Projet Cannonball

Intégration d'une caméra auto-directive

Rapport technique

Auteurs : Mickaël Cottin Bizonne, Azzam Alwan et Jérôme Laurent

Tuteur pédagogique : Ahmad Farhat

Date : Mars 2015

I. Remerciements

Nous tenons à remercier en premier lieu **Ahmad Farhat** pour son suivit et ses conseils tout au long de ce projet.

Nous sommes reconnaissants de l'aide que **Forian Thonnat** nous a apporté, il nous a permis de trouver des solutions efficaces lors de la réalisation matérielle.

Nous souhaitons également exprimer notre gratitude au **FabLab** pour leur support matériel.

Enfin nous remercions toutes les personnes qui nous ont permis d'avancer de près ou de loin sur ce projet.

II. Introduction

La robotique est aujourd'hui omniprésente dans la vie de l'homme, sous plusieurs formes, elle permet la réalisation de taches fastidieuses ou répétitives. Avec de nombreux progrès sur les capteurs et les systèmes embarqués, il existe des solutions permettant aux robots de s'orienter dans l'espace.

Le drone Pixy développé par l'IRD de Montpellier est lui un exemple de robots permettant la photographie vertical aérienne. Tout d'abord, utilisé dans un but scientifique, il a été adapté aux applications audio-visuelles. Récemment sont apparu d'autres drones tels qu'OnyxStar ou encore DJI Phatom accessible au grand public avec un prix abordable (500€).

Figure 1: le drone OnyxStar

Notre travail s'inspire du concept de caméra embarquée sur un drone. Nous avons développé un dispositif permettant de filmer une cible de manière automatique. Ce projet est donc l'association de cette nouvelle idée avec le projet Cannonball.

Le projet Cannonball a été lancé par le Gipsa Lab il y a 3 ans, il consiste à instrumenter une voiture de course radiocommandée afin de la rendre autonome dans ses déplacements. La voiture, équipée d'une caméra et d'une tablette, est capable de reconnaitre des QR codes[1] grâce à un algorithme de traitement d'image. La première application utilisant ce concept consiste à guider la voiture par une seconde qui est équipée d'un QR code à l'arrière. La seconde application défini un parcours sous forme de portes chacune composée de 2 QR codes. Les QR codes contenant l'information du côté de la porte (gauche ou droit) sont détectés par la voiture qui calcule la direction à prendre pour passer à travers la porte.

La nouvelle application que nous avons choisi de développer ne s'inscrit pas dans la lignée des précédentes car elle ne va pas affecter le déplacement de la voiture. Pour réaliser cette fonction de caméra auto-directive, nous avons choisi un concept d'émetteur/récepteur ultrason. La cible à filmer est équipée d'un émetteur qui propage un signal ultrason à une certaine fréquence tandis que la voiture est équipée de trois récepteurs ultrasons qui ont pour but de capter le signal émis. Le positionnement des capteurs les uns par rapport aux autres permet, à partir des signaux reçus, de calculer de manière géométrique la position de l'émetteur. Ce calcul est effectué par la carte Nucleo STM32 qui est alimentée par la tablette déjà présente sur la voiture. Après avoir calculé la position de l'émetteur, la carte Nucleo STM32 commande un servomoteur[2] S125 6T D, sur lequel est positionnée la caméra, pour qu'il soit orienté vers l'émetteur et qu'il puisse le filmer. Le schéma suivant présente l'association de notre travail avec les fonctionnalités déjà existantes :

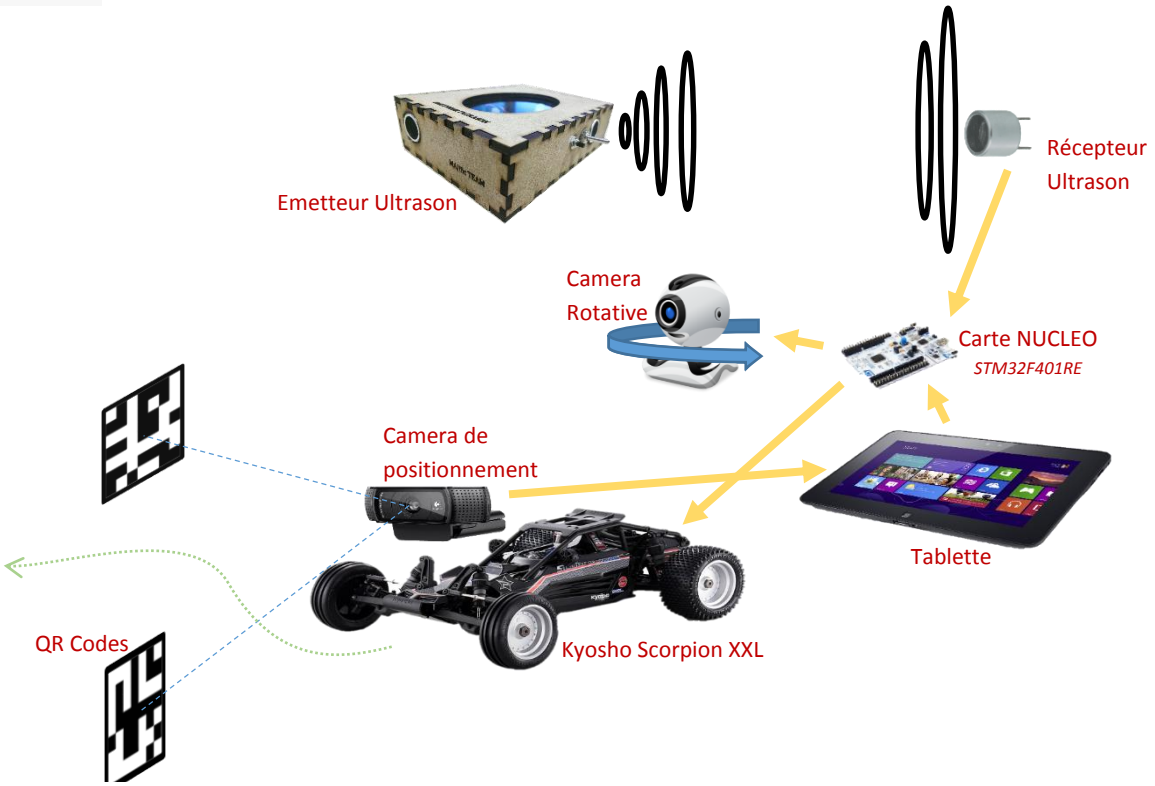


Figure 2 : Fonctionnement général du robot Cannonball

Ainsi, ce rapport présente en trois parties les détails techniques composant ce système :

- La partie hardware
- La partie software
- La partie mécanique

Chaque partie détaille suffisamment les solutions adoptées de manière à les rendre exploitables dans de futurs projets. Dans cette optique, les problèmes rencontrés sont également exposé à dans la suite du rapport ainsi que les moyens qui ont été mis en œuvre pour les surmonter.

III. Hardware

1. Choix technologique et contraintes

Afin de diriger la caméra pour qu'elle cible un objet en mouvement, il faut trouver une solution technique de mesure de position à distance. On cherche à définir la position d'un objet et pour cela deux solutions s'offrent à nous :

- Le traitement d'image, ceci suppose un système de calcul puissant. La tablette utilisée au moment où nous écrivons ces lignes n'a pas la capacité de traitement suffisant pour gérer deux flux vidéo en temps réel. L'avantage de cette méthode est son fonctionnement 100% logiciel avec une grande possibilité d'adaptation.
- Détection par triangulation[3] de signaux, cette méthode repose sur la mesure du temps de parcourt d'un signal. Les deux types de signaux susceptible de nous intéresser car ils se propagent dans l'air, sont soit de nature électromagnétique soit de nature acoustique.

Après l'étude de ces options, nous avons choisi d'utiliser la propagation de l'onde ultrasonore[4] comme support signal.

Avantages :	Inconvénients :
• Vitesse du son lente (340m.s^{-1}) • Instrumentation simple. • Peu cher • Angle d'ouverture des capteurs large (30°) • Portée jusqu'à 3m.	• Sensibilité à d'autres sources sonores et aux vibrations mécaniques • Grande réverbération de l'ultrason sur des objets, créant des sources parasite • Présence d'un émetteur sur la cible à détecter • Traitement du signal en sortie de capteur

Le système comprendra :

- L'émetteur, une source ultrason puissante, multidirectionnelle et autonome: source d'alimentation dédiée, 4 haut-parleurs ultrason, impulsions toutes les 20ms d'un signal à 40kHz.
- Le Récepteur, une détection sensible et autonome: source d'alimentation dédiée, 3 microphones, traitement du signal indépendant, information de sortie du type numérique (tension TTL 0-5V).

2. Elaboration du montage

a) Emetteur

Afin de respecter le cahier des charges décrit dans le point précédent, l'émetteur sera constitué d'un circuit électronique dédié. Ce système doit être le plus petit possible par soucis d'encombrement et il est composé de plusieurs fonctions distinctes :

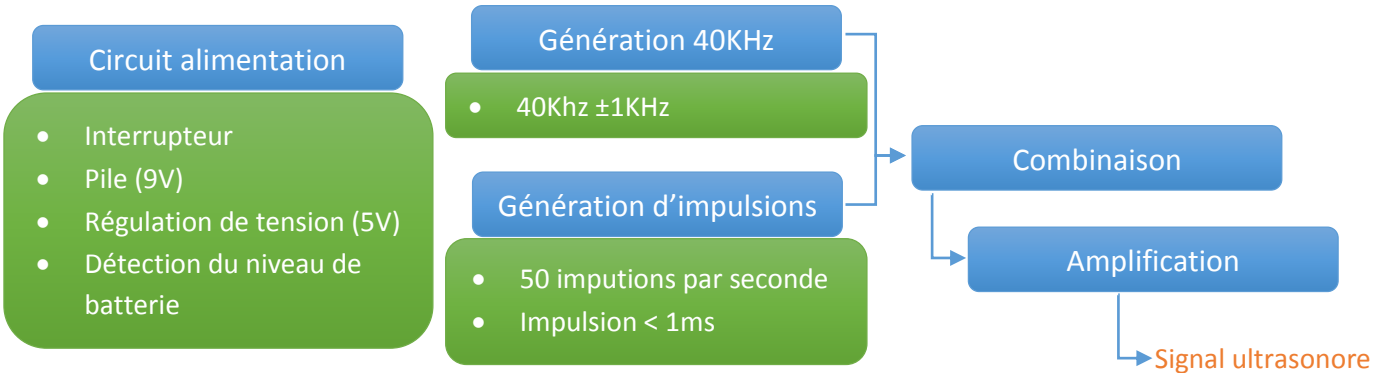


Figure 3 : schéma fonctionnel de l'émetteur

Le schéma complet du montage est disponible en annexe (X.2).

- Génération 40KHz

Afin de générer un signal périodique de 40KHz, le choix s'est porté vers le montage nommé astable. L'amplificateur opérationnel fonctionne en saturation et plus précisément en comparaison. Le principe de l'oscillation repose sur la charge et la décharge du condensateur C1 au travers de la résistance R3. Le système a donc deux états, R1 et R2 permettent de générer une tension stable :

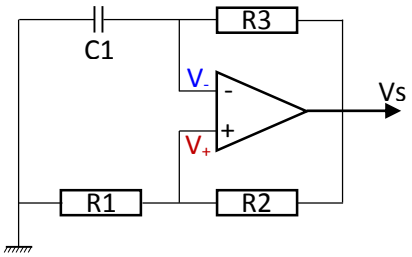


Figure 4 : montage astable

$$V+= \pm \frac{R1}{R1+R2}.Vsat \qquad (1)$$

Dans le cas idéal où la tension de sortie de l'AOP est égale à la tension d'alimentation et l'impédance de sortie de l'AOP est nulle alors la tension Vs oscille à $\pm V_{alim}$ à une fréquence F :

$$\frac{1}{F} = C1.R3.\ln\left[\frac{2(R1+R2)}{R1}\right] \qquad (2)$$

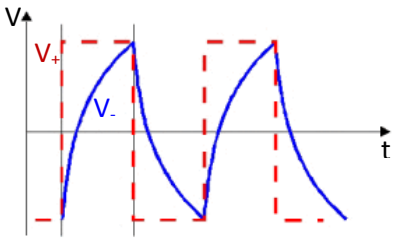


Figure 5 : évolution des signaux interne

De cette manière, il est possible de déduire la valeur des composants. Ces valeurs ont été légèrement modifiées dans le cas pratique pour que le fonctionnement avec l'AOP TL082 soit correct.

