

# Rapport de séance: Projet industriel 19

## Drone de cartographie

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

Rédacteur: Vincent

Erwann:

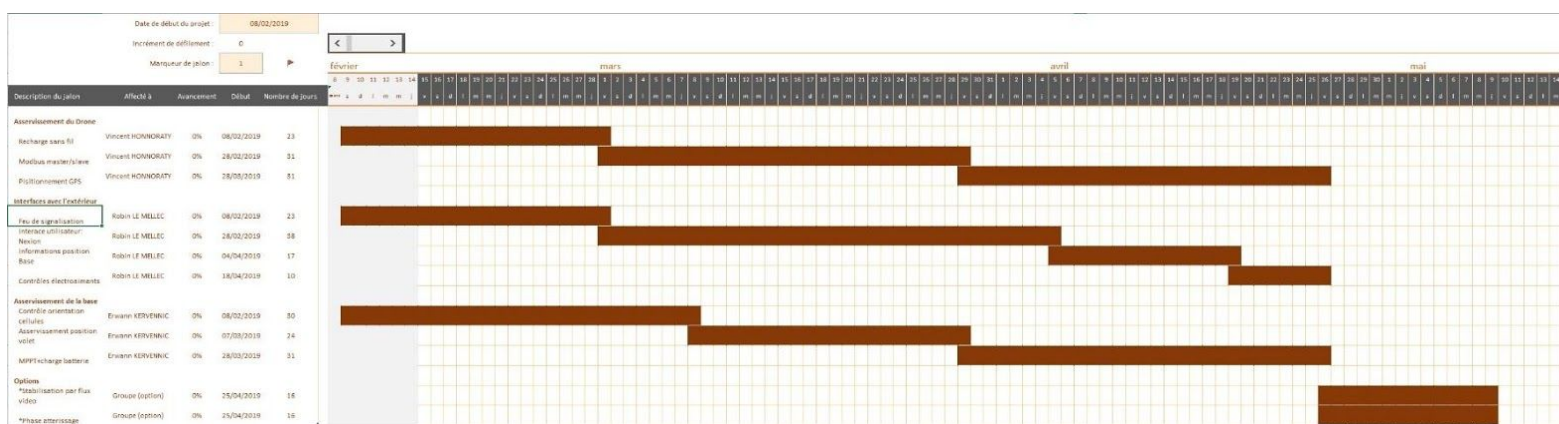
Définition mathématique de l'orientation des cellules solaires:

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.

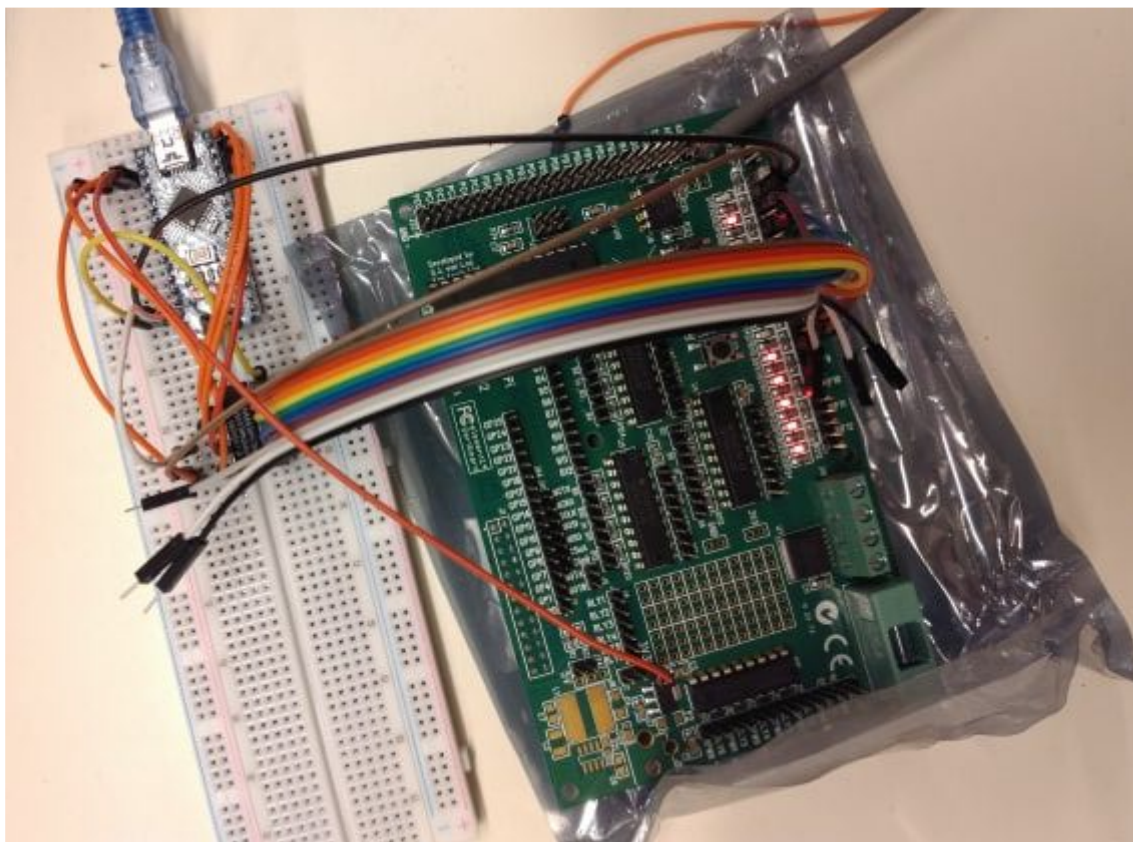
Programmation du atmega328p via FTDI232, pour utilisation de l'IDE arduino.

