



Filière génie électrique, orientation électronique

RAPPORT DE PROJET PS5

EUROBOT 2012

Absolute Local Positioning System

Auteur :

Michaël DEFFERRARD

Superviseurs :

Dr. Michaël ANSORGE
Dr. Wolfram LUTHARDT

26 avril 2012

Table des matières

1	Introduction	3
1.1	Plan du document	4
2	Spécifications	5
2.1	Présentation du problème	5
2.2	Description de l'environnement	6
2.3	Description des entrées/sorties	7
2.3.1	Entrées	7
2.3.2	Sorties	7
2.4	Spécifications fonctionnelles	8
2.4.1	Fonction principale	8
2.4.2	Fonction auxiliaire	8
2.5	Spécifications opératoires	8
2.6	Spécifications technologiques	8
2.6.1	Boîtiers	8
2.6.2	Modules et balises	8
2.6.3	Conditions d'utilisation	8
2.6.4	Signalisations optiques	9
2.6.5	Connexions	9
2.7	Procédure de test	9
2.8	Informations complémentaires	9
3	Concept	10
3.1	Principe d'opération	10
3.1.1	Techniques basées sur des balises	10
3.1.2	Émetteur/récepteur directionnel ou omnidirectionnel	11
3.1.3	Vecteur d'information	11
3.1.4	Balises émettrices ou réceptrices	12
3.1.5	Résumé	12
3.2	Balises et modules	13
3.3	Architecture matérielle	14
3.3.1	Optique	15
3.3.2	Barrière optique	15
3.3.3	Batteries	16
3.3.4	Radio et antenne	16
3.3.5	Moteur et contrôleur	16
3.3.6	Électronique module	16
3.3.7	Électronique balise	17
3.4	Signaux	17
3.5	Flux de données	18
4	Signaux	20
4.1	Modulation des signaux	20
4.1.1	Porteuses	20
4.1.2	Balises	20
4.1.3	Attribution des fréquences	21
4.1.4	Paramètres t_0	23
4.2	Spectre complet des fréquences	24

4.3	Génération des signaux	24
4.4	Schéma bloc acquisition et traitement du signal	25
4.5	Chaîne d'acquisition et de conditionnement analogique	26
4.5.1	Photodiode	26
4.5.2	Pré-amplificateur à transconductance	26
4.5.3	Filtre passe-haut et changement de référence	28
4.5.4	Filtre passe-bas anti-repliement et gain programmables	29
4.5.5	Convertisseur analogique / numérique	30
4.5.6	Simulation	31
4.6	Traitement numérique du signal	34
4.6.1	Principe détection et identification balise	34
4.6.2	Transformée de Fourier discrète à fenêtre glissante (SDFT)	34
4.6.3	Simulation Matlab	35
4.6.4	Transformée en Z	39
4.6.5	Analyse de la complexité algorithmique	39
4.6.6	Étude de la stabilité	41
5	Implémentation	42
5.1	Matériel	42
5.1.1	Schéma bloc de l'électronique des modules	42
5.1.2	Mécanique et optique	43
5.2	Logiciel	46
5.2.1	Présentation du MCU / DSP TI F28069	46
5.2.2	Architecture logicielle	48
5.2.3	Coprocasseur DSP (CLA)	50
6	Conclusion	52
6.1	Gestion du projet	53
6.2	Améliorations et travaux futurs	53
6.3	Remerciements	54
	Références	56
A	Code de la simulation Matlab du traitement numérique du signal	57
B	Schéma électronique des modules	61
C	Schéma électronique des balises	63

1 Introduction

Dans le cadre de l'équipe Eurobot 2012 de l'EIA-FR, nous développons un robot autonome capable d'accomplir une mission spécifique qui change chaque année. En principe, le robot se déplace sur une table afin de collecter ou manipuler divers objets. Le concours se déroule en duels, où deux équipes s'affrontent. Cette année, chaque équipe a droit à deux robots. Il peut donc y avoir jusqu'à quatre robots ainsi que divers obstacles sur la table.

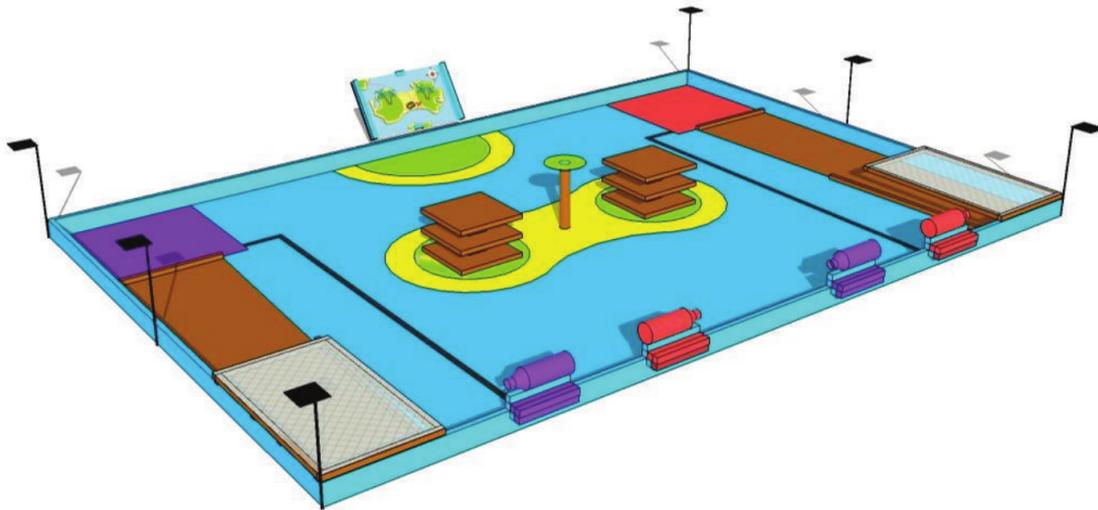


FIGURE 1 – Table Eurobot 2012 [1]

Une tâche qui revient chaque année et qui est indépendante de la mission à accomplir est la localisation du robot et de l'adversaire. En effet, afin d'éviter les obstacles et collisions, et de se déplacer de manière optimale sur le plan de jeu, nous devons en permanence connaître notre position et celle de notre adversaire sur celui-ci. La première approche a été d'introduire l'odométrie, il y a deux ans déjà, à l'aide de roues codeuses sur les roues du robot. Cette technique fonctionne bien mais souffre d'une dérive dans le temps. En effet, l'erreur de mesure est intégrée tout au long du match et devient trop importante vers la fin de celui-ci pour permettre au robot d'accomplir sa mission.

Nous avons donc besoin d'un système capable de mesurer la position actuelle du robot de manière directe. Un système de positionnement global existant (tel le GPS), ne peut être utilisé. Nous devons donc développer notre propre système de positionnement local (LPS : Local Positioning System).

1.1 Plan du document

Dans ce document nous traiterons en premier lieu les spécifications établies en accord avec l'équipe, puis le choix d'un concept de positionnement après étude de l'état de l'art. Suivra un approfondissement de ce concept puis la définition, la génération, la réception et le traitement des signaux choisis. Enfin l'implémentation électronique, mécanique, optique et logicielle du système.

Ce projet ayant été accompli dans le cadre d'un projet de semestre et dans le cadre de ma participation à l'équipe Eurobot de l'EIA-FR, ce document ne décrit qu'une partie du projet représentant le travail de semestre. Certaines parties telle que la conception et réalisation optique et mécanique, l'algorithme de triangulation, la commande moteur, l'alimentation électrique, le contrôleur USB, la communication CAN et radio ainsi que leurs protocoles respectifs ne seront donc pas présentés en détails dans ce rapport.

2 Spécifications

2.1 Présentation du problème

Il convient de créer un système de positionnement absolu fiable, exact et dynamique pour un robot mobile. Ce système sera physiquement composé de deux entités distinctes : les balises et les modules.

Le système disposera de trois balises (actives ou passives) dont les positions sur le terrain sont fixées par le règlement du concours [1]. Chaque module sera capable de déterminer individuellement sa position ainsi que son angle sur le terrain de jeu, tout en étant en mouvement permanent.

Les modules communiqueront entre eux, de sorte que chaque module connaîtra sa position et son angle ainsi que ceux des autres, installés sur robot principal, secondaire, adverse. Il peut y avoir jusqu'à quatre robots sur la table.

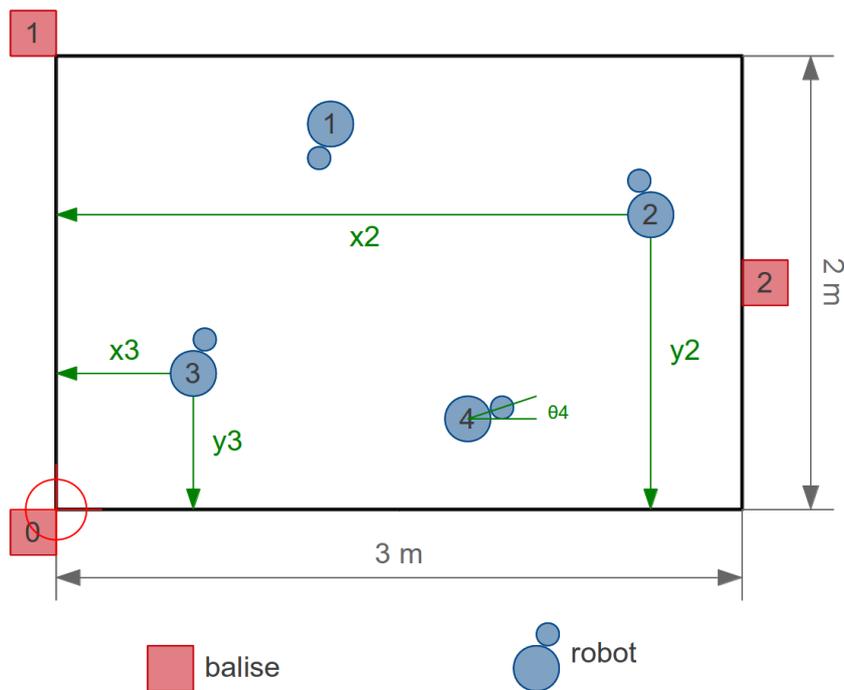


FIGURE 2 – Schéma représentant la localisation présent dans [2]

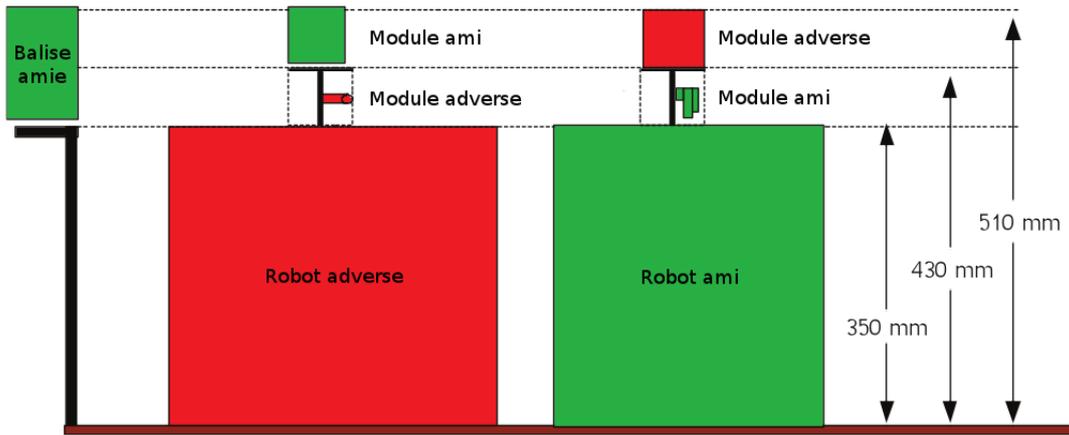


FIGURE 3 – Hauteurs imposées par le règlement du concours [1]

Nous voyons sur la FIGURE 3 que l’alignement des balises et des modules est garanti par le règlement du concours [1]. De même, les robots et objets sont limités en hauteur, nous permettant une ligne de vue directe.

2.2 Description de l’environnement

Les interactions entre le système à concevoir et son environnement sont modélisées ainsi :

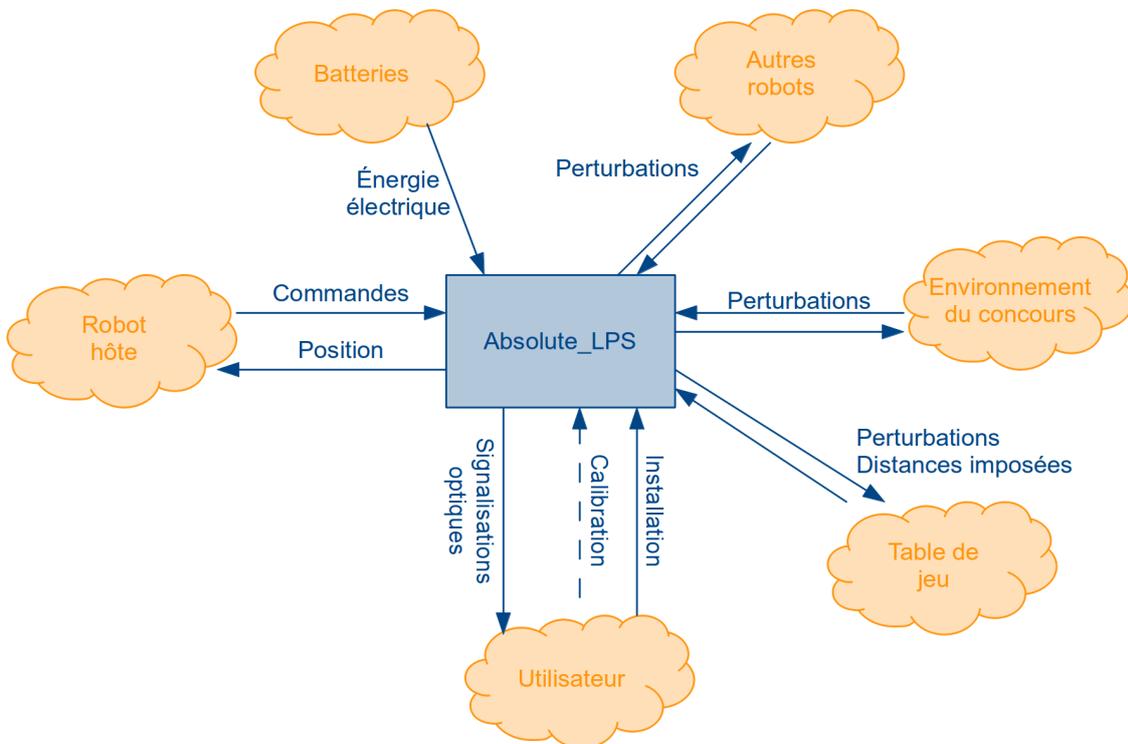


FIGURE 4 – Modélisation de l’environnement de l’Absolute_LPS

Diverses entités interagissent :

Batteries : Elles fournissent l'énergie électrique. Soit directement dans le cas des balises et modules autonomes, soit par l'intermédiaire de l'alimentation du robot hôte.

Robot hôte : Robot sur lequel un module est installé. Nos robots (pas les robots adverses) communiquent avec leur module afin d'acquérir leur position ainsi que celles des autres robots portant un module.

Utilisateur : Il installe les parties mobiles du système de positionnement avant le début du concours, procède si nécessaire à une calibration et reçoit des informations sur l'état du système par l'intermédiaire de signalisations optiques.

Table de jeu : Elle nous impose la position des balises ainsi que les positions possibles des robots. Elle peut également perturber notre système, par exemple à causes des réflexions sur les éléments de jeu.

Environnement du concours : Il apporte également son lot d'interférences : lumière ambiante, communications Wifi etc. Notre système perturbera également cet environnement.

Autres robots : Les autres robots présents vus du robot hôte. Ici aussi il y a échanges de perturbations, notamment dues aux systèmes de communication, positionnement etc. Il y a des limites à ne pas dépasser, voir le règlement [1].

2.3 Description des entrées/sorties

2.3.1 Entrées

Énergie électrique : Alimentation DC nécessaire au fonctionnement du système.

Commandes : Elles sont envoyées par le robot hôte pour demander une information de position ou configurer le module.

Installation : Elle intervient au début du match, lorsqu'un membre de l'équipe installe les parties mobiles du système : balises et modules sur robots adverses.

Calibration : Elle intervient après l'installation. Idéalement elle devrait ne pas être nécessaire ou être automatique. Cette entrée n'est pas définitive et dépend de l'implémentation qui sera choisie.

Perturbations : Elles peuvent être électromagnétiques (ondes radio, lumière etc.), acoustiques (notamment ultrason) ou autres. Elles devront être prises en compte et ne pas altérer le bon fonctionnement du système de positionnement.

2.3.2 Sorties

Position : Retourne au robot hôte la position et l'angle actuel de tous les modules présents sur la table de jeu.

Signalisations optiques : Indique à l'utilisateur l'état dans lequel se trouve l'Absolute_LPS.

Perturbations : Elles peuvent être électromagnétiques (ondes radio, lumière etc.), acoustiques (notamment ultrason) ou autres. Voir le règlement [1] pour les limites.

2.4 Spécifications fonctionnelles

2.4.1 Fonction principale

La fonction primaire de l'Absolute_LPS est de retourner au robot hôte sa position absolue et son angle sur la table de jeu lorsque celui-ci la demande. Il devra également lui fournir la position et l'angle des autres robots présents sur lesquels un module est installé, notamment les adversaires.

2.4.2 Fonction auxiliaire

Une signalisation optique indiquera à l'équipe opérant le robot dans quel état se trouve ce sous-système. Ceci afin de détecter si l'installation en début de partie a été faite correctement. Par exemple : Power ON, non-calibré, calibration en cours, prêt. Cette liste est à adapter selon l'implémentation choisie.

2.5 Spécifications opératoires

La résolution de la position sera égale ou inférieure à un centimètre. La résolution angulaire sera quant à elle égale ou inférieure à un degré. Le taux de rafraîchissement sera au minimum de 10 Hz. La distance possible entre un module et une balise est de zéro à quatre mètres. Le système complet pourra être installé et calibré (si nécessaire) en moins de trois minutes, temps mis à disposition pour cette tâche avant le début du duel. L'autonomie sera au minimum d'une dizaine de duels (trois minutes de préparation et nonante secondes de jeu).

2.6 Spécifications technologiques

2.6.1 Boîtiers

Les balises doivent être contenues dans un volume de 8x8x16 centimètres (Lxlxh). Les modules auront une taille maximale de 8x8x8 centimètres (Lxlxh). Les balises seront fixées sur leur support par du velcro. Les modules qui seront installés sur les robots adverses également. Les modules de nos robots seront quant à eux fixés à l'aide de vis.

2.6.2 Modules et balises

Toutes les balises ainsi que tous les modules seront physiquement identiques, de sorte qu'il sera possible de les permuter aisément en cas de panne. Une différenciation par DIP switch ou autre méthode sera nécessaire pour discriminer leur lieu d'installation. Au moins une balise et un module supplémentaire seront livrés à cet effet. Le système final comprendra donc cinq modules et quatre balises.

2.6.3 Conditions d'utilisation

Le système sera utilisable en intérieure, à température et humidité standards.

2.6.4 Signalisations optiques

Les diverses signalisations optiques seront implantées à l'aide de LEDs de couleur sur les modules et/ou les balises. Un label permettra d'identifier rapidement ce qu'elles indiquent.

2.6.5 Connexions

L'interconnexion entre l'Absolute_LPS et le robot hôte se fera à l'aide d'un bus CAN. Le protocole de communication a été défini au sein de l'équipe et ne sera pas présenté dans le cadre de ce rapport. L'alimentation fournie aux modules par l'intermédiaire de nos robots est de 5V DC (par l'intermédiaire du bus CAN) et de 9V DC (puissance, lignes de masse et VCC). La connexion avec les batteries (pour les balises et les modules sur robots adverses) est libre.

2.7 Procédure de test

1. Les balises sont installées à leur emplacement prévu, au bord du terrain de jeu.
2. Les modules sont installés sur le ou les robots adverses.
3. Une calibration manuelle est exécutée si nécessaire.
4. Ces trois opérations doivent pouvoir être exécutées en moins de trois minutes.
5. Le robot hôte doit pouvoir acquérir la position et l'angle de tous les modules présents sur la table de jeu avec la précision et le taux de rafraîchissement demandés sur toute la surface de la table, même lorsqu'ils sont en mouvement.

2.8 Informations complémentaires

Il n'y a pas de restriction de budget. Cependant les dépenses doivent rester en accord avec les financements disponibles pour le robot.

Un prototype devrait être livré pour la Coupe de Belgique afin que l'équipe puisse travailler avec une information de position, notamment pour le développement et les tests de stratégie. Un démonstrateur fonctionnel sera livré pour des tests en conditions réelles lors de la Coupe de Belgique, les 14 et 15 avril 2012. Le système final sera quant à lui prêt pour la Coupe de Suisse, les 4 et 5 mai 2012.

Une documentation concise sera fournie pour la prise en main et l'installation rapide du système.

3 Concept

Ce chapitre présente les concepts retenus pour la réalisation du sous-système de positionnement du robot de l'équipe RTFM (Robot Team Fribourg Morat) qui participera aux concours Eurobot 2012. Cette implémentation respecte les spécifications précédemment établies.

3.1 Principe d'opération

Après étude de la littérature dans le domaine du positionnement local, il ressort plusieurs techniques principales. Nous allons les analyser et choisir la plus adaptée à nos besoins.

Il existe des systèmes qui ne reposent pas sur des balises actives, tel que l'analyse de scène (avec ou sans cibles optiques), mais ils sont moins fiables, plus lents et/ou moins précis. Ils ont été développés pour opérer là où il est difficile d'implanter des balises. Ce n'est pas notre cas, nous avons trois supports de balises à disposition et un temps de préparation suffisant pour les installer. Voyons donc ces différentes méthodes.

3.1.1 Techniques basées sur des balises

Il existe plusieurs techniques permettant de retrouver sa position à partir de balises dont l'emplacement est connu d'avance. Nous allons en citer trois ainsi que leurs possibles implémentations et ferons rapidement le point pour chacune d'entre elles.

Trilatération : TOF (Time Of Flight), TOA (Time Of Arrival), TDOF (Time Difference Of Flight), TDOA (Time Difference Of Arrival)

Basé sur la vitesse : FDOA (Frequency Difference Of Arrival), DD (Differential Doppler)

Triangulation : AOA (Angle Of Arrival), DAOA (Differance Angle Of Arrival)

La trilatération est basée sur une mesure de (différences de) distances entre l'objet à localiser et les balises fixes. Le GPS utilise par exemple l'une de ses méthodes. Leur principal désavantage dans notre cas est de ne pas fournir d'information d'angle. Il est néanmoins possible d'obtenir un angle de déplacement de manière indirect par dérivation du chemin parcouru. Cette technique peut être suffisante pour un objet qui ne se déplace que dans une direction. Ce n'est pas applicable à notre robot qui dispose d'une base holonomique lui permettant de se déplacer dans toutes les directions. Une solution possible serait de déterminer la position de deux endroits distincts du robot et de calculer l'angle après coup.