Pour R1 = 1,8KΩ ; R2 = 15KΩ ; R3 = 1,2KΩ ; C1 = 22nF, la fréquence d'oscillation mesurée sur la structure finale est de 40,0kHz.

- Génération d'impulsions

La génération d'impulsions est générée aussi par un montage astable légèrement différent du montage précédent car le temps de charge et de décharge du condensateur est contrôlé afin d'avoir le rapport cyclique α voulu. L'impulsion doit être brève α < 10% et la fréquence F doit être proche de 50Hz.

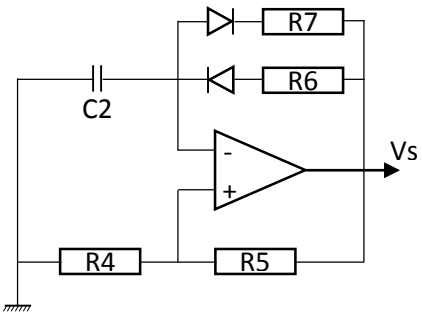


Figure 6 : montage astable, contrôle du rapport cyclique

La fréquence et le rapport cyclique sont donnés par ces formules :

$$\frac{1}{F} = C2.(R7+R6).\ln\left[\frac{2(R4+R5)}{R4}\right] ; \alpha = \frac{R7}{R7+R6} \qquad (3) ; (4)$$

Dans la pratique, les défauts de l'AOP nous obligent à choisir des valeurs de composants légèrement différentes soit R4 = 1KΩ ; R5 = 1,5KΩ ; R6 = 17,8KΩ ; R7 = 10Ω ; C2 = 1µF.

On mesure α = 8,57 pour une fréquence F = 41,6Hz.

- Mise en forme, combinaison et amplification

Les signaux issus des deux oscillateurs sont combinés par un ET logique afin d'avoir le signal utile. Or à la sortie des oscillateurs, les signaux oscillent à ±Vsat des AOP, ce n'est pas des signaux logiques (0-5V). Il convient donc d'adapter ce signal.

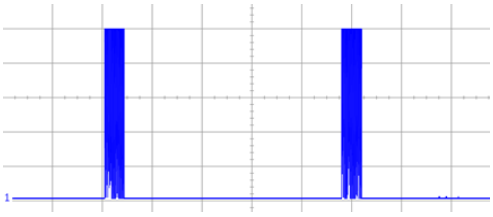


Figure 7 : visualisation du signal de sortie, envoyé aux haut-parleurs

Ceci est réalisé par une fonction électronique contenant un transistor NPN qui est passant à +Vsat et bloqué à –Vsat. La combinaison s'effectue simplement grâce aux NON-ET logique du composant 74LS00. Finalement, afin de délivrer le maximum de puissance aux haut-parleurs, le signal est amplifié par un montage à transistor.

- Circuit PCB

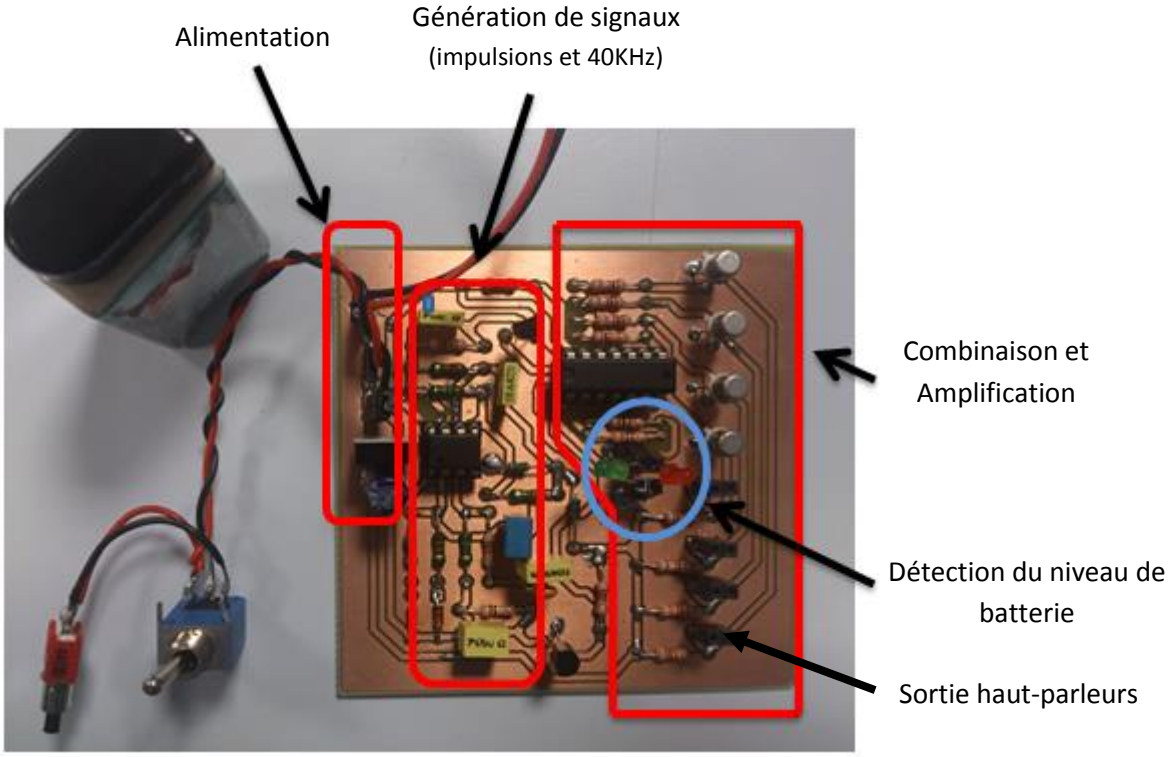


Figure 8 : circuit PCB de l'émetteur

b) Récepteur

Afin de respecter le cahier des charges, le récepteur doit intégrer 3 microphones et doit être capable de détecter l'information utile soit les impulsions à 40KHz. Ce traitement est effectué de manière analogique avec doit être très sélectif afin de limiter les bruits parasites. Le schéma complet du montage est disponible en annexe (X.4).

- Processus de traitement

Une fois le signal obtenu grâce aux capteurs, il est amplifié pour être puis filtré à la bonne fréquence pour obtenir une réponse correcte. L'objectif est ensuite de détecter l'instant où le signal est détecté par le capteur. L'enveloppe du signal est comparée à un seuil de détection, le front de tension correspond ainsi à la détection d'un signal ultrasonore. Pour finir, le montage assure une sortie TTL pour que le signal logique varie entre 0 et 5 Volts. De cette manière, un microcontrôleur à la capacité de récupérer les données.

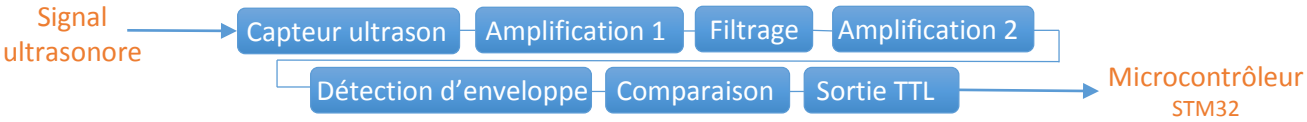


Figure 9 : Schéma fonctionnel du Récepteur

- Amplification

Les fonctions électroniques décrites ci-dessus sont alimentées en +5 Volts. Elles nécessitent pour la plupart, un ou plusieurs amplificateurs opérationnels, la chaine de traitement est composée d'un total de 7 AOP[5]. Afin de limiter le nombre de composant dans le circuit, nous utilisons un LM324N intégrant 4 AOP, un TL082 intégrant 2 AOP puis un LF356N intégrant un AOP.

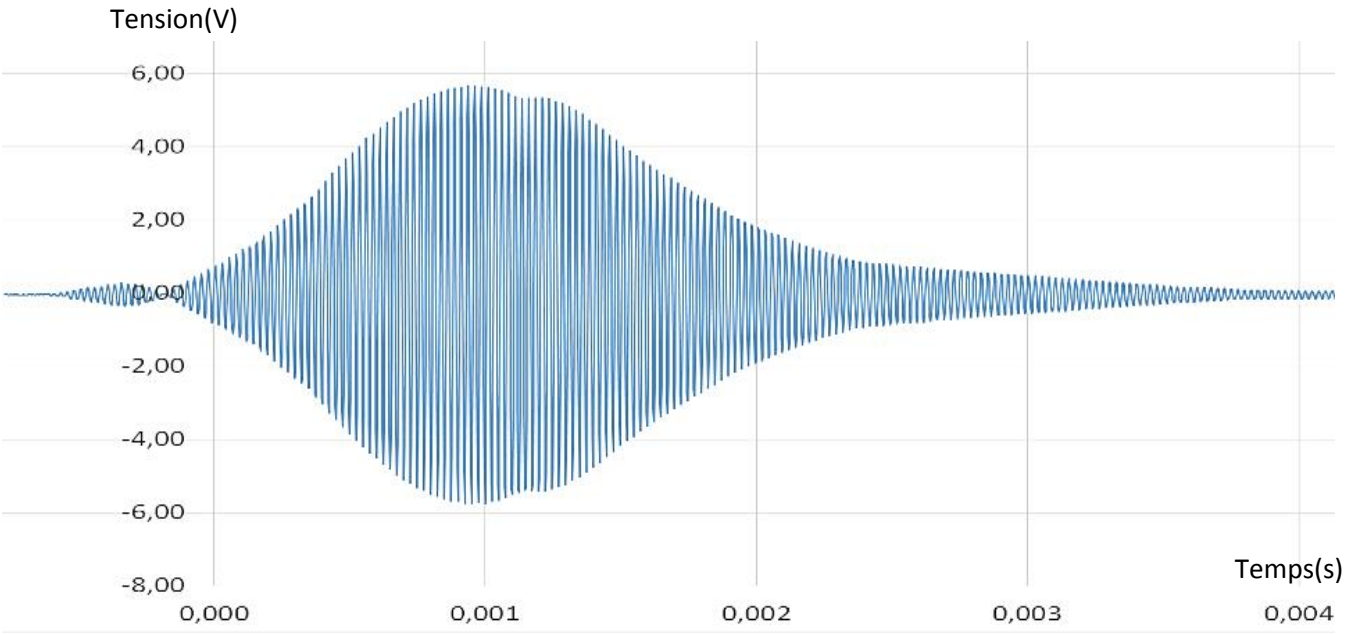


Figure 10 : Visualisation du signal amplifié

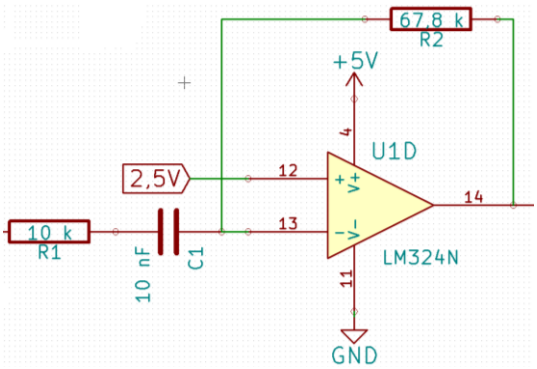


Figure 11 : Amplification n°1 de la carte récepteur

Les paramètres des amplificateurs ont été déterminés après une série de mesures du signal reçu par les capteurs ultrasons lorsque l'émetteur se trouve à 3m de distance. Ces mesures nous ont révélé qu'un gain de 45 dB serait nécessaire pour exploiter ce signal. Pour notre application, le produit gain-bande[6] serait de : 40kHz * 45dB soit 7 MHz. Un tel produit gain-bande ne peut être assuré par un seul amplificateur, c'est pourquoi nous avons placé plusieurs amplifications dans la chaine de traitement.

