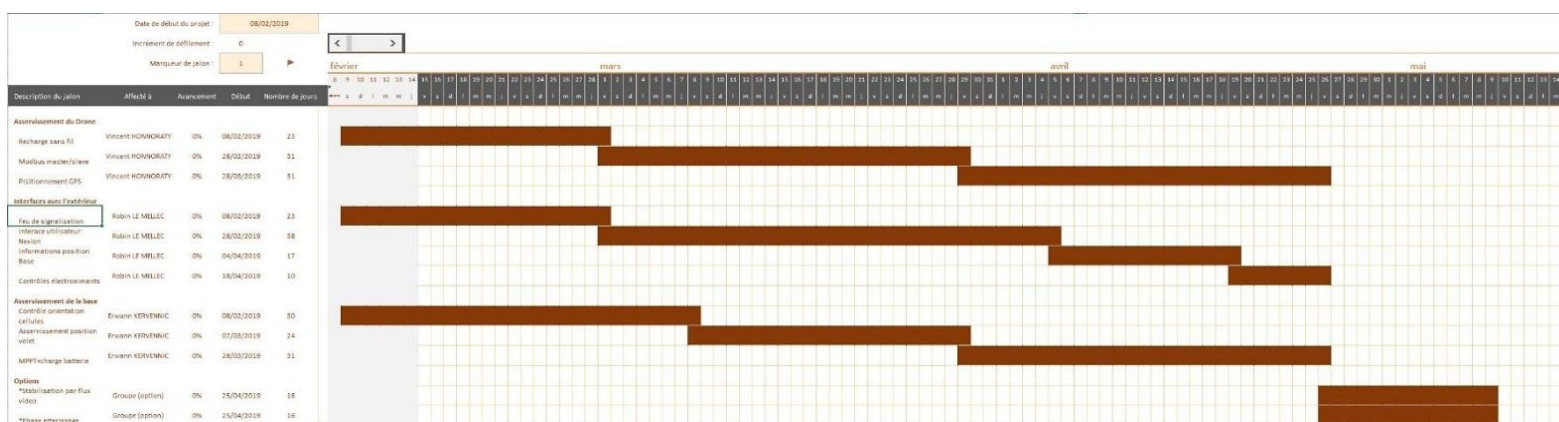
Définition mathématique de l'orientation des cellules solaires:

Recherche des équations des descriptions angulaires des volets entre eux.

L'objectif est qu'un volet sera asservi directement, les autres cellules devront s'orienter en fonction, de telle manière à qu'ils soient tous parallèles au même plan.

Rédaction sur feuille, simulation sur tableur excel.

Planification sous forme d'un gantt pour le semestre à venir.



Prochaine étape: implémentation des équations dans une arduino, test via terminale série avant d'être ajouté à la base solaire (quand les moteurs y seront installé, probablement dans une ou deux semaine), en l'attendant: réalisation du capteur solaire (tracking du Soleil)

Robin:

Programmation des feux de signalisation:

Découverte de la Getboard compatible raspberry pi, compréhension des fonctions présentes.

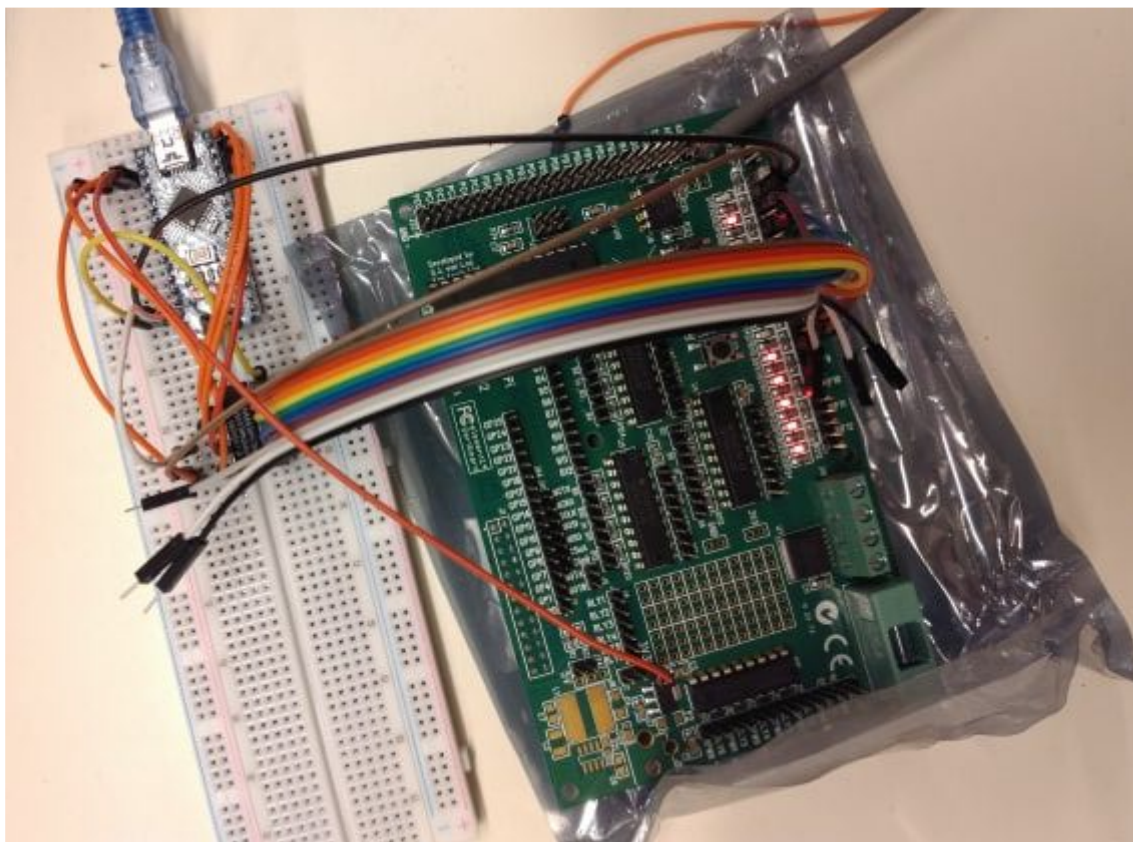
Programmation du atmega328p via FTDI232, pour utilisation de l'IDE arduino.