Une autre méthode est l'utilisation de l'effet Doppler : on émet à une fréquence connue et on mesure la fréquence reçue. La différence nous permet de retrouver la vitesse de l'objet puis la distance par intégration. Cette technique est adaptée aux corps célestes mais très peu à notre robot qui travail à basse vitesse.

La triangulation est basée sur une mesure d' (différence d') angles entre des balises fixes observées depuis l'objet à localiser. Pour peu qu'elle soit AOA, la triangulation fournit une information d'angle.

Après évaluation de ces différentes méthodes, la triangulation semble la plus adaptée à notre problème.

3.1.2 Émetteur/récepteur directionnel ou omnidirectionnel

Pour garder la communication sur l'ensemble du terrain de jeu les éléments fixes (balises) doivent être omnidirectionnels, qu'ils soient émetteurs, récepteurs ou les deux. Pour obtenir une information d'angle, les éléments mobiles doivent donc être directionnels.

3.1.3 Vecteur d'information

Nous avons donc besoin d'un vecteur d'information qui puisse être émis ou reçu de manière directionnelle. Si la réception est directionnelle alors l'émission doit être omnidirectionnelle et vice-versa.

Deux types d'ondes sont à notre disposition : les ondes sonores et les ondes électromagnétiques. Les ondes sonores sont éliminées car elles subissent beaucoup de réflexions et sont d'ailleurs utilisées pour les capteurs de proximité de notre robot. Il vaut également mieux éviter de nous perturber nous-même. Les ultrasons conviennent très bien pour des mesures de distances mais semblent peu utilisés par les systèmes de positionnement.

Reste les ondes électromagnétiques :

Basses fréquences : Peu étudié, une solution développée par le Massachusetts Institute of Technology. [7]

Radio fréquences : Émission ou réception directionnelle difficile (éventuellement grille d'antennes avec mesure des différences de phase).

UWB (Ultra Wide Band) : Technologie prometteuse, plusieurs publications et solutions développées. [4] [10] [8] Uniquement exploitable par trilatération.

Infrarouge : L'émission et la réception peuvent être directionnelles ou omnidirectionnelles.

Lumière visible : Idem à l'infrarouge.

Ultraviolet : Idem.

Les ondes lumineuses semblent bien adaptées. Le règlement du concours [1] interdit cependant d'utiliser les couleurs présentes sur la table de jeux, éléments fixes ou mobiles. Cette restriction réduit fortement les couleurs disponibles pour l'utilisation de lumière visible, d'autant plus que la table et les couleurs la composant changent chaque année. L'ultraviolet est peu répandu pour ce type d'application et n'a pas été étudié plus en détail.

La solution la plus adaptée semble alors être l'infrarouge. Elle peut éventuellement être perturbée par des sources de chaleur ou de l'éclairage à incandescence. Dans tous les cas les principales sources de perturbations proviendront des robots adverses. Le système devra impérativement être protégé contre toute perturbation externe.

3.1.4 Balises émettrices ou réceptrices

Nous avons donc un vecteur d'information pouvant être émis ou reçu de manière directionnelle. L'émission directionnelle par laser est proscrite due aux difficultés d'homologation des lasers pour le concours. Nous avons donc une émission omnidirectionnelle et une réception directionnelle.

Pour une communication unidirectionnelle, l'un des deux éléments est émetteur, l'autre récepteur. Puisque les balises doivent être omnidirectionnelles, elles seront émettrices. Les éléments mobiles seront donc récepteurs.

3.1.5 Résumé

Beaucoup de choix se sont offerts à nous, il a fallu en faire un. Nous avons donc une communication unidirectionnelle : balises actives omnidirectionnelles émettrices d'infrarouge et éléments mobiles actifs directionnels récepteurs d'infrarouge.

3.2 Balises et modules

La table de jeu telle que définie par le règlement [1] du concours. Les proportions ne sont pas respectées. Nous distinguons :

Les balises : Elles envoient en continu un signal infrarouge sur toute la table. Leur position est fixe et déterminée par le règlement [1]. Leur volume de 8 x 8 x 16 cm est également fixé.

Les modules : Ils sont montés sur les robots, reçoivent les signaux infrarouges de manière directionnel et déterminent leur localisation : position et orientation. Ils communiquent par radio afin que chaque module connaisse la localisation de tous les modules. Lorsque le robot hôte est ami, ces informations seront transmises à celui-ci par bus CAN. Leur volume de 8 x 8 x 8 cm est également fixé.

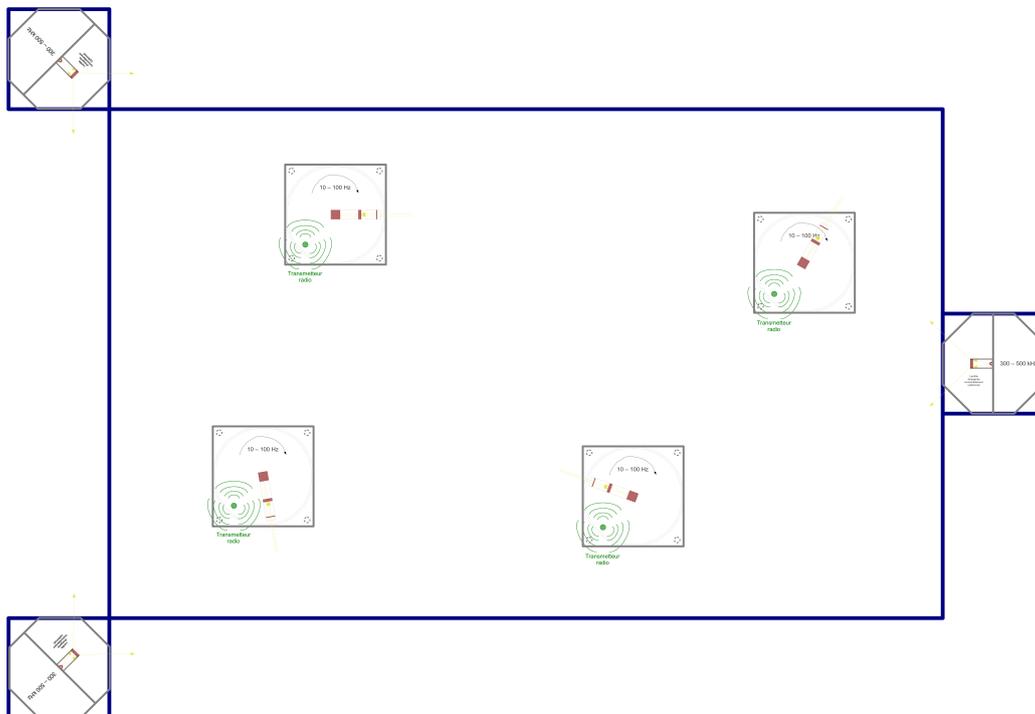


FIGURE 5 – Schéma de la table de jeu

Les balises ont une base octogonale afin de s'aligner facilement sur leur support quel que soit leur emplacement. Elles sont toutes physiquement identiques et leur position peut être assignée facilement. Elles ne sont pas synchronisées et ne communiquent pas entre elles.

Les modules sont également identiques et configurables : robot principal, robot secondaire, robot adverse principal, robot adverse secondaire etc.

La partie optique des modules tourne sur elle-même, à une fréquence allant de 10 à 100 Hz, permettant une dynamique suffisante vis-à-vis de la rapidité des mouvements du robot. Le système détecte lorsque l'optique *pointe* sur une balise et calcule les différences d'angle. Ces angles sont exploités par un algorithme de triangulation afin de déterminer la localisation.

3.3 Architecture matérielle

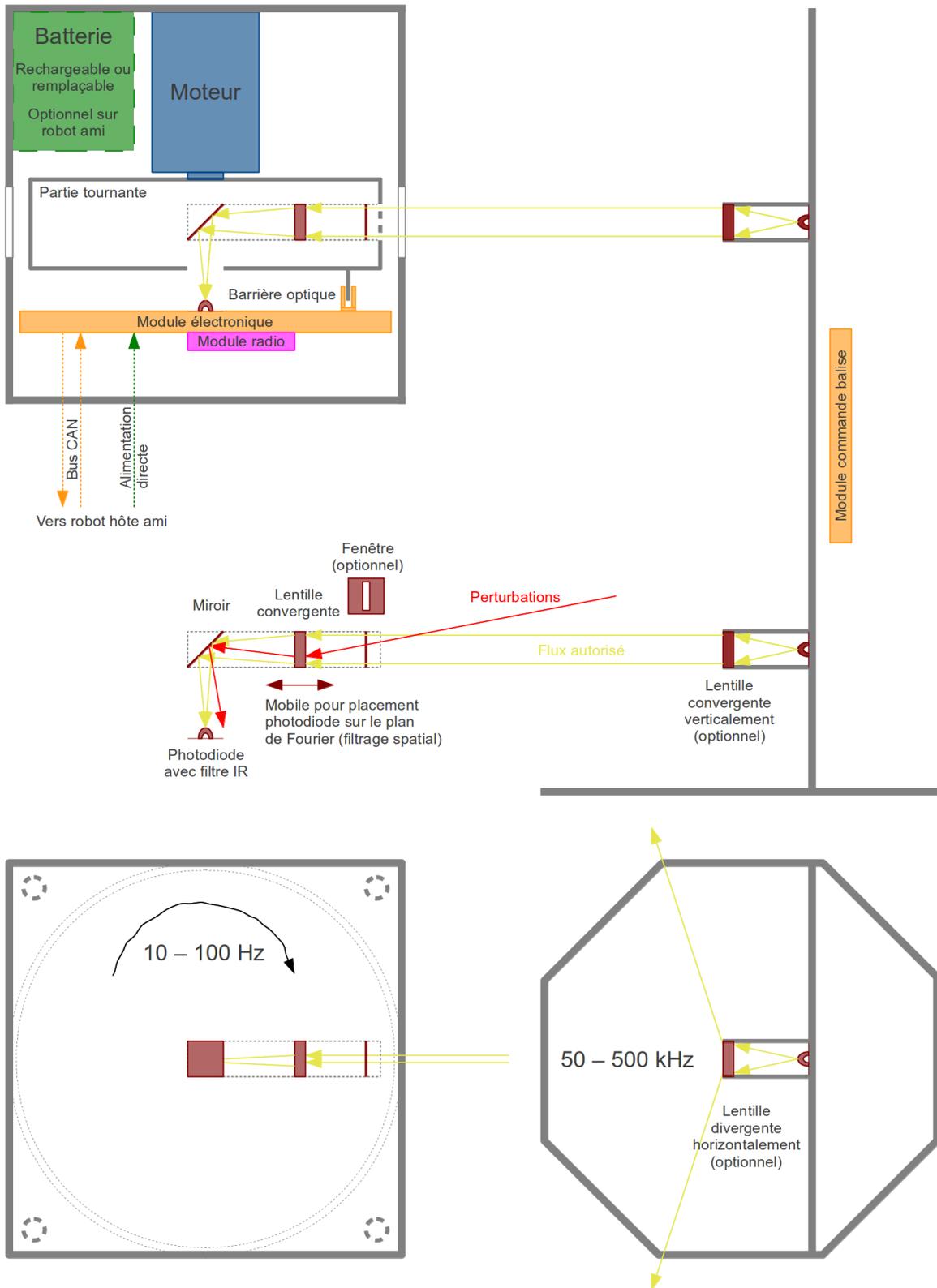


FIGURE 6 – Architecture matérielle complète

3.3.1 Optique

La partie optique est inspirée du système réalisé par une équipe belge présenté dans ce document [6]. Plusieurs modifications et améliorations y ont été apportées.

Fenêtre : Une fenêtre à l'entrée du système optique permettrait de recevoir plus de lumière sur le plan vertical que sur le plan horizontal. Ceci afin de permettre une tolérance de fabrication verticale et une plus grande directivité horizontale. N'a pas été implémenté pour l'instant.

Modulation : La lumière émise par chacune des balises est modulée de manière unique ce qui permet de les identifier sans synchronisation. La modulation permet également de discriminer nos balises du bruit IR.

Récepteur directif : Lorsque le récepteur est en face d'un émetteur, il ne doit voir que celui-ci afin que le bruit IR (par exemple d'une puissante lampe à incandescence) ne puisse pas le couvrir. Nous utilisons pour cela un filtre spatial. Une lentille convergente dirige tous les rayons perpendiculaires à elle-même sur sa focale. La photodiode IR, qui est placée sur le plan de Fourier (où se trouve la focale), fait office de filtre passe-bas sur les fréquences spatiales. Le miroir plan sert à rediriger le flux lumineux à l'extérieur de la partie tournante. La lentille fait également office d'amplificateur.

Lentille sur balise : Elle devrait idéalement être convergente sur le plan vertical afin d'y concentrer les rayons et divergente sur le plan horizontal afin de les disperser sur une surface maximale. La hauteur du flux dépend de la place disponible dans les modules, de l'ordre de 2cm, permettant une tolérance d'alignement suffisante. Un réflecteur conique remplirait également cette tâche. Cette lentille n'est pas implémentée à cause de son coût et des difficultés supplémentaires de montage. Une LED IR grand angle peut la remplacer, malgré la dispersion et donc les pertes sur le plan vertical.

Filtres polarisants : La polarisation de la lumière change en cas de réflexion. Ces filtres éliminent les rayons reflétés mais tout de même perpendiculaire des LEDs IR. Ces filtres ne seront pas implémentés car ils sont trop chers pour un risque négligeable, la puissance des rayons reflétés étant nettement plus faible que les originaux.

Fréquence de modulation et de rotation

La relation entre la fréquence de modulation de la LED IR et la vitesse de rotation est décrite par la relation suivante :

$$f_{modulation} = \frac{360^\circ}{\text{ouv}_{angulaire}} \cdot N_{periodes} \cdot f_{rotation} \quad (1)$$

Elle dépend de l'ouverture angulaire du récepteur directif et du nombre de périodes que nous voulons pouvoir mesurer afin d'identifier la balise émettrice.

3.3.2 Barrière optique

Elle est utile comme référence (0°) pour la détermination de l'orientation du robot ainsi que le contrôle de la fréquence de rotation.

3.3.3 Batteries

La batterie devra alimenter le circuit électronique, un transmetteur radio ainsi qu'un moteur. La consommation de ses parties est à définir. Sur nos robots (pas les adverses), cette batterie pourra être remplacée par une alimentation directe.

La batterie des balises alimentera le petit module de contrôle ainsi que les LEDs IR.

Un interrupteur permettra de découper les batteries afin de les économiser. L'autonomie devra être de plusieurs matchs.

3.3.4 Radio et antenne

Un émetteur / récepteur (transceiver) sera acheté. Le débit nécessaire reste à définir mais il est faible et ne sera pas déterminant. Le temps de transmission sera quant à lui déterminant afin de ne pas introduire une latence inacceptable. Distance maximale de cinq mètres et consommation minimale.

L'antenne sera préférablement intégrée au module radio. Sinon il est possible d'utiliser une antenne omnidirectionnelle monopole (quart d'onde) orientée pour une polarisation verticale. La polarisation verticale est idéale pour la communication avec les autres modules se trouvant sur le même plan horizontal. L'antenne est placée vers le centre du module afin de rayonner de manière optimale. Le cylindre extérieur du module sera réalisé en plexiglas pour ne pas gêner la transmission. Seule la partie tournante ainsi que les plaques carrées haute et basse du module seront réalisées en aluminium.

La bande de fréquences sera de 2.4 ou 5 GHz pour une taille d'antenne minimale. Une modulation de type spread spectrum : FHSS (Frequency Hop Spread Spectrum) ou DSSS (Direct Sequence Spread Spectrum) est préférée aux traditionnelles modulations narrowband pour une robustesse accrue.

3.3.5 Moteur et contrôleur

Le moteur (probablement de type DC brushed) fait tourner la partie optique. Il doit être capable d'une rotation régulière de 10 à 100 Hz. Un contrôleur externe pourrait être nécessaire à une régulation fiable. Sa consommation et son volume devront être minimaux.

3.3.6 Électronique module

Ce module comportera une partie analogique pour l'acquisition des signaux et une partie numérique pour le traitement de ceux-ci.

Les tâches à accomplir sont les suivantes :

- Acquisition des signaux provenant de la barrière optique et de la photodiode.
- Filtrage, amplification, numérisation et traitement de ces signaux.
- Identification des balises.
- Calculs d'orientation et de position.
- Éventuellement régulation de la vitesse de rotation du moteur.
- Collecte des données provenant des modules montés sur les différents robots.

- Interfaçage avec le robot hôte (bus CAN).
- Gestion de l'alimentation électrique.

3.3.7 Électronique balise

Ce module commande les LEDs IR. Il doit être capable de générer la modulation choisie. Aucune synchronisation nécessaire entre balises.

3.4 Signaux

Il y a deux signaux d'entrées :

Barrière optique : Donne un pic (positif ou négatif) lorsqu'une découpe interrompt le flux lumineux à chaque tour du moteur. Permet de mesurer la référence 0° du robot. Rends également possible une mesure de la vitesse de rotation du moteur.

Récepteur IR : Signal proportionnel à l'intensité des ondes IR l'atteignant. À noter que la modulation représentée sur la FIGURE 7 n'est pas utilisée. Elle est représentée ainsi pour nous permettre une discrimination aisée des modulations uniques de chaque balise.

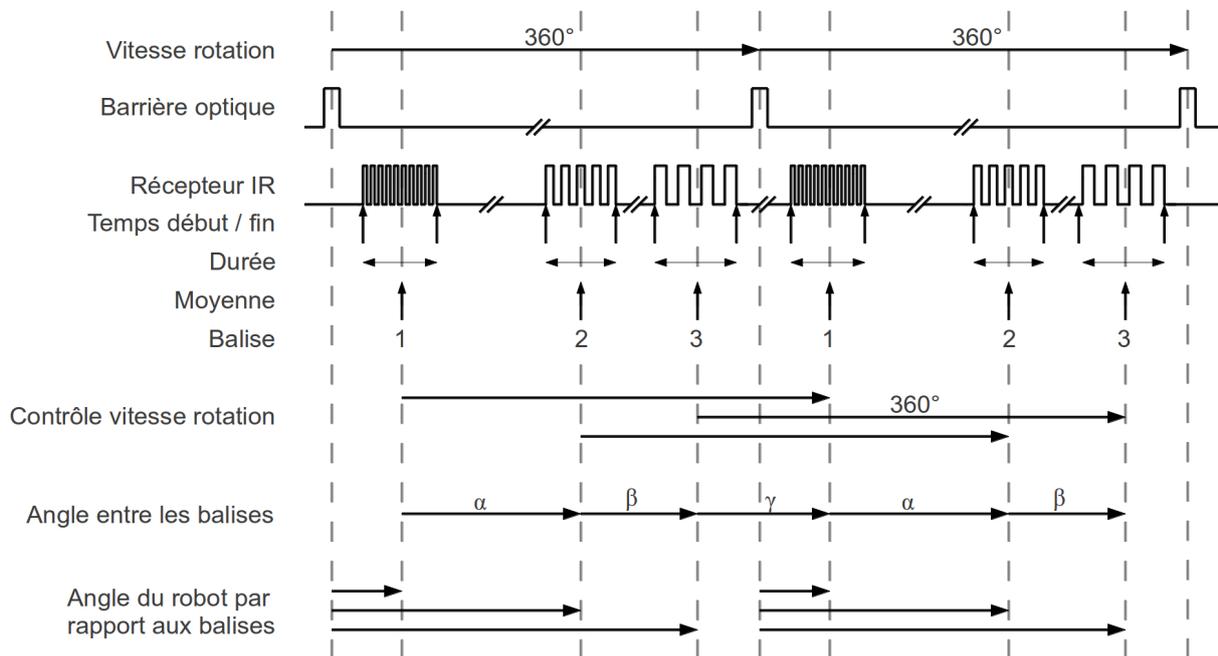


FIGURE 7 – Signaux reçus et interprétations

Les impulsions de la barrière optique permettent de mesurer la vitesse de rotation du moteur.

Le signal infrarouge émis par chaque balise est modulé de manière unique. Voir section 4.1 pour les détails.

On mesure le temps absolu de début et de fin du burst. La moyenne nous donne le temps absolu représentant le moment où nous sommes en face d'une balise.

L'analyse de la durée d'un burst permet de vérifier que le signal provient d'une balise et non d'une réflexion ou d'un autre émetteur d'infrarouge.

On analyse la modulation des bursts afin de déterminer de quelle balise ils proviennent.

La mesure du temps séparant la réapparition d'une même balise permet de vérifier la cohérence entre vitesse de rotation du moteur et vitesse de rotation réelle. Un déplacement ou une rotation rapide du robot entraînerait une incohérence.

Les différences relatives entre les temps absolus d'apparition des balises représentent les angles entre elles. Un algorithme de triangulation permet d'en déduire notre position et orientation sur le terrain de jeu.

La différence entre un pic de la barrière optique et le temps absolu d'apparition d'une balise représente son décalage par rapport à la référence (0°) du robot.

3.5 Flux de données

Le diagramme de la bscFigure 8 représente le flux des données depuis leur acquisition par les capteurs physiques jusqu'à leur envoi au robot hôte. Ces données subissent divers traitements et transformations. Les textes en *italique* représentent des actions qui ne seront pas implémentées en priorité mais qui ajoutent une robustesse supplémentaire au système.