La première amplification est réalisée par le LM324N (pattes 12, 13, 14). Le signal ultrasonore perçu par le capteur est centré en 0. Pour réaliser une amplification correcte, on transpose la valeur moyenne de ce signal en 2,5 Volts (soit la moitié de notre tension d'alimentation) en appliquant une tension équivalente sur la patte « + » de l'AOP. On place également en amont de chaque étage un condensateur en série afin de supprimer des basses fréquences parasites particulièrement dues au 50Hz. Cette première amplification apporte un gain de :

$$\frac{R2}{R1} = \frac{67{,}8k}{10k} = 6{,}78\ dB \qquad (5)$$

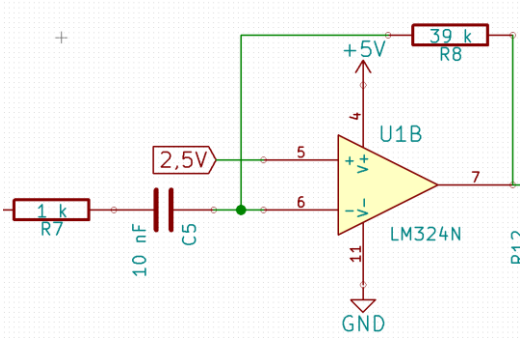


Figure 12 : Amplification n°2 de la carte récepteur

La deuxième amplification est également réalisée par le LM324N (pattes 5, 6 et 7). Elle est placée en sortie du filtre à 40kHz (détaillé plus bas). Ici, le signal est beaucoup plus propre, on réalise donc une amplification plus importante (toujours centrée en 2,5 Volts) :

$$\frac{R2}{R1} = \frac{39k}{1k} = 39\ dB \qquad (6)$$

- Filtrage

Nous ajoutons également un filtre passe bande afin de limiter des éventuelles perturbations sur la réception des signaux ultrasons. Nous avons choisi de reproduire le filtre réalisé sur la carte HC-SR04.

- Détection d'enveloppe

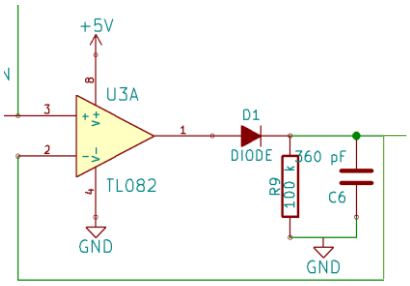


Figure 13 : Schéma de la détection d'enveloppe

La détection de crête est réalisée par les deux ampli-op intégré du TL082. Sur le schéma ci-contre, le condensateur C6 joue le rôle de mémoire. La diode D1 conduira dès que La tension d'entrée sera supérieure à la tension aux bornes du condensateur. Dans ce cas, le condensateur sera chargé avec la nouvelle valeur de la tension d'entrée. On rajoute un deuxième AOP pour réduire le courant au niveau du condensateur et une résistance en parallèle du condensateur pour régler le temps de décharge de ce dernier.

- Comparateur

Le comparateur récupère le signal en sortie du détecteur d'enveloppe et fourni un signal carré dépendant du réglage de la tension de seuil sur la patte « - » de l'amplificateur (patte n°2 du LM324N). La tension de seuil peut prendre une valeur allant de 2,5 Volts à 5 Volts, elle est ajustable grâce à un potentiomètre ayant une résistance maximale de 100 kOhms. Cette valeur nous permet d'avoir une précision suffisante concernant le réglage de la sensibilité des capteurs.

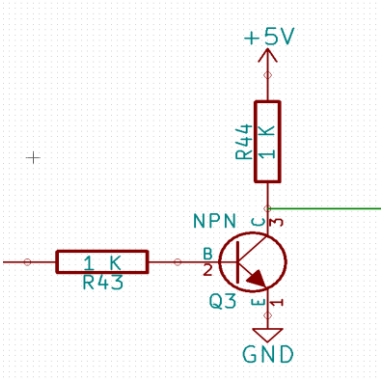


Figure 14 : Sortie TTL

- Sortie TTL

On place en sortie du montage une sortie TTL dans le but d'obtenir un signal carré variant de 0 Volts à 5 Volts (au lieu des tensions +Vsat et –Vsat du LM324N). Notre avons choisi d'utiliser un transistor PNP, la détection des signaux sera sur front descendant.

- Circuit PCB

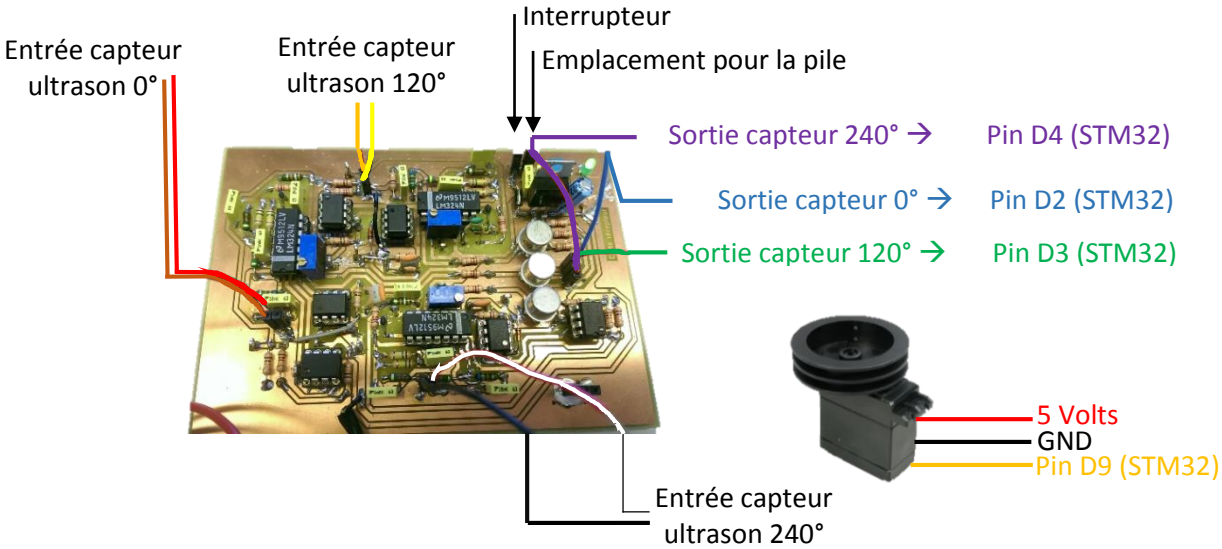


Figure 15 : Schéma de câblage Récepteur

3. Réalisation du PCB

Ces montages ont été élaborés avec la version 2013 du logiciel Kicad, celui-ci nous a permis d'imprimer le PCB[7] des cartes pour notre projet. Le logiciel Kicad est une suite open source pour la réalisation de schémas électroniques, Il dispose d'un environnement de développement intégré avec éditeur de schémas, il inclut des bibliothèques de composants, et permet également de créer notre propre composant. Il existe des outils de routage automatique, mais payants. Nous avons choisi pour des raisons économiques d'optimiser manuellement le routage de nos cartes afin d'adapter leurs dimensions à notre projet.

Les deux cartes sont imprimées en simple face et sont de géométrie 80x80mm pour l'émetteur et de 89x130mm pour le récepteur. Les pistes de cuivres sont de 1 mm de large, les pins ont été collées afin d'assurer une certaine résistance à l'usure.

IV. Software

1. Choix technologique et contraintes

L'objectif du projet étant d'ajouter une nouvelle fonctionnalité, il est nécessaire de trouver une solution pour préserver ce qui a été déjà fait et donc de satisfaire le contrôle de la voiture (vitesse et direction) et le contrôle de la caméra pivotante.

Le contrôle de la voiture est initialement supporté par une carte Arduino Uno. En vue de l'ensemble des améliorations à apporter qui demandent une puissance de calcul conséquente (calcul trigonométrique à virgule flottante, voir partie IV.2.c), nous avons décidé de changer de carte. Ainsi, pour la partie contrôle/commande du système, notre choix s'est porté vers le microcontrôleur STM32F401RE présent sur la carte NUCLEO pour plusieurs raisons :

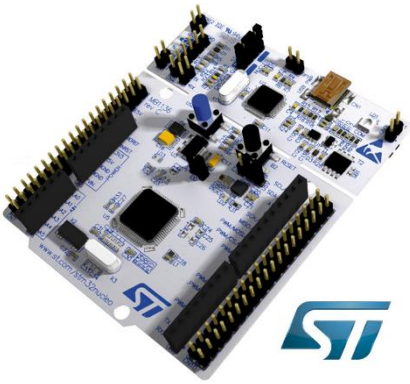


Figure 16 : STM32 NUCLEO Board

- Prix faible de la carte (~10€)
- Caractéristique technique suffisante (512KB Flash, ARM-CPU 84MHz, Communication USB)
- Robuste, adapté à la réalisation de prototype
- Similaire à aux carte Arduino (facilité pour le portage du code)
- Facilité d'utilisation avec la bibliothèque C++ mbed
- Possibilité d'utiliser un système d'exploitation et le Multithread[1]

Pour le support informatique, notre choix c'est porté sur la plateforme µvision 5.14 utilisant STLINK 2 comme driver. Le programme est basé sur 3 bibliothèques (http://developer.mbed.org) :

- « mbed.h », ajout d'une surcouche logicielle pour plus d'abstraction du matériel
- « rtos.h », Real-Time Open Système basé sur une implémentation RTX permettant la gestion des taches.
- « DriverSerial.h », transmission de donnée USB avec buffer.

2. Elaboration logiciel

Dans le but d'obtenir une programmation intelligible et efficace, le programme a été séparé en un maximum de sous-parties chacune contenu dans des threads indépendant. L'ensemble du code est présent en Annexe « main.cpp » et aussi sur la plateforme GitHub « https://github.com/Jerome-Laurent/CannonBall-Nucleo-STM32F401 »

a) Définition I/O

Les ports d'entrée/sorties ont été définit de la manière suivante:

- D0, D1 : Utilisés pour communication USB.
- D2, D3 et D4 : Signaux TTL[8] provenant de la carte Réception Ultrason (III.2.b), actif sur front descendant.

- D5 et D6 : Commande PWM[9] pour le servomoteur (D5 pour la direction, D6 pour l'accélération).
- D9 : Commande PWM pour le servomoteur permettant la rotation de la camera.
- D13 : Led présente sur la carte d'origine

b) Contrôle de la voiture

Il s'agit de traduire le code déjà créé sur l'Arduino pour qu'il soit implémenté sur le microcontrôleur STM32.

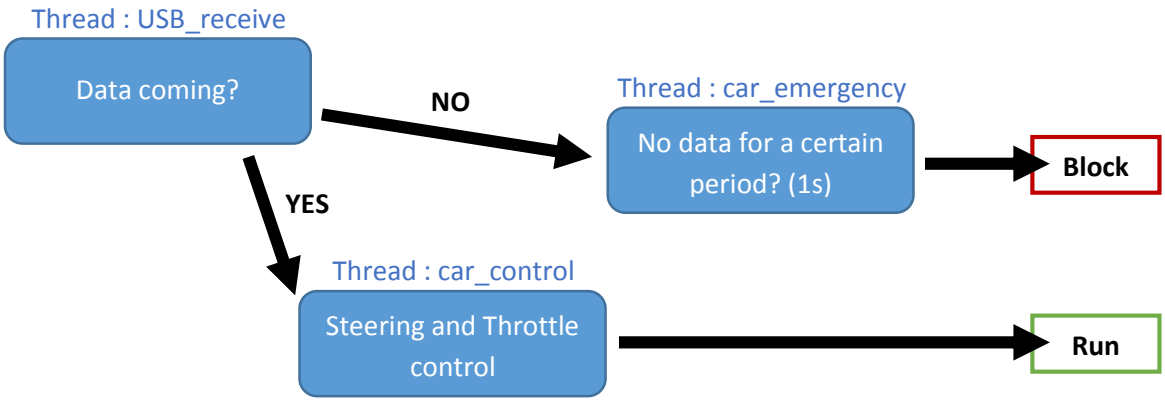


Figure 17 : Contrôle de la voiture

c) Commande camera pivotante

Pour le contrôle de la caméra, le système se décompose en trois états successifs détaillés ci-dessous.