Les 2 LEDs RGB seront alimentés en parallèle (pin verte LED de gauche avec pin verte LED de droite, de même pour rouge et bleu), pareille pour les LEDs arrière. Choix fait pour limiter le nombre de sortie.

Finalement utilisation d'un arduino nano, de shift register et de la Getboard.

Utilisation du 74HC595 (shift register standard) avec librairie shiftout(pinData, pinVerrou, data), data sur un octet, relier au buffer de sortie pour alimenter convenable les LEDs (~15 mA par canaux).

Programmation de fonction d'accès aux LEDs indépendamment et affichage. Première implémentation concluante.



Prochaine étape: contrôle des LEDs avec différents mode: clignotant, changement de couleur, contrôle via port série (qui sera relié au raspberry pi)

Vincent:

Inventaire du matériel reçus

Rédaction du rapport de séance

Guidage de Robin pour programmer la Getboard et envoie du bootloader sur atmeg328p, permettant de la programmer via liaison série directement sans passer par ISP.

Attributions des tâches pour les prochaines séances

Transmission d'énergie sans fil, premiers tests:

- Bobinages des 2 bobines: 50 cm de diamètre, 5 tours, 1.5mm², 0.03 mH et 0.4 ohme
- Montage sur breadboard du pont en H
- Simulation avec oscilloscope et GBF -> fonctionne
- Préparation de l'émetteur et du récepteur:

Etablir LC résonnant à 250KHz:

$$- f_r = \frac{1}{2\pi\sqrt{CL}}$$

- Fréquence souhaitée : 250 KHz

$$\sqrt{C} = \frac{1}{2\pi f_r \sqrt{L}} = \frac{1}{2\pi \times 250E3 \times \sqrt{0.03E-3}} = 1.16E-4$$
$$C = 1.35E-8F = 13.5nF$$

Donc bobine en série avec condensateur de 13.5nF pour avoir des bobine résonnante à 250 KHz

Pour des raisons de possible variation de fréquence de résonance dû aux perturbations provoquent par le drone en aluminium qui agira comme un noyau de transformateur ou l'échauffement des câble, il sera peut être nécessaire d'ajouter un feedback de la réel fréquence de résonance du transmetteur à l'oscillateur. Cela devrait relativement être facile à mettre en place, puisque l'oscillateur est basé sur une boucle à verrouillage de phase, donc possibilité d'extraction de la porteuse du feedback qui doit être théoriquement au bon fr via le même circuit (CD4046), et de la réinjecté dans le circuit.

<Ajouter ici photo montage>

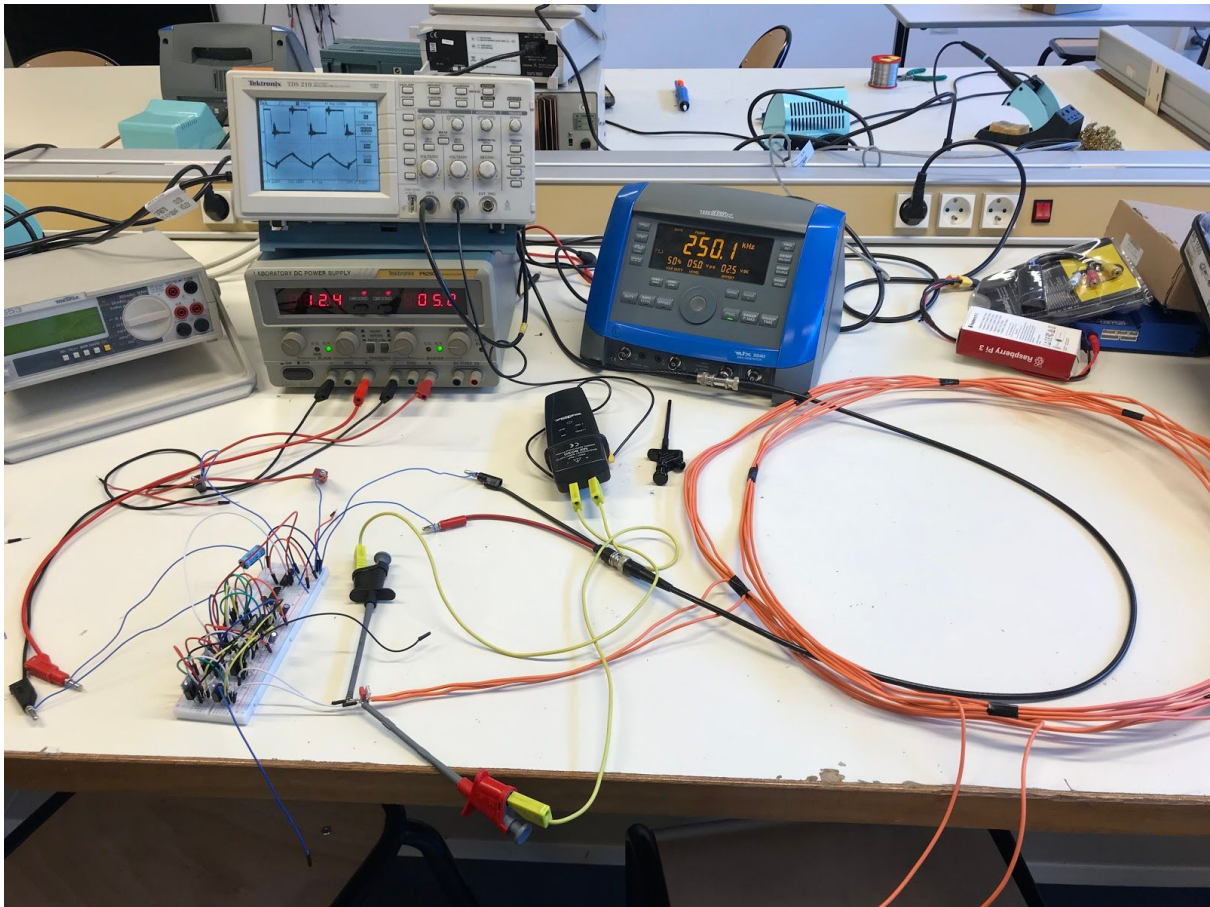
Prochaine étape: montage du circuit récepteur (redressage HF + filtrage HF), puis teste du rendement en fonction de la distance.