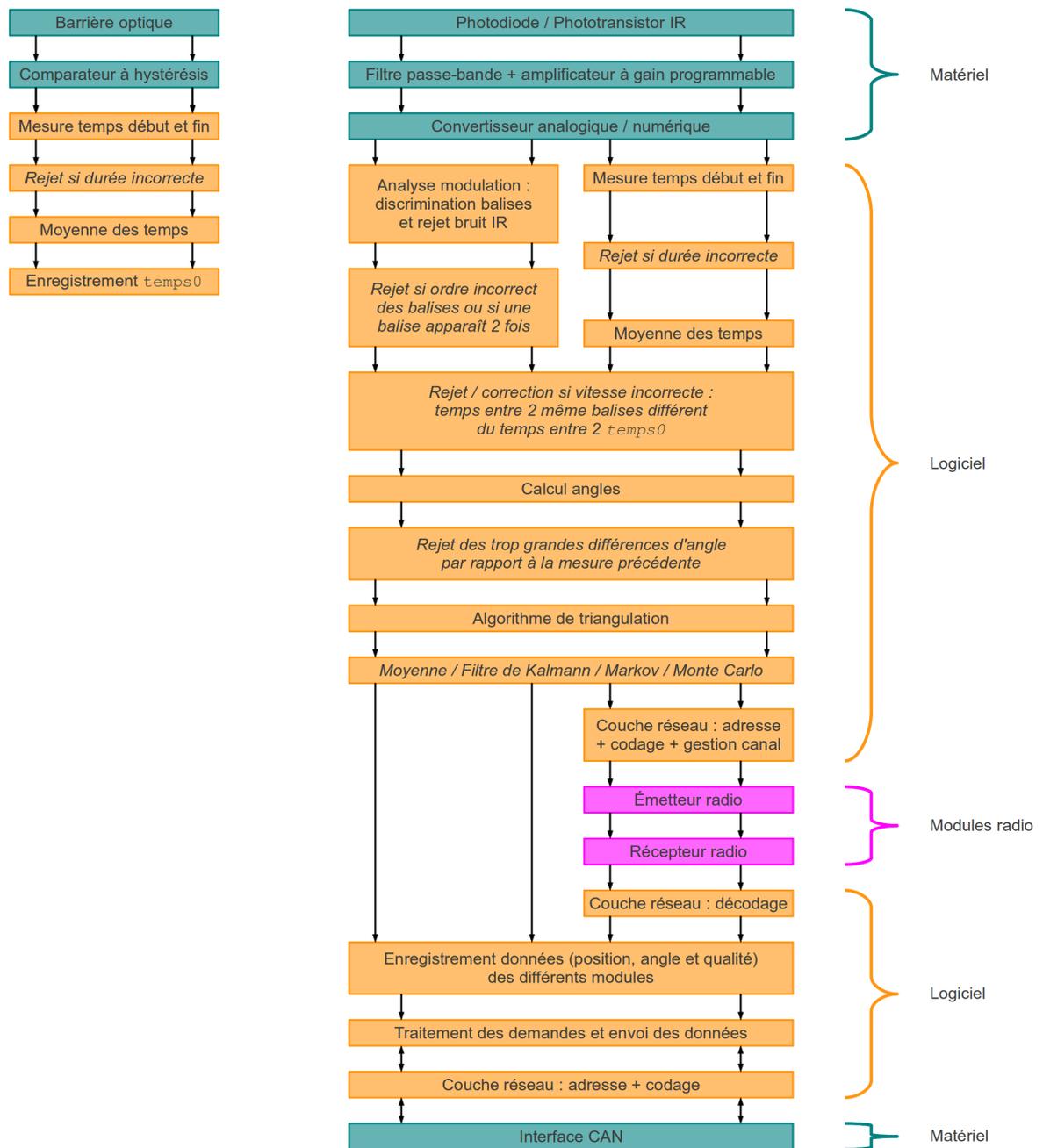


FIGURE 8 – Représentation des flux de données

Suivant le prix de la plate-forme matérielle nécessaire à l’implémentation de l’algorithme de triangulation, la transmission radio pourrait être déplacé avant ce calcul. Elle transmettrait donc uniquement les informations d’angle. La localisation serait calculée après réception uniquement par les modules équipés de cette plate-forme. Il est cependant préférable que les PCB restent identiques.

4 Signaux

4.1 Modulation des signaux

4.1.1 Porteuses

Nous utiliserons une modulation DTMF¹ pour différencier les balises. Pour avoir au moins trois paires de fréquences (une par balise), nous avons besoins d'au moins trois porteuses. Nous en utilisons le minimum afin de diminuer au maximum la complexité du traitement numérique.

$$x_{p0}(t) = A \cdot \sin(2\pi f_0(t - t_{00})) \quad (2)$$

$$x_{p1}(t) = A \cdot \sin(2\pi f_1(t - t_{01})) \quad (3)$$

$$x_{p2}(t) = A \cdot \sin(2\pi f_2(t - t_{02})) \quad (4)$$

Dans le domaine fréquentiel :

$$X_{p0}(f) = j\frac{A}{2} \cdot [\delta(f + f_0) - \delta(f - f_0)] \cdot e^{-j2\pi f t_{00}} \quad (5)$$

$$X_{p1}(f) = j\frac{A}{2} \cdot [\delta(f + f_1) - \delta(f - f_1)] \cdot e^{-j2\pi f t_{01}} \quad (6)$$

$$X_{p2}(f) = j\frac{A}{2} \cdot [\delta(f + f_2) - \delta(f - f_2)] \cdot e^{-j2\pi f t_{02}} \quad (7)$$

4.1.2 Balises

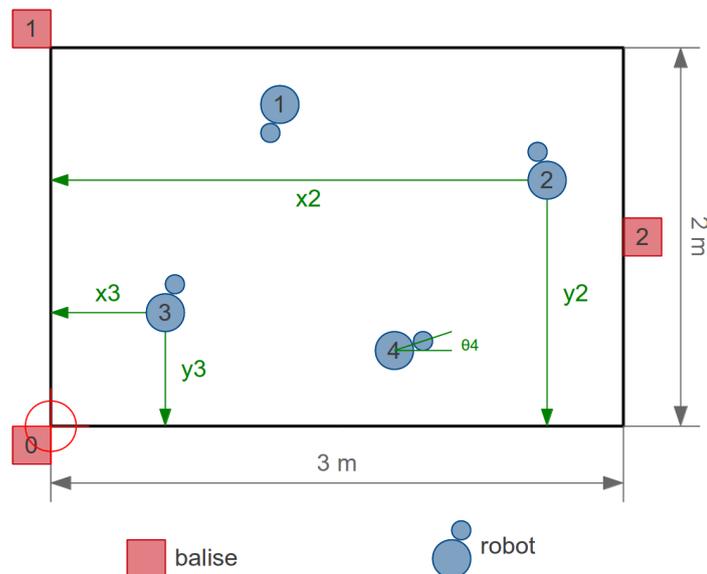


FIGURE 9 – Emplacement et numérotation des balises

1. Dual-tone multi-frequency

Nous établissons une grille avec les porteuses sur laquelle nous plaçons le numéro des balises. Chaque balise se voit attribuer deux des trois porteuses.

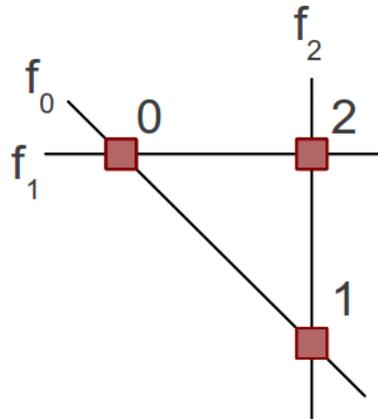


FIGURE 10 – Attribution des porteuses

Équations dans le domaine temporel :

$$x_{b0}(t) = x_{p0}(t) + x_{p1}(t) \quad (8)$$

$$x_{b1}(t) = x_{p0}(t) + x_{p2}(t) \quad (9)$$

$$x_{b2}(t) = x_{p1}(t) + x_{p2}(t) \quad (10)$$

Dans le domaine fréquentiel :

$$\Phi_{x_{b0}}(f) = \frac{A^2}{4} \cdot [\delta(f \pm f_0) + \delta(f \pm f_1)] \quad (11)$$

$$\Phi_{x_{b1}}(f) = \frac{A^2}{4} \cdot [\delta(f \pm f_0) + \delta(f \pm f_2)] \quad (12)$$

$$\Phi_{x_{b2}}(f) = \frac{A^2}{4} \cdot [\delta(f \pm f_1) + \delta(f \pm f_2)] \quad (13)$$

4.1.3 Attribution des fréquences

Le système numérique peut acquérir en entrée les fréquences allant de $1.6kHz$ à $150kHz$, les autres étant filtrées analogiquement avant la conversion. Afin d'avoir une dynamique suffisante (RPM du capteur optique), nous choisissons une plage de fréquences utiles allant de $60kHz$ à $140kHz$. Ceci correspond à la plage de fréquences qui peuvent être générées par les balises analogiques.

La modulation DTMF peut s'apparenter à une modulation AM car :

$$\sin(\alpha) + \sin(\beta) = 2 \cdot \sin\left(\frac{\alpha + \beta}{2}\right) \cdot \cos\left(\frac{\alpha - \beta}{2}\right) \quad (14)$$

Graphiquement :

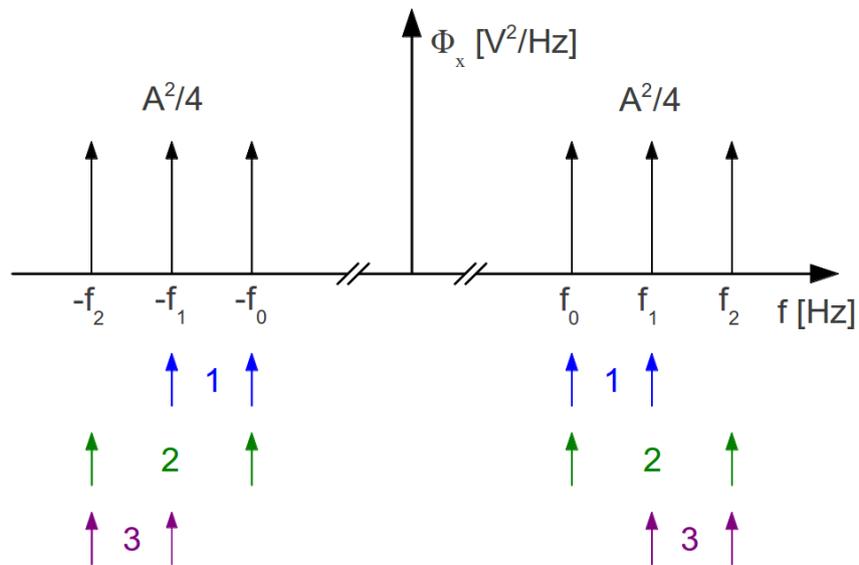


FIGURE 11 – Distribution et répartition des porteuses

Représentations temporelles avec $A = 1V$, $f_0 = 90kHz$, $f_1 = 110kHz$, $f_2 = 130kHz$ et $t_{00} = t_{01} = t_{02} = 0$:

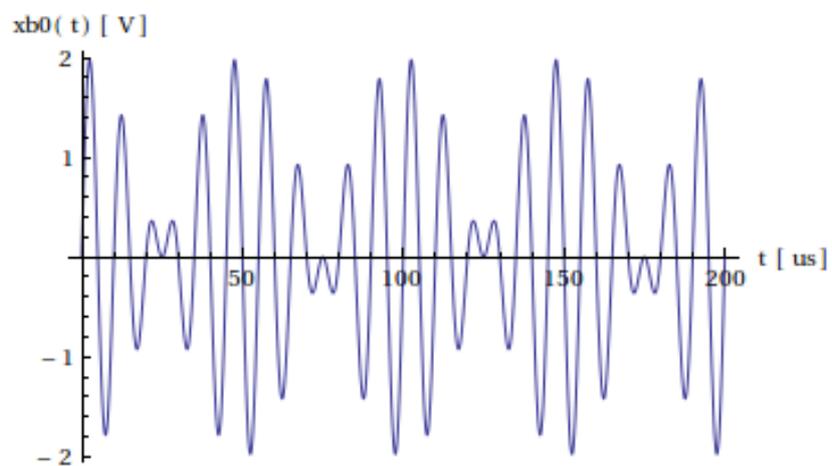


FIGURE 12 – Représentation temporelle du signal émis par la balise 0

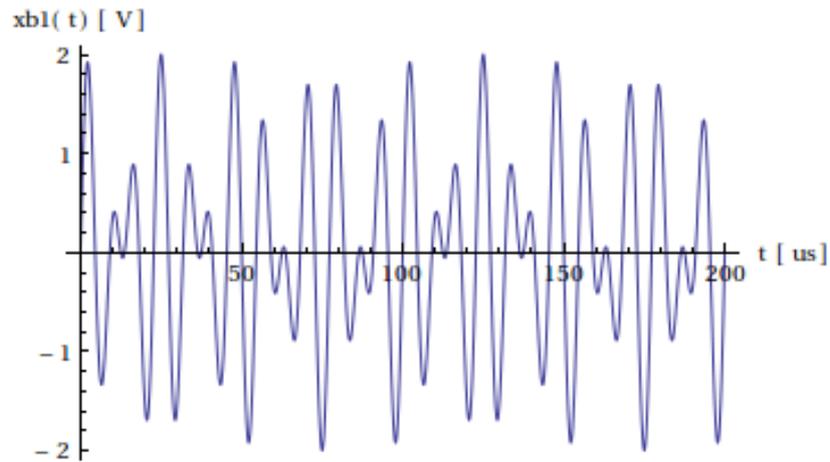


FIGURE 13 – Représentation temporelle du signal émis par la balise 1

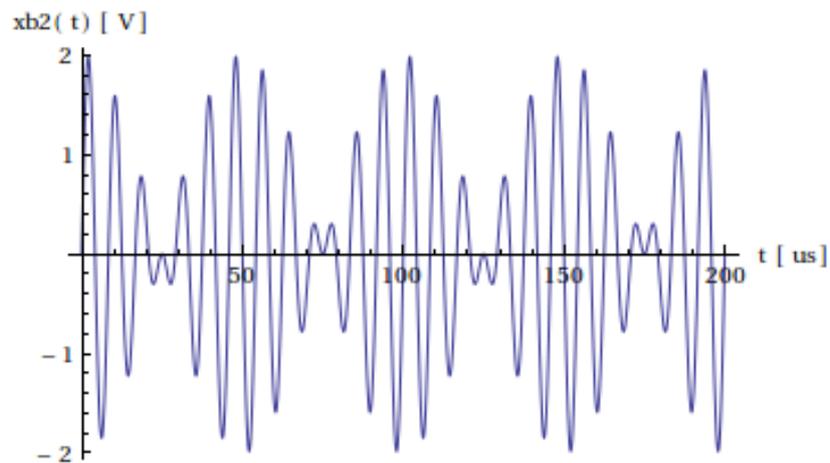


FIGURE 14 – Représentation temporelle du signal émis par la balise 2

À l'heure actuelle, les valeurs numériques définitives des fréquences des porteuses n'ont pas encore été définies. L'attribution dépend de la répartition statistique du bruit dans le domaine fréquentiel ainsi que des possibilités d'émissions des balises. Ce paramètre étant facilement ajustable des deux côtés, des mesures empiriques sont effectuées.

4.1.4 Paramètres t_0

t_{00} , t_{01} et t_{02} sont indéterminés. Ils pourraient éventuellement être choisis et définis pour modifier la période et l'évolution temporelle de l'onde.

Étant donnée la construction analogique actuelle des balises, une telle synchronisation est difficilement réalisable. Éventuellement pour une future version exploitant la synthèse numérique.

4.2 Spectre complet des fréquences

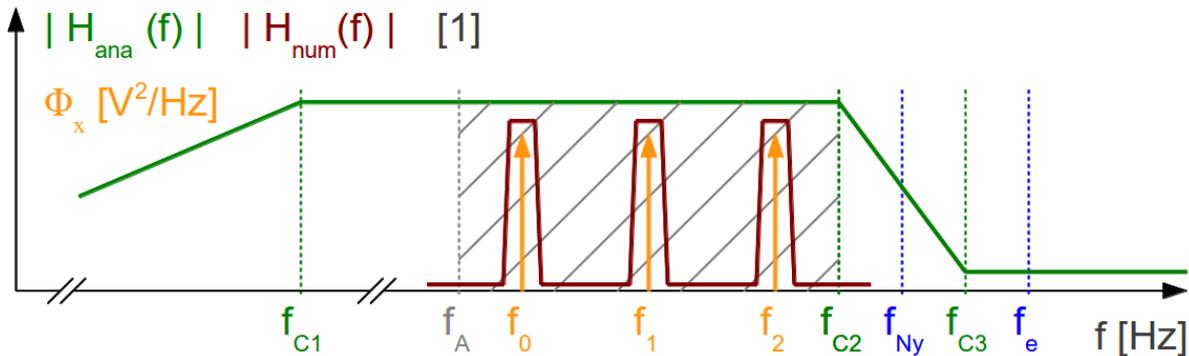


FIGURE 15 – Spectre des fréquences

Nom	Désignation	Fréquence [kHz]	Référence
Coupure du passe-haut Butterworth	f_{C1}	1.6	Équation (21)
Borne inférieure du signal utile	f_A	60	Section 4.1.3
Porteuse 0	f_0	indéfini	Section 4.1.3
Porteuse 1	f_1	indéfini	Section 4.1.3
Porteuse 2	f_2	indéfini	Section 4.1.3
Coupure du passe-bas LTC1564	f_{C2}	150	Section 4.5.4
-100dB du passe-bas LTC1564	f_{C3}	400	Section 4.5.4
Fréquence de Nyquist	f_{Ny}	250	–
Fréquence d'échantillonnage	f_e	500	Section 5.2.3

TABLE 1 – Définition des fréquences

4.3 Génération des signaux

Le principe est le suivant : nous générons deux sinus à des fréquences définies, les additionnons et commandons une LED infrarouge en courant. En courant pour une meilleure linéarité, nous mesurons aussi en courant à la réception.

Le générateur d'onde sinusoïdal peut être réalisé principalement de deux manières :

Analogiquement : la *vielle* méthode. À l'aide d'oscillateurs à ampli-op : pont de Wien, retard de phase etc. ou à l'aide de circuits intégrés : MAX038, XR2206, ICL8038. Ces circuits sont cependant obsolètes.

Numériquement : la méthode moderne, la DDS (Direct Digital Synthesis). Là aussi deux méthodes principales : calcul numérique de la somme des sinus avec convertisseur D/A ou utilisation de générateurs sinusoïdaux à haute précision, par exemple les circuits d'Analog Device.

Avantages et inconvénients de la méthode analogique :

- Rapidité de mise en oeuvre : assemblage de composants
- Réglage fréquence à précision arbitraire
- Distorsion harmonique (THD) élevée
- Dérive dans le temps

Avantages et inconvénients de la méthode numérique :

- Distorsion harmonique très faible (Analog Device)
- Temps de développement élevé : élaboration et fabrication PCB, programmation (éventuellement prise en main du MCU, des circuits oscillateurs)
- Choix des fréquences à générer par DIP switch et non potentiomètres : gain de temps lors d'éventuelles reconfigurations en environnement de concours

Le temps de développement et de réalisation étant un critère d'importance pour ce projet (les balises n'étant qu'une partie du système), l'implémentation analogique a été choisie. Le travail de développement et de fabrication a principalement été fait par le bureau de construction et ses apprentis.

Une nouvelle version de l'Absolute_LPS pourrait éventuellement faire usage de synthèse numérique pour les prochains concours Eurobot.

Le schéma électronique des balises se trouve en annexe.

4.4 Schéma bloc acquisition et traitement du signal

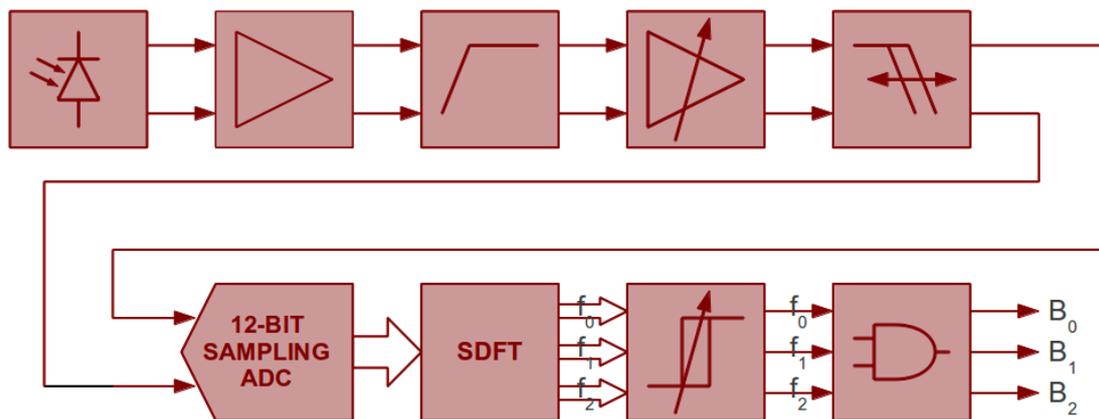


FIGURE 16 – Schéma bloc de la chaîne d'acquisition

Photodiode IR : Transforme la lumière infrarouge en courant électrique.

Pré-amplificateur à transconductance : Transforme le courant en tension avec amplification.

Filtre passe-haut : Supprime le DC et modifie la référence.

Amplificateur à gain programmable : Permet de varier le gain analogique de manière numérique et par logiciel.

Filtre anti-repliement : Évite un repliement du spectre sur les fréquences utiles.

Convertisseur analogique/numérique : Quantifie et discrétise le signal pour le traitement numérique.

SDFT : Analyse le spectre pour les trois fréquences des porteuses.

Seuils : Détermine si la porteuse est active ou non.

Combinatoire : Détermine quelle balise est active en fonction des porteuses actives.

4.5 Chaîne d'acquisition et de conditionnement analogique

4.5.1 Photodiode

La photodiode VBPW34FAS de Vishay a été sélectionnée pour son importante vitesse de fonctionnement et sa haute sensibilité.

Sa surface sensible de 7.5mm^2 nous offre une bonne marge mécanique quant à son centrage sur le faisceau lumineux.

Son demi-angle de 65° est plus que suffisant pour notre application. Nous utilisons en effet une lentille de diamètre 10mm et de focale 30mm. Le demi-angle utile est ainsi de $\arctan(5/30) = 9.5^\circ$ maximum.

Sa bande passante est comprise entre 780 et 1050 nm avec un maximum à 950 nm. Nous utiliserons cette longueur d'onde à l'émission.