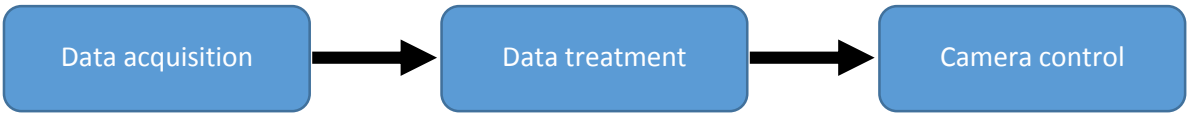


Figure 18 : Contrôle de la camera

- **<u>Data acquisition</u>** :

Les 3 signaux correspondant à la détection impulsionnelle ultrason vue dans la partie précédente sont actifs à l'état bas. Ainsi, les fronts descendant correspondent au moment de référence à mémoriser appelé $\mathbf{T_1}$, $\mathbf{T_2}$ et $\mathbf{T_3}$. Grâce aux moments obtenus, nous pouvons déduire la position de la source par le système de triangulation expliqué ci-dessous.

On considère dans un premier temps la distance entre l'émetteur et le récepteur grande devant la distance de chaque point de réception (les 3 microphones). On est dans une situation d'onde plane c'est-à-dire qu'il existe un plan perpendiculaire à la propagation des signaux. La géométrie choisie pour la disposition des microphones est un triangle équilatéral.

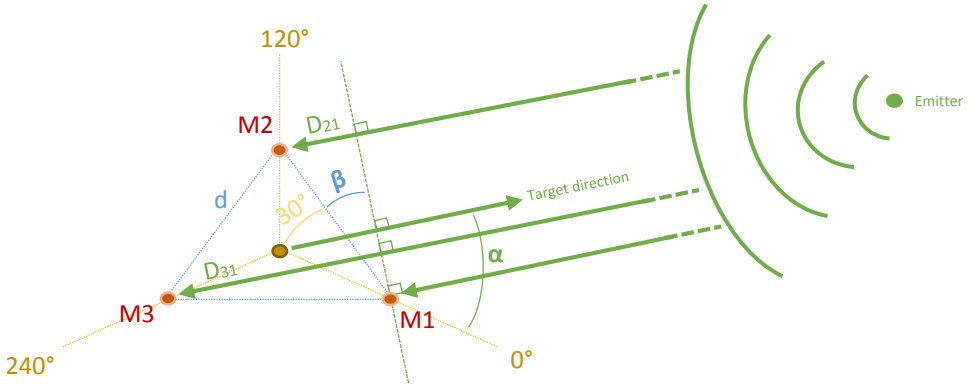


Figure 19 : Schéma théorique de la détection de présence

- **<u>Data treatment</u>** :

L'objectif est de trouver la valeur de l'angle α. Afin de déterminer cette valeur, il faut interpréter le schéma ci-dessus en y ajoutant 2 contraintes supplémentaires :

- Il n'y a pas obligatoirement de réception sur les 3 microphones car l'ouverture du transducteur est limitée.
- Si les 3 microphones sont atteint par l'onde sonore, nous allons dans le cas pratique ignorer le celui qui atteint en dernier. Une réverbération[10] possible due à un objet proche peut compromettre le bon résultat de la mesure.

Il va y avoir 7 cas différent à traiter :

- Cas 1 : M1 est atteint → angle = 0°
- Cas 2 : M1 et M2 sont atteint → angle = 60° - $\frac{180}{\pi}\left(\frac{V*(T_1-T_2)}{d}\right)$
- Cas 3 : M2 est atteint → angle = 120°
- Cas 4 : M2 et M3 sont atteint → angle = 180° - $\frac{180}{\pi}\left(\frac{V*(T_2-T_3)}{d}\right)$
- Cas 5 : M3 est atteint → angle = 240°
- Cas 6 : M3 et M1 sont atteint → angle = 300° - $\frac{180}{\pi}\left(\frac{V*(T_3-T_1)}{d}\right)$
- Cas 7 : M1, M2 et M3 sont atteint → angle = cas 2, cas 4 ou cas 6 suivant

Avec **V**, vitesse du son dans l'air soit 340m.s^{-1} dans des conditions normales de pression et d'humidité. Et **d** est la distance entre chaque microphone soit 0,20m.

Il y a donc deux catégories distinctes :

- Seul un point est atteint l'angle prend dans ce cas la valeur du point soit 0°, 120° ou 240°.
- Deux points sont atteints, l'angle doit être calculé par la méthode ci-dessous :

On se place dans la situation de la figure 18 ce qui correspond au cas n°2 :

$\sin\beta = \frac{D_{21}}{d}$ avec $D_{21} = V \times (T_1 - T_2)$ or on a $\alpha = 90 - 30 - \beta$

$$\boxed{\alpha = 60 - \sin^{-1}\left(\frac{V * (T_1 - T_2)}{d}\right)} \quad (7)$$

On peut approximer le calcul car $\sin^{-1}(X) \approx X$ pour des petits angles ($|\alpha| < 30°$ → 7% d'erreur)

$$\boxed{\alpha \approx 60 - \frac{180}{\pi}\left(\frac{V * (T_1 - T_2)}{d}\right)} \quad (8)$$

Une fois l'angle trouvé, on applique un filtre passe bas RII (Réponse Impulsionnel Infinie) afin de gagner en précision et en souplesse :

$$\alpha_n = \frac{\alpha_n}{4} + \frac{3 * \alpha_{n-1}}{4} \quad (9)$$

- **<u>Camera control</u>** :

Le contrôle du servomoteur S 125 (6 tours) permettant de faire pivoter la camera s'effectue grâce à une PWM. Lorsque la PWM = 0.05, l'angle est à 0° (0 tour); lorsque la PWM = 0.1, l'angle est à 2160° (6 tours). La conversion se fait par cette formule :

$$\boxed{PWM = \frac{\alpha}{43200} + 0.05} \quad (10)$$

V. Intégration physique sur la voiture

La solution technologique étant développée, il est nécessaire d'intégrer les cartes sur la voiture de manière stable afin de les protéger. Pour cela, nous avons choisi de concevoir nos pièces sur mesure en utilisant le logiciel Sketchup (version 2015) et l'imprimante 3D *Maker Replicator 2* de l'école Polytech Grenoble.

Les éléments à intégrer sur la voiture sont les suivants :

- La carte électronique de réception du signal
- Le servomoteur S125 6T D
- Les 3 capteurs ultrasons
- La carte STM32

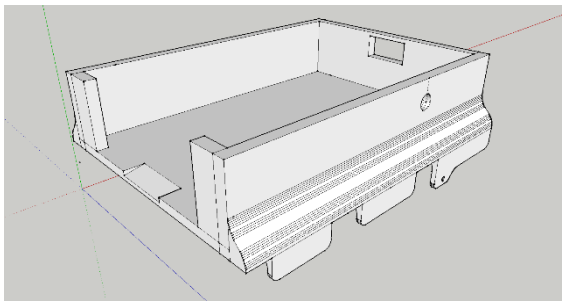
Cette intégration présente plusieurs contraintes que nous devons prendre en compte :

- Coût de la matière première pour l'impression
- Respect de la géométrie des capteurs ultrasons
- Intégration facile et stabilité des structures
- Respect des éléments déjà présent sur la voiture

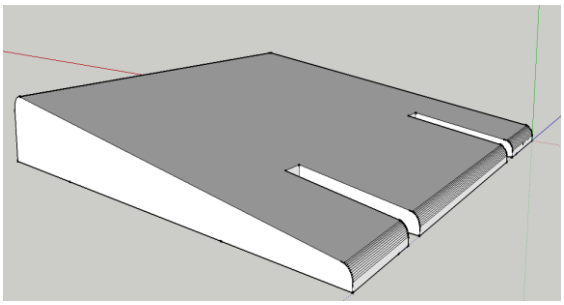
Nous proposons une solution comprenant 3 pièces. La première pièce accueille la carte « récepteur », elle est disposée à l'arrière de la voiture en raison de ses dimensions importantes (131mm x 90mm) et est fixée sur les barres du toit de la voiture grâce à deux vis de serrage (*Pièce 1*).

La seconde pièce est disposée sur le toit de la voiture, elle épouse les formes de la structure et permet d'obtenir une surface plane et stable au-dessus de la voiture (*Pièce 2*).

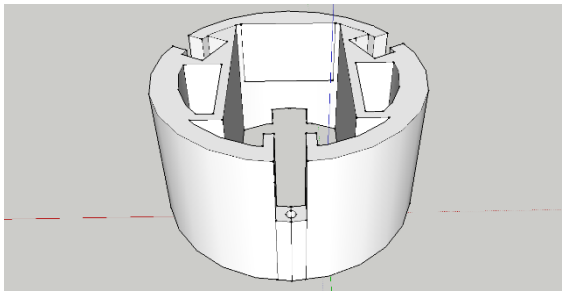
La troisième pièce est collée au-dessus de la deuxième, c'est un assemblage d'une base cylindrique (*Pièce 3*) creuse et de trois bras (*Pièce 4*) accueillant chacun un capteur ultrason en son extrémité. Les bras sont encastrés puis vissés sur le contour de la base cylindrique et sont chacun espacé de 120° pour respecter le calcul de triangulation.



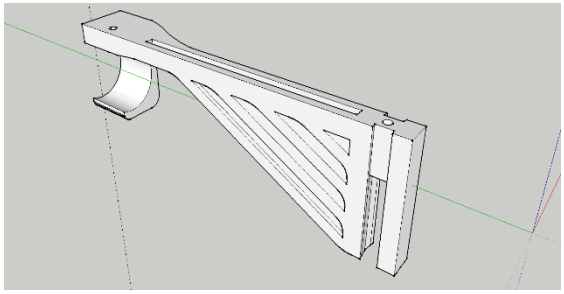
Pièce 4: Support carte récepteur



Pièce 4 : Plaque d'adaptation



Pièce 4: Base cylindrique



Pièce 4 : Bras

Figure 20 : Support physique du capteur

La distance entre chaque capteur est de 200 millimètres. Plus cette distance est grande, plus la précision de détection sera meilleure. En revanche, nous sommes limités par les dimensions de l'imprimante 3D, c'est pourquoi nous avons réalisé un compromis entre ces deux caractéristiques.

La solution finale est donc facilement intégrable à la structure et elle respecte le placement des capteurs pour le calcul de l'orientation de la caméra. Les dimensions de chaque pièce sont disponibles en annexe. Cependant, la solution occupe l'intégralité du toit de la voiture qui était déjà occupé par la première webcam. Une solution serait de créer une plateforme à l'avant de voiture qui accueillerait la caméra directive.