Séance du 15 / 02 / 2019: LEDs, énergie sans fils, suivie du Soleil

Rédacteur: Robin, Erwann, Vincent

Vincent:

Test transmission d'énergie sans fil:

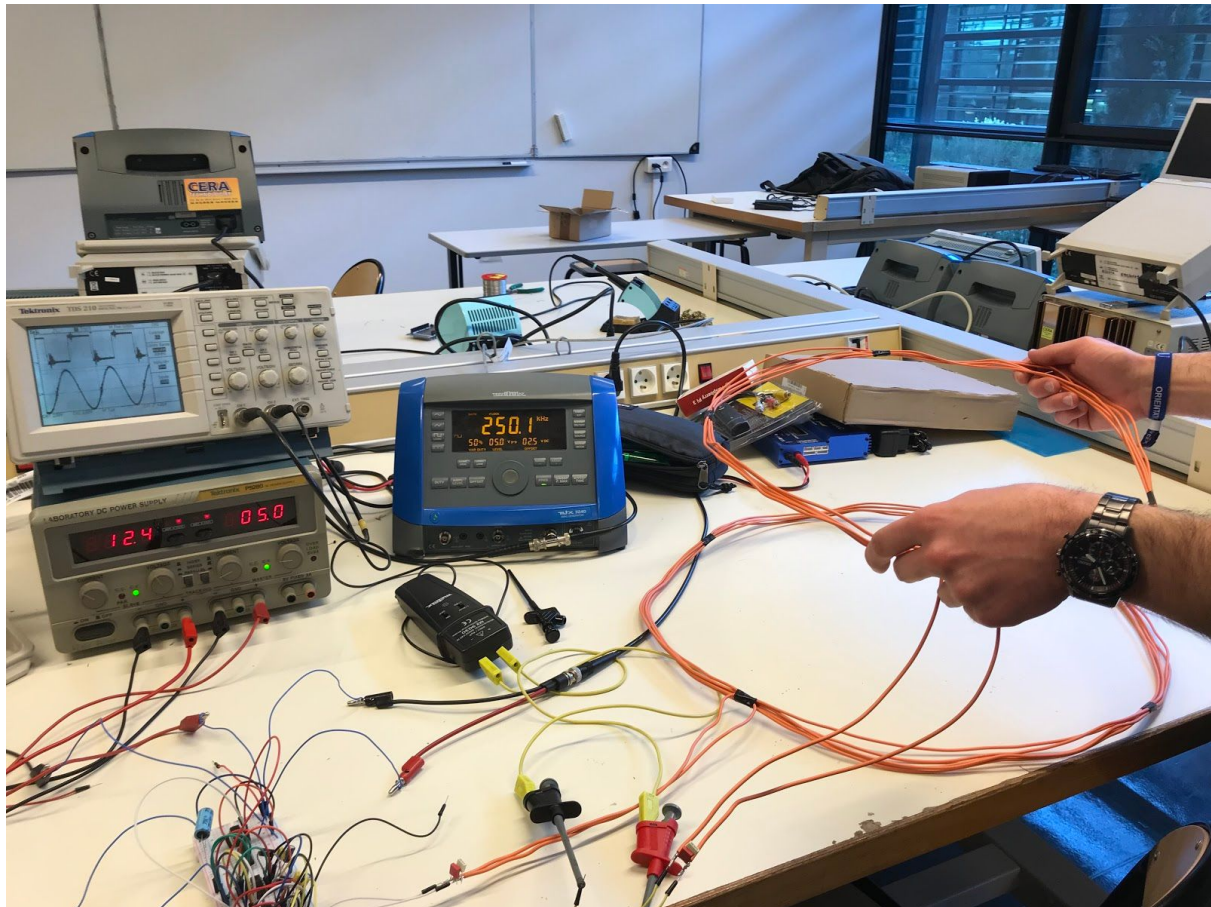


Soudure des condensateurs approprié précédemment calculer : 13.3nF

Finalement apparition d'harmonique dans le circuit émetteur: fréquence de résonance incorrecte.

Donc augmentation de la capacité série à 33nF pour s'approcher de la fréquence de résonance réelle.

Transmission d'énergie réussie à 250kHz .