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, 1.5mm<sup>2</sup>, 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>

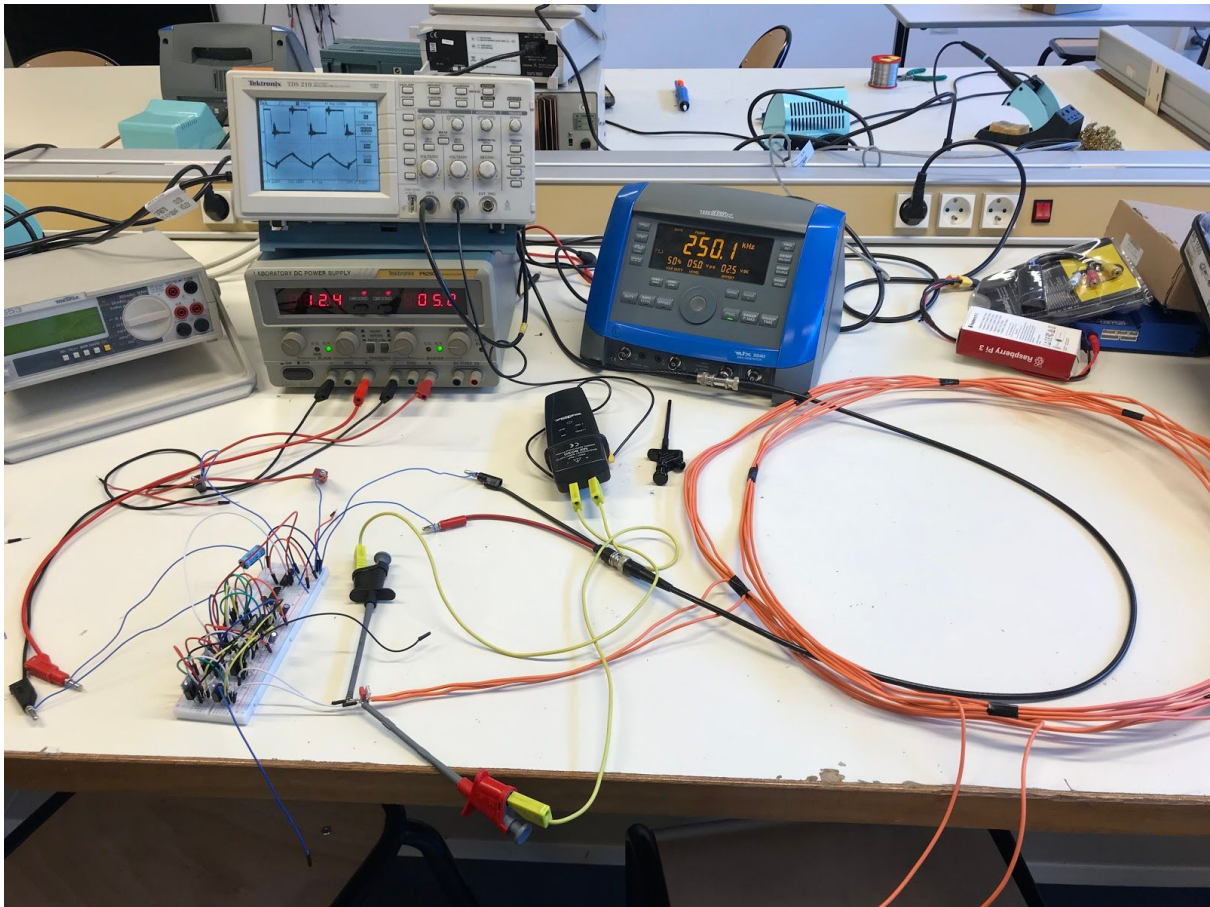
Prochaine étape: montage du circuit récepteur (redressage HF + filtrage HF), puis teste du rendement en fonction de la distance.

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

Rédacteur: Robin, Erwann, Vincent

Vincent:

Test transmission d'énergie sans fil:



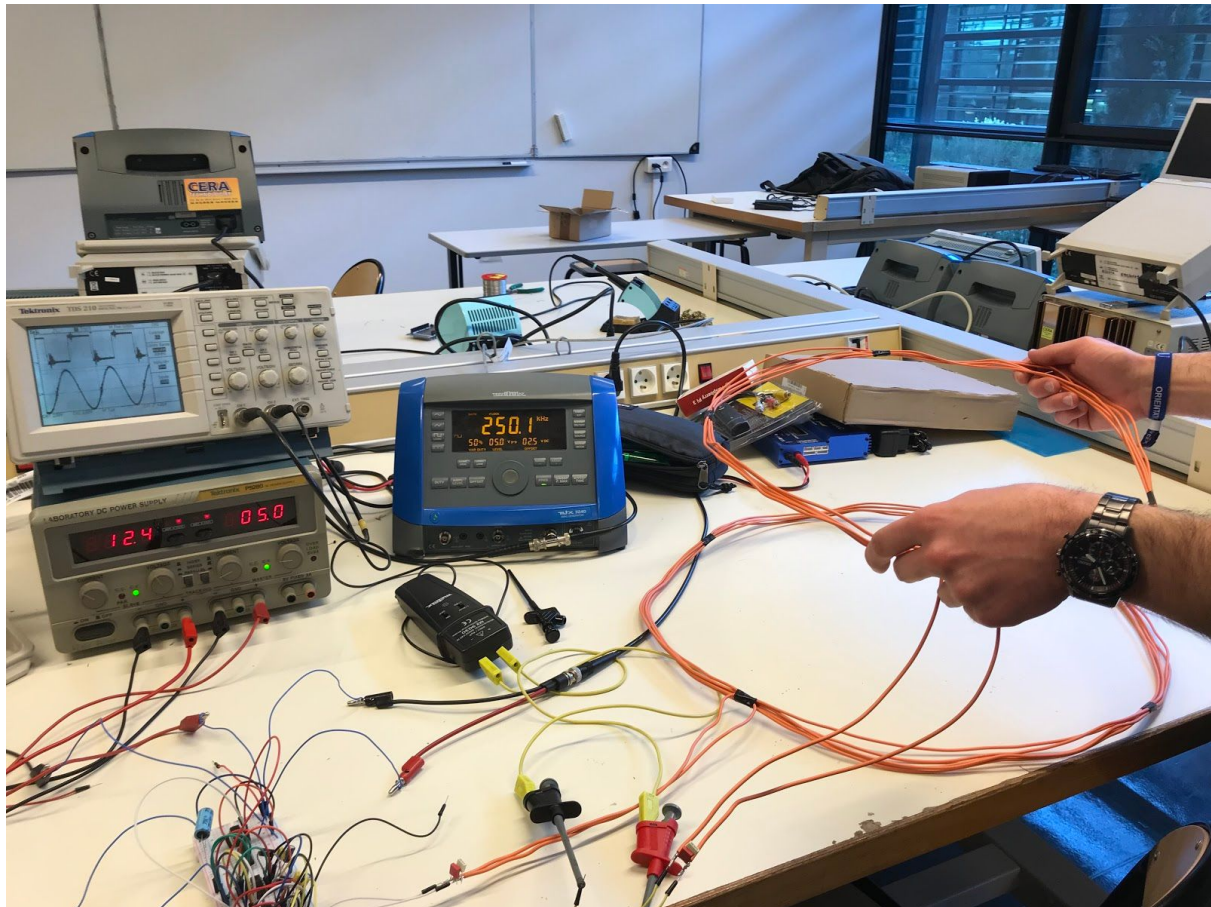
Soudure des condensateurs approprié précédemment calculer :  $13.3\text{nF}$

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

Donc augmentation de la capacité série à  $33\text{nF}$  pour s'approcher de la fréquence de résonance réelle.