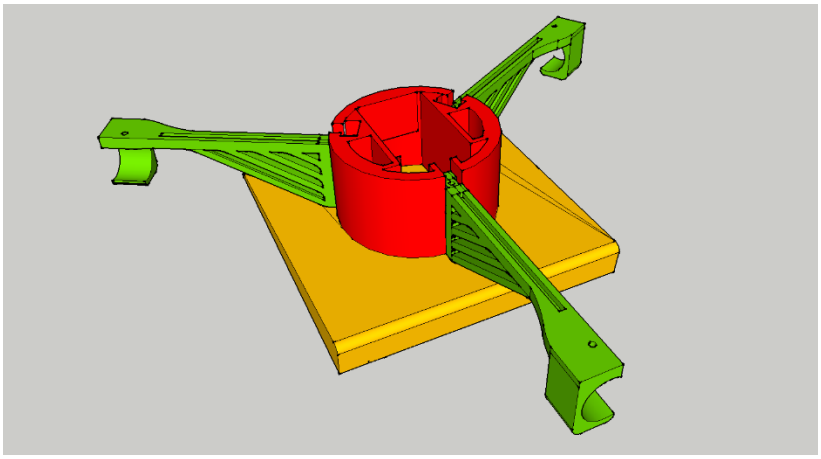


Figure 21 : Assemblage des pièces 2,3 et 4

VI. Résultats & corrections

Après assemblage des parties mécaniques, hardware et software, le test du fonctionnement de l'ensemble du dispositif a révélé quelques points sensibles à corriger. Le mouvement du servomoteur ne correspondait pas à celui de la source ultrasonore. Nous avons donc établi une liste d'hypothèses concernant l'origine de la perturbation du mouvement du servomoteur et réaliser des tests permettant de les vérifier.

a) Hypothèses

Après avoir mesuré les signaux en sortie de la carte « récepteur » à l'oscilloscope, nous avons défini les hypothèses suivantes comme étant des sources possibles de perturbation :

1. Différence entre les 3 capteurs utilisés
2. Sensibilité différente entre les 3 capteurs
3. Différence entre les 3 traitements électroniques
4. Perturbation électrique, électromagnétiques et mécanique dues au servomoteur

b) Mesures & Solutions

- Caractérisation des capteurs

Pour répondre à cette question, nous avons analysé la réponse de différents capteurs en fonction d'un même signal ultrasonore, en prenant soin de garder la même distance entre l'émetteur et le récepteur lors de chaque mesure. Le tableau ci-dessous montre l'amplitude crête à crête perçue par 7 différents récepteurs pour un même signal émis à 40kHz. Nous avons sélectionné suite à ce test les capteurs les plus sensibles dans le but d'avoir un rapport signal sur bruit plus élevé soit les numéros 2,3 et 6.

Numéro du capteur	1	2	3	4	5	6	7
Amplitude cc [Volts]	1,01	2,40	2,54	1,96	0,91	2,10	1,1

- Sensibilité

Les potentiomètres utilisés pour les comparateurs dans les 3 chaines de traitement (partie III.2.b) jouent un rôle important pour le bon fonctionnement du dispositif.

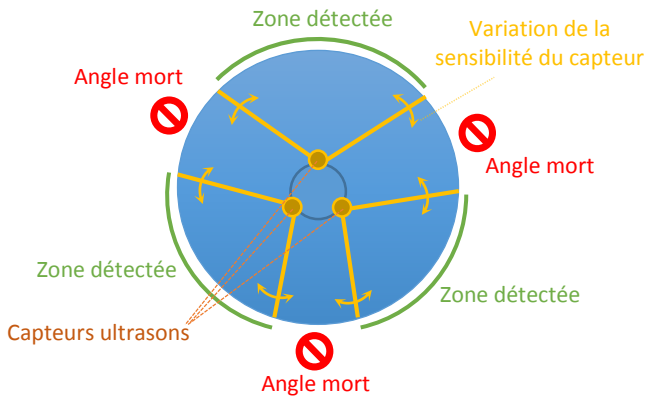


Figure 22 : Schéma du fonctionnement de la détection de signaux

Chaque comparateur permet de régler la sensibilité du capteur qui lui est associé, c'est-à-dire régler la zone de détection du capteur. Plus la sensibilité d'un capteur est élevée, plus sa zone de détection est large. Cependant, augmenter la sensibilité d'un capteur le rend plus vulnérable aux différentes perturbations du système décrites ci-dessous.

Dans le cas contraire, une faible sensibilité induit plusieurs angles morts dans lesquels il est impossible de détecter un signal. Le nombre de positions angulaires permises à la caméra se retrouve limité. Ce paramètre sera ajusté de manière à ce qu'aucune perturbation ne soit détectée pour ne pas altérer le fonctionnement du système tout en gardant une sensibilité maximale.

- Traitement électronique

Pour que le calcul de la position angulaire de la caméra soit bien effectué, il est nécessaire d'avoir un comportement identique entre les trois chaines de traitement électronique (outre le réglage de la sensibilité). Pour vérifier cela, nous avons comparé, à tour de rôle, la réponse des trois circuits électroniques chacun équipé du même capteur ultrason pour la réception du signal. Il s'est avérée que les trois circuits n'avaient pas la même réponse, ce phénomène s'explique par l'imprécision des composants du circuit.

- Perturbations dues au servomoteur

Comme nous l'avons précisé plus haut, le mouvement du servomoteur ne répondait pas parfaitement aux critères imposés lors des tests. Des perturbations étaient présentes lorsque le servomoteur était en mouvement, nous avons donc coder un petit programme faisant tourner le servomoteur en continu dans un sens puis dans l'autre sur toute sa course pour mieux observer ces perturbations. Nous avons constaté des impulsions parasites hautes fréquences détectées et amplifiés par les circuits électroniques. Nous en avons conclu que la source de ces perturbations pouvait être électrique, électromagnétique ou encore mécanique.

La perturbation électrique se traduit par le fort courant de démarrage demandé par le moteur induit des perturbations de la tension d'alimentation. Pour vérifier cela, nous avons branché le servomoteur sur une source d'alimentation externe et comparé les deux fonctionnements. Après mesures, nous avons conclu que la perturbation d'ordre électrique n'était pas majoritairement responsable des perturbations observées.

La base cylindrique (partie V, pièce 3) a été conçue pour accueillir les fils reliant les capteurs ultrasons à la carte en son centre. Nous avons supposé que ces câbles, étant très proche du servomoteur, étaient sensibles à des perturbations électromagnétiques. Nous avons donc déplacé ces câbles hors de la structure cylindrique et observé une certaine amélioration sur les signaux en sortie de la carte « récepteur ».

Chaque capteur ultrason est encastré dans le bras qui lui est associé. Les trois bras sont également encastrés dans la base cylindrique qui elle-même est fixée au servomoteur par deux vis de serrage. En observant les impulsions parasites, nous avons pensé qu'une telle structure propagerait les vibrations causées par le servomoteur jusqu'aux capteurs ultrasons. Nous avons donc tenté d'isoler au maximum les capteurs en insérant de la mousse entre ces derniers et la structure mécanique. N'ayant pas le temps de repenser la

structure mécanique pour accueillir une partie isolant les vibrations, cette méthode nous a permis d'améliorer le signal en sortie de la carte cependant, le circuit détectait encore des perturbations.

- Conclusion

Il est évident que des améliorations sont à apporter sur le montage la carte de réception afin de la rendre moins sensible aux bruits. Une autre solution serait d'ajouter des capteurs afin de supprimer les angles morts.

Figure 23 : Vu final du robot

VII. Conclusion

Le projet Cannonball, étant maintenant le fruit de 3 ans d'idée et de travail, a été augmenté d'une fonctionnalité supplémentaire. Les possibilités d'utilisation du robot ont été élargies, lui permettant dorénavant de cibler un objet en mouvement. Ceci a été possible grâce à l'étude et la réalisation d'un système basé sur la physique de l'ultrason incluant des circuits électronique, un microcontrôleur et des supports mécanique. L'objectif définit a été atteint, les tests ont prouvé que la caméra était capable de s'orienter convenablement de façon autonome. Dans le futur, nous espérons que ce projet sera l'objet de nouvelles recherches. Il reste de nombreuses améliorations et innovations envisageables notamment sur la technique de détection de présence ou encore sur de nouvelles fonctionnalités.

VIII. Définitions

1 – **QR Code** : Type de code-barres en deux dimensions constitué de modules noirs disposés dans un carré à fond blanc

2 – **Servomoteur** : Moteur capable de maintenir une opposition à un effort statique et dont la position est vérifiée en continu et corrigée en fonction de la mesure

3 – **Triangulation** : Technique permettant de déterminer la position d'un point en mesurant les angles entre ce point et d'autres points de référence dont la position est connue.

4 – **Onde ultrasonore** : Onde mécanique et élastique qui se propage au travers de supports fluides, solides, gazeux ou liquide ayant une gamme de fréquence allant de 20 000 à 10 000 000 Hertz, trop élevée pour être perçues par l'oreille humaine.

5 – **AOP** : « amplificateur opérationnel » Amplificateur électronique qui permet l'amplification de la différence de potentiel électrique présente à ses entrées.

6 – **Produit Gain Bande** : Caractéristique d'un amplificateur opérationnel permettant de connaitre son gain maximal à une fréquence d'utilisation donnée.

7 – **PCB** : « Printed Circuit Board » Nom anglais désignant un circuit imprimé.

8 – **Signal TTL** : « Transistor-Transistor Logic » Signal défini comme niveau logique bas entre 0 et 0,5 V, et comme niveau logique haut entre 2,4 V et 5 V

9 – **Commande PWM** : Traduction anglaise pour « Modulation de largeur d'impulsions ». Technique couramment utilisée pour synthétiser des signaux continus à l'aide de circuits à fonctionnement tout ou rien.

10 – **Réverbération** : (écho) Phénomène acoustique de réflexion du son.

IX. Bibliographie

[Précédent travaux]
http://fablab.ensimag.fr/index.php/Cannonball_de_voitures_RC/FicheSuivi

[Support STM32]
http://developer.mbed.org/platforms/ST-Nucleo-F401RE/

[Support Capteur]
http://www.generationrobots.com/fr/content/65-les-capteurs-a-ultrasons-pour-les-robots

[Support Electronique]
http://www.elektronique.fr/documents-pour-le-site/documentation/Tutorial-Kicad-FR.pdf
http://uglyduck.ath.cx/ep/archive/2014/01/Making_a_better_HC_SR04_Echo_Locator.html
http://electronique.aop.free.fr/