Montage de l'oscillateur à verrouillage de phase: 4046

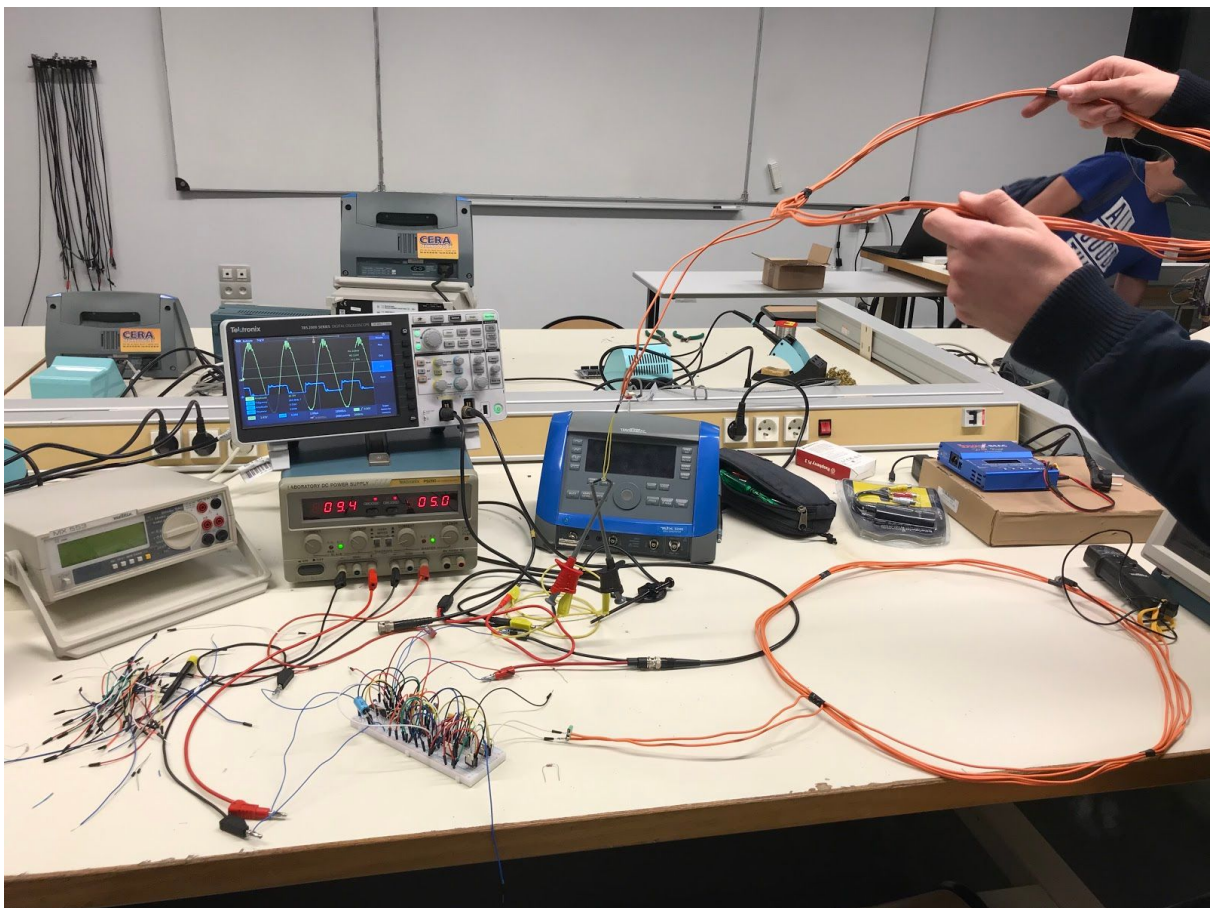
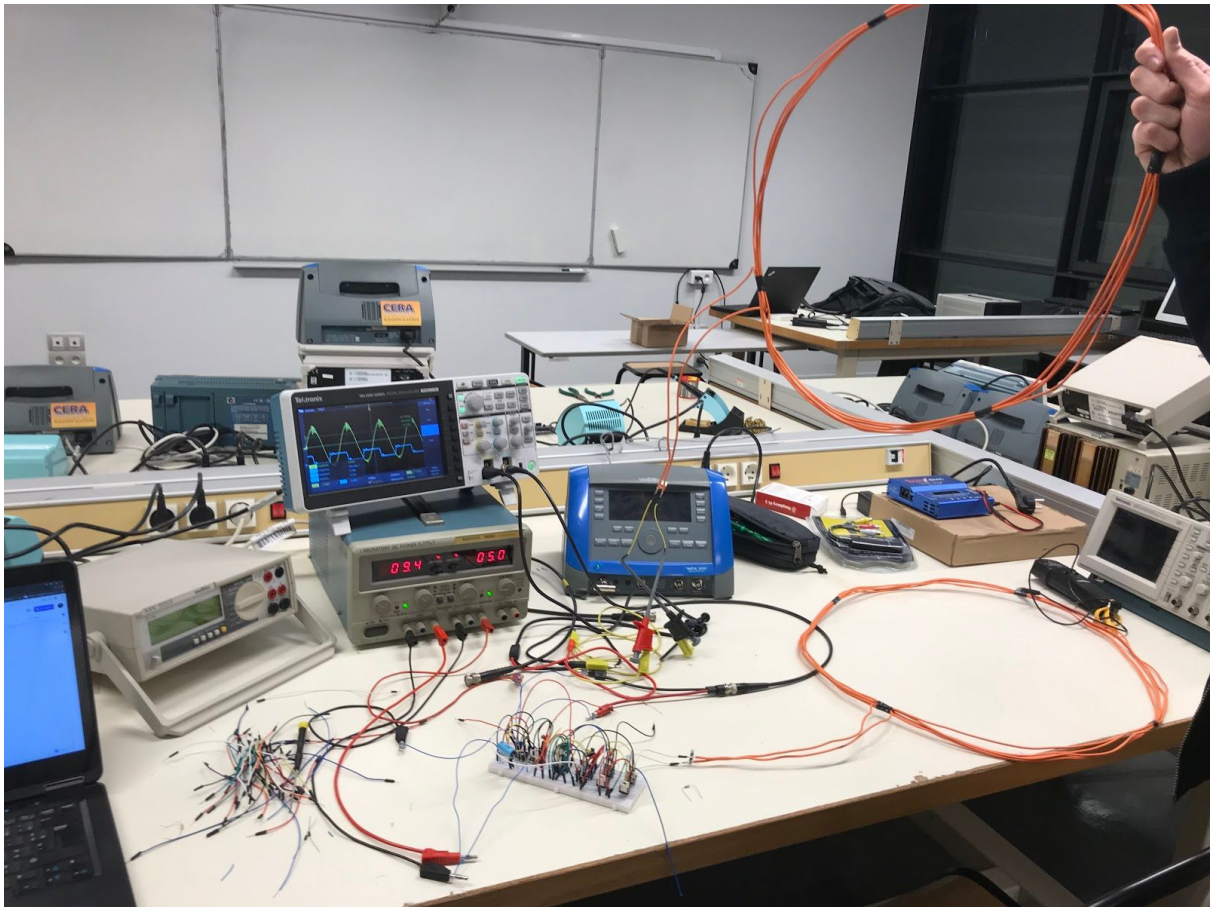
Calibration de la sortie entre 230KHz et 350KHz: $R1 = 10\text{ K}\Omega$, $R2 = 20\text{ K}\Omega$
 Ajustement de la fréquence par potentiomètre en attendant d'avoir une feedback de la porteuse transmise (fréquence de résonance du récepteur). Le feedback sera réalisé par une bobine de paramètre semblable à la réceptrice qui sera bobiné avec la bobine émettrice (même tuyau).

Courant moyen consommé de 500 mA en 12 V, puissance "émise" : 6 W.

Augmentation de la puissance => augmentation de la tension d'alimentation => driver côté gauche grillé (2x fois toujours le même côté, second fois dû à une fréquence de résonance réajusté).

Non présence de diode de roue libre sur les MOSFETs, ce qui implique peut être des surtension inverse dans les drivers. Piste à étudier + ajout de diode de roue libre (fast).

A cette puissance les MOSFETs même sans radiateur ne chauffe pas du tout.
 Composant dimensionné pour 100 W d'émission.



Piste amélioration: inspecter la voie du ZVS pour encore limiter les pertes coté hachage. ZVS forcé difficilement implémentable via cette topologie de contrôle, mais ZVS "libre" possible en profitant de la réponse de la bobine sur le système: sinusoïde -> passage à 0 -> ajuster fréquence pour avoir le switch à l'instant du passage à 0 de la réponse du circuit LC.