Transmission d'énergie réussie à  $250\text{kHz}$ .





Montage de l'oscillateur à verrouillage de phase: 4046

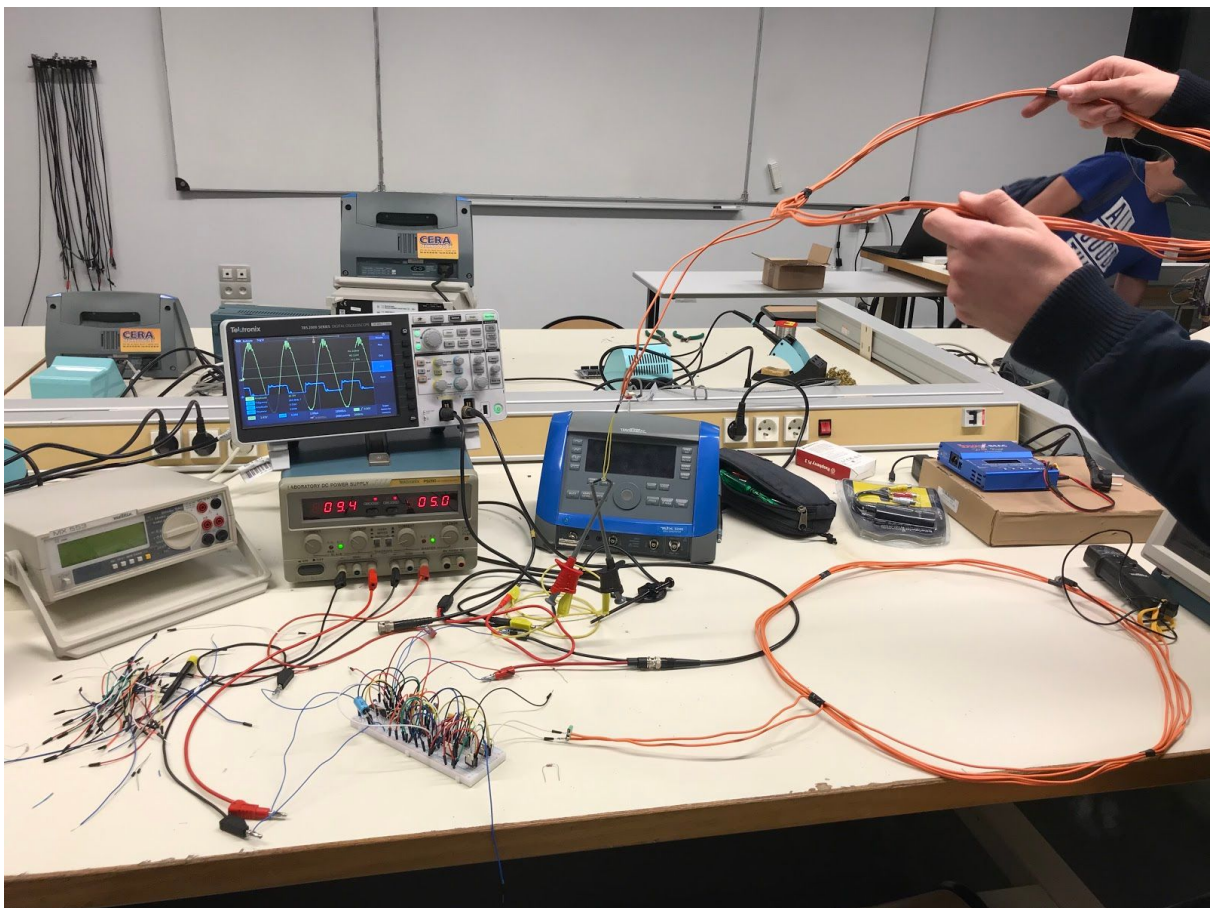
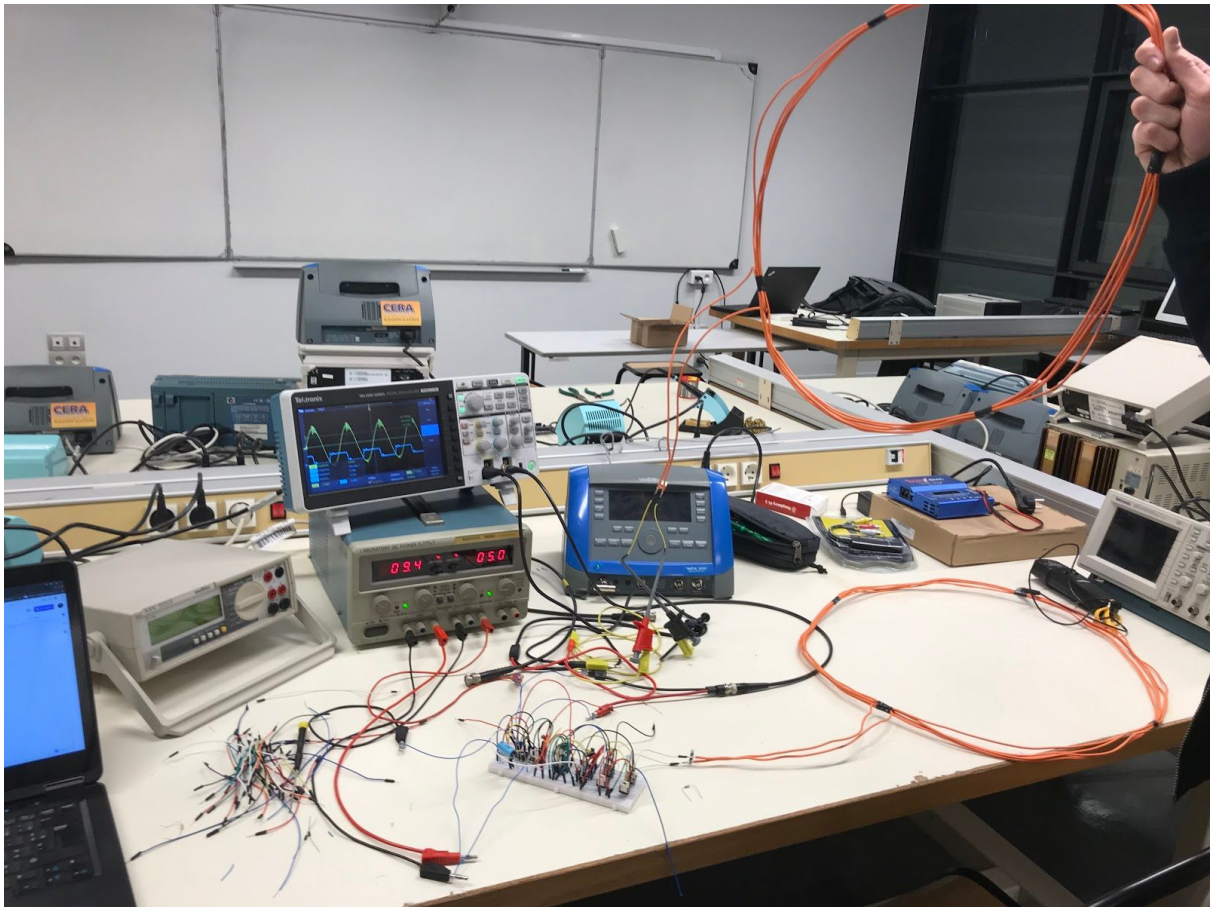
Calibration de la sortie entre 230KHz et 350KHz:  $R_1 = 10\text{ K}\Omega$ ,  $R_2 = 20\text{ K}\Omega$   
 Ajustement de la fréquence par potentiomètre en attendant d'avoir une feedback de la porteuse transmise (fréquence de résonance du récepteur). Le feedback sera réalisé par une bobine de paramètre 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.