X. Annexes

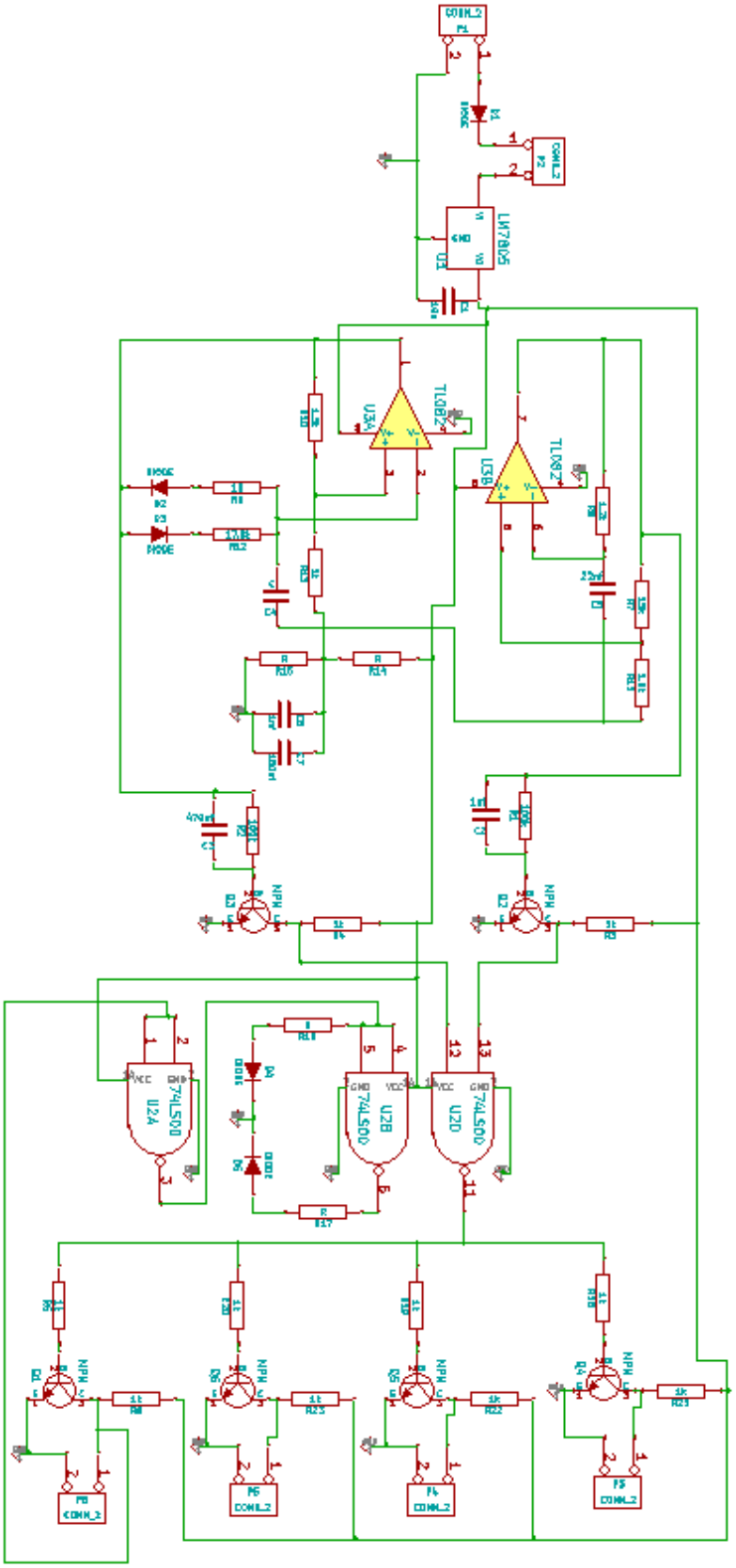
1. Main.cpp

```
/*********************************************
Project CONNONBALL using NUCLEO STM32F401RE
Year 2015
Created by MAJiiiC Team
@mail jerome_laurent16@yahoo.fr
*********************************************

//Include Library
#include "mbed.h"
#include "rtos.h"
#include "SerialDriver.h"

// DU1>--------------------- angle = 0 + i*360 ;
// DU2 >----------------------------- angle = 120 + i*360;
// DU3>------------- angle = 240 + i*360;
const float SuD = (57.3 * 340) / (0.2f * 1000000) ;
// Speed divided by the distance between 2 points
// 57.3 = 180 / pi to have degrees
// 1000000 to convert in µs

//IO definition
SerialDriver usb(USBTX, USBRX,2,2);
InterruptIn detect1(D2);
InterruptIn detect2(D3);
InterruptIn detect3(D4); // set ultrasonic connectors
PwmOut servo(D9); //PWM on the D9 connector
PwmOut STEERING_SERVO_PIN(D5);
// set servo-motor connector for steering
PwmOut THROTTLE_SERVO_PIN(D6);
// set servo-motor connector for throttle
DigitalOut led(LED1);

//Timers
Timer timer1;
Timer timer2;

//Variables
unsigned int time_data_check;
unsigned int time1,time2,time3;
uint8_t reach1,reach2,reach3;
uint8_t steeringTarget;
uint8_t throttleTarget;

//declaration of the Threads
Thread * USB_receive_thread;
Thread * car_control_thread;
Thread * car_emergency_thread;
Thread * camera_control_thread;

// Definition of triggers for the interrupts
void trigger1();
void trigger2();
void trigger3();

//definition of functions used in threads
void USB_recieve(void const *argument);
void car_emergency(void const *argument);
void car_control(void const *argument);
void camera_control(void const *argument);
void treatement(uint16_t &angle, const uint8_t
&combination, uint8_t &round);
void set_camera(const uint16_t &angle);

/*********************************************/
//Car control
void USB_recieve(void const *argument) {
 usb.baud(115200);
 while(true) {
    time_data_check=timer1.read_ms();
    usb.putc(steeringTarget = usb.getc());
    usb.putc(throttleTarget = usb.getc());
    car_control_thread->signal_set(0x01);
    //there is a data
    car_emergency_thread->signal_set(0x01);
    //free the emergency
  }
}

void car_emergency(void const *argument) {
 while(1) {
  if ((timer1.read_ms() - time_data_check > 5000)) {
  //the delay is to hight
  led = true;
   if (throttleTarget < 89) //high speed
   STEERING_SERVO_PIN.write(float(130.0f/255.0f));
   //brake
  else
   THROTTLE_SERVO_PIN.write(float(91.0f/255.0f));
   //only slow down, not to go reverse
   Thread::signal_wait(0x01);
   // waiting for a new transmition
   led = false;
   time_data_check = timer1.read_ms();
   }
  else  Thread::wait(100);
 }
}

void car_control(void const *argument) {
 STEERING_SERVO_PIN.write(float(90.0f/255.0f));
 //initialisation
 THROTTLE_SERVO_PIN.write(float(91.0f/255.0f));
 timer1.start();
 time_data_check = timer1.read_us();
```

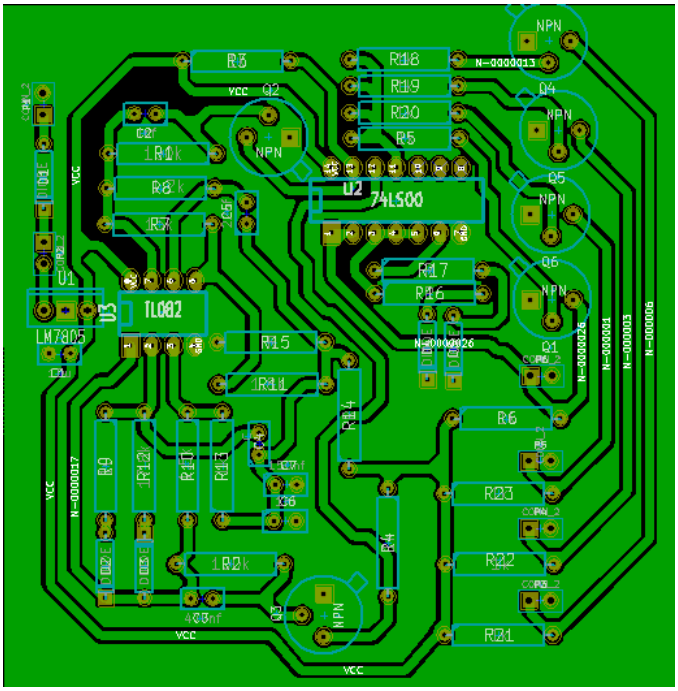
```
 USB_receive_thread = new Thread(USB_recieve);
car_emergency_thread = new Thread(car_emergency);
 while(1) {
   Thread::signal_wait(0x01); //If there is a data;
   STEERING_SERVO_PIN.write(steeringTarget/255.0f);
   THROTTLE_SERVO_PIN.write(throttleTarget/255.0f);
 }
}

/*******************************************/
//Camera_control

void trigger1() {
 time1 = timer2.read_us();
 reach1 = true;
 detect1.disable_irq(); // disable to prevent multi rises
}
void trigger2() {
 time2 = timer2.read_us();
 reach2 = true;
 detect2.disable_irq(); // disable to prevent multi rises
}
void trigger3() {
 time3 = timer2.read_us();
 reach3 = true;
 detect3.disable_irq(); // disable to prevent multi rises
}

void camera_control(void const *argument) {
 // using in a thread
 unsigned int time_r;
 uint8_t combination = 0;
 uint16_t angle_get[2] = {0,0};
 // get angle from the sensor (0 to 360)
 uint16_t angle_set[2] = {1080,1080};
 // set angle (at the middle) to the camera (0 to 2160)
 uint8_t round = 3;
 set_camera(360*3);
 Thread::wait(2000); //set in middle and wait ²
 timer2.start();
 while (true) {
   // ********* RAZ
   time1 = time2 = time3 = 0;
   reach1 = reach2 = reach3 = false;
   timer2.reset();
   // ********* Detection of signal
   detect1.enable_irq(); // allows interrupt
   detect2.enable_irq();
   detect3.enable_irq();
   detect1.fall(&trigger1);
   detect2.fall(&trigger2);
  detect3.fall(&trigger3);
  while( ! (reach1 || reach2 || reach3) ) {
    Thread::wait(2);}
    // Wait for a receive
    Thread::wait(5);
    //5ms more to wait for delayed signals
    detect1.disable_irq(); // for those not reached
    detect2.disable_irq();
    detect3.disable_irq();
    // ********** Treatment
   combination = (reach3 << 2) + (reach2 << 1) + reach1 ;
    //the 3 lasts bits are all the reach's possibilities
    treatement(angle_get[0],combination,round);
    // ********** Add turn parameter
if( ((angle_get[1] - angle_get[0]) > 180) && (round < 6) )
{
round++; //set the number of rotation
}
else if( ((angle_get[1] - angle_get[0]) < -180) && (round
> 0) ) {
round--;
}
  angle_get[1] = angle_get[0]; //memorize the last value
  angle_set[0] = angle_get[0] + 360 * round;
// ********** Smooth
   if(timer2.read() < 1 ) {
     angle_set[0] = angle_set[0] / 4 + angle_set[1] * 3 / 4 ;
   }
  // *********** Set the camera
    set_camera(angle_set[0]);
    Thread::wait(abs(angle_set[1] - angle_set[0]) << 3 );
   // wait for time rotation
   //led = false;
   angle_set[1] = angle_set[0];
   // *********** Regulation, time loop control
   time_r = timer2.read_ms();
   if(time_r < 60) { // wait if less than 50ms loop
   Thread::wait(60 - time_r);
    }
  }
}


void treatement(uint16_t &angle, const uint8_t
&combination, uint8_t &round) {
 switch(combination) {
   case 1 : // DU1 is reached
      angle = 0;
      break;
   case 2 : // DU2 is reached
     angle = 120;
     break;
  case 3 : // DU1 and DU2 are reached
     angle = 60 - SuD * (time1 - time2);
     //determined in the report , 57.3 = 180 / pi
     break;
  case 4 : // DU3 is reached
     angle = 240;
     break;
  case 5 : // DU1 and DU3 are reached
      angle = 300 - SuD * (time3 - time1);
     //determined in the report
```

```
  break;
case 6 : // DU2 and DU3 are reached
  angle = 180 - SuD * (time2 - time3);
  //determined in the report
  break;
case 7 : // DU1 and DU2 and DU3 are reached
   if(time1 >= time2 & time1 >= time3)
      // DU1 is rejected, DU2 and DU3 are reached
      angle = 180 - SuD * (time2 - time3); // as case 6
   else if(time2 >= time1 & time2 >= time3)
      // DU2 is rejected, DU1 and DU3 are reached
        angle = 300 - SuD * (time3 - time1); // as case 5
   else // DU3 is rejected, DU1 and DU2 are reached
         angle = 60 - SuD * (time1 - time2); // as case 3
   break;
case 0 : // no reach
default :
   angle = 0;
      break;
   }
}

void set_camera(const uint16_t &angle) {
   if( angle < 2160) { // between 0 and 2160 (= 6*360°)
      servo = (float(angle) / 43200.0f) + 0.05f;
      //initialy 0.05 for 0T, 0.10 for 6T
   }
}

/*****************************************/
//Main function
int main() {
car_control_thread = new Thread(car_control);
camera_control_thread = new Thread(camera_control);
while(1);
}
```

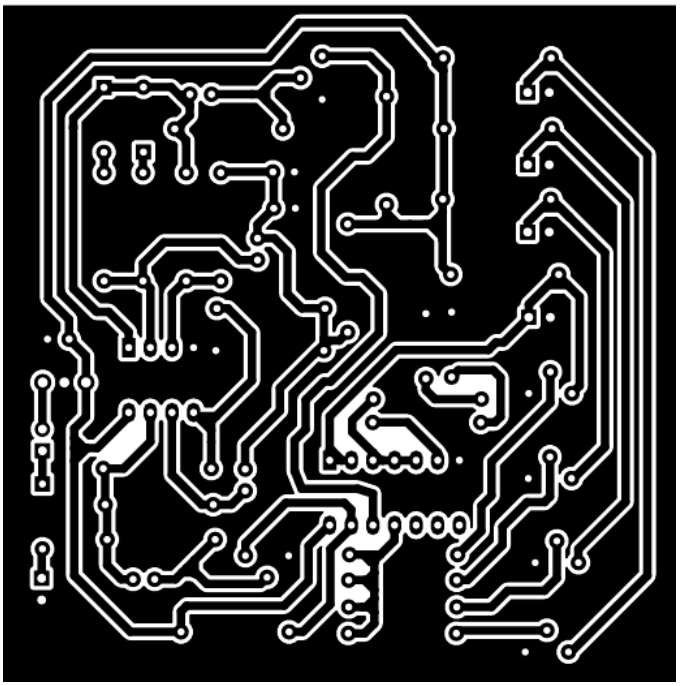
2. Schéma électrique de la carte émetteur