Prochaine étape: Trouver la raison pour laquelle le driver de MOSFET gauche grille si la puissance augmente (fort possible: diodes roue libre non présente), réaliser bobine de feedback. Si résultat concluant: routage du PCB sur protoboard.

Polytech aboard

Robin:

Programme de clignotement des LEDs du drones. Gestion de la durée de clignotement et de la couleurs de LEDS. Nous avons utilisés la bibliothèque `time_scheduler` pour pouvoir gérer le clignotement sans utiliser des `delay()`.

Le drone a 8 LEDs:

- 2 LEDs à l'avant de couleur vertes, fréquences de clignotement 2Hz
- 2 LEDs sur les cotés de couleurs bleues, fréquences de clignotement 4Hz
- 2 LEDS à l'arrière de couleur rouges, pas de clignotement toujours allumées

Test du programme sur carte nano Arduino. Consultation des datasheet des différents composants utilisés.

Test à l'oscilloscope du signal de donnée transmis à travers les composants et vérification visuelle à l'aide de LEDs sur la breadboard.

Problème avec les shifts registers: valeur envoyez non présente sur la sortie parallèle, pourtant fonctionne correctement en mode manuel.

Prochaine étape: Si le shift register fonctionne passage au routage et l'implémentation sur le drone.

Erwann:

Ecriture d'un programme Arduino permettant l'orientation des volets et des cellules de la base de rechargement.

L'orientation des volets est contrôlée par un moteur à charbon. L'orientation des cellules est contrôlée par des servo-moteurs.

Asservissement des volets et des cellules par PID: développement d'un programme sur arduino qui asservie en position via 3 PIDs les 3 volets. Le contrôle des moteurs

à charbon est fait via des pont en H: L298, le retour d'information sur l'orientation sera fait par des potentiomètre relié directement à l'arbre du moteur.

Fin de la rédaction du rapport de répartition des tâches pour le projet.

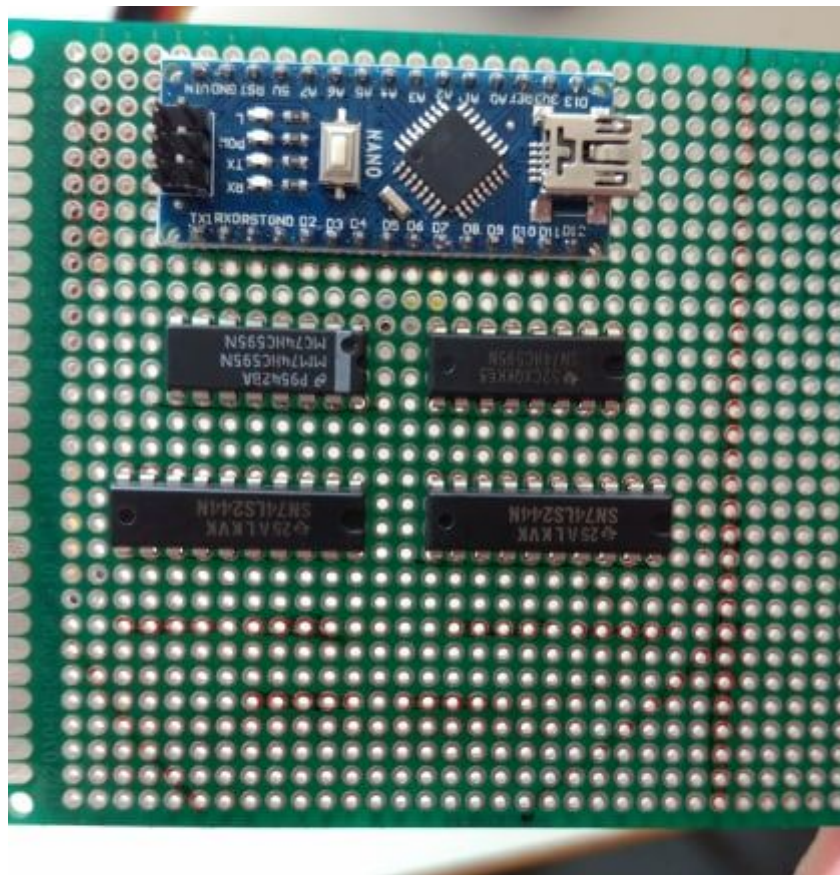
Séance du 1 / 03 / 2019: Prototypage, asservissement et tests !

Robin:

Implémentation du contrôleur LEDs sur la protoboard

Trouver la zone où placer BCB sur le drone, choix de la forme du PCB pour optimiser l'espace occupé et les branchements.

Positionnement des composants sur BCB avec support adapté pour chaque composants. Support 20 broches pour les buffers SN74HC244 et support 16 broches pour les registres à décalage 74HC595.



Support 28 broches pour la carte Arduino Nano.

Choix du positionnement des 16 sorties du contrôleur LEDs. 4 sorties pour chaque LEDs à contrôler représentant la couleur rouge, verte et bleue + GND. Les sorties seront des pins 2.54mm mâles.

Branchement des composants

Branchements des 2 registres à décalage avec la carte Arduino Nano et branchement entre 2 les registres à décalages et les 2 buffers.

Les supports nous permettrons de pouvoir changer les composants en cas de dysfonctionnement au lieu de souder directement les composants sur le BCB. Il faut un support particulier pour la carte Arduino Nano qui a des pins 2.54mm. Pour permettre au pins d'être insérer facilement dans le support nous avons tailler les pins en pointes.

Soudure des composants au BCB

Réalisation de la soudure des support sur le BCB puis ensuite réalisation de la soudure des fils qui serviront à l'alimentation des composants via la pin 5V de la carte Arduino Nano ainsi que la pin GND de celle-ci.

La soudures des composants a été réalisée de manières à minimiser la longueur des fils. Câble management très important vu le nombre important de connexion à réaliser.

Respecter le code couleur des gaines des fils utilisés. Les gaines bleues seront utilisés pour la masse et les gaines rouges pour le 5V et toutes les autres connexions.