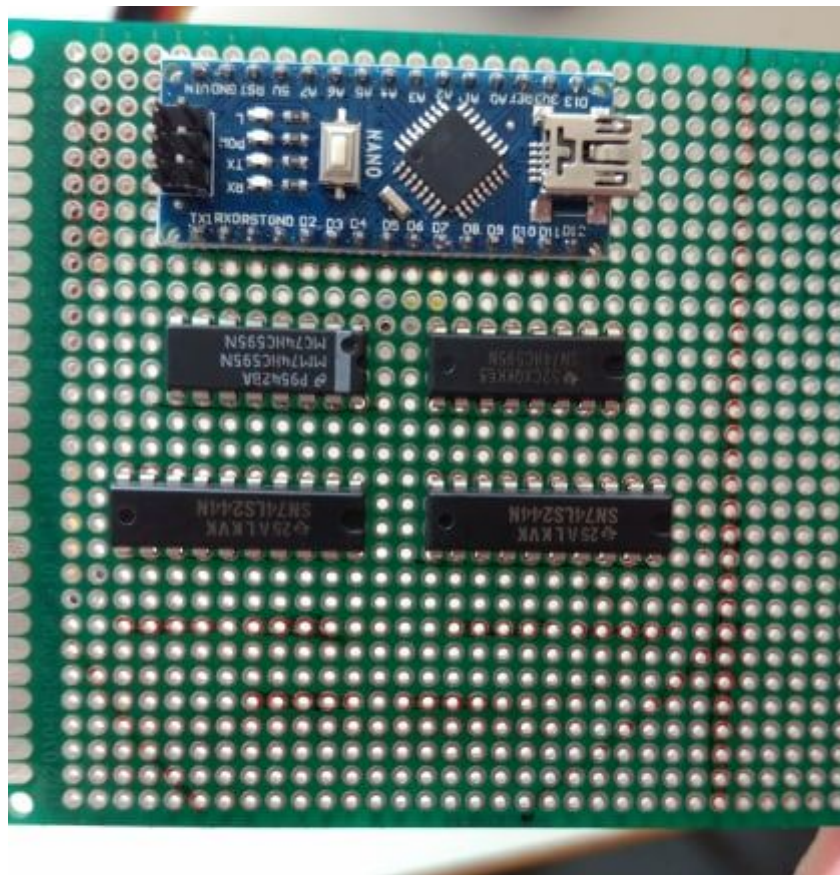
Séance du 1 / 03 / 2019: Prototypage, asservissement et tests !

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 permettront 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 insérés facilement dans le support nous avons taillé les pins en pointes.

### **Soudure des composants au BCB**

Réalisation de la soudure des supports 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 soudure des composants a été réalisée de manière à minimiser la longueur des fils. Câble management très important vu le nombre important de connexions à réaliser.

Respecter le code couleur des gaines des fils utilisés. Les gaines bleues seront utilisées 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:

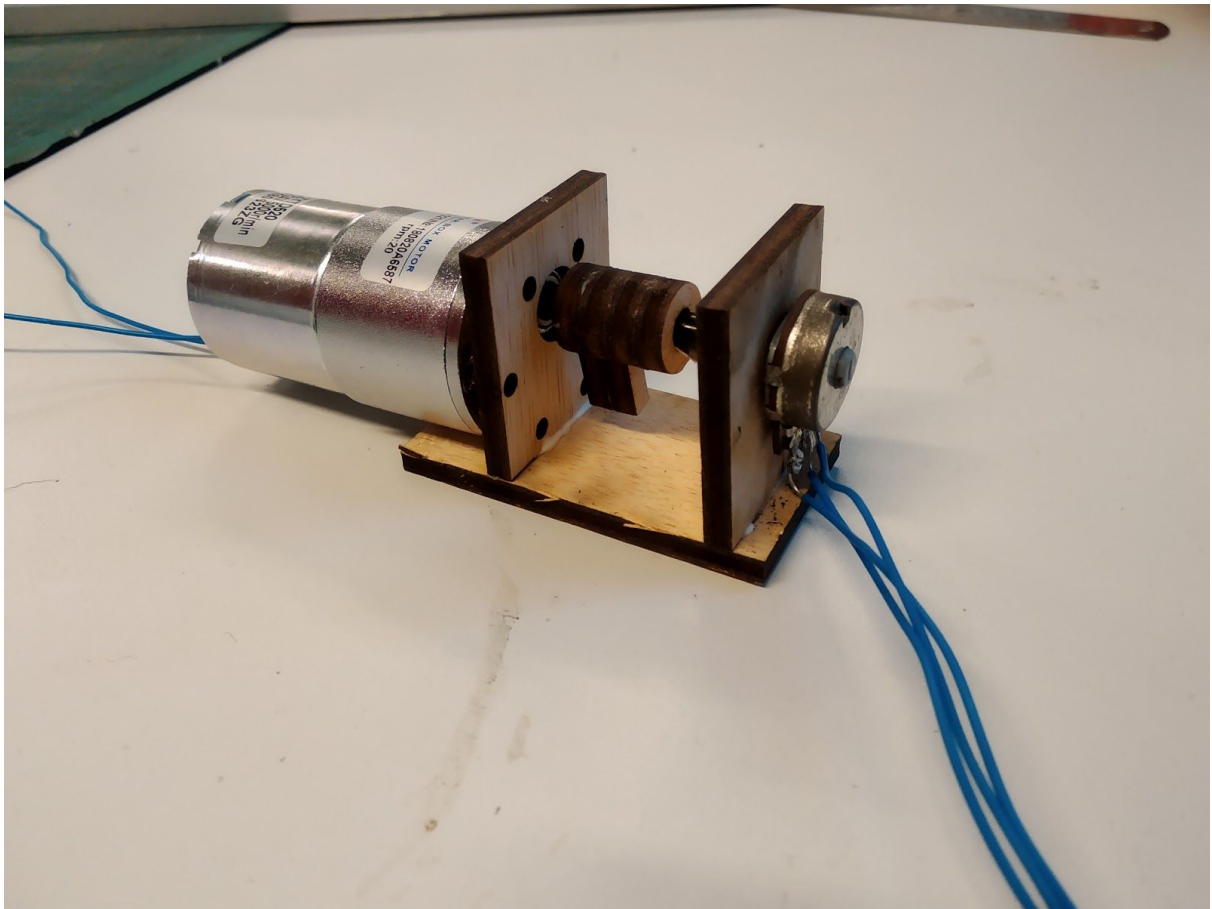
- Souder connecteur mal 2.54mm x4 pour chaque LED RGB
- **Respecter code couleur avec gaine thermo**
- Trouver meilleur passage pour les fils

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)

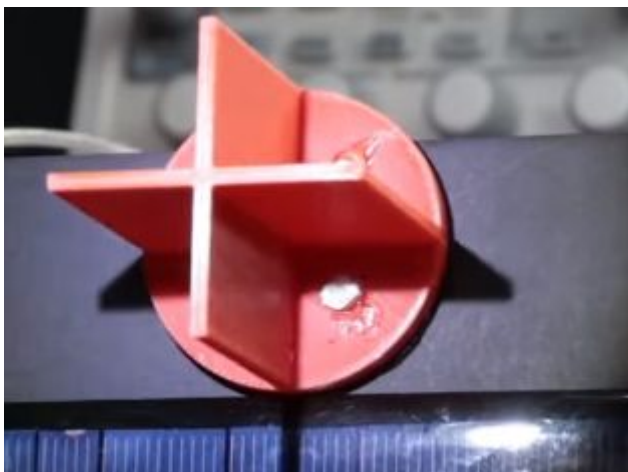
Erwann:

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.

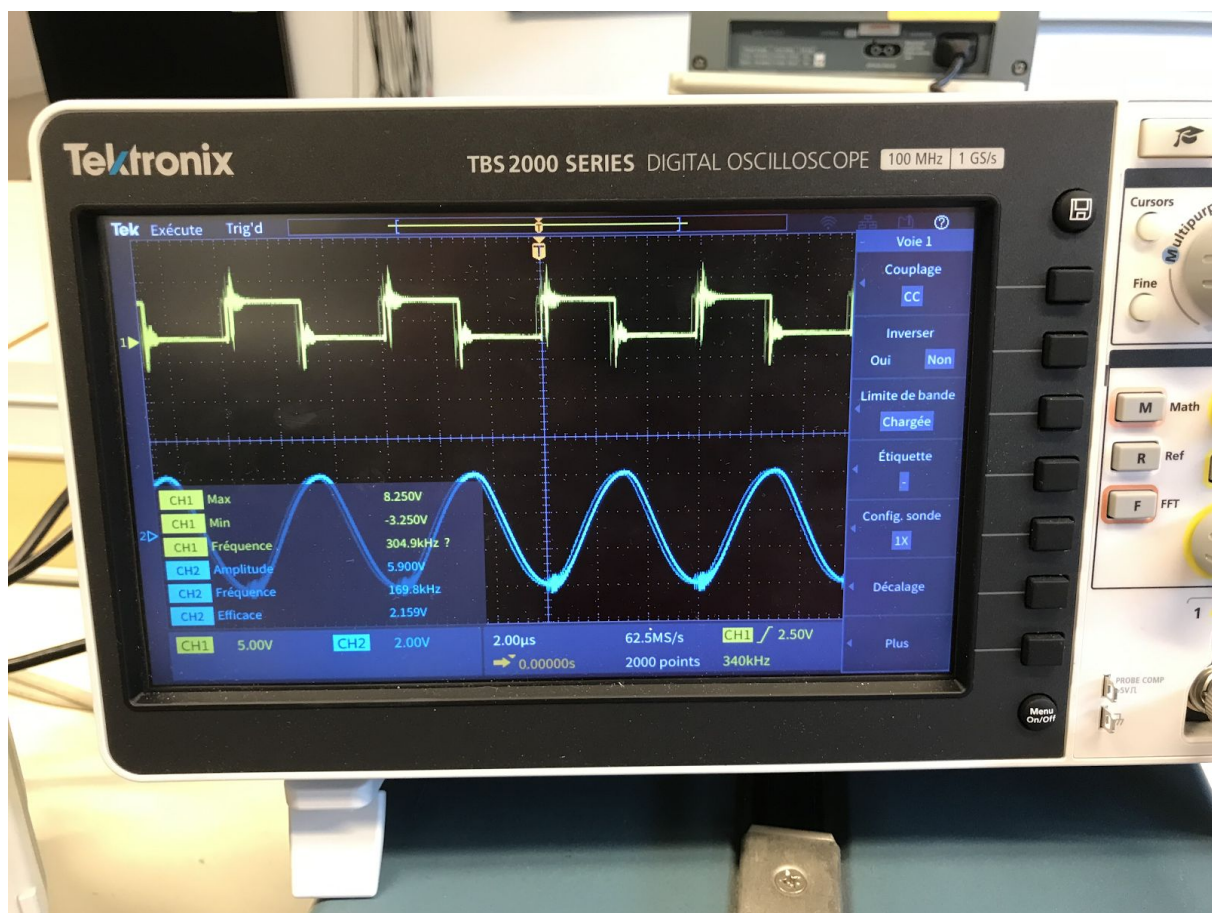
Vincent:

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)

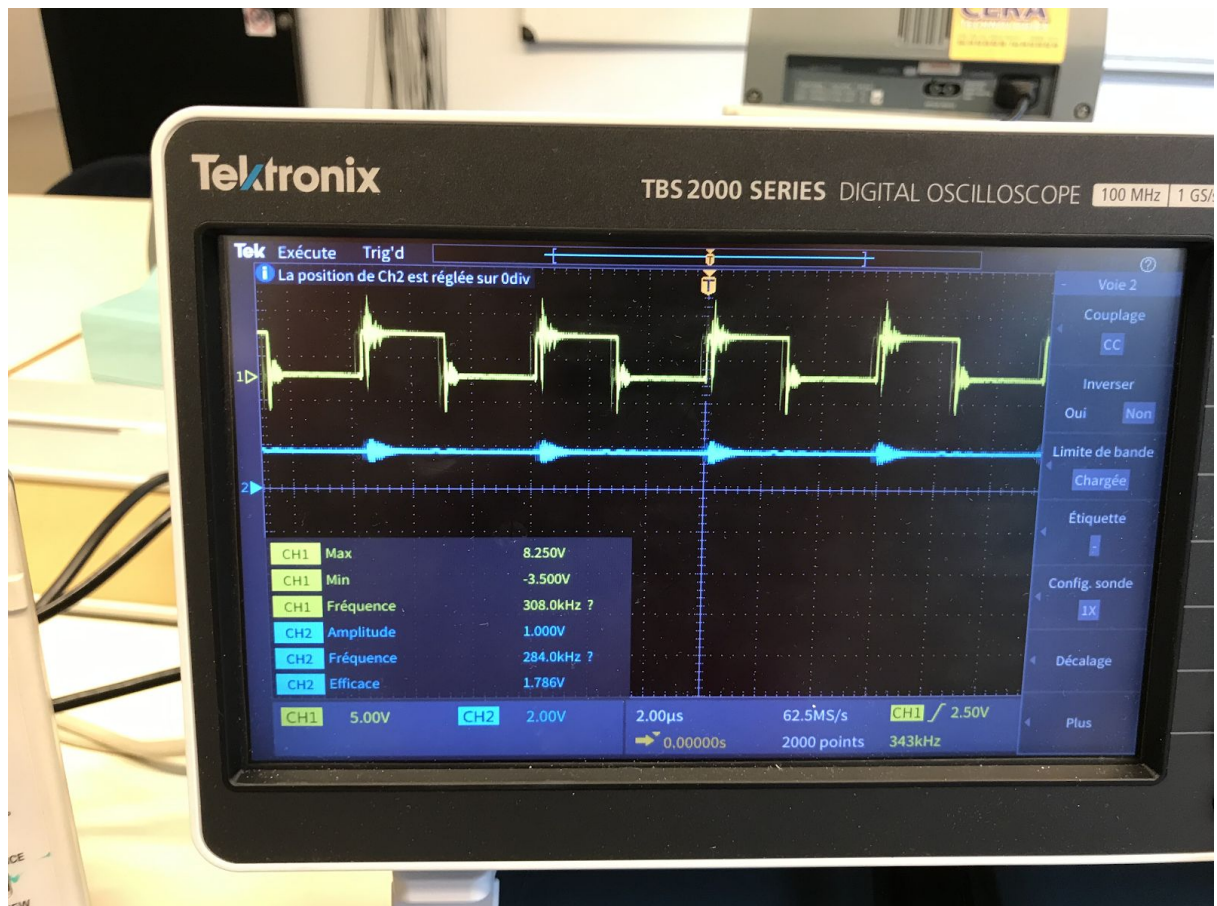


Réception 21.5V efficace

Channel 1: Sortie de la PLL

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



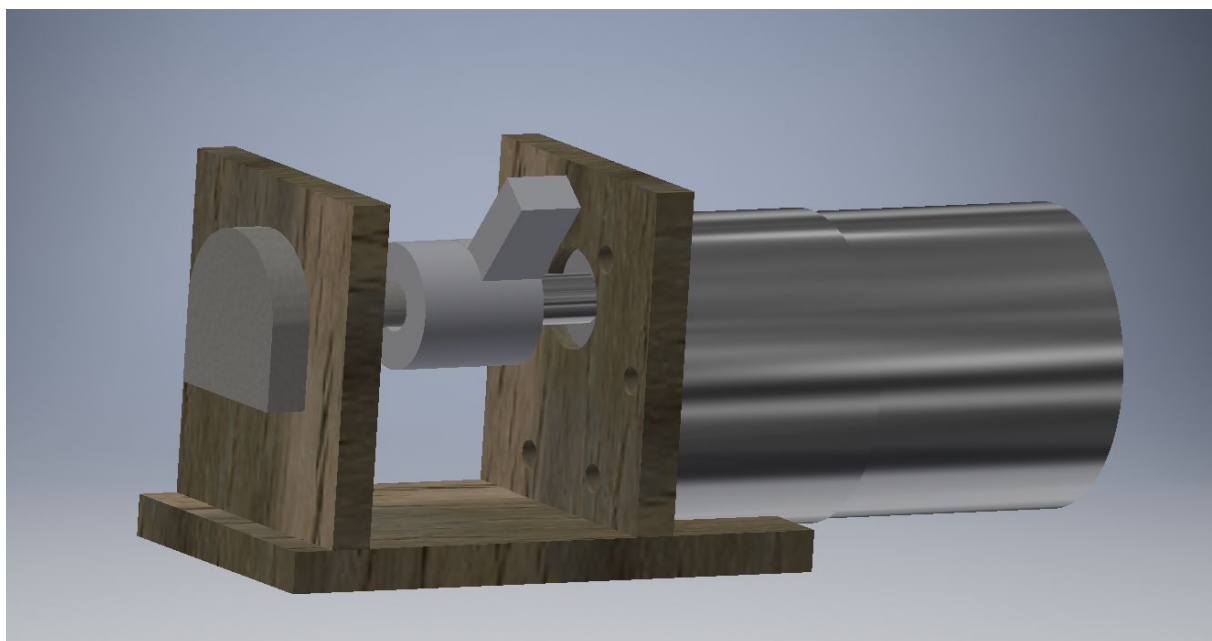


Réception 17.6 V dc.

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

Solution à étudier.