3. PCB de la carte émetteur

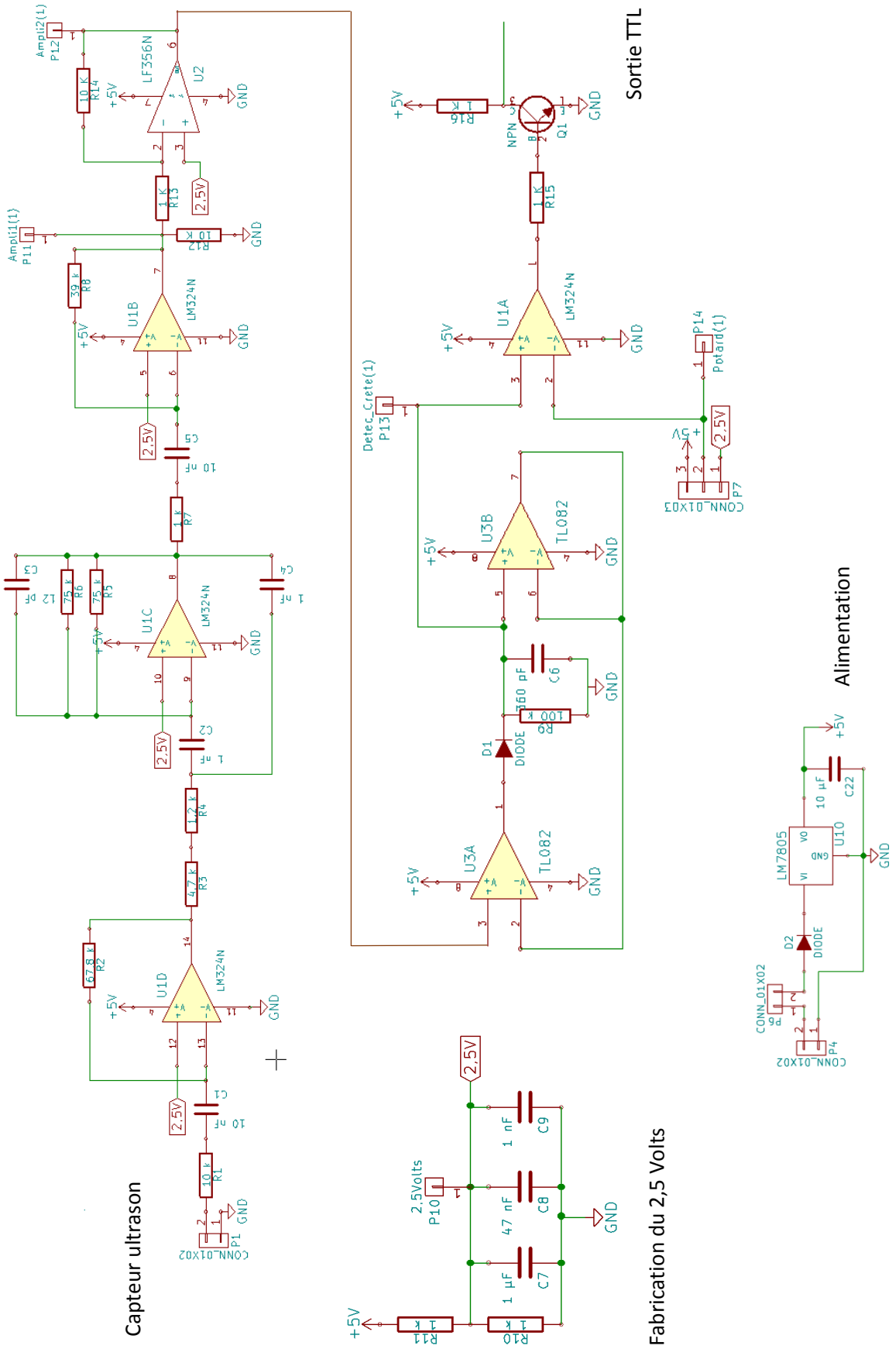


(Vue KiCad)

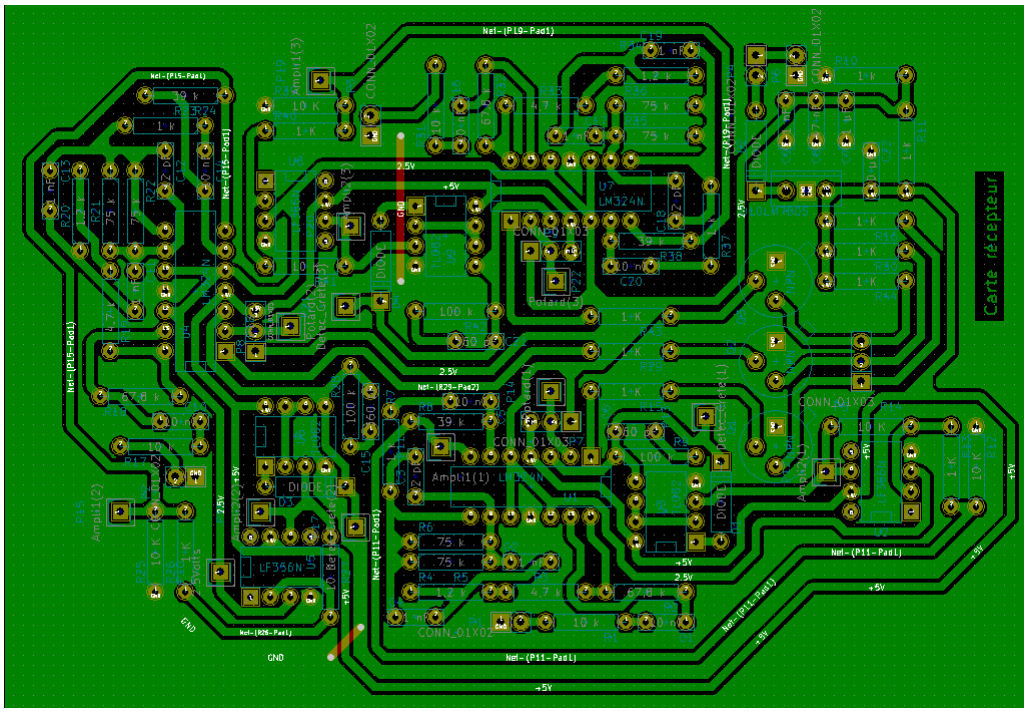


(vue GSview)

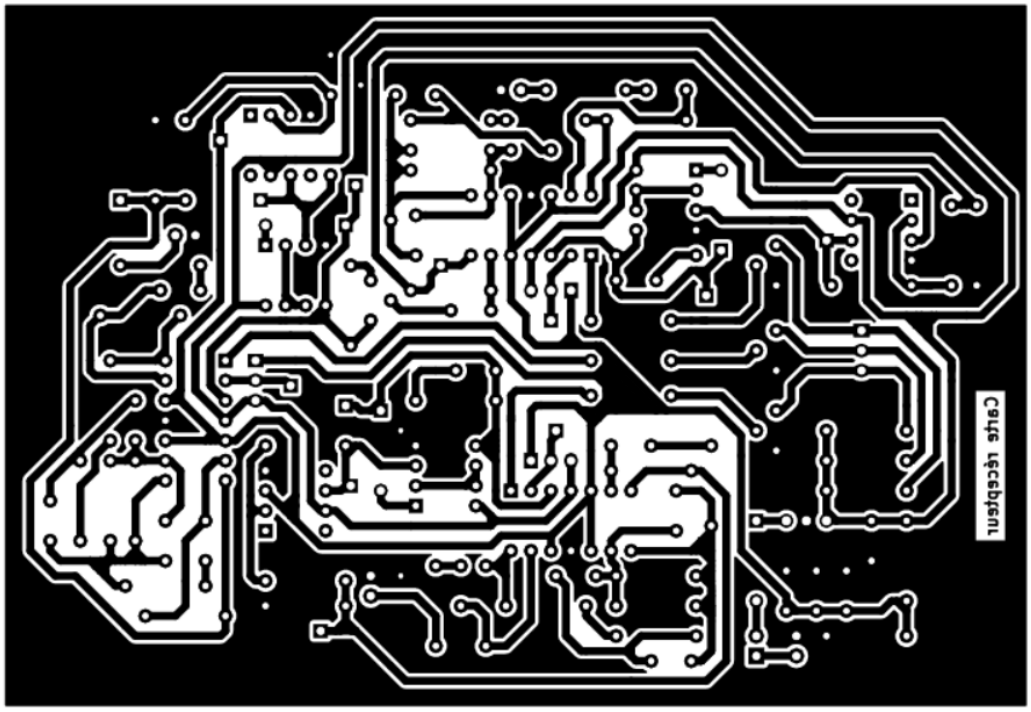
4. Schéma électrique de la carte de réception (une chaine de traitement)



5. PCB de la carte de réception

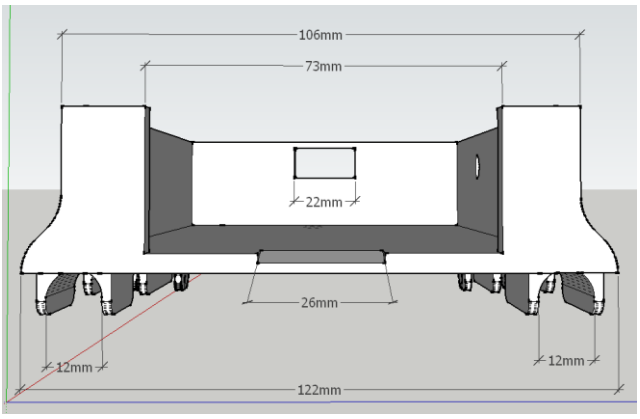


(Vue KiCad)

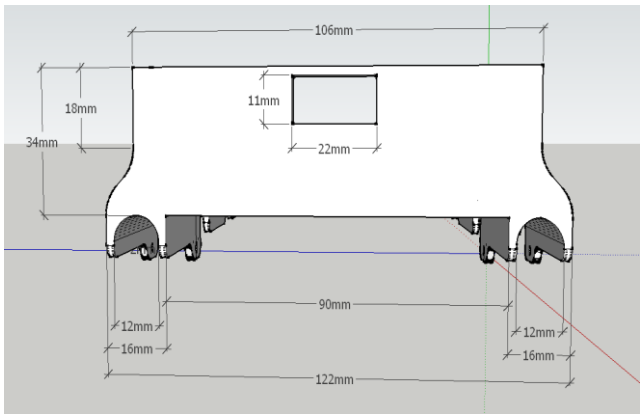


(vue GSview)

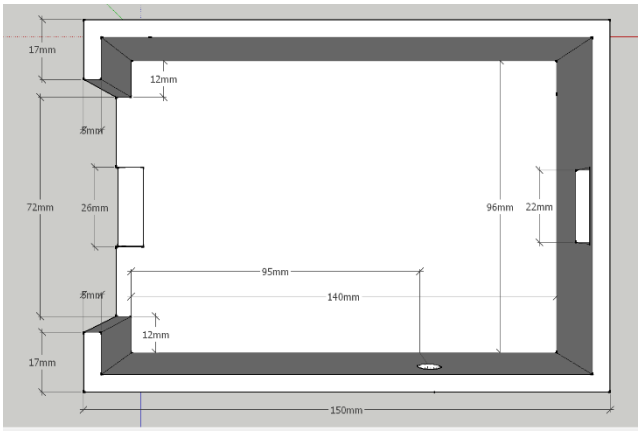
6. Vues et dimensions du support de la carte récepteur