Prochaine fois:

- Branchement LED via pin femelle 2.54mm x4 (GND, R, G, B)

Cable management des fils émaillés:

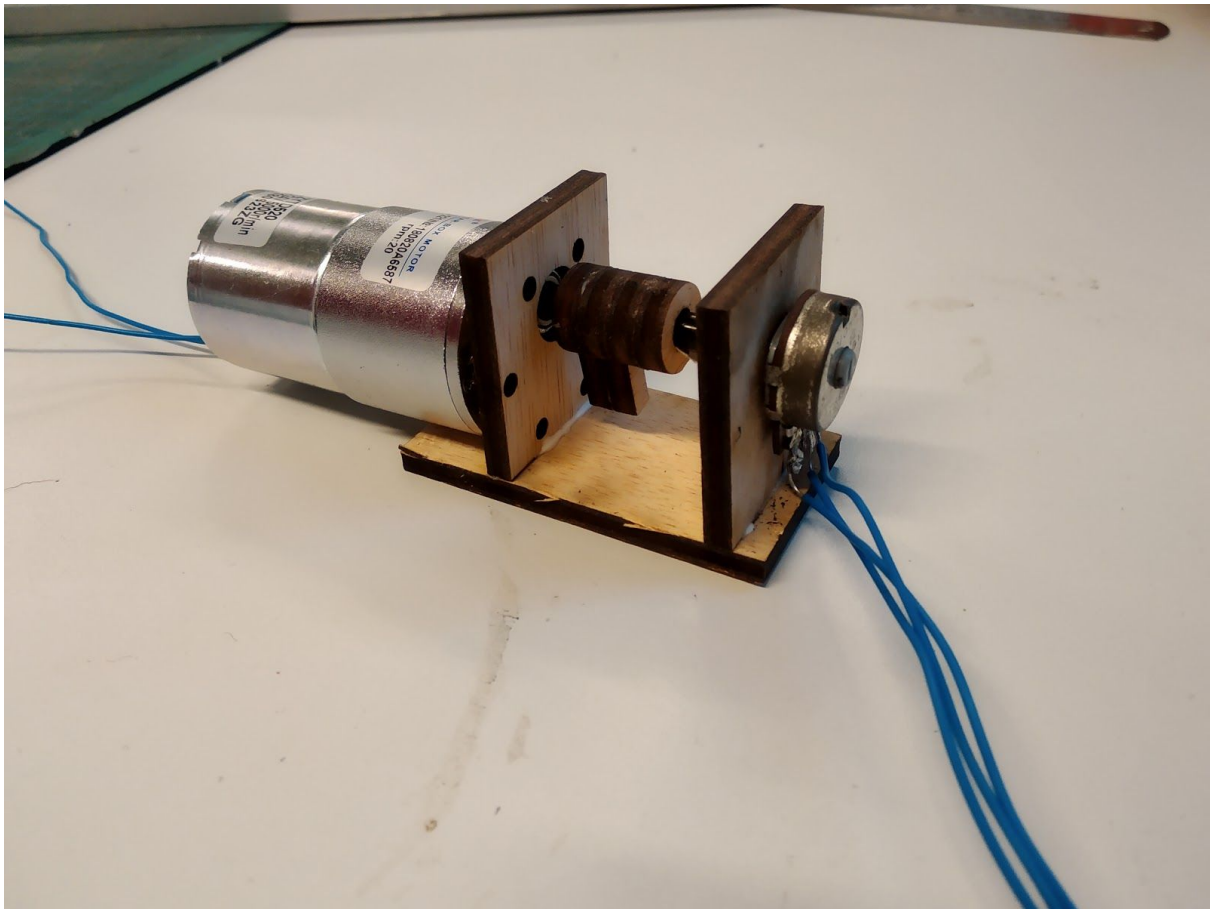
- Souder connecteur mal 2.54mm x4 pour chaque LED RGB
- **Respecter code couleur avec gaine thermo**
- Trouver meilleur passage pour les fils

Tester programme sur drone

Si tout fonctionne: Vincent ajoutera MODBUS + explication registre à Robin
En fonction programmation type affichage sélectionnable par commande reçu (état de registre)

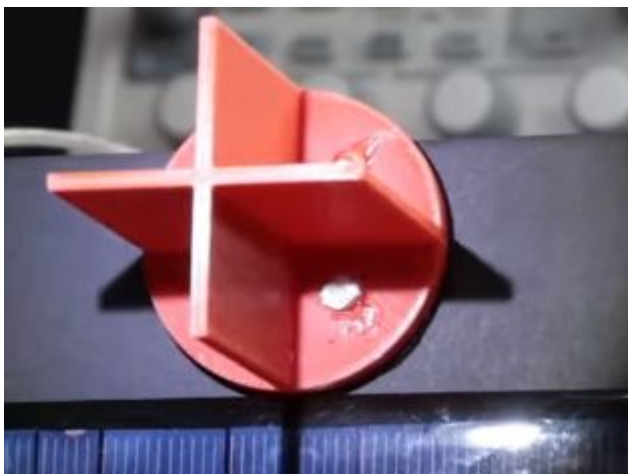
Erwann:

Avec Vincent fabrication d'un petit support et moyeux (Fab lab) pour l'asservissement en position d'un moteur avec le programme Arduino.



La colle n'étant pas sèche, nous n'avons pas pu tester cette séance.

Compréhension du fonctionnement du **tracker solaire** : Les panneaux solaires doivent être orientés vers le soleil. Utilisation de 4 photodiodes et du principe des ombres.



Les panneaux solaire sont bien orientés lorsqu'il y a n'y a plus d'ombre (Plus les photodiodes reçoivent de lumière plus leur résistance est faible).

Les photo résistances étant identiques, il nous suffit de comparer leur valeur de résistance et de modifier l'orientation des panneaux solaires en fonction.
Ecriture du code arduino.

Prochaine fois:

Maquette : **monter LDR** sur support, tester asservissement en position des moteurs et tester le programme le programme tracker solaire.