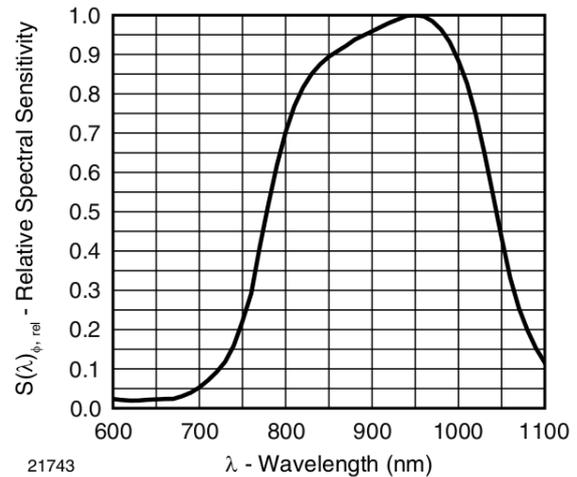
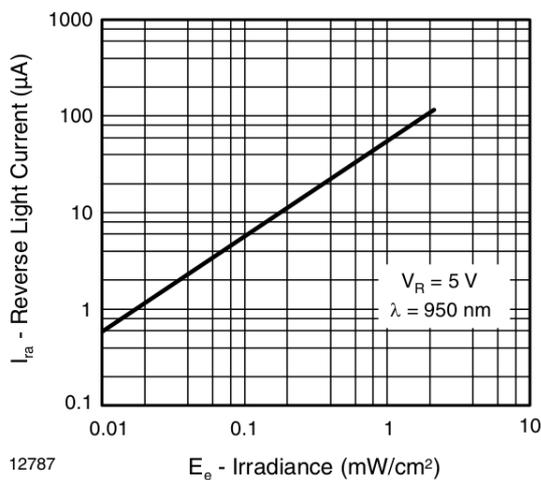


FIGURE 17 – Reverse Light Current vs. Irradiance FIGURE 18 – Spectral Sensitivity vs. Wavelength

Fonction de transfert :

$$\log(I_{ra}) = G \cdot \log(E_e) \quad (15)$$

$$I_{ra} = E_e^G \quad (16)$$

4.5.2 Pré-amplificateur à transconductance

Buts :

Transconductance : Transformer la variation de courant fournie par la photodiode en variation de tension.

Adaptation d'impédance : Haute impédance d'entrée et faible impédance de sortie.

Amplification : Très faible courant vers tension raisonnable.

Polarisation photodiode : Pour accélérer la réaction.

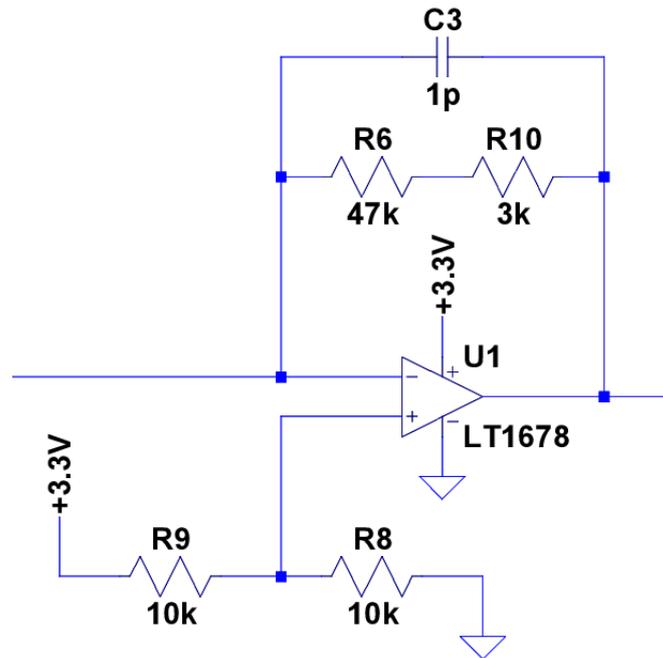


FIGURE 19 – Schéma du pré-amplificateur à transconductance

Il a fallu faire un compromis entre gain et tension de polarisation.

Gain :

$$G_{A1} = 47k\Omega + 3k\Omega = 50k\Omega \quad (17)$$

Courant maximum mesuré expérimentalement : $I_{Pmax} = 28\mu A$. Tensions min/max de sortie :

$$V_{A1min} = 3.3V \cdot \frac{R_8}{R_8 + R_9} = 3.3V \cdot \frac{10k\Omega}{10k\Omega + 10k\Omega} = 1.65V > 0V \quad (18)$$

$$V_{A1max} = V_{A1min} + (R_6 + R_{10}) \cdot I_{Pmax} = 1.65V + (47k\Omega + 3k\Omega) \cdot 28\mu A = 3.05V < 3.3V \quad (19)$$

C_3 permet de stabiliser l'amplificateur en évitant son oscillation. La fréquence de coupure de ce filtre est de $f_c = \frac{1}{2\pi R_{A1} C_3} \approx 3MHz$. Cette valeur a été choisie en faisant un compromis entre l'atténuation de la fréquence de résonance du circuit (voir section 4.5.6) et la déformation (amplitude et phase) du signal utile.

Fonction de transfert :

$$H_{preampli}(s) = \frac{1}{\frac{1}{G_{A1}} + sC} = \frac{G_{A1}}{1 + sG_{A1}C} = \frac{5 \cdot 10^4}{1 + 5 \cdot 10^{-8}s} \quad (20)$$

4.5.3 Filtre passe-haut et changement de référence

Buts :

Filtrage : Supprimer le DC et le 100Hz des lampes à incandescence.

Changement de référence : Référencer le signal par rapport à "LTC1564_AGND" (qui correspond à $\frac{3.3V}{2} = 1.65V$) et non plus à "OV". Ceci permettra une amplification linéaire et sans déformation du signal par le LTC1564.

Amplification : Doubler l'amplitude précédemment limitée par la polarisation de la photodiode.

Le filtre retenu est un Butterworth du deuxième ordre pour sa fonction de transfert constante en bande passante, sa pente et sa relative simplicité d'implémentation à l'aide de la topologie de Sallen-Key.

La fréquence de coupure choisie est volontairement largement inférieure à la bande utile du signal. Ceci pour éviter toute déformation en amplitude ou en phase de celui-ci.

Un suiveur a été ajouté à la sortie AGND du circuit LTC1564 pour compenser sa relative haute impédance de sortie.

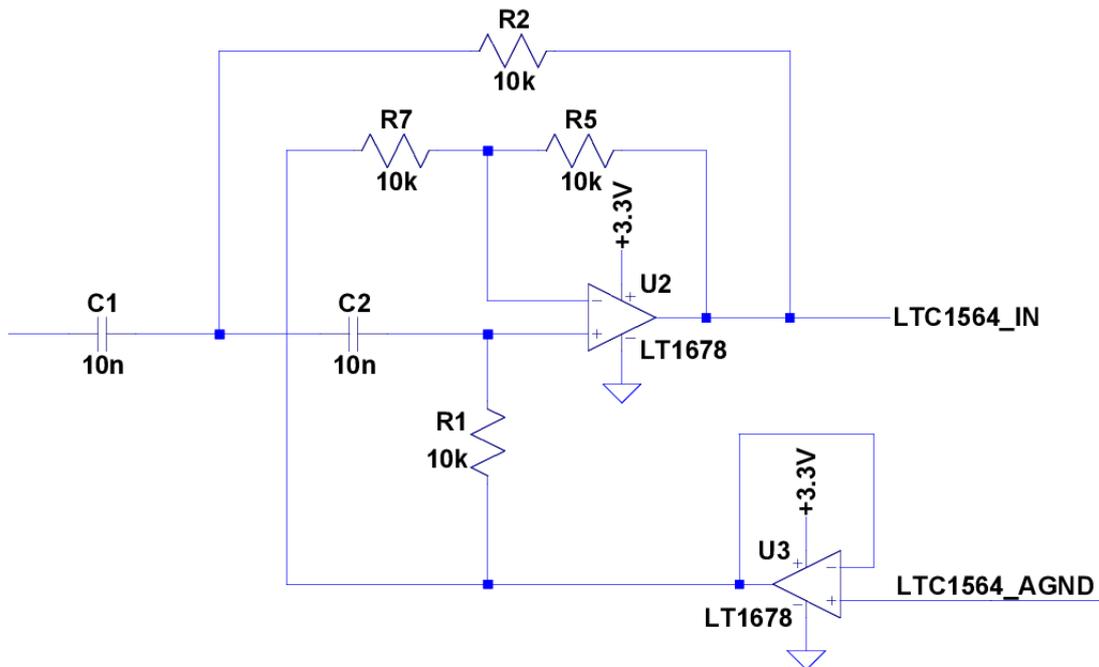


FIGURE 20 – Schéma du filtre passe-haut

Valeurs caractéristiques :

$$f_{C1} = \frac{1}{2\pi \cdot \sqrt{C1 \cdot R2 \cdot C2 \cdot R1}} = \frac{1}{2\pi \cdot \sqrt{10nF \cdot 10k\Omega \cdot 10nF \cdot 10k\Omega}} \approx 1.6kHz \quad (21)$$

$$G_{A2} = 1 + \frac{R5}{R7} = 1 + \frac{10k\Omega}{10k\Omega} = 2 \quad (22)$$

Tensions min/max de sortie :

$$V_{A2min} = V_{LTC1564_AGND} - \frac{V_{A1max} - V_{A1min}}{2} \cdot G_{A2} \tag{23}$$

$$V_{A2min} = 1.65V - \frac{3.05V - 1.65V}{2} \cdot 2 = 0.25V > 0V \tag{24}$$

$$V_{A2max} = V_{LTC1564_AGND} + \frac{V_{A1max} - V_{A1min}}{2} \cdot G_{A2} \tag{25}$$

$$V_{A2max} = 1.65V + \frac{3.05V - 1.65V}{2} \cdot 2 = 3.05V < 3.3V \tag{26}$$

Fonction de transfert :

$$H_{passe-haut}(s) = G_{A2} \cdot \frac{C_1 C_2 R_1 R_2 s^2}{1 + C_1 R_2 s + C_1 C_2 R_1 R_2 s^2} = \frac{2 \cdot 10^{-8} s^2}{1 + 10^{-4} s + 10^{-8} s^2} \tag{27}$$

4.5.4 Filtre passe-bas anti-repliement et gain programmables

Buts :

Filtrage : Supprimer les fréquences supérieures à la fréquence d'échantillonnage afin d'éviter un repliement du spectre en-dessous de la fréquence de Nyquist.

Amplification : Offrir un gain analogique programmable numériquement afin de compenser la dynamique élevée du signal et offrir une capacité de modification logicielle du gain analogique.

Cette fonction est assurée par le circuit intégré LTC1564 de la maison Linear Technologies. Il s'agit d'un filtre anti-repliement à fréquence de coupure et gain programmables. Nous utiliserons toujours la fréquence de coupure maximale de $150kHz$. C'est un passe-bas du huitième ordre et au vu de sa réponse fréquentielle, il doit s'agir d'un Chebyshev inverse (la fonction de transfert n'est pas donnée). Le gain est programmable de 1 à 16 par pas de 1.

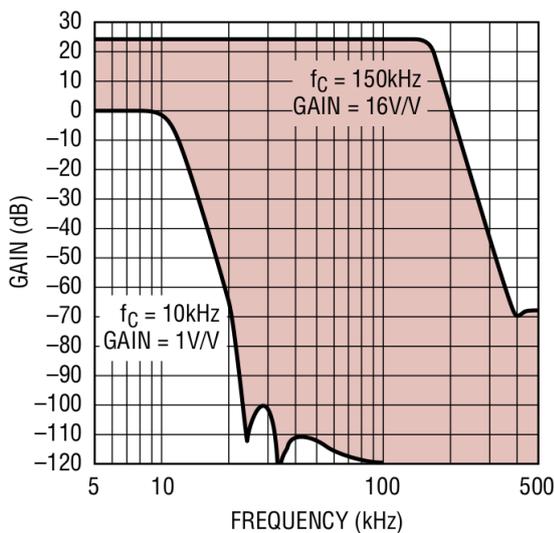


FIGURE 21 – Plage programmable du LTC1564

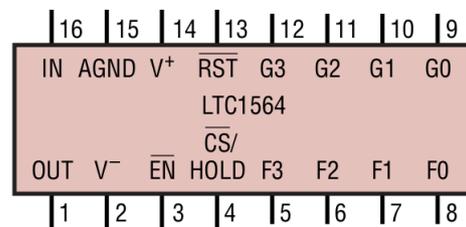


FIGURE 22 – Schéma du circuit LTC1564

D'après l'équation (33), nous savons que le rapport signal sur bruit maximal de l'A/D est de $70dB$. Dans le cas le plus critique ($f_c = 150kHz$ et $G = 16$), la fréquence d'échantillonnage doit être supérieure

à $300kHz$ pour garantir une atténuation en bande bloquée supérieure à $70dB$. La fréquence de Nyquist est donc d'au moins $150kHz$, ce qui est supérieure à la bande de fréquences utile.

4.5.5 Convertisseur analogique / numérique

Nous utilisons pour l'acquisition du signal un convertisseur analogique/numérique 12 bits, intégré au MCU TI Piccolo F28069. Sa dynamique est fixée par la tension d'alimentation, soit $3.3V$. Nous pouvons en déduire son pas et la puissance de son bruit de quantification :

$$\Delta = \frac{V}{2^n} = \frac{3.3V}{2^{12}} \approx 806\mu V \quad (28)$$

$$\sigma_q^2 = \int_{-\Delta/2}^{\Delta/2} n_q^2 \cdot p(n_q) dn_q = \int_{-\Delta/2}^{\Delta/2} n_q^2 \cdot \frac{1}{\Delta} dn_q = \frac{\Delta^2}{12} = \frac{1}{12} \cdot \left(\frac{V}{2^n}\right)^2 = \frac{1}{12} \cdot \frac{(3.3V)^2}{2^{24}} \approx 54nV^2 \quad (29)$$

L'A/D n'est évidemment pas parfait et a également des erreurs d'offset et de gain. Nous n'en tiendrons pas compte ici.

Les mesures empiriques ont montrées que la dynamique (amplitude max sur min) du signal était de 100, ce qui correspond à $40dB$.

Deux résistances permettent de fixer le gain du pré-amplificateur à transconductance. Nous ajusterons celles-ci afin d'avoir un signal en pleine échelle lorsque l'amplitude est maximale, c'est-à-dire lorsque la distance entre une balise et le module est minimale. Afin de garder une marge, nous fixons l'amplitude maximale du signal à $3V$. Avec une dynamique de $40dB$, cela nous donne :

$$7.5mV \leq A \leq 750mV \quad (30)$$

Nous pouvons ainsi calculer la puissance du signal utile :

$$\sigma_x^2 = P_x = \int_{-\infty}^{\infty} \Phi_x(f) df = \int_{-\infty}^{\infty} \frac{A^2}{4} \cdot [\delta(f \pm f_{0,1,2}) + \delta(f \pm f_{0,1,2})] df = A^2 \quad (31)$$

Nous calculons maintenant le rapport signal sur bruit minimal et maximal :

$$\xi_{qdB} = 10 \cdot \log_{10}(\xi_q) = 10 \cdot \log_{10}\left(\frac{\sigma_x^2}{\sigma_q^2}\right) = 10 \cdot \log_{10}\left(\frac{12 \cdot A^2 \cdot 2^{2n}}{V^2}\right) \quad (32)$$

$$\xi_{qdB}|_{A=750mV} = 10 \cdot \log_{10}\left(\frac{12 \cdot (750mV)^2 \cdot 2^{24}}{(3.3V)^2}\right) \approx 70dB \quad (33)$$

$$\xi_{qdB}|_{A=7.5mV} = 10 \cdot \log_{10}\left(\frac{12 \cdot (7.5mV)^2 \cdot 2^{24}}{(3.3V)^2}\right) \approx 30dB \quad (34)$$

Si nécessaire, nous avons à disposition un gain analogique $1 \leq G_A \leq 16$ pour augmenter ce rapport. Son utilisation demande évidemment un effort de programmation supplémentaire. Les équations deviendraient alors :

$$\xi_{qdB} = 10 \cdot \log_{10} \left(\frac{12 \cdot (G_A \cdot A)^2 \cdot 2^{2n}}{V^2} \right) \quad (35)$$

$$\xi_{qdB}|_{A=750mV, G_A=1} = 10 \cdot \log_{10} \left(\frac{12 \cdot (1 \cdot 750mV)^2 \cdot 2^{24}}{(3.3V)^2} \right) \approx 70dB \quad (36)$$

$$\xi_{qdB}|_{A=7.5mV, G_A=16} = 10 \cdot \log_{10} \left(\frac{12 \cdot (16 \cdot 7.5mV)^2 \cdot 2^{24}}{(3.3V)^2} \right) \approx 54dB \quad (37)$$

4.5.6 Simulation

La première partie de la chaîne analogique a été simulée à l'aide de l'outil LTspice, comprenant la description de tous les amplificateurs opérationnels LT. Les signaux produits par la photodiode sont modélisés par des sources de courant. La sortie AGND du filtre anti-repliement LTC1564 est modélisée par son diviseur de tension interne ainsi que la capacité de stabilisation externe.

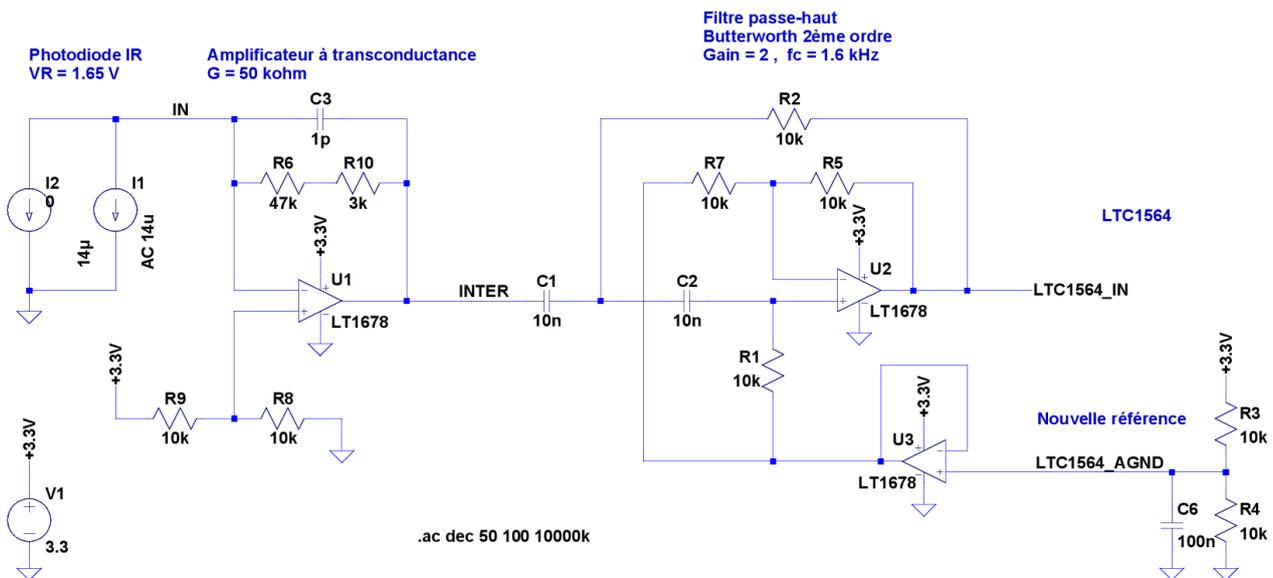


FIGURE 23 – Schéma de simulation LTspice IV

La FIGURE 24 est une simulation avec un signal sinusoïdal afin de se représenter la fonctionnalité du conditionneur en bande passante.

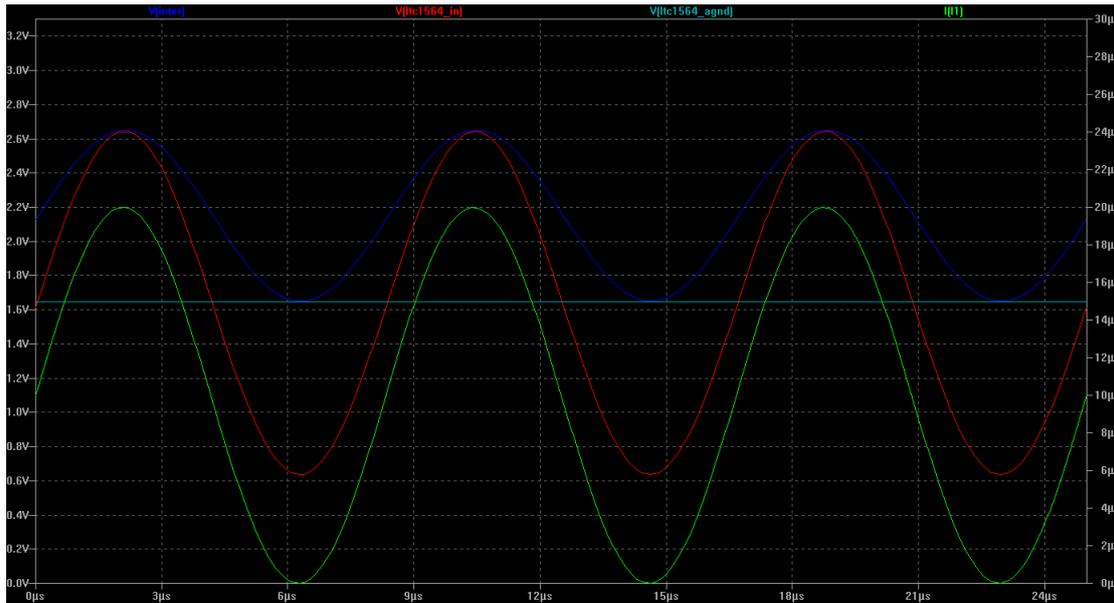
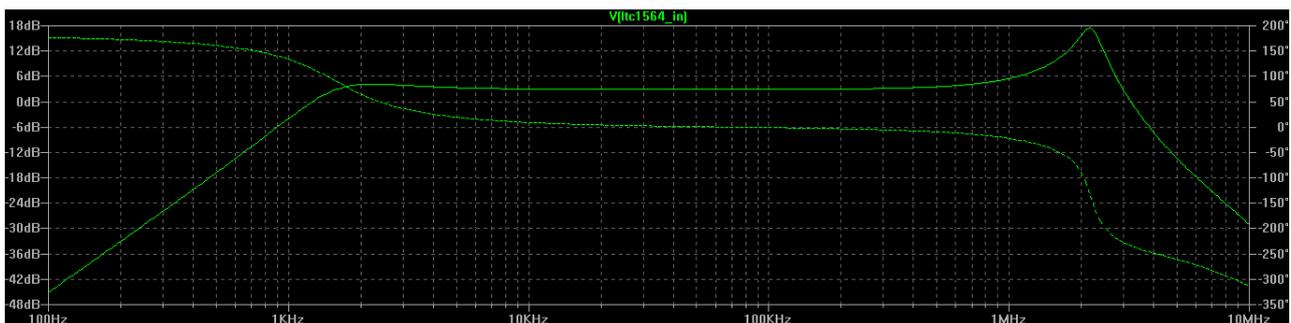
FIGURE 24 – Exemple avec un sinus à 120kHz d'amplitude $A = 10\mu A$ 