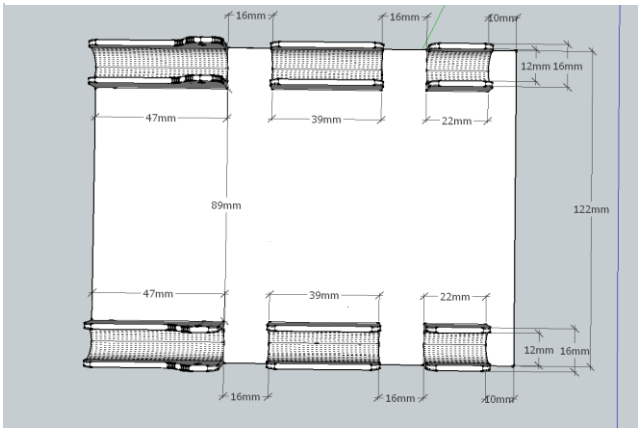
Vue de face



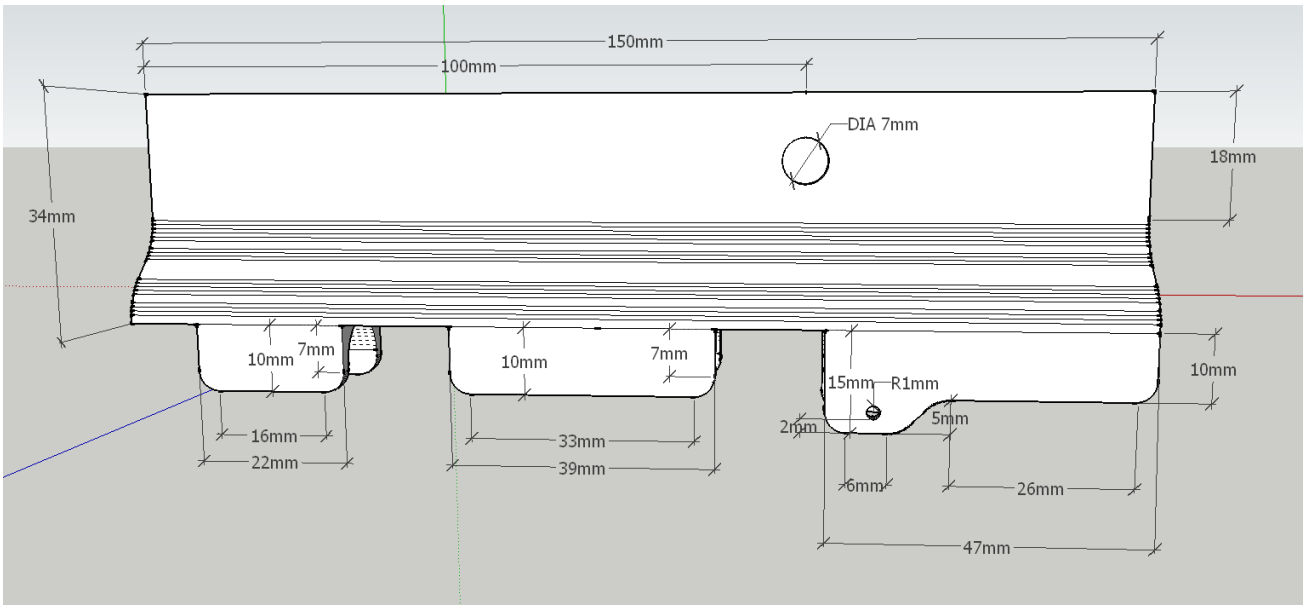
Vue de derrière



Vue de dessus

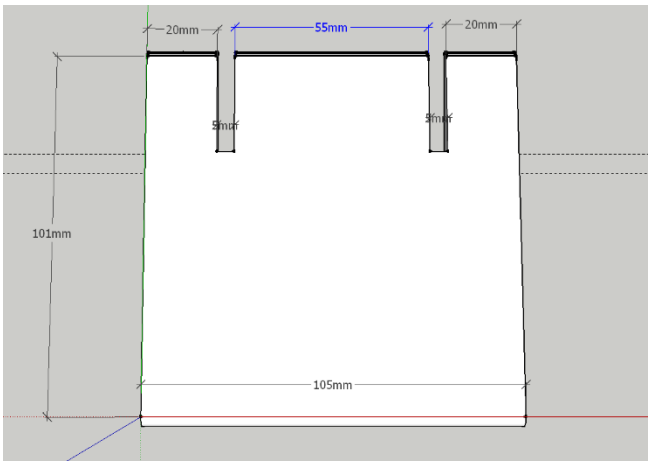


Vue de dessous

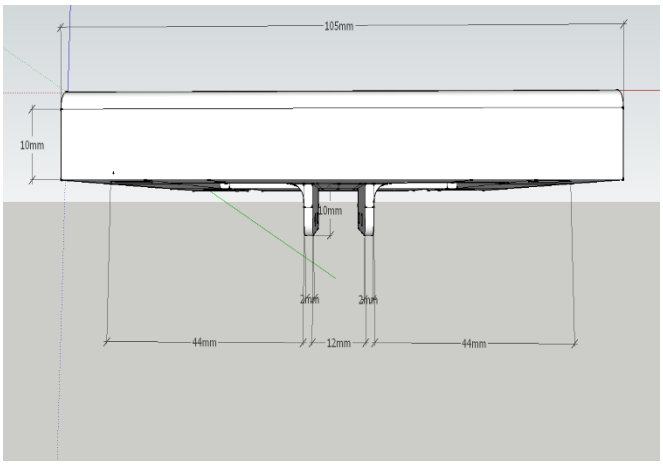


Vue latérale

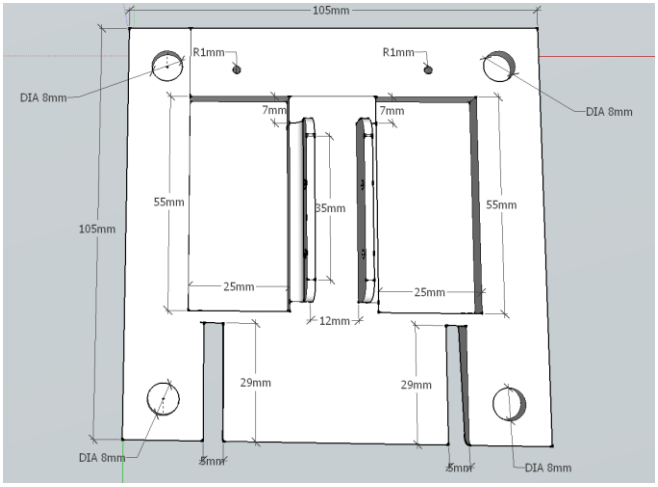
7. Vues et dimensions de la plaque d'adaptation



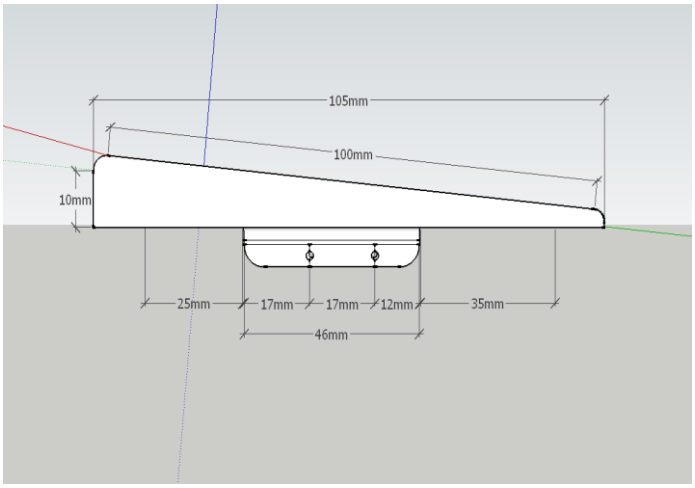
Vue de dessus



Vue de face

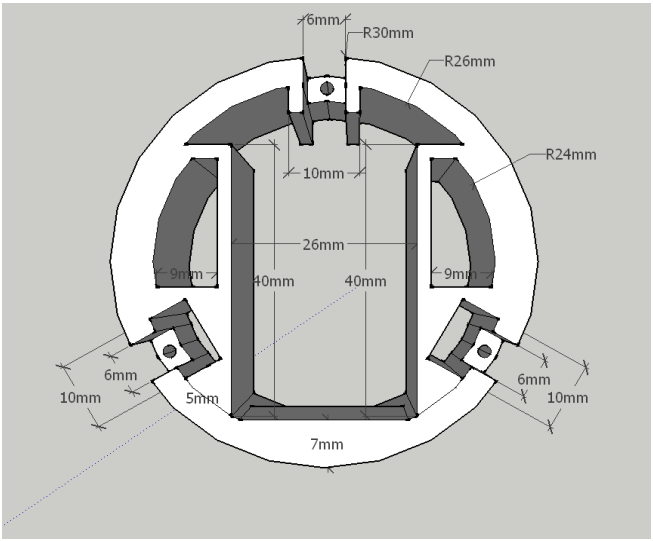


Vue de dessous

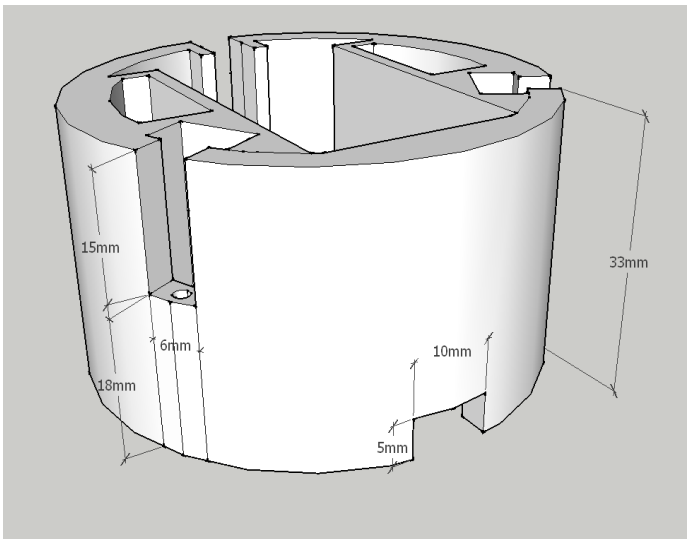


Vue latérale

8. Vues et dimensions de la base cylindrique

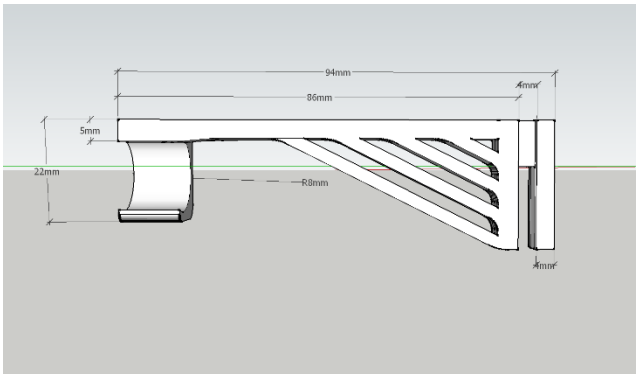


Vue de dessus

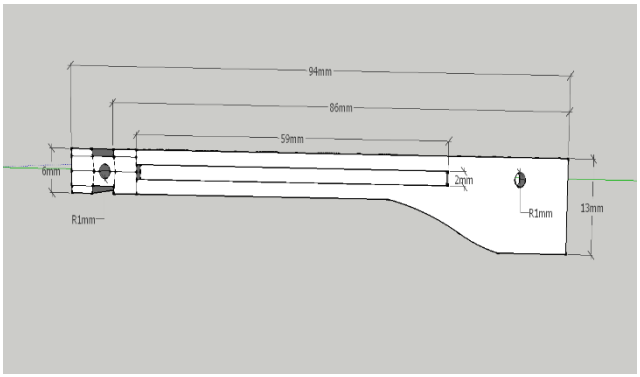


Vue Latérale

9. Vues et dimensions du bras



Vue latérale



Vue de dessus