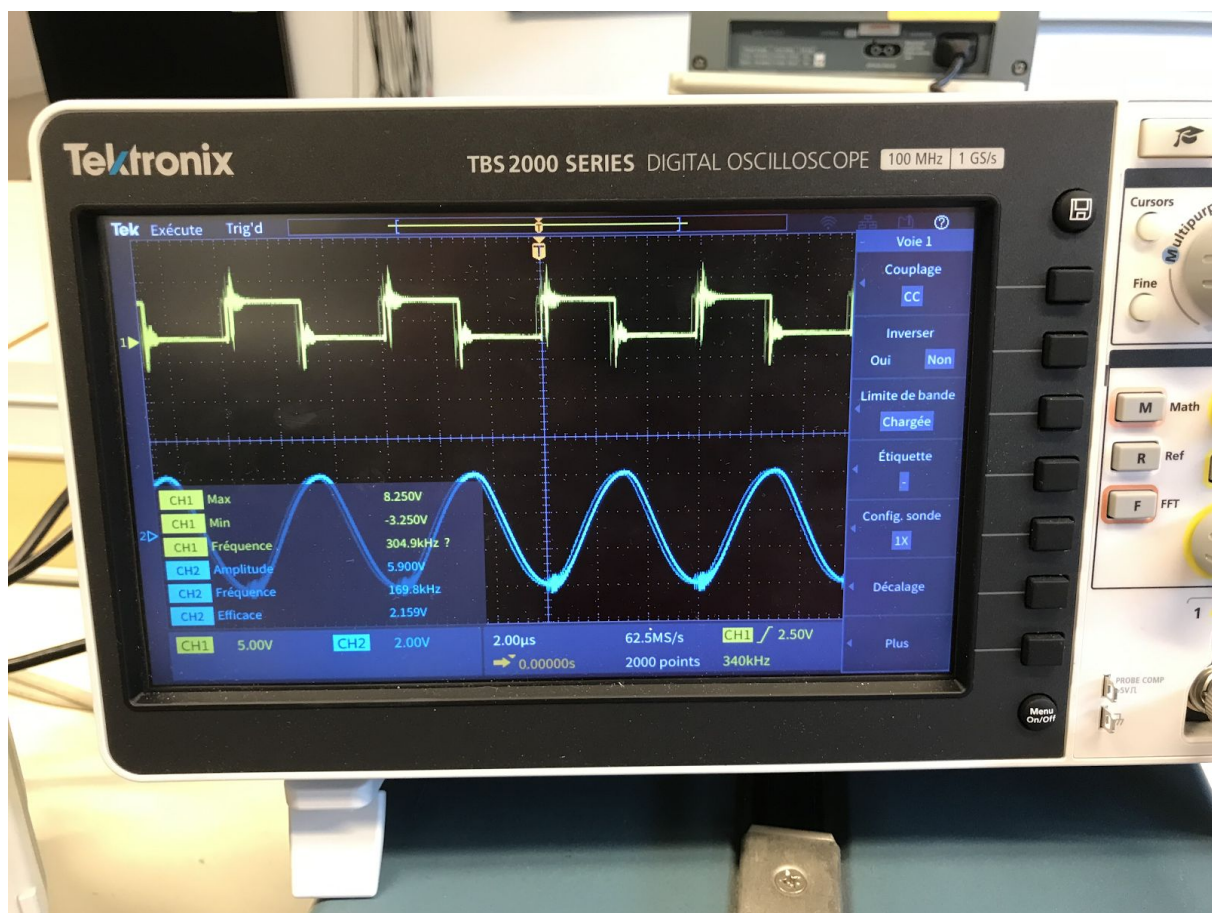
Vincent:

Conception du redresseur HF avec une double diode ultrafast 30A.

Teste du montage:

Channel 1: Sortie de la PLL

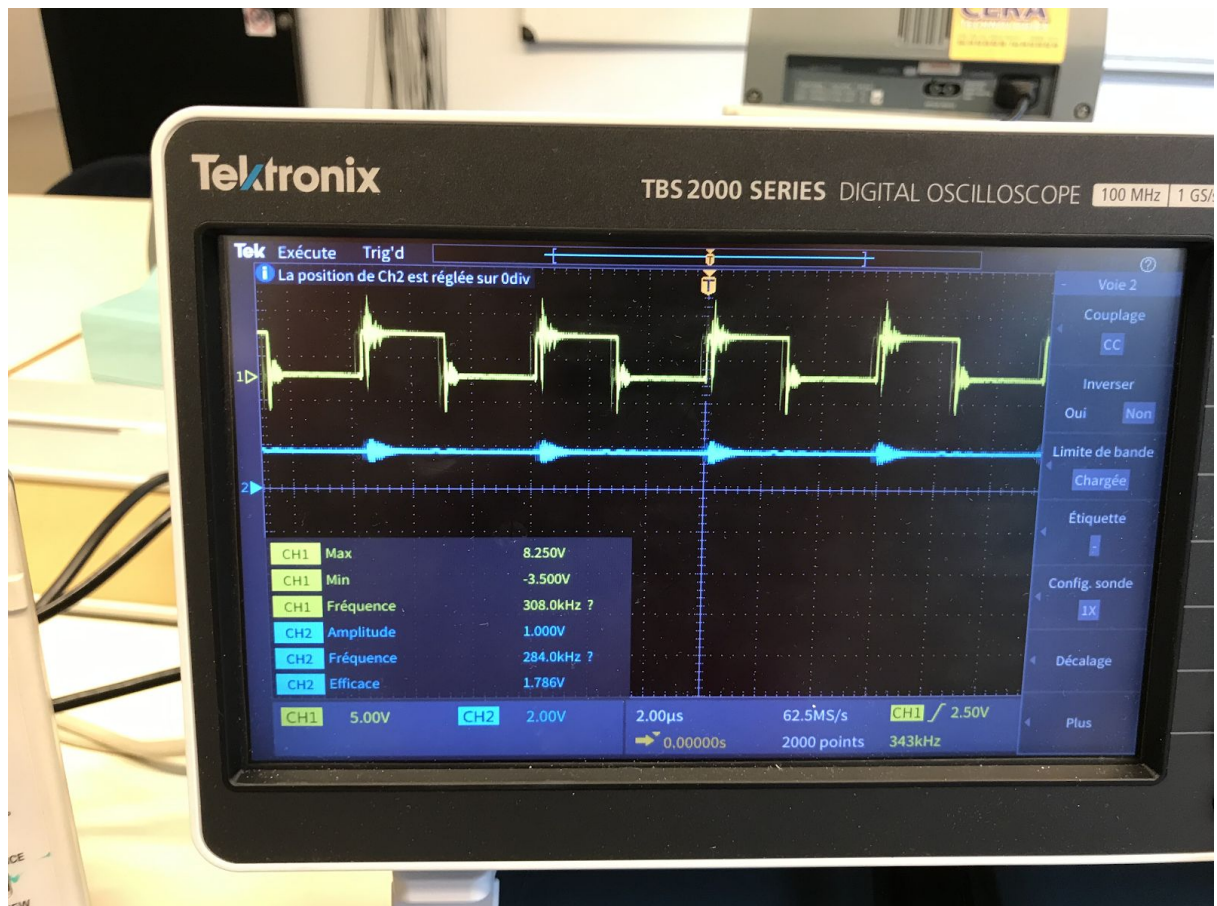
Channel 2: Borne bobine réceptrice (atténuateur de 20dB)