FIGURE 25 – Réponse fréquentielle complète

L'amplification à $2MHz$ n'est pas souhaitable et pourrait être supprimée en diminuant la fréquence de coupure du filtre passe-bas, c'est-à-dire en augmentant la valeur de la capacité C_3 . La fréquence de coupure se rapprochant de la bande utile, ceci augmenterait le déphasage et le group delay des signaux en bande passante. Une autre solution aurait été d'utiliser des AOP avec un GBW plus élevé. Ceux-ci présentant d'autres caractéristiques moins bonnes, notamment l'erreur d'offset ou le bruit. Cette bande de fréquence sera coupée par le filtre anti-repliement LTC1564.

$$\Delta_G = G|_{f=140kHz} - G|_{f=100kHz} \approx 23.8mdB \quad (38)$$

$$\Delta_\varphi = \varphi|_{f=100kHz} - \varphi|_{f=140kHz} \approx 1.03^\circ \quad (39)$$

$$\Delta_{groupdelay} = t|_{f=100kHz} - t|_{f=140kHz} \approx 12.3ns \quad (40)$$

$$t_{groupdelay} = \frac{t|_{f=100kHz} + t|_{f=140kHz}}{2} \approx 71.4ns \quad (41)$$

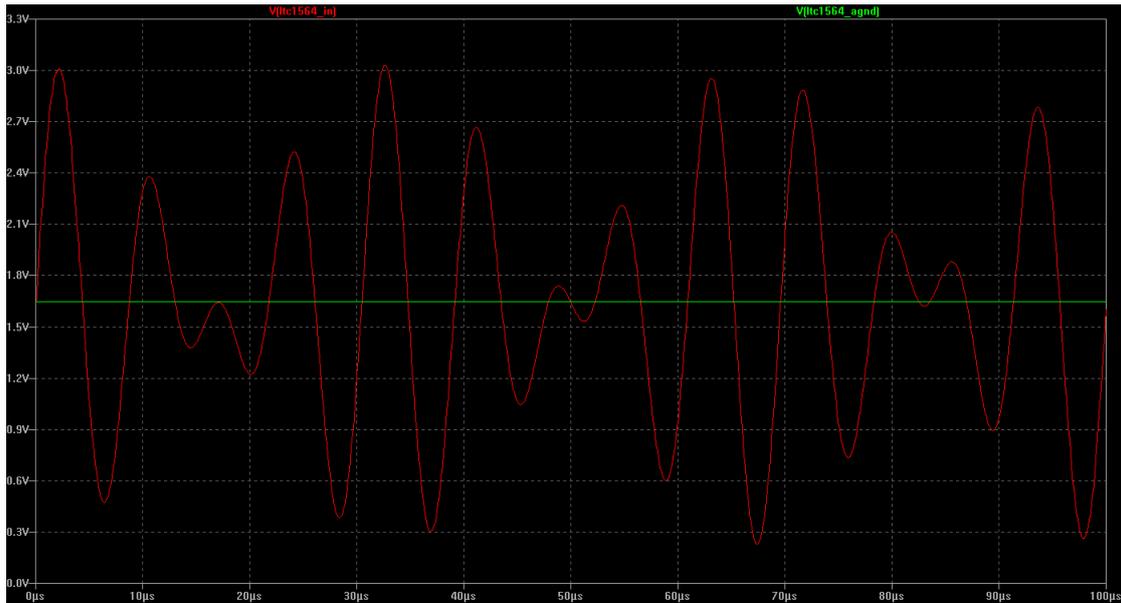
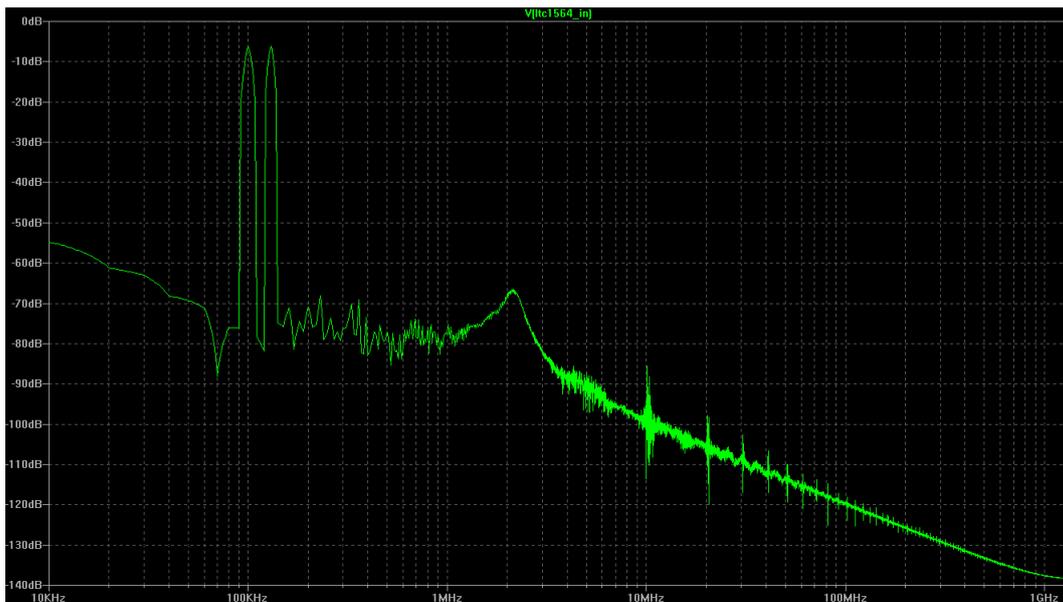
FIGURE 26 – Simulation d'une période du signal reçu d'une balise ($A = 10\mu A$)

FIGURE 27 – Simulation de la FFT du signal reçu d'une balise

4.6 Traitement numérique du signal

4.6.1 Principe détection et identification balise

Nous mesurons l'énergie (ou la puissance) des bandes de fréquence comprenant les porteuses à l'aide de la SDFT. Un comparateur à hystérèse avec des seuils absolus permet de détecter la présence ou l'absence d'une porteuse. Nous mesurons également l'énergie du signal dans la fenêtre d'analyse temporelle et comparons l'énergie des porteuses à celle-ci. Pour vérification de la présence d'une balise, environ 50% de l'énergie doit se trouver sur une porteuse et 50% sur l'autre.

4.6.2 Transformée de Fourier discrète à fenêtre glissante (SDFT)

La transformée de Fourier discrète à fenêtre glissante permet d'analyser l'amplitude d'une bande de fréquence en continue. Nous n'utilisons pas de FFT car celle-ci est trop coûteuse puisque seule trois bandes de fréquence nous intéressent. Une DFT classique est envisageable pour analyser une suite finie d'échantillons. Pour analyser des échantillons en continu, il est moins coûteux d'utiliser la SDFT qui demande très peu de calcul par échantillon une fois ses registres remplis. Il existe également l'algorithme de Goertzel, qui est toutefois plus gourmand que la SDFT.

Pour plus de détails, notamment la formulation mathématique ainsi que la comparaison avec d'autres algorithmes, voir le chapitre 13.18 de [3].

Ci-dessous la transformée en Z de la SDFT. Celle-ci peut être décomposée en deux parties : le filtre en peigne et le résonateur complexe. Le filtre en peigne ne dépend pas de la fréquence à analyser et peut donc être implémenté une fois pour toute. Le résonateur complexe doit être implémenté pour chaque fréquence à analyser.

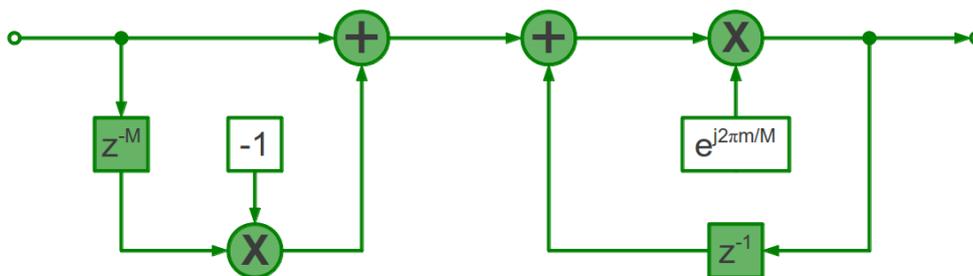


FIGURE 28 – Transformée en Z de la SDFT

Ci-dessus le résonateur complexe est représenté à l'aide de nombres complexes. Bien sûr ceux-ci ne peuvent pas être implémentés sur un DSP. Il convient donc de le transformer pour une implémentation à l'aide de nombres réels. Nous intégrons directement la mise au carré et l'addition des parties réelles et imaginaires permettant de mesurer l'énergie et non l'amplitude du signal.

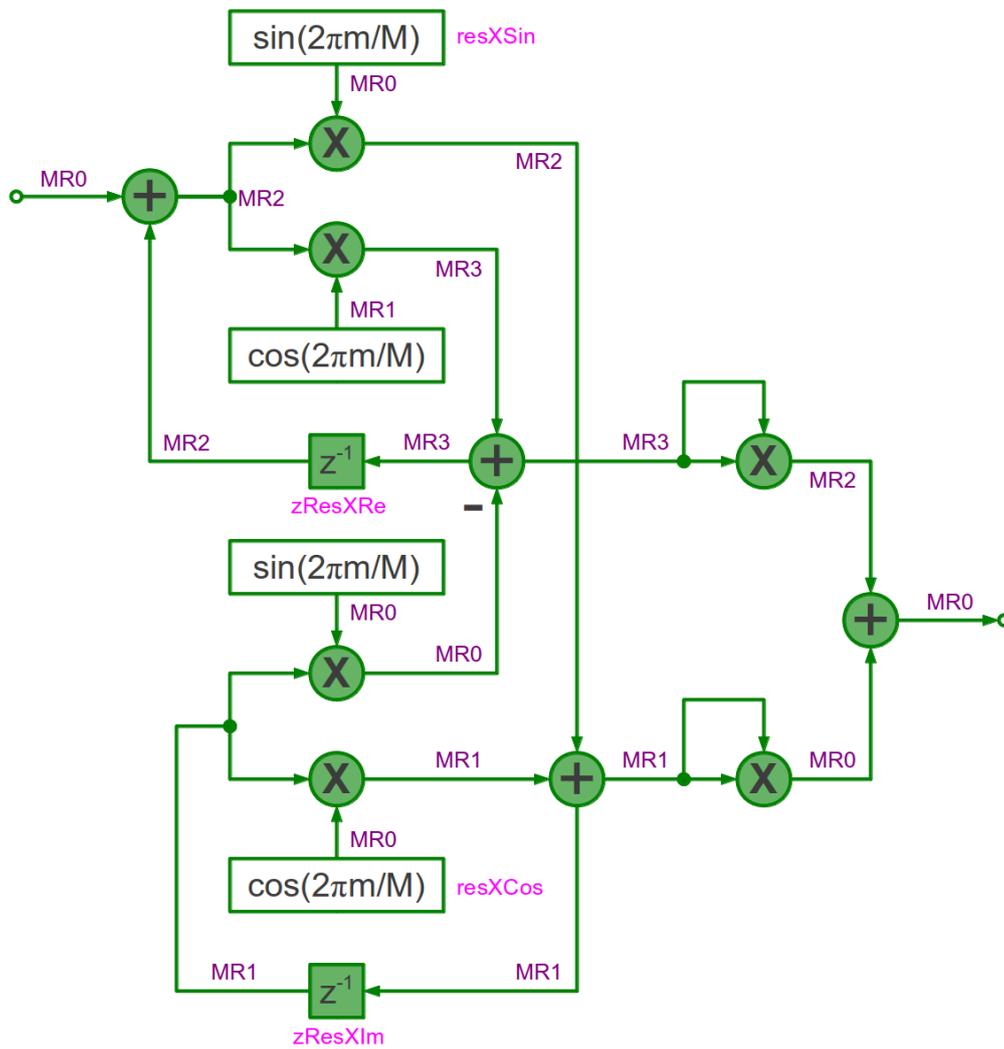


FIGURE 29 – Transformée en Z du résonateur complexe implémenté à l'aide de nombres réels

Les annotations correspondent à des noms de variables et registres permettant de s'y retrouver dans le code assembleur de l'algorithme.

L'implémentation réelle du résonateur complexe de la SDFT comprends 6 multiplications, 4 additions et 2 registres pour chaque fréquence analysée. À cela il faut ajouter un filtre en peigne comprenant une addition et un registre de longueur M. M correspondant à la taille de la fenêtre d'analyse.

4.6.3 Simulation Matlab

Voici quelques figures illustrant les simulations du traitement numérique effectuées avec Matlab. Le code source se trouve en annexe.

Tout d'abord une représentation temporelle de l'apparition et disparition de deux porteuses ($f_0 = 87.5kHz$ et $f_1 = 112.5kHz$) représentant la balise 0 avec addition de bruit blanc (tension crête à crête de $20mV$) :

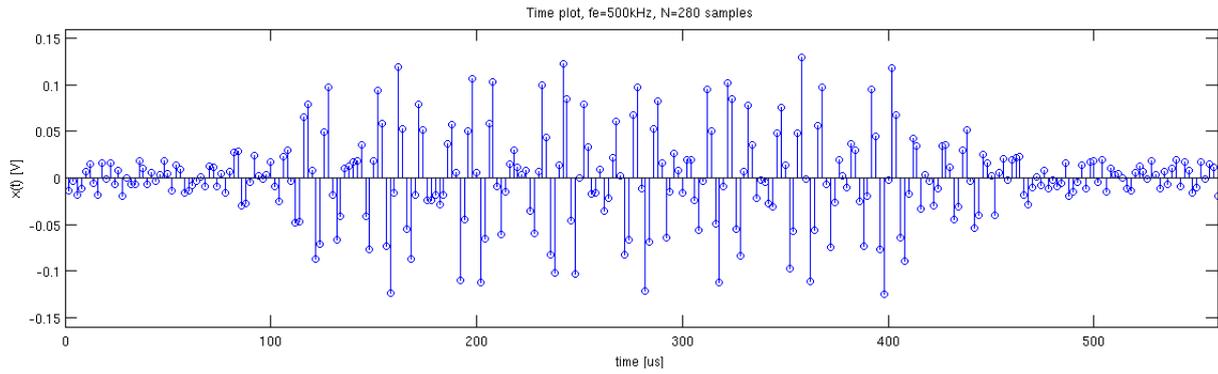


FIGURE 30 – Représentation temporelle

Une FFT vers le milieu du signal dont la fenêtre d'analyse est identique à celle des SDFT afin de se rendre compte de l'aspect du spectre :

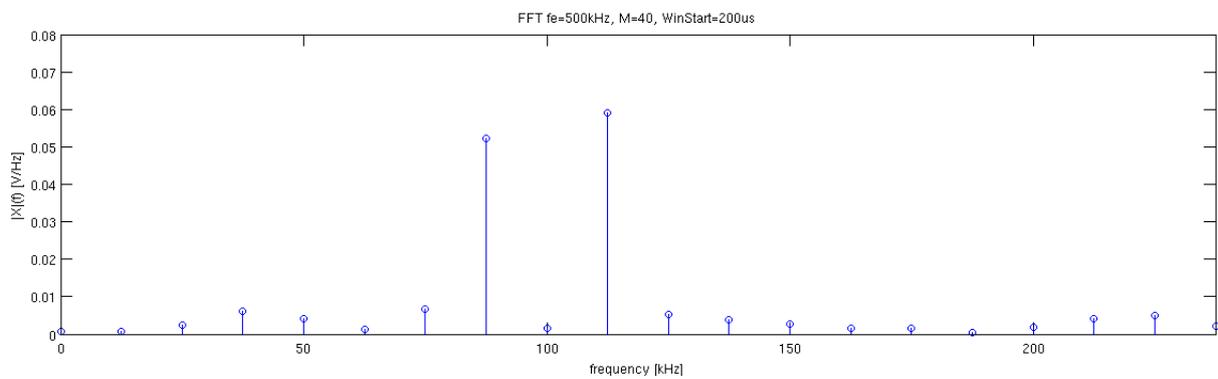


FIGURE 31 – FFT du signal

Le résultat temporel de la première SDFT analysant la fréquence de la porteuse 0 :

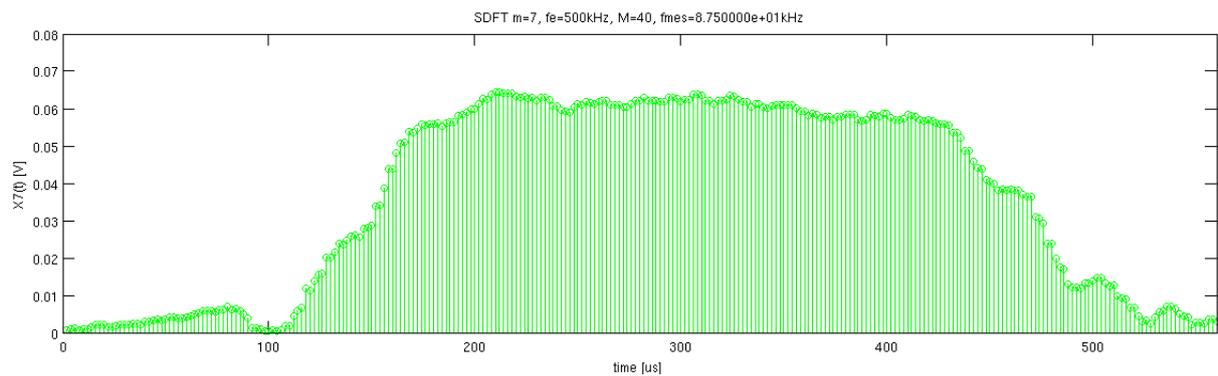


FIGURE 32 – SDFT porteuse 0

Le résultat temporel de la seconde SDFT analysant la fréquence de la porteuse 1 :

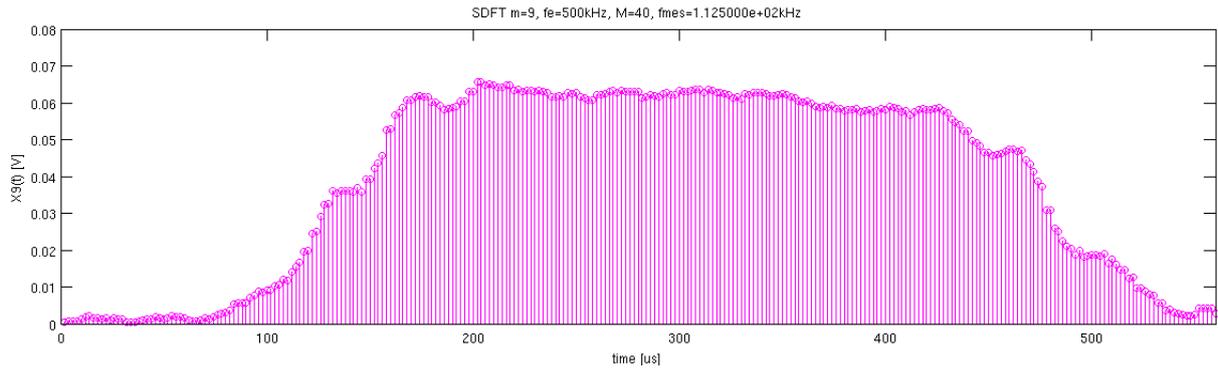


FIGURE 33 – SDFT porteuse 1

Le résultat temporel de la troisième SDFT analysant la fréquence de la porteuse 2 :

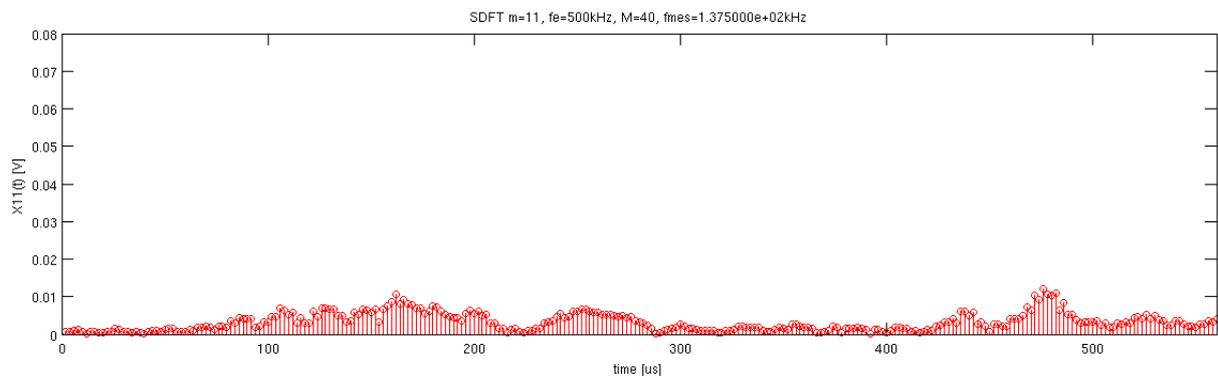


FIGURE 34 – SDFT porteuse 2

L'énergie totale du signal, de la porteuse 0, 1 et 2 :

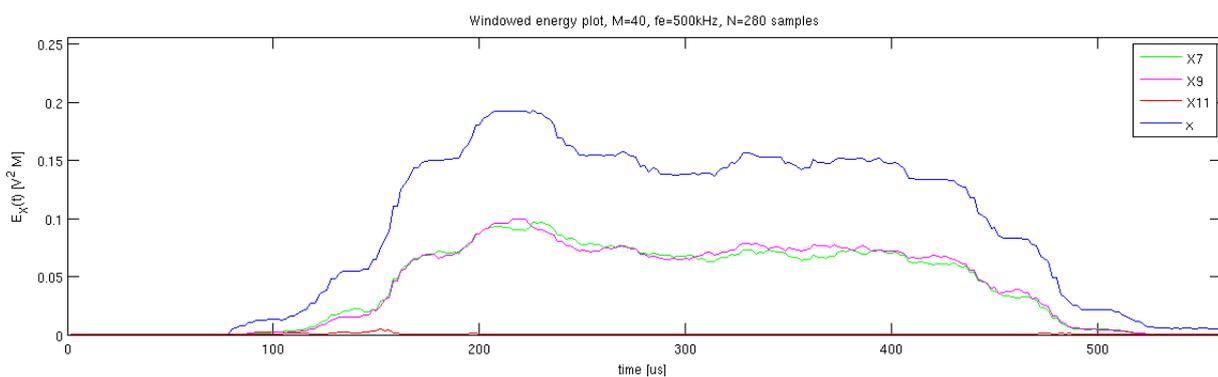


FIGURE 35 – Énergies absolues

L'énergie relative de la porteuse 0, 1 et 2 par rapport à l'énergie totale du signal :

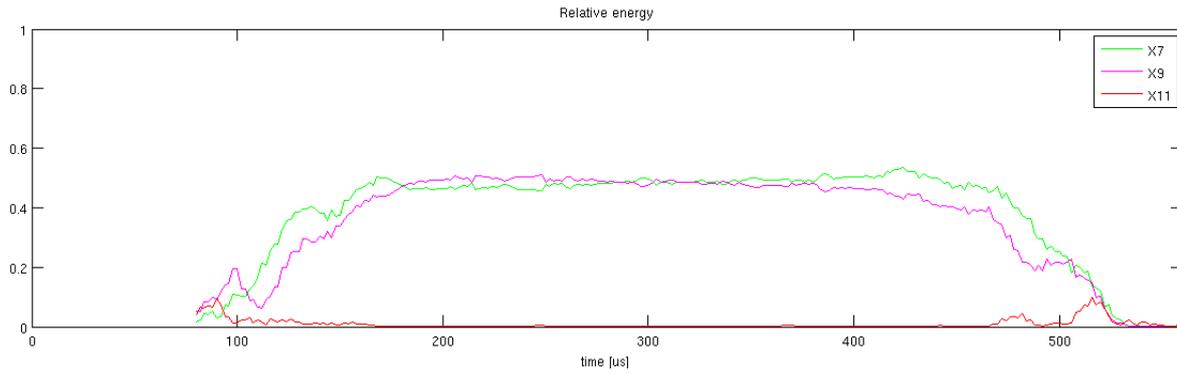


FIGURE 36 – Énergies relatives

Détection binaire de la présence ou non des porteuses :

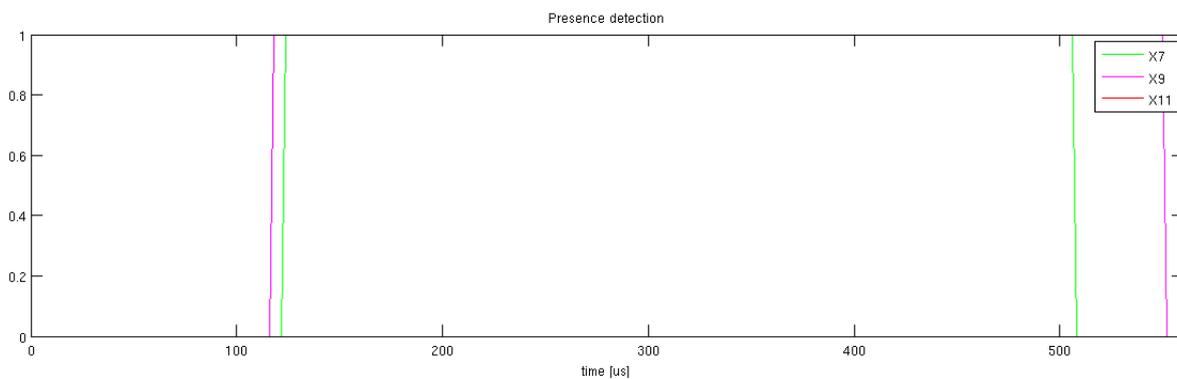


FIGURE 37 – Présences des porteuses

Ces simulations avec Matlab ont été très utiles pour caractériser les algorithmes et déterminer les divers paramètres. Il est également possible d'extraire les tensions mesurées par l'A/D et de les insérer dans la simulation afin de disposer de signaux réels.