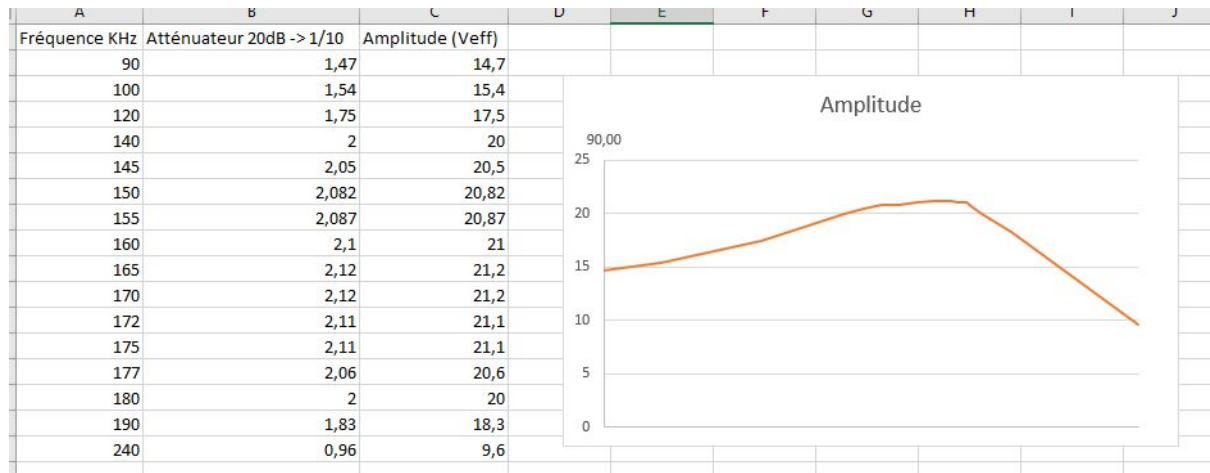
Réalisation avec Erwann du banc d'essai de l'asservissement en position. Désigne sur Inventor:



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 (consomme 10 mA en marche)

Tester sonde courant ACS712 avec arduino

Commencer code de contrôle de l'émetteur: MODBUS

## Séance du 8 / 03 / 2019: Test des premiers prototypes

Vincent:

Mise au propre du schéma électronique partie puissance et générateur d'horloge

<Photo schéma>

Vérification de l'étage de découplage => amélioration en ajoutant des couples condensateur électrochimique et film:

- Gros étage de découplage alimentation pont en H <ajouter valeur capacité totale>
- Découplage efficace sur les drivers de MOSFET (partie 12V)
- Découplage alimentation 4046
- Petit découplage 100nF sur inverseuse 7404

Mise en place de diode de roue libre <noter référence> en parallèle des MOSFET

Routage partie puissance réalisé avec fil étamé puis souder sur pastille pour former des pistes de puissance (puissance + limitation rayonnement électromagnétique). Méthode qui demande beaucoup de temps pour réaliser le routage, mais est très important dans ce genre de montage. (L'autre solution aurait été de faire graver un PCB)

<Photo bottom>

<Photo top>

Réflexion à propos de la topologie:

Condensateur monté en parallèle avec la bobine n'est pas le plus efficace pour réaliser des switch en ZVS, en effet si la fréquence dévie de  $f_r$ , risque de hacher la sinus formé par le résonateur, ce qui entraîne des HF.

Ces HF vont être absorbé par le condensateur (court-circuit pour les HF), ce qui va le faire s'échauffer et risquer de les endommager.

Phénomène rencontré lors d'un test prolongé, mais je n'avais pas pensé à ce comportement lors de l'apparition de l'effet.

Donc solution placer condensateur en série avec la bobine, en théorie le courant devrait être sinusoïdale à travers l'émetteur (sur ce genre de montage on n'observe pas l'évolution de la tension, mais le courant pour déduire le comportement du système).



Par contre le courant qui passera à travers la bobine est le même que celui à travers le condensateur, donc redimensionnement du condensateur pour avoir la même capacité mais capable de faire passer un courant relativement important (5-10A).  
=> Condensateur film faible valeur mis en parallèle

Prochaine fois:

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

Tester sonde courant ACS712 avec arduino

Commencer code de contrôle de l'émetteur: MODBUS

Erwann:

Vérification de l'asservissement moteur

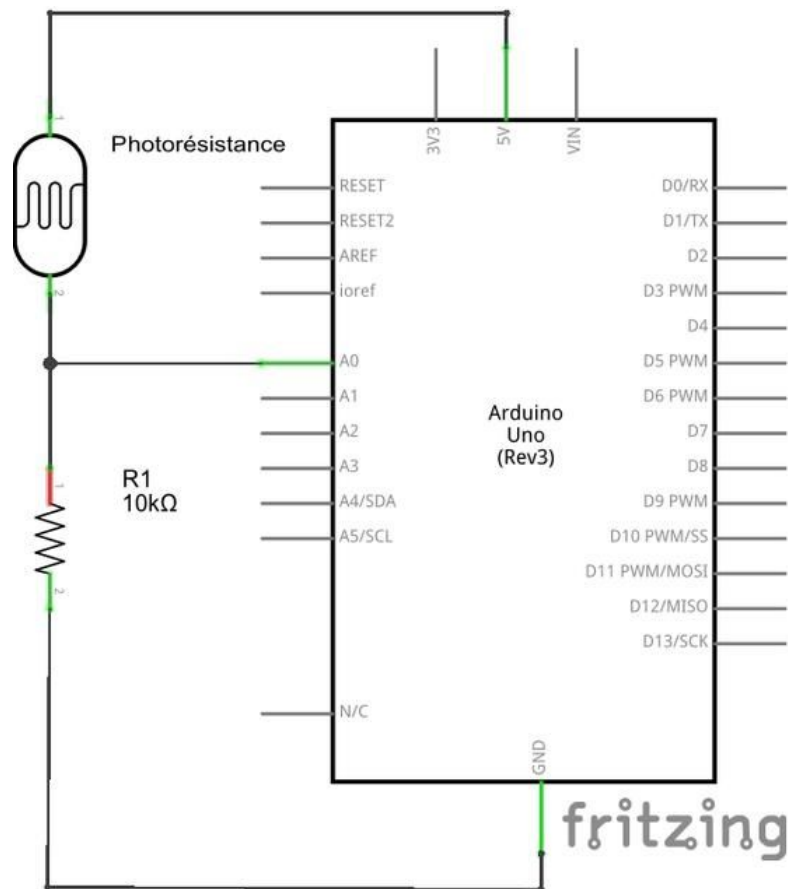
Création d'un programme de test en arduino