Réception 21.5V efficace

Channel 1: Sortie de la PLL

Channel 2: Sortie redresseur récepteur (atténuateur de 20dB)

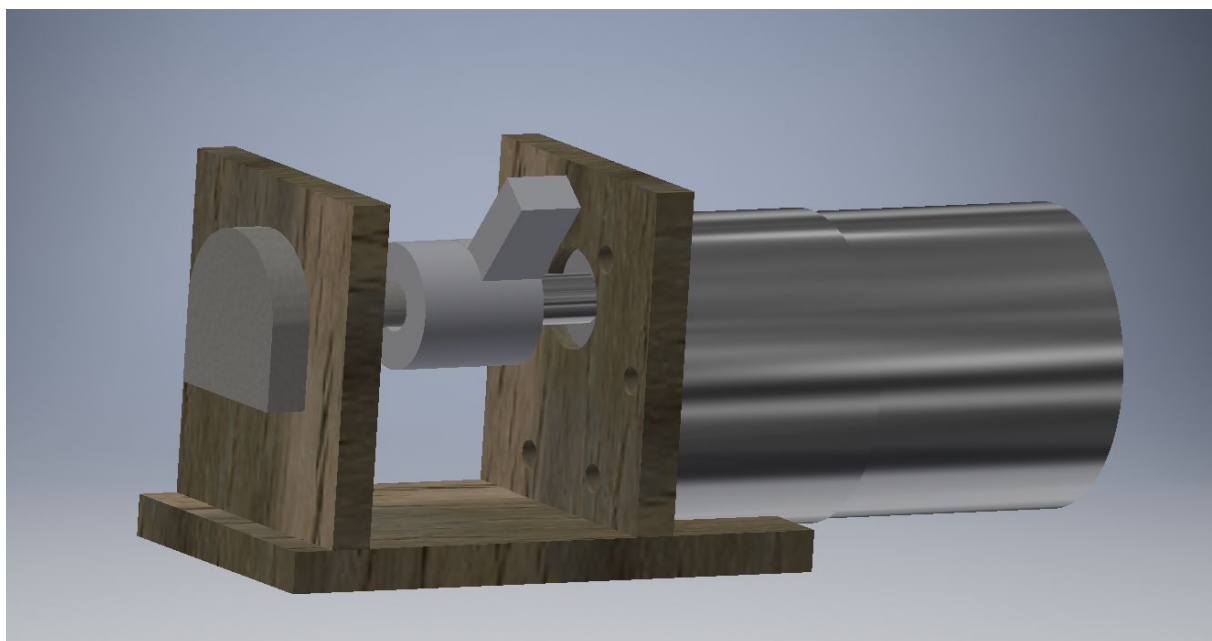


Réception 17.6 V dc.

Résidu d'HF, une inductance série pourrait limiter leur passage dans le BMS.

Solution à étudier.

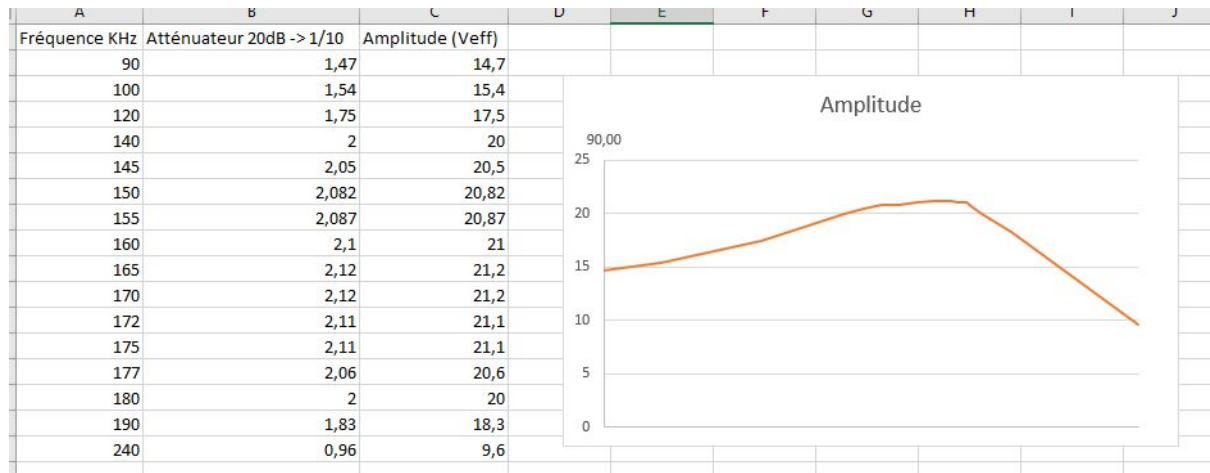
Réalisation avec Erwann du banc d'essai de l'asservissement en position. Désigne sur Inventor:



Exportation des face en fichier DXF, passage par LibreCad pour convertir en DXF r13 compatible avec la découpeuse LASER.

Découpe au FabLab support dans plaque de contreplaqué de 5mm d'épaisseur, puis assemblage à la colle à bois. La colle ayant besoin de plus d'un jour se séchage nous n'avons pas pu tester l'asservissement.

Recherche du gain maximum:



Gain maximum pour 170KHz.

Mode veille des IR2110: pin 11 (off: 5V, on: 0V)

Début de l'implémentation du circuit sur PCB: Placement des MOSFET, support des circuit intégrés et des condensateurs.

Utilisation de STP40NF10: 100V, 50A, Rdon = 0.025 ohmes

Prochaine fois:

Finir implémentation sur protoboard.

Tester asservissement en fréquence du transmetteur: bobine de feedback !

Checker rendement grâce à une résistance électronique

Comprendre pourquoi IR2110 très sensible au surtension dans ce montage, vérifier pics de tension induit

Penser à ajouter diode de roue libre!!

Mode veille sur 4046

Tester sonde courant ACS712 avec arduino

Commencer code de contrôle de l'émetteur: MODBUS