4.6.4 Transformée en Z

Ci-dessous la transformée en Z de la chaîne de traitement numérique du signal avec SDFT et calculs d'énergie.

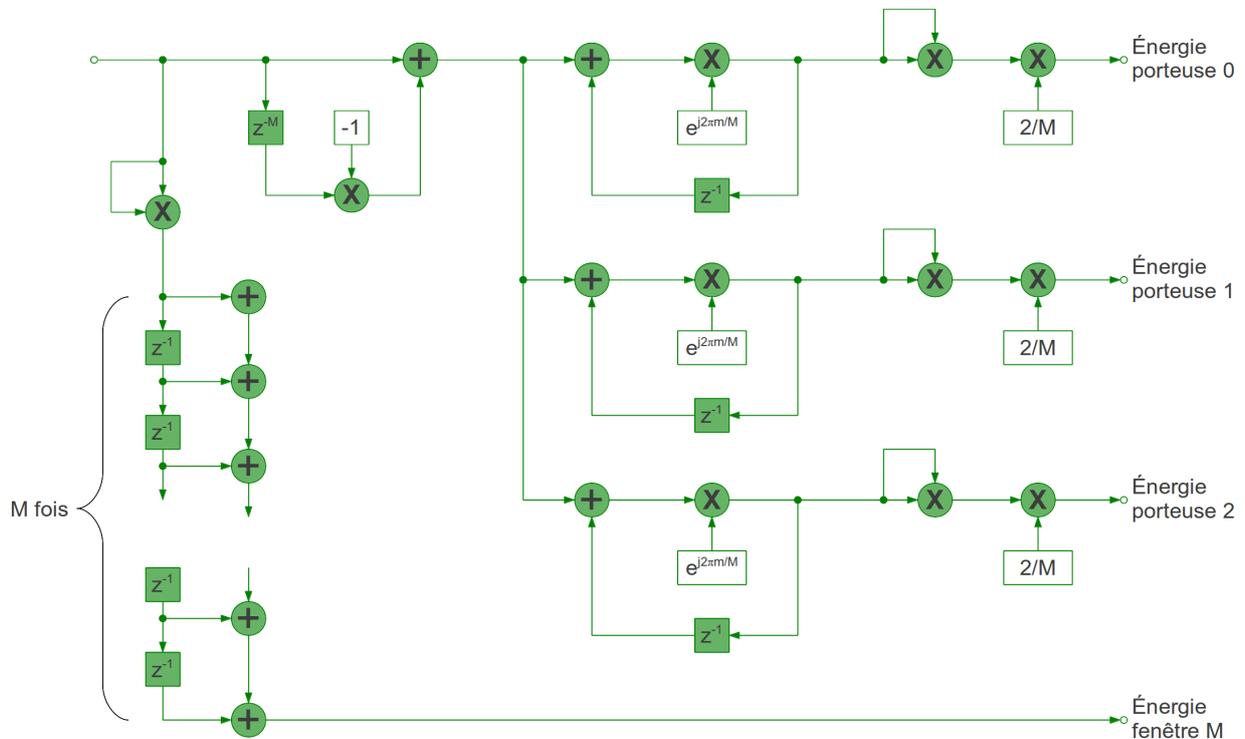


FIGURE 38 – Transformée en Z

4.6.5 Analyse de la complexité algorithmique

Analysons la complexité algorithmique avant optimisation :

Multiplications : $3 \cdot 6 + 3 + 2 = 23$

Additions : $3 \cdot 4 + 1 + M + 1 = M + 14$

Registres : $3 \cdot 2 + 2 \cdot M = 2M + 6$

Nous avons beaucoup de mémoire à disposition mais peu de ressources de calcul allouées à chaque période d'échantillonnage. Nous allons donc optimiser l'algorithme pour diminuer le nombre d'opérations quitte à augmenter le nombre de registres utilisés si nécessaire.

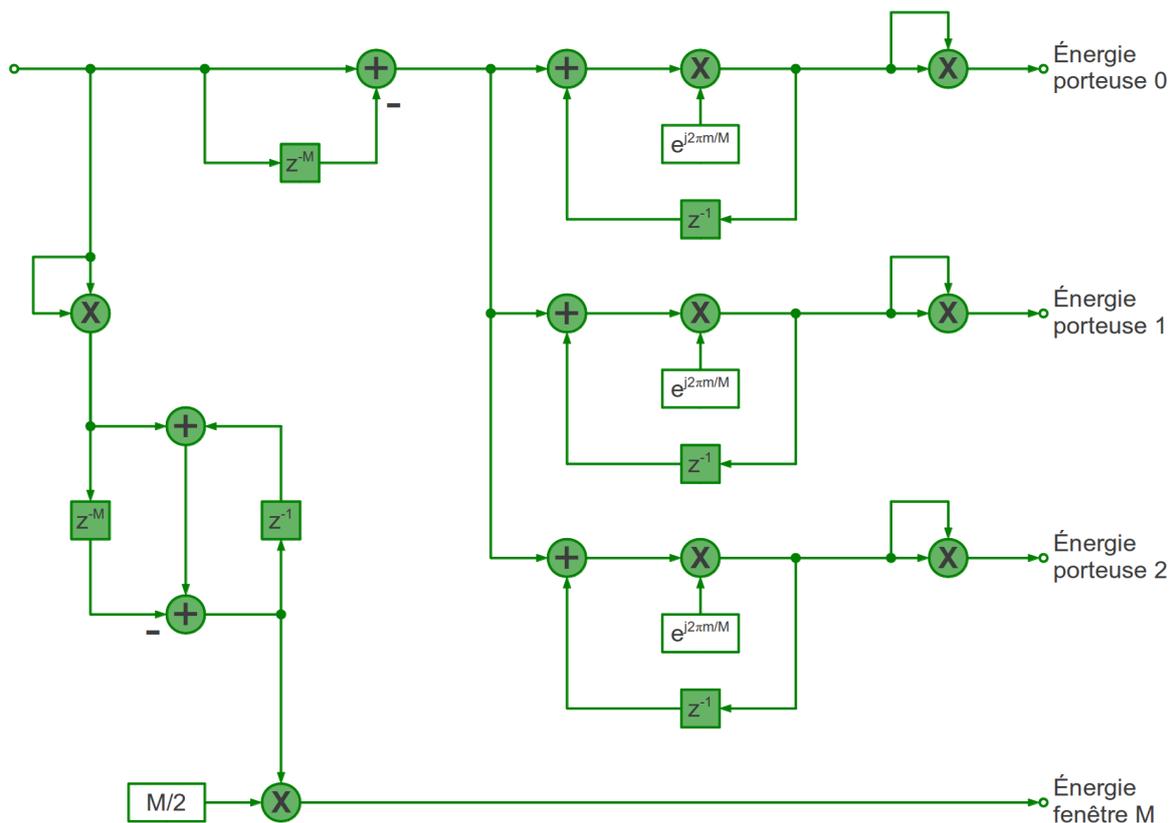


FIGURE 39 – Transformée optimisée selon contraintes

Bilan :

Multiplications : $3 \cdot 6 + 2 = 20$

Additions : $3 \cdot 4 + 3 = 15$

Registres : $3 \cdot 2 + 2 \cdot M + 1 = 2M + 7$

La complexité après optimisation est plus faible mais l'algorithme moins stable due aux inévitables erreurs d'intégrations lors du calcul de l'énergie dans le domaine temporel. Malgré une représentation des nombres en virgule flottante 32 bits certaines additions peuvent être *perdues* (c'est-à-dire non soustraite par la suite) lorsque la valeur contenue dans le registre de somme est nettement plus grande que la valeur à soustraire. Ce problème n'est cependant pas gênant car il doit également être traité pour le résonateur complexe. Voir section 4.6.6.

4.6.6 Étude de la stabilité

L'algorithme présentée ci-dessus pose deux problèmes de stabilité : la SDFT et l'intégrateur permettant la mesure d'énergie dans le domaine temporel.

L'impact est ici limité par la représentation des nombres en virgule flottante, donc avec une très grande dynamique comparée à une représentation en virgule fixe. Ce format n'a pas été choisi, mais imposé par la plate-forme d'implémentation (le co-processeur mathématique) qui ne travaille qu'en 32 bits flottants.

Une étude plus poussée de la stabilité serait nécessaire à la détermination exacte de son influence.

Pour s'affranchir définitivement de ce problème sans entacher les performances de l'algorithme (par l'ajout de multiplications supplémentaires, par exemple, pour rester à l'intérieur du cercle unité), nous avons choisi de remettre périodiquement tous les registres à zéro. Ceci peut se faire après identification d'une balise car nous savons que deux balises ne peuvent pas immédiatement se suivre.

5 Implémentation

5.1 Matériel

5.1.1 Schéma bloc de l'électronique des modules

La FIGURE 40 présente le schéma bloc de l'électronique des modules. Un micro-contrôleur de la maison Texas Instrument dirige les opérations. Celui-ci sera présenté plus en détail dans la section 5.2.1. On y voit bien sûr apparaître la chaîne analogique d'acquisition et de conditionnement du signal déjà présentée dans la section 4.5. Les autres éléments ont été succinctement présentés dans le chapitre 3. Ils ne font pas partie du projet de semestre et ne sont donc pas présentés plus en détail dans ce document.

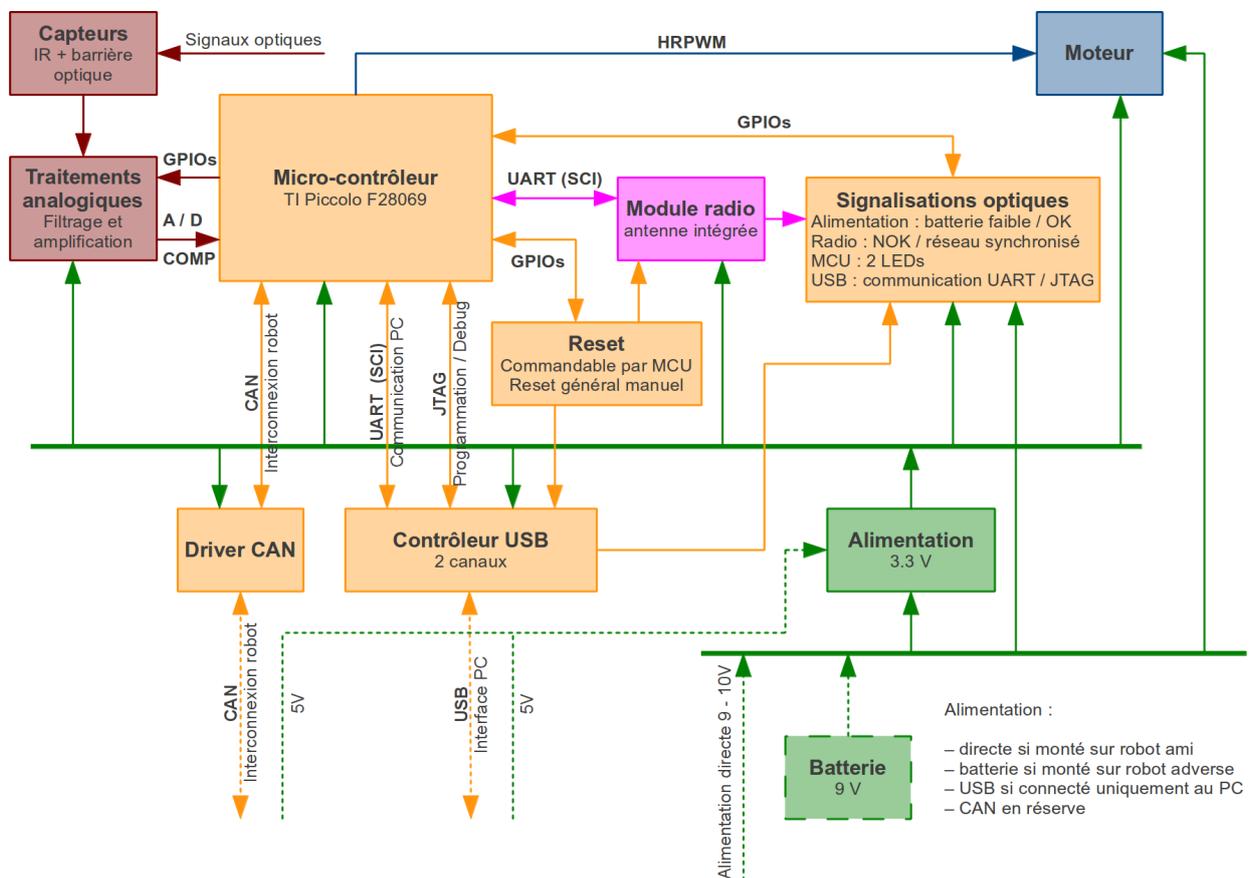


FIGURE 40 – Schéma bloc de l'électronique des modules

5.1.2 Mécanique et optique

La conception et la réalisation mécaniques et optiques ne faisant pas parties du projet de semestre, nous verrons uniquement quelques photos des éléments terminés pour se faire une idée de la réalisation physique.

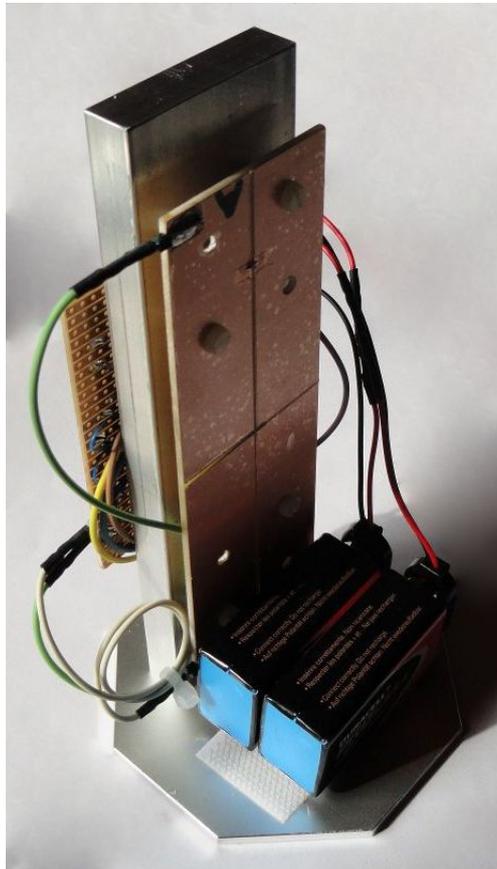


FIGURE 41 – Balise vue de l'avant avec ses LEDs infrarouges

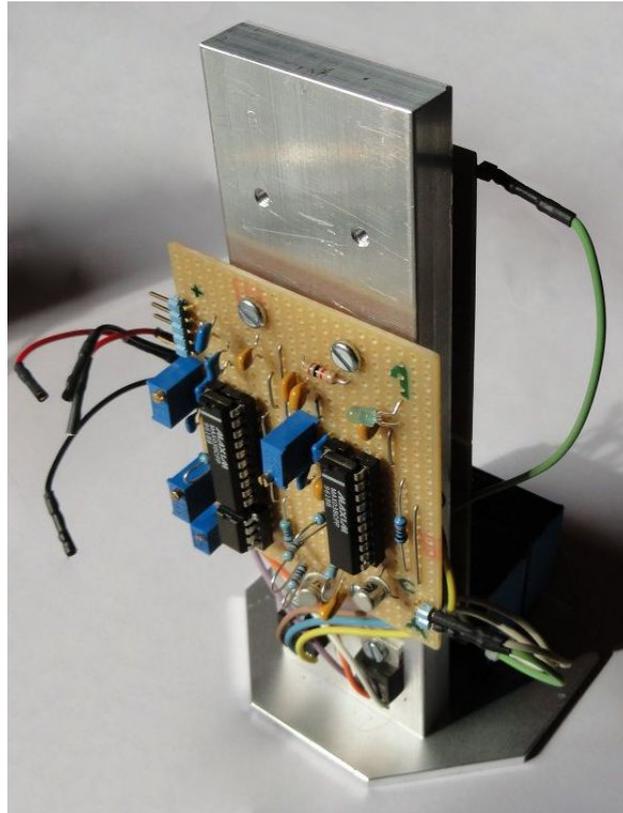


FIGURE 42 – Balise vue de l'arrière avec son circuit générateur

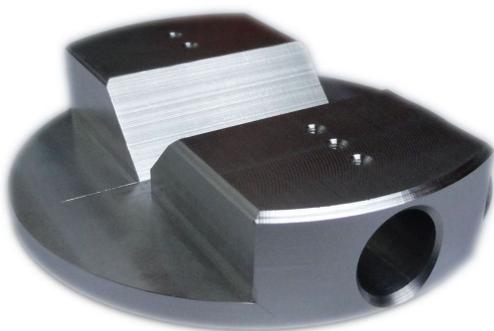


FIGURE 43 – Partie tournante



FIGURE 44 – Support moteur



FIGURE 45 – Support moteur avec miroir, lentille, moteur et partie tournante montée sur axe

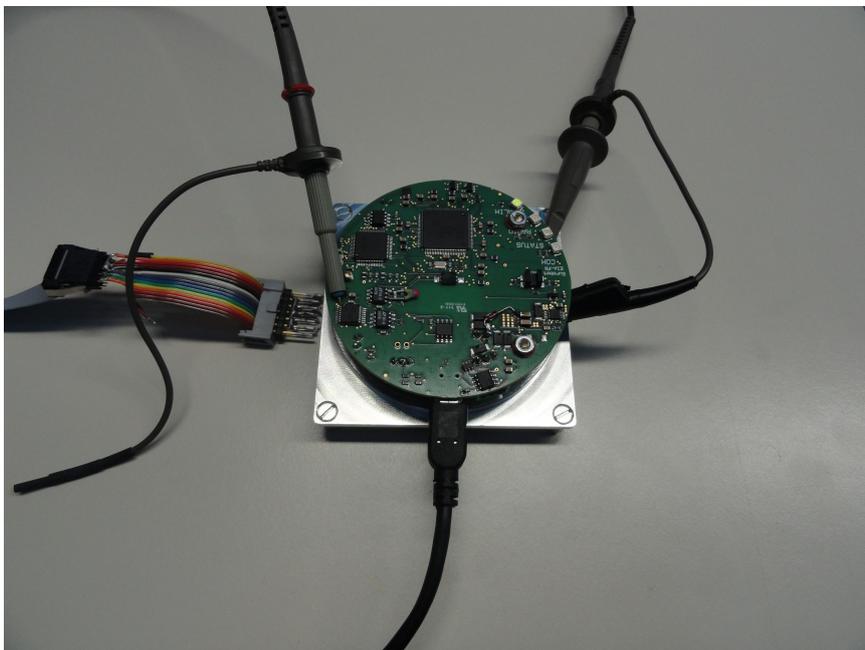


FIGURE 46 – Carte électronique montée sur son support (ici lors d'une série de tests)

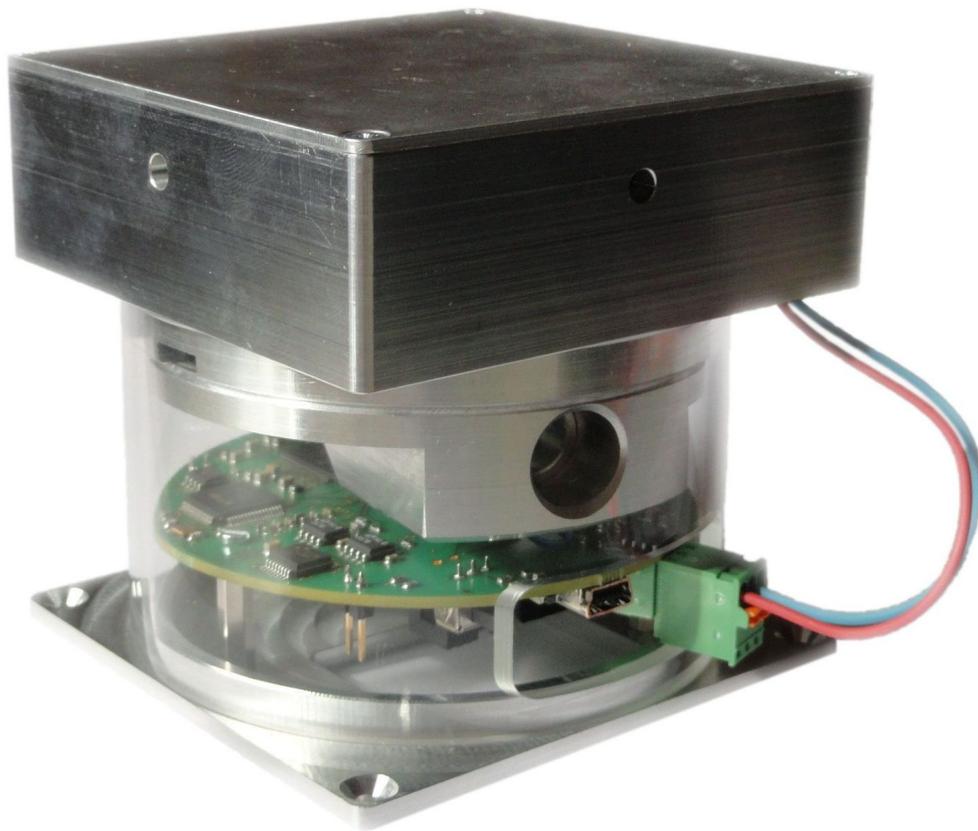


FIGURE 47 – Module complètement assemblé

5.2 Logiciel

5.2.1 Présentation du MCU / DSP TI F28069

Le micro-contrôleur 32bits F28069 de TI dispose de deux cœurs d'exécution complètement indépendants : le CPU et le CLA, un coprocesseur faisant office de DSP.