Essais avec consigne fixé

Puis essais avec consigne variable par potentiomètre

Matage des LDR sur le support imprimé

Pont diviseur de tension, avec une résistance de 10KOhm



$$U(R1) = (R1 / (R1 + R(\text{Photodiode}))) * 5V$$

$$\Rightarrow R(\text{Photodiode}) = (5 * R1 / U(R1)) - R1$$

Conclusion: Quand  $U(R1)$  augmente,  $R(\text{Photodiode})$  diminue, Donc la luminosité est grande, on peut en déduire le mouvement des panneaux, sans faire le calcul de résistance

Création du programme de test pour les photodiodes

Finir le montage pour teste avec les 4 photodiodes

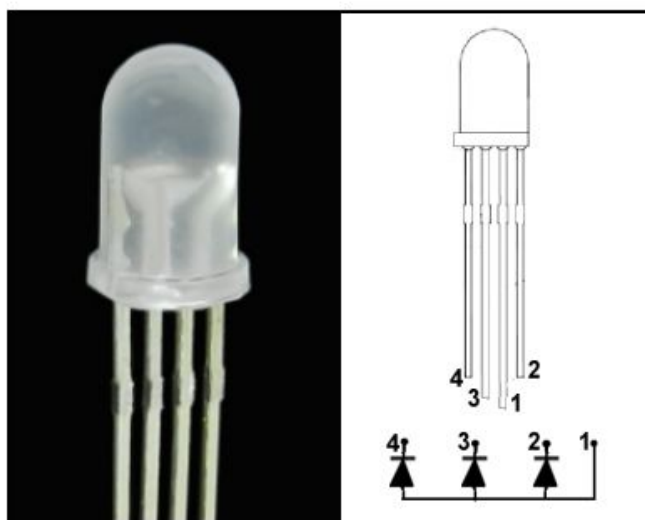
Robin:

Finition des soudures des soudures sur le BCB. Connexion entre la sorties des buffers et les ports femelles de sortie. Pour visualiser plus facilement les différentes LEDs qui vont être contrôlées on place les sorties en 6 groupes de 4 PINs. Chaque groupe de 4 PINs correspond au signal Rouge, Bleu, Vert et la masse (GND, R, G, B). Les sorties de contrôles des 2 LEDs avant et arrières sont reliées en parallèle car on veut contrôler ces deux groupes de LEDs de la même manière. Les PINs femelles utilisées sont de d'espace 2.54mm. Nous sommes restés attentif au cable management étant donné le grand nombre de connexion à rajouter au BCB.

## Calcul des résistances de sortie des buffers:

En sortie des buffers avant les PINs de sortie du BCB on place des résistances en série sur chaque entrées pour obtenir le courant optimale pour avoir la plus grande intensité lumineuse sur nos LEDs sans les endommager et sans surcharger le buffer.

De plus nous utilisons des LEDs RGB à anode commune pour les faire fonctionné en logique inverse, puisque le buffer que nous utilisons peu drainé bien plus de courant vers la masse (état bas) que en fournir (état haut). Pour calculer la valeur des résistance à mettre en série on utilise les caractéristiques des LEDs ci-dessous:



Référence	Couleur	Tension de fonctionnement (V)			Imax	Angle de vue	Intensité lumineuse (mcd) à 20mA
		Min.	Typ.	Max.			
718-D	Rouge 625nm		2	2,2	30mA	100°	800
	Bleu 465nm		3	3,2	30mA		400
	Vert 525nm		3,2	3,4	30mA		800

On voit que les LEDs bleu et verte ont des tensions de fonctionnement semblables: au maximum 3.2V et 3.4V respectivement. C'est pourquoi on décide d'utiliser les même resistances pour ces 2 couleurs. On prends une tension de fonctionnement  $V_{f_{blue,green}} = 3.2V$ . Les LEDs rouges seront contrôlés à part on choisit une tension de fonctionnement  $V_{f_{red}} = 2V$ . Le courant maximum admissible par les LEDs est 30mA cependant nous devons voir le courant maximum que peuvent débiter les buffers SN74LS240 en sortie. La documentation de ce composant ci-dessous nous indique que le courant maximum est de  $I_{max} = 24mA$  en mode de fonctionnement Low-level output current. On va garder une marge de sécurité et on prend  $I_{max} = 15mA$  pour calculer les résistances.



### 6.3 Recommended Operating Conditions

over operating free-air temperature range (unless otherwise noted)

		MIN	NOM	MAX	UNIT	
V <sub>CC</sub>	Supply voltage <sup>(1)</sup>	SN54xS24x	4.5	5	5.5	V
		SN74xS24x	4.75	5	5.25	
V <sub>IH</sub>	High-level input voltage	2			V	
V <sub>IL</sub>	Low-level input voltage	SN54LS24x	0.7			V
		SN54S24x, SN74xS24x	0.8			
I <sub>OH</sub>	High-level output current	SN54xS24x	−12			mA
		SN74xS24x	−15			
I <sub>OL</sub>	Low-level output current	SN54LS24x	12			mA
		SN54S24x	48			
		SN74LS24x	24			
		SN74S24x	64			
External resistance between any input and V <sub>CC</sub> or ground (SNx4S24x only)		40			kΩ	
T <sub>A</sub>	Operating free-air temperature <sup>(2)</sup>	SN54xS24x	−55	125		°C
		SN74xS24x	0	70		

Ce qui nous donne avec une tension Vcc de 5V pour notre circuit les valeurs de résistances suivantes:

$$R_{red} = (V_{CC} - V_{fred}) / I = (5 - 2) / 0.015 = 200 \text{ ohms}$$

$$R_{blue,green} = (V_{CC} - V_{fblue,green}) / I = (5 - 3.2) / 0.015 = 120 \text{ ohms}$$

En tout nous avons besoin de 8 résistances 120 ohms et 4 résistances 200 ohms. On effectue le test des valeurs de toutes les résistances avant de les souder. Variations de +-2 ohms par rapport à la valeur théoriques.

Après que les soudures soit tous terminées en test si les composants sont correctement alimentés en 5V. On place tous les composants sur leur support puis on test aussi les valeurs de tension en sortie du circuit ou seront connectés les 4 PINs de chaque LEDs. Tout fonctionne comme prévu.

Il restera à tester les valeurs de sortie des signaux R, G, B pour chaque sortie qui serviront à obtenir les couleurs de LEDs désirés sur notre drone.

Prochaine fois:

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