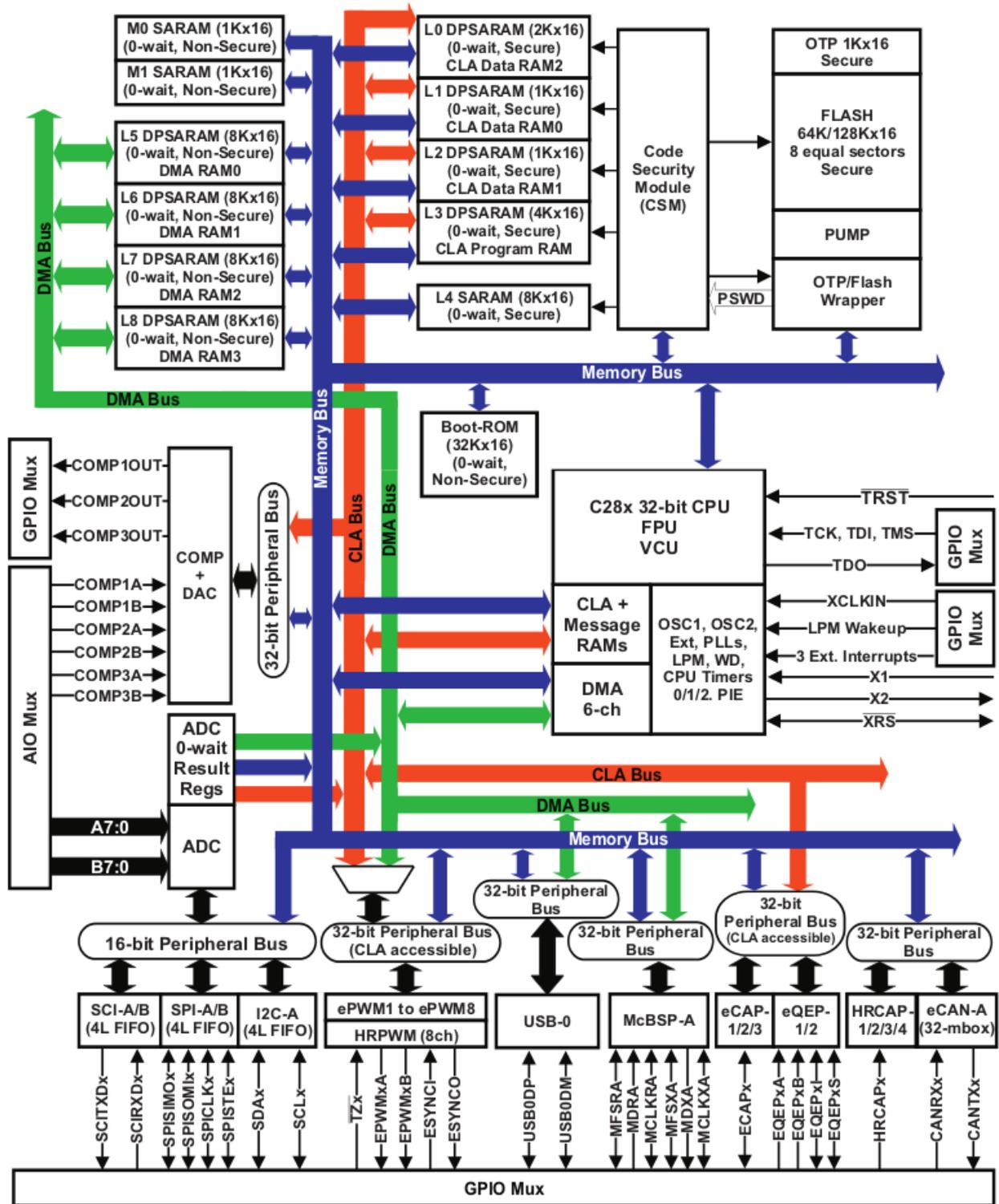


FIGURE 48 – Schéma bloc du micro-contrôleur F28069

Le CPU dispose d'une FPU (Floating Point Unit) et de divers périphériques : ADC 12 bits, contrôleur CAN, comparateurs analogiques, UART/SCI (pour radio et contrôleur USB), Timers, PWM, PIE (permet de multiplexer les interruptions).

Le CLA est un cœur d'exécution complètement indépendant. Sa configuration, sa mémoire de programme et de données sont néanmoins initialisées par le CPU.

5.2.2 Architecture logicielle

Notre application est essentiellement séquentielle et ne nécessite pas une multitude de processus ou threads s'exécutant en pseudo-parallèles. Une implémentation sans OS (Operating System) a donc été choisie.

Nous pouvons séparer notre application en deux sections distinctes : une avec des contraintes hard realtime, l'autre soft realtime. La distinction se fait ainsi : lors du dépassement d'une contrainte hard realtime, les performances du système se dégradent de manière notables, notamment une perte de précision dans la mesure de la position. Un dépassement soft realtime ne doit pas pénaliser ainsi le système. Il peut par exemple s'agir d'un retard dans la livraison d'une réponse à une requête CAN ou d'envoi de message radio.

Nous classons donc ainsi les tâches à réaliser :

Hard realtime : Traitement numérique du signal

Soft realtime : – Calcul des angles

- Détection et éventuellement correction des erreurs
- Triangulation
- Communication CAN
- Gestion radio

La partie hard realtime est implémentée sur le CLA en assembleur afin de maîtriser et optimiser les timings. Il dispose de l'accès aux périphériques nécessaires à cette tâche : ADC et Timer. Le temps maximal de réponse et d'exécution de l'algorithme ne doit pas dépasser la période d'échantillonnage car il doit être exécuté pour chaque valeur mesurée par l'ADC.

La partie soft realtime est quant à elle implémentée sur le CPU qui dispose de l'accès à tout les périphériques. Il aura également pour tâche d'initialiser le CLA au démarrage. Le langage choisi est le C pour optimiser le temps d'implémentation et de modification ultérieure. Nous éviterons d'interrompre le CPU à tout moment, c'est-à-dire pour chaque échantillon mesuré. Tous les calculs qui sont nécessaires sur chaque échantillon seront donc exécuté par le CLA. Le CPU sera interrompu uniquement sur les changements d'état qui nécessitent son action.

Le polling est évité de manière à diminuer au maximum les latences. Cette tâche est facilitée par l'offre pléthorique d'interruptions de ce MCU. La seule qui a manqué est une interruption sur le changement d'état des comparateurs analogiques (barrière optique et niveau batterie), nous obligeant à faire ici du polling, mais de manière limitée.

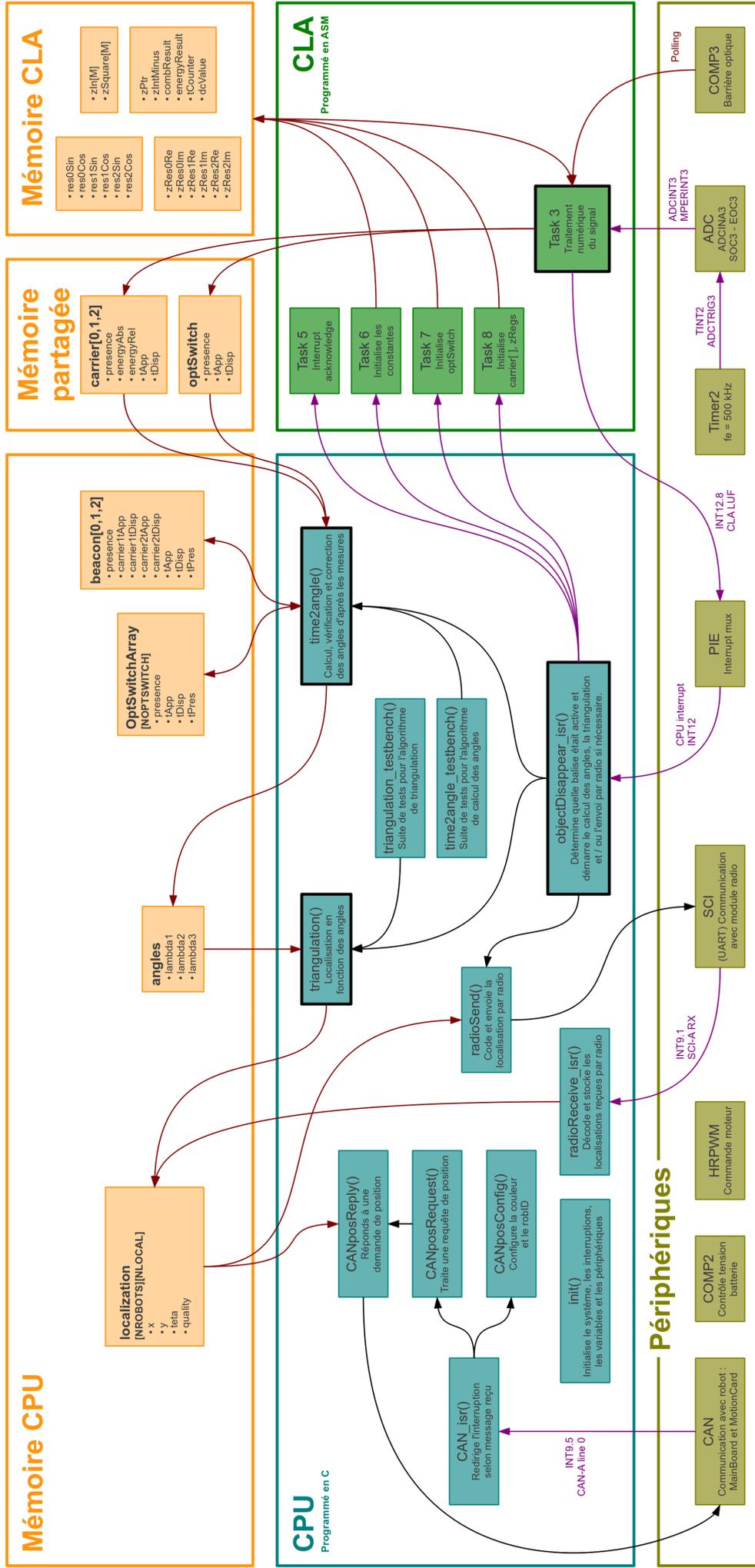


FIGURE 49 – Schéma simplifiée de l'architecture logicielle

Nous voyons sur la FIGURE 49 qu'il y a trois points d'entrée dans le CPU correspondant à trois interruptions : disparition d'un objet (porteuse ou barrière optique) signalée par le CLA, réception d'un message CAN et réception d'un message radio. Le CPU est ainsi programmé pour réagir avec une latence minimale à une demande.

Le CLA possède quant à lui cinq points d'entrées, également toutes par interruptions. La première, déclenchée après chaque conversion de l'A/D, analyse le signal à la recherche des porteuses. Les algorithmes utilisés sont décrits dans la section 4.6. Les quatre autres interruptions servent à réinitialiser une partie de la mémoire du CLA. Par exemple les constantes à l'initialisation ou les registres après détection d'une balise.

Un soin particulier a été pris lors de l'accès à la mémoire partagée pour éviter les races conditions.

5.2.3 Coprocesseur DSP (CLA)

Le CLA (Control Law Accelerator) possède 8 tâches programmables qui sont déclenchées par interruption (périphérique ou logicielle).

Quatre de ces tâches sont des initialisations de la RAM de données du CLA : mise à zéro des registres z, initialisation des constantes, des pointeurs sur les structures de données etc. Ces tâches sont exécutées par interruptions logicielles (envoyées par le CPU) à l'initialisation. Après chaque détection de balise, l'une de ces tâches réinitialise les registres pour garantir la stabilité (voir section 4.6.6).

Une autre tâche, dédiée au traitement numérique du signal, est exécutée sur chaque échantillon de l'ADC (interruption sur fin de conversion). Elle se décompose ainsi :

- Time counter incrementation and interrupt flag clearance
- Initialize pointers to M-stage z registers
- Load and store ADC result and his squared value in M-stage z registers
- SDFT comb
- Time domain window's energy calculation
- **carrier 0** : - SDFT complex resonator
 - Frequency domain energy calculation
 - Absolute and relative energy comparison and carrier detection
 - Time positioning and CPU interrupt generation
- **carrier 1** : - SDFT complex resonator
 - Frequency domain energy calculation
 - Absolute and relative energy comparison and carrier detection
 - Time positioning and CPU interrupt generation
- **carrier 2** : - SDFT complex resonator
 - Frequency domain energy calculation
 - Absolute and relative energy comparison and carrier detection
 - Time positioning and CPU interrupt generation
- Time positioning and CPU interrupt generation (optical switch)

Cette tâche, décrite en assembleur, possède 158 instructions. Avec un cycle par instruction et deux cycles nécessaires pour le changement et démarrage de la tâche, 160 cycles maximum sont nécessaires à l'exécution. Avec une fréquence d'horloge maximale de 80 MHz, nous obtenons une fréquence d'échantillonnage maximale garantie de :

$$f_{emaxgar} = \frac{80MHz}{160} = 500kHz \quad (42)$$

Exploitation des probabilités de branchement

Le code se compose de 7 branchements, ce qui réduit le nombre d'instructions minimal à exécuter à 102. Ce chemin d'exécution est le plus probable, étant donné que les signaux produits par les balises sont le plus souvent absents. Étudions ces probabilités.

Considérons un angle d'ouverture du récepteur de 10° , ce qui est une bonne estimation. Avec 3 balises actives sur un tour, on trouve une balise active environ 10% du temps. Une balise est composée de deux porteuses actives, chacune nécessitant 18 cycles supplémentaires pour son traitement.

Considérons la barrière optique également active 10% du temps. Cette estimation est largement surdimensionnée mais peu influente, uniquement 2 cycles supplémentaires étant nécessaires.

Le nombre de cycle moyen nécessaire devient :

$$0.8 \cdot 104 + 0.1 \cdot (104 + 2 \cdot 18) + 0.1 \cdot (104 + 2) = 107.8 < 108 \quad (43)$$

On en déduit une fréquence d'échantillonnage maximale de :

$$f_{emax} = \frac{80MHz}{108} > 740kHz \quad (44)$$

Dans le pire des cas, une tâche nécessite $104 + 2 * 18 + 2 = 142$ cycles pour son exécution alors qu'elle n'en a que 108 à disposition avant l'arrivée du prochain échantillon et donc d'une interruption. Si aucune précaution n'est prise, il y aura rapidement un débordement du flag d'interruption (1 bit). Il est donc nécessaire de stocker les échantillons provenant de l'A/D dans une mémoire tampon FIFO.

Cette optimisation n'a pas encore été implémentée par manque de temps mais pourrait l'être prochainement pour son gain d'environ 50% sur la résolution temporelle (et donc angulaire) de la position des balises et de la barrière optique.

6 Conclusion

Le système n'est à l'heure actuelle pas encore terminé et donc non fonctionnel. Une critique objective est donc difficile à émettre pour l'instant. Nous pouvons néanmoins dire que le temps de développement et de réalisation a été plus long qu'escompté. Le système n'était pas prêt pour la Coupe de Belgique, ce qui a causé préjudice à l'équipe qui s'attendait à un système fonctionnel d'après la planification interne. Heureusement rien de grave, beaucoup de sous-systèmes n'étant pas encore opérationnels à ce moment-là.

Il est à rappeler ici que ce projet dépasse du cadre d'un simple projet de semestre en se rattachant à la participation à l'équipe Eurobot. Ceci implique que le projet a été commencé avant le départ des projets de semestre et se terminera après ceux-ci. Si le système se révèle fonctionnel et efficace, il sera gardé et amélioré au sein de l'équipe Eurobot lors de la préparation des futurs concours et pourrait faire l'objet d'autres travaux de semestre. Ceci implique également que toutes les facettes du projet n'ont pas été développées et décrites dans le cadre de ce document.

Un point qui m'a particulièrement marqué est la difficulté de la conception d'une plate-forme matérielle parallèlement au développement des algorithmes et techniques à utiliser. Les besoins sont alors encore méconnus et peuvent être sous-estimés ou sur-estimés.

Avoir plus de temps à disposition pour définir, dimensionner et simuler toutes les variables aurait certainement abouti à une solution plus optimisée, voire plus performante.

Ce projet devrait finalement apporter un système de positionnement précis et fiable au robot de l'équipe RTFM pour ses prochaines participations aux concours Eurobot. Ce système sera probablement encore amélioré et optimisé lors de la préparation des prochaines éditions.

De mon côté ce projet m'a permis de mettre en pratique une grande partie des connaissances acquises durant ma formation bachelor à l'EIA-FR et à la Hochschule München. Notamment en réunissant de nombreuses matières : traitement numérique du signal, conception analogique, RF, systèmes embarqués, programmation C et assembleur, simulation, validation et optimisation des algorithmes à l'aide de Matlab, conception et réalisation d'un circuit imprimé, recherche de solutions et composants, gestion de projet etc. Bref tout ce qu'un ingénieur en électronique est amené à faire au quotidien.

6.1 Gestion du projet

La gestion de projet fût particulièrement importante tout au long du projet due au grand nombre de personnes impliquées et à la synchronisation nécessaire, que ce soit au niveau du temps que de la technique. Une coordination a également été nécessaire avec les autres membres de l'équipe Eurobot, notamment au niveau des interfaces, des délais et des buts à atteindre.

Personnes impliquées dans le projet :

Michaël ANSORGE : Monsieur Ansonge a suivi le projet du début à la fin, nous nous sommes vus quasiment tous les mercredis. Ses conseils, idées et avis tant au niveau traitement du signal que dans d'autres domaines ont beaucoup apportés au projet.

Wolfram LUTHARDT : Monsieur Luithardt a plutôt œuvré du côté Eurobot (dont il est le responsable vis-à-vis de l'EIA-FR) que du projet de semestre. Il a toutefois participé aux discussions quant au choix d'une plate-forme matérielle et a régulièrement pris des nouvelles quant à l'avancement du projet.

Yves BIRBAUM : Monsieur Birbaum a dessiné le schéma électronique à l'aide des outils Mentor, conçu et monté le PCB. Il m'a également aidé pour le dépannage de la carte et m'a donné de nombreux conseils notamment sur la préservation de l'intégrité du signal. Il a participé avec Monsieur Chassot à l'élaboration des balises.

Alfred NEUHAUS : Monsieur Neuhaus a aidé Monsieur Birbaum pour le montage des PCBs et a fait les modifications sur les cartes.

Paul-Hervé CHASSOT : Monsieur Chassot a conçu et fabriqué les balises.

Gérard GUEX : Monsieur Guex a fait des modifications mécaniques sur les modules et m'a aidé au montage de la partie optique.

Kim BRUGNETTI : Monsieur Brugnetti a conçu la mécanique d'après mes besoins, l'a dessiné puis l'a mise en plan. Il a également fait usiner les pièces.

6.2 Améliorations et travaux futurs

Il y a encore de nombreux points qui nécessiteraient un approfondissement et de futures investigations. Il sera nécessaire de faire à nouveau le point à la fin du projet, c'est-à-dire après les concours Eurobot, afin de pouvoir reprendre et continuer le projet l'année prochaine lors de la préparation des concours 2013. Nous pouvons déjà en citer quelques-uns :

Le moteur DC devrait être commandé à l'aide de la modulation PWM prévu. Il est actuellement commandé en continue dû à du bruit analogique provenant du signal PWM qui dégrade l'intégrité du signal reçu par la photodiode IR. Ce bruit est difficile à mesurer, la commande PWM a donc été remplacée le temps d'identifier, d'évaluer, de caractériser et d'éventuellement diminuer les autres sources de bruit. Une piste pour diminuer le bruit est de rapprocher physiquement l'amplificateur à transconductance de la photodiode IR dont la position est contrainte. Le courant traversant les lignes les reliant est très faible et du bruit est créé par couplage capacitif et / ou inductif.

La communication entre modules par radio reste à implanter. Le matériel est prêt, le protocole de communication reste à définir et le logiciel à concevoir.

Étude des avantages et inconvénients de l'utilisation d'un moteur pas-à-pas vis-à-vis d'un moteur DC. Notamment vibration, commande, stabilité de la vitesse de rotation, facilité et fiabilité de la mesure de

l'angle de l'axe du moteur. L'utilisation d'un codeur est envisageable. Meilleur équilibrage mécanique et axe plus stable, éventuellement utilisation d'un roulement. Actuellement monté directement sur l'axe moteur.

Le choix des porteuses ainsi que la taille M de la fenêtre d'analyse devrait faire l'objet d'une étude plus approfondie afin de déterminer les éventuelles valeurs optimales en fonction de la répartition statistique du bruit dans les fréquences et des possibilités des balises génératrices.

Chercher un remplaçant pour les amplificateurs opérationnels qui possèdent un GBW trop faible. L'AOP de l'amplificateur à transconductance peut ainsi se mettre à osciller. Une capacité de contre-réaction a été anticipée lors de la phase de conception pour éviter ce problème. Celle-ci s'est révélée sous-dimensionnée malgré les simulations et l'AOP a commencé à osciller. Sa valeur a été modifiée de 1 à 3 pF. La fréquence de coupure de ce filtre est ainsi abaissée et il affaiblit le haut de la bande utile.

La génération des porteuses par les balises devraient être faite par synthèse numérique afin de diminuer la distorsion harmonique et d'améliorer la stabilité dans le temps par rapport à la génération analogique actuelle. Une distorsion harmonique est également introduite par la source de tension à transistor bipolaire actuellement utilisée (voir schéma en annexe) dont la tension est trop dépendante du courant la traversant. Un PCB devrait également être produit qui donnera un aspect plus soigné que le veroboard.

L'étude et l'implantation d'un filtre (type Kalman [5], Markov ou Monte Carlo [9]) permettant la fusion d'informations bruitées et imprécises (positions provenant des mesures directes et odométriques) dans le but de produire une estimation statistiquement optimale de l'état dynamique du système. Le concept global de navigation du robot devra probablement être revu pour permettre cette intégration.

Comme nous venons de le voir, il y a encore un certain nombre de travaux à effectuer. Certains sont de courtes durées et seront réalisés prochainement alors que d'autres, tels que les deux derniers, prendront plusieurs mois. La difficulté et les connaissances nécessaires à ces travaux sont également variables et devront être prises en compte avant d'entreprendre l'un d'eux.

6.3 Remerciements

La réalisation de ce projet n'aurait pas été possible sans la disponibilité de nombreuses personnes. C'est pourquoi je souhaite remercier particulièrement mes deux professeurs co-responsables : Monsieur Michaël ANSORGE et Monsieur Wolfram LUITHARDT. Je souhaite également remercier le personnel du bureau de construction : Messieurs Yves BIRBAUM, Alfred NEUHAUS, Paul-Hervé CHASSOT et Gérard GUEX ainsi que leurs apprentis ayant participé au projet. Monsieur Kim BRUGNETTI, de l'entreprise Johnson Electric, pour la conception des pièces mécaniques et ses apprentis pour leur réalisation. Merci également à tous les professeurs et collaborateurs qui ont répondu à mes interrogations durant le projet.

Table des figures

1	Table Eurobot 2012 [1]	3
2	Schéma représentant la localisation présent dans [2]	5
3	Hauteurs imposées par le règlement du concours [1]	6
4	Modélisation de l'environnement de l'Absolute_LPS	6
5	Schéma de la table de jeu	13
6	Architecture matérielle complète	14
7	Signaux reçus et interprétations	17
8	Représentation des flux de données	19
9	Emplacement et numérotation des balises	20
10	Attribution des porteuses	21
11	Distribution et répartition des porteuses	22
12	Représentation temporelle du signal émis par la balise 0	22
13	Représentation temporelle du signal émis par la balise 1	23
14	Représentation temporelle du signal émis par la balise 2	23
15	Spectre des fréquences	24
16	Schéma bloc de la chaîne d'acquisition	25
17	Reverse Light Current vs. Irradiance	26
18	Spectral Sensitivity vs. Wavelength	26
19	Schéma du pré-amplificateur à transconductance	27
20	Schéma du filtre passe-haut	28
21	Plage programmable du LTC1564	29
22	Schéma du circuit LTC1564	29
23	Schéma de simulation LTspice IV	31
24	Exemple avec un sinus à 120kHz d'amplitude $A = 10\mu A$	32
25	Réponse fréquentielle complète	32
26	Simulation d'une période du signal reçu d'une balise ($A = 10\mu A$)	33
27	Simulation de la FFT du signal reçu d'une balise	33
28	Transformée en Z de la SDFT	34
29	Transformée en Z du résonateur complexe implémenté à l'aide de nombres réels	35
30	Représentation temporelle	36
31	FFT du signal	36
32	SDFT porteuse 0	36
33	SDFT porteuse 1	37
34	SDFT porteuse 2	37
35	Énergies absolues	37
36	Énergies relatives	38
37	Présences des porteuses	38
38	Transformée en Z	39
39	Transformée optimisée selon contraintes	40
40	Schéma bloc de l'électronique des modules	42
41	Balise vue de l'avant avec ses LEDs infrarouges	43
42	Balise vue de l'arrière avec son circuit générateur	44
43	Partie tournante	44
44	Support moteur	44
45	Support moteur avec miroir, lentille, moteur et partie tournante montée sur axe	45
46	Carte électronique montée sur son support (ici lors d'une série de tests)	45
47	Module complètement assemblé	46

48	Schéma bloc du micro-contrôleur F28069	47
49	Schéma simplifiée de l'architecture logicielle	49

Références

- [1] Eurobot. *EUROBOT 2012 rules*. Planète Sciences, 2011.
- [2] Michaël DEFFERRARD. *Système de positionnement – Cahier des charges pour PS5*. EIA-FR, 2011.
- [3] Richard G. LYONS. *Understanding digital signal processing*. Prentice Hall, 2010.
- [4] C. MEIER, A. TERZIS, and S. LINDENMEIER. *A high precision wideband local positioning system at 24GHz*. IEEE, 2006.
- [5] Rudy NEGENBORN. *Robot Localization and Kalman Filters*. Utrecht University, 2003.
- [6] Vincent PIERLOT and Marc VAN DROOGENBROECK. *A Simple and Low Cost Angle Measurement System for Mobile Robot Positioning*. INTELSIG, Laboratory for Signal and Image Exploitation, Montefiore Institute, University of Liège, Belgium, 2009.
- [7] Matthew Stephen REYNOLDS. *Low Frequency Indoor Radiolocation*. Massachusetts Institute of Technology, 2003.
- [8] Jens SCHROEDER, Stefan GALLER, and Kyandoghene KYAMAKYA. *A Low-Cost Experimental Ultra-Wideband Positioning System*. IEEE, 2005.
- [9] Sebastian THRUN, Dieter FOX, Wolfram BURGARD, and Frank DELLAERT. *Robust Monte Carlo Localization for Mobile Robots*. Carnegie Mellon University, 2000.
- [10] P. TOMÉ, C. ROBERT, R. MERZ, C. BOTTERON, A. BLATTER, and P.-A. FARINE. *UWB-based Local Positioning System : from a small-scale Experimental Platform to a large-scale Deployable System*. IEEE, 2006.

A Code de la simulation Matlab du traitement numérique du signal

```

1 clear all; close all; clc;
3 PlotActive = 1 ; % 1: plot active, 0: plot inactive
5 %% Frequency and amplitude definitions
7 f0 = 87.5 * 10^3 ; % carrier frequencies
8 f1 = 112.5 * 10^3 ;
9 f2 = 137.5 * 10^3 ;
11 A = 0.060 ; % sinus amplitude [V] (max 3.3V / 2)
12 NoiseLevel = 20 ; % white noise level [mV peak-peak]
13
14 fe = 500 * 10^3 ; % sampling frequency
15
16 M = 40 ; % window size : used for FFT and SDFT
17 m = 1:1:M ;
18
19 m0 = 7 ; % 7/40 * 500 = 87.5 kHz = f0
20 m1 = 9 ; % 9/40 * 500 = 112.5 kHz = f1
21 m2 = 11 ; % 11/40 * 500 = 137.5 kHz = f2
22
23 %% Index definitions
24
25 N = 280 ; % total number of samples
26 n = 1:1:N ;
27 t(n) = n / fe * 10^6 ; % time in us
28
29 %% Signals' definition
30
31 xp0(n) = A * sin(2*pi * f0/fe * n) ; % carriers
32 xp1(n) = A * sin(2*pi * f1/fe * n) ;
33 xp2(n) = A * sin(2*pi * f2/fe * n) ;
34
35 xb0(n) = xp0(n) + xp1(n) ; % beacons
36 xb1(n) = xp0(n) + xp2(n) ;
37 xb2(n) = xp1(n) + xp2(n) ;
38
39 xb(n) = xb0 ; % beacon's signal to analyse
40 xbmov(n) = (rand(1,N)-0.5)/500 * NoiseLevel ; % create some noise
41
42 for k = 1:1:N % create ramp signal (emulate beacons appearance and disappearance)
43     if k >= 40 && k < 80
44         xbmov(k) = xbmov(k) + (k-29) / 40 * xb(k) ;
45     elseif k >= 80 && k < 200
46         xbmov(k) = xbmov(k) + xb(k) ;
47     elseif k >= 200 && k < 240
48         xbmov(k) = xbmov(k) + (1 - (k-199)/40) * xb(k) ;
49     end
50 end
51
52 x(n) = round(2^12*xbmov(n))/2^12 ; % 12 bits quantification error
53
54 %% FFT with sliding window (spectrum verification)
55
56 WinStart = 100 ; % start of the analysis' window
57 Xfft(m) = fft(x(WinStart:WinStart+M-1)) ;
58 f(m) = (m-1) * fe / M / 10^3 ; % frequency in kHz

```

```

59 %% Sliding DFT
61 X(3,N) = 0 ;
63 for l = 1:1:3
65     switch l
67         case 1, m = m0 ;
68         case 2, m = m1 ;
69         case 3, m = m2 ;
71     end
73     X(1,1) = exp(2i*pi*m/M) * (0 + x(1) - 0) ;
75     for k = 2:1:N
76         if k > M
77             X(1,k) = exp(2i*pi*m/M) * (X(1,k-1) + x(k) - x(k-M)) ;
78         else
79             X(1,k) = exp(2i*pi*m/M) * (X(1,k-1) + x(k) - 0) ;
80         end
81     end
83 %% Time domain and frequency domain energy measurment
85 Ex(N) = 0 ;
86 for k = M+1:N
87     Ex(k) = sum(x(k-M+1:k).^2) ;
88 end
89 EX(3,N) = 0 ;
90 for k = 1:3
91     EX(k,:) = abs(X(k,:)).^2 / M * 2 ;
92 end
94 %% Relative energy computation
96 Er(3,N) = 0 ;
97 for k = 1:3
98     Er(k,:) = EX(k,:) ./ Ex ;
99 end
101 %% Presence validation
103 P(3,N) = 0 ;
104 for l = 1:3
105     P(1,1) = 0 ;
106     for k = 2:1:N
107         if P(1,k-1) == 1
108             if Er(1,k) >= 0.15 && Er(1,k) <= 0.6
109                 P(1,k) = 1 ;
110             else
111                 P(1,k) = 0 ;
112             end
113         else
114             if Er(1,k) >= 0.25 && Er(1,k) <= 0.6
115                 P(1,k) = 1 ;
116             else
117                 P(1,k) = 0 ;
118             end
119         end
120     end
121 end

```

```

121     end
122 end
123 %% Beacon detection and time positioning
125 Pt(3,2) = 0 ;
127 for l = 1:3
129     for k = 1:1:N
131         if P(1,k) == 1 && Pt(1,1) == 0
132             Pt(1,1) = k ;
133         elseif P(1,k) == 0 && Pt(1,1) ~= 0 && Pt(1,2) == 0
134             Pt(1,2) = k-1 ;
135         end
136     end
137 end
138 Beacon = sum(sum(Pt))/4 / fe * 10^6 ;
139 if Pt(1,1) ~= 0 && Pt(2,1) ~= 0 && Pt(3,1) == 0
141     sprintf('Beacon 1 was detected at %f us', Beacon)
142 end
143 if Pt(1,1) ~= 0 && Pt(2,1) == 0 && Pt(3,1) ~= 0
144     sprintf('Beacon 2 was detected at %f us', Beacon)
145 end
146 if Pt(1,1) == 0 && Pt(2,1) ~= 0 && Pt(3,1) ~= 0
147     sprintf('Beacon 3 was detected at %f us', Beacon)
148 end
149
151 %% Plots
153 if PlotActive == 1
155     Aplot = A + NoiseLevel/1000 ;
157     subplot(4,2,1)                % time
158     stem(t,x)
159     title(sprintf('Time plot, fe=%dkHz, N=%d samples', fe/1000,N))
160     xlabel('time [us]'), ylabel('x(t) [V]')
161     axis([0 max(t) -2*Aplot 2*Aplot])
163     subplot(4,2,2)                % FFT
164     stem(f(1:M/2),abs(Xfft(1:M/2))/M*2) % /M
165     title(sprintf('FFT fe=%dkHz, M=%d, WinStart=%dus', fe/1000,M, WinStart/fe*10^6))
166     xlabel('frequency [kHz]'), ylabel('|X|(f) [V/Hz]')
167     axis([0 max(f(1:M/2)) 0 Aplot])
169     subplot(4,2,3)                % SDFT on f0
170     stem(t,abs(X(1,:))/M*2,'g')
171     title(sprintf('SDFT m=%d, fe=%dkHz, M=%d, fmes=%dkHz',m0, fe/1000,M,m0/M*fe/1000))
172     xlabel('time [us]'), ylabel(sprintf('X%d(t) [V]',m0))
173     axis([0 max(t) 0 Aplot])
175     subplot(4,2,4)                % SDFT on f1
176     stem(t,abs(X(2,:))/M*2,'m')
177     title(sprintf('SDFT m=%d, fe=%dkHz, M=%d, fmes=%dkHz',m1, fe/1000,M,m1/M*fe/1000))
178     xlabel('time [us]'), ylabel(sprintf('X%d(t) [V]',m1))
179     axis([0 max(t) 0 Aplot])

```

```

181 subplot(4,2,6) % SDFT on f2
stem(t,abs(X(3,:))/M*2,'r')
183 title(sprintf('SDFT m=%d, fe=%dkHz, M=%d, fmes=%dkHz',m2,fe/1000,M,m2/M*fe/1000))
xlabel('time [us]'), ylabel(sprintf('X%d(t) [V]',m2))
185 axis([0 max(t) 0 Aplot])

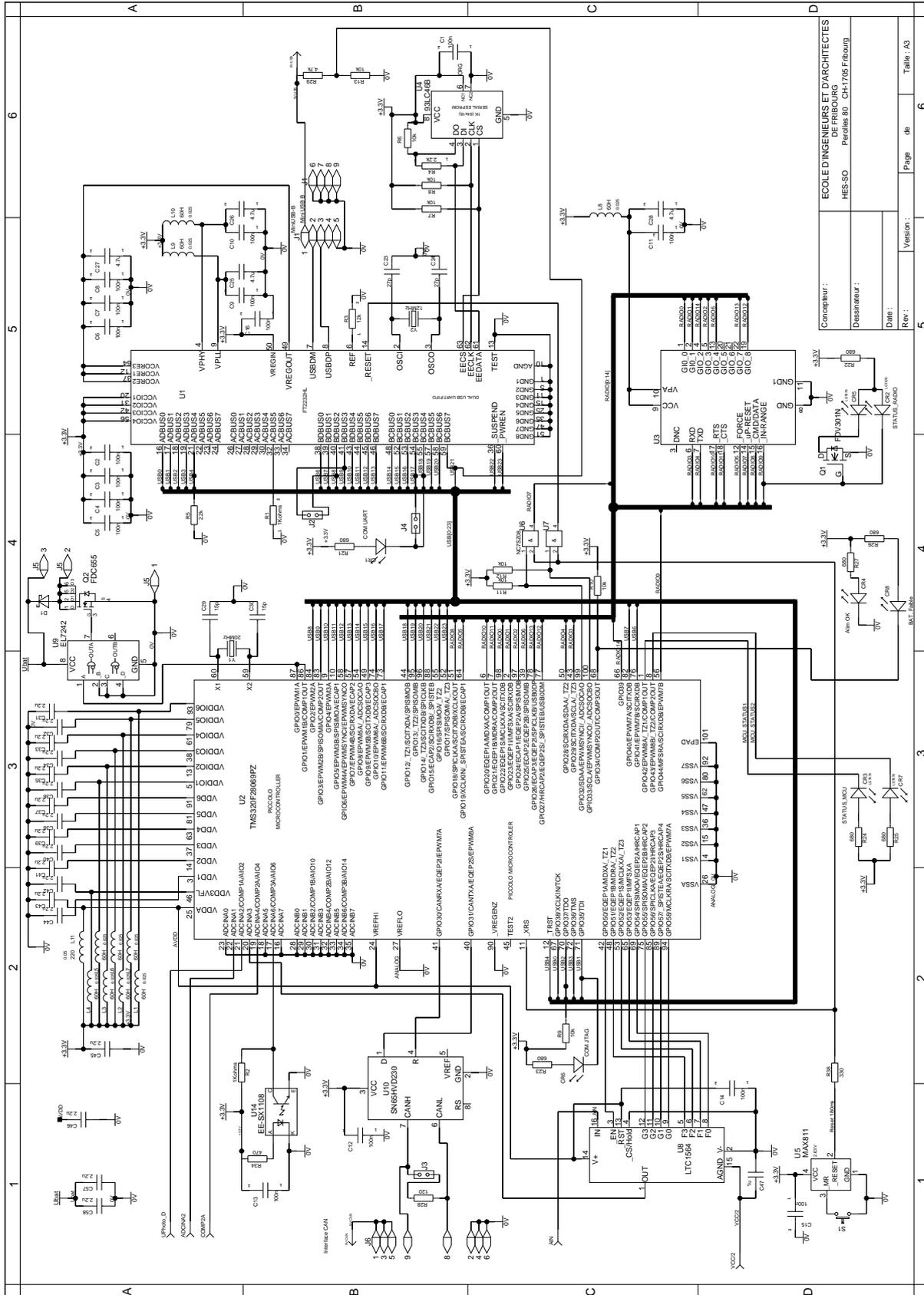
187 subplot(4,2,5); % energy
plot(t,EX(1,:), 'g',t,EX(2,:), 'm',t,EX(3,:), 'r',t,Ex, 'b')
189 legend(sprintf('X%d',m0),sprintf('X%d',m1),sprintf('X%d',m2),'x')
title(sprintf('Windowed energy plot, M=%d, fe=%dkHz, N=%d samples',M,fe/1000,N))
191 xlabel('time [us]'), ylabel(texlabel('E_{X}(t) [V^{2}*M]'))
axis([0 max(t) 0 Aplot^2*M])

193 subplot(4,2,7); % relative energy
195 plot(t,Er(1,:), 'g',t,Er(2,:), 'm',t,Er(3,:), 'r')
legend(sprintf('X%d',m0),sprintf('X%d',m1),sprintf('X%d',m2))
197 title('Relative energy')
xlabel('time [us]'), ylabel('')
199 axis([0 max(t) 0 1])

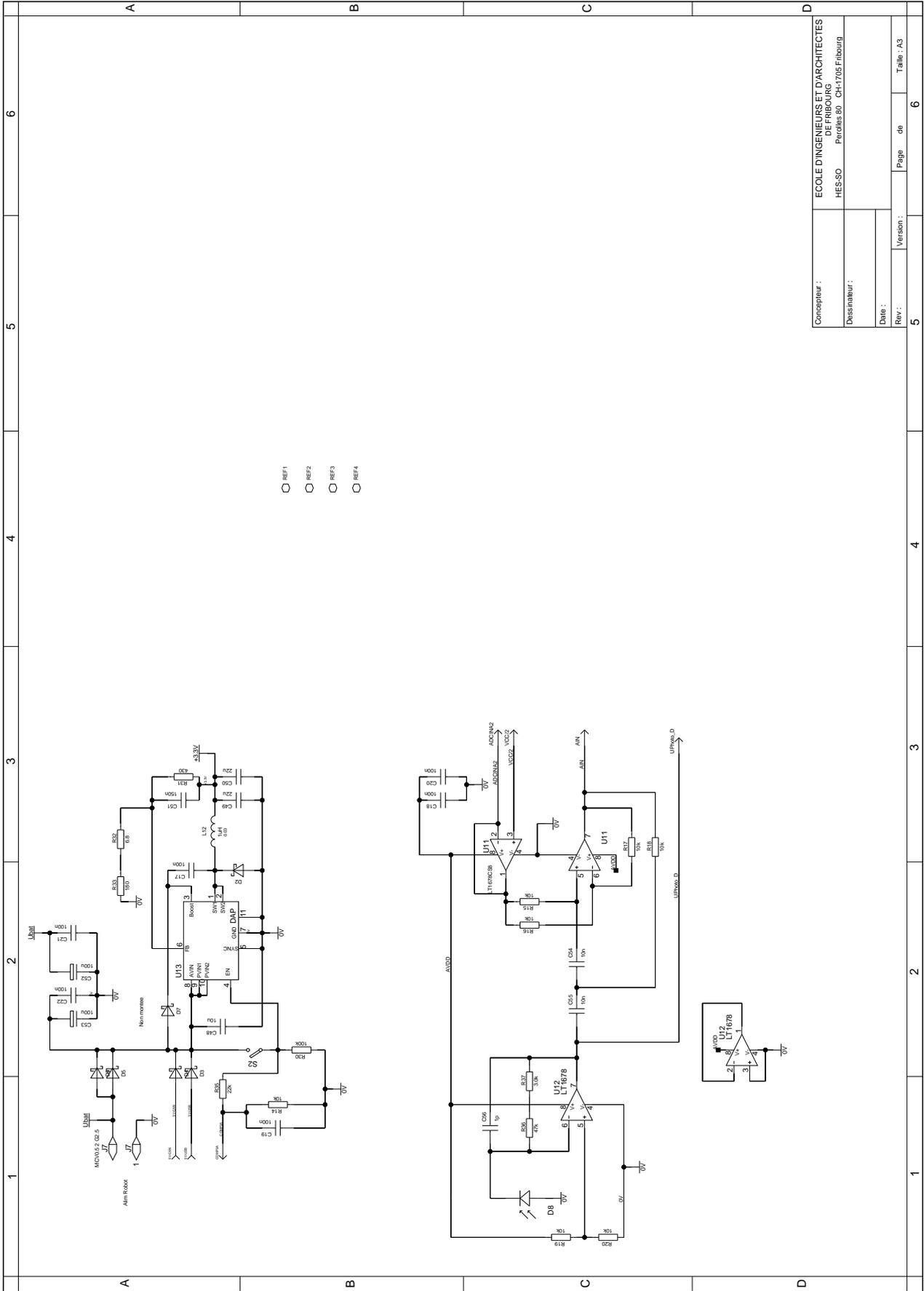
201 subplot(4,2,8); % presence
plot(t,P(1,:), 'g',t,P(2,:), 'm',t,P(3,:), 'r')
203 legend(sprintf('X%d',m0),sprintf('X%d',m1),sprintf('X%d',m2))
title('Presence detection')
205 xlabel('time [us]'), ylabel('')
axis([0 max(t) 0 1])
207
end

```

B Schéma électronique des modules



Concepteur :
Disseinateur :
Date :
Rev :
Version :
Page de 6
Taille : A3



C Schéma électronique des balises

