

Rapport de projet

Géolocalisation du robot ARES

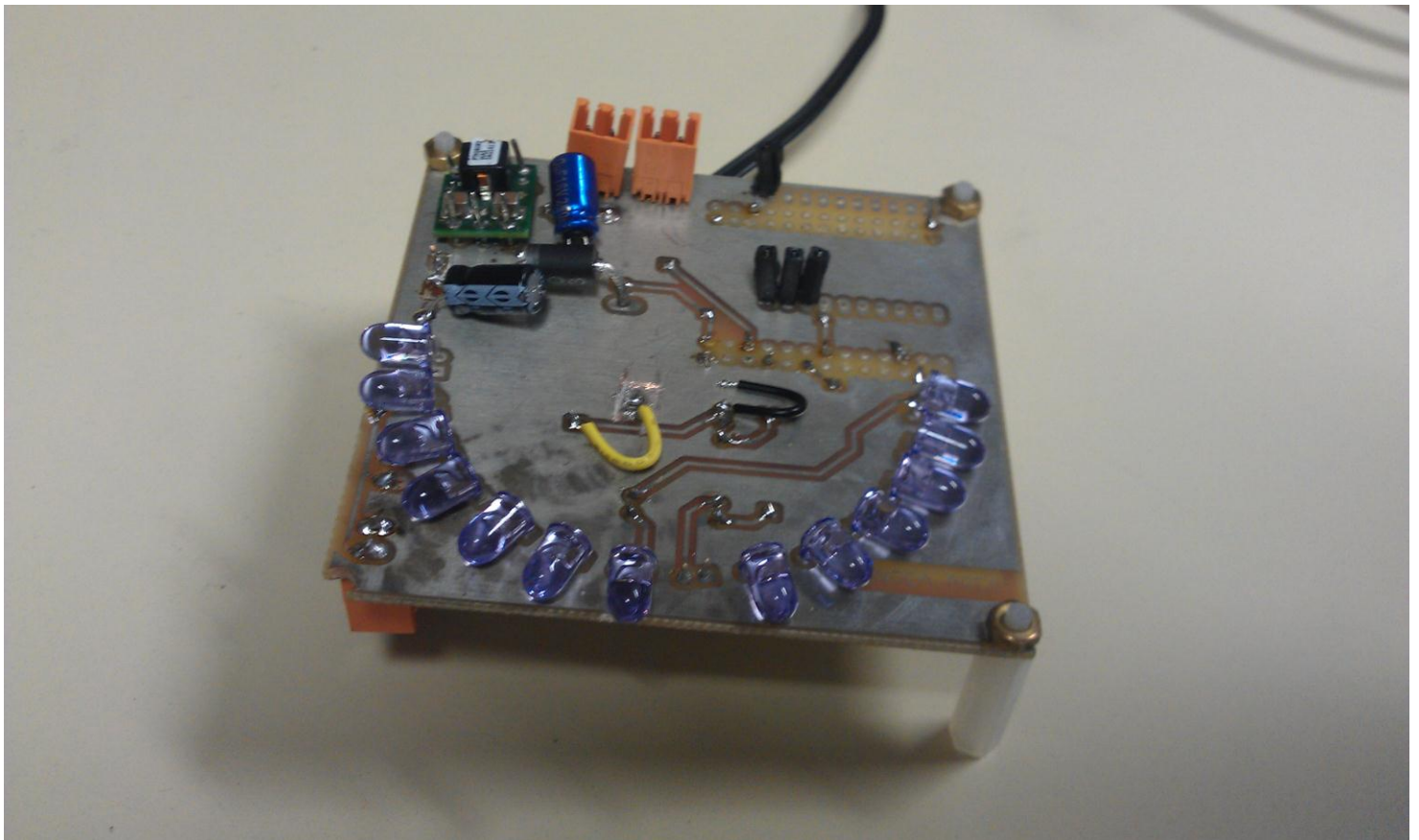


Table des matières

Table des matières	2
Partie I : Préambule	4
Partie II : La coupe de robotique 2011 : Chess up !	4
Partie III : Objectif	5
Partie IV : Cahiers des charges	5
1) Cahier des charges spécifique à la coupe	5
a) Contraintes balises « terrain »	5
b) Contraintes balises embarquées	6
c) Contraintes technologiques	6
2) Cahier des charges spécifique au projet ARES	7
Partie V : Etude préliminaire	7
1) Technologies retenues	8
a) Ultrasons	8
b) Infrarouges	9
c) Télémètre laser	10
2) Choix de la technologie	11
Partie VI : Conception hardware des balises	13
1) Choix des diodes IR	13
2) Choix des photodiodes IR	14
3) Balise émettrice	15
a) Première approche	15
b) Seconde approche	16
c) Mise en œuvre	17
4) Balise réceptrice	18
a) Première approche	18
b) Seconde approche	20
Partie V : Programmation	22
1) Approche concrète	22
2) Approche avancée	23
Partie VI : Quelles sont les raisons qui ont fait que le projet n'ait pas abouti ? Qu'aurions-nous du faire autrement ?	26
Partie VII : Les points essentiels pour faire ce genre de projet	28
Conclusion	29

ANNEXE	30
Annexe : Schématique cartes réception (balises embarquées)	30
Annexe : Schématique cartes émission (balises terrain).....	32
Annexe : Configuration des X-Bee.....	33

Partie I : Préambule

Notre projet de deuxième année entre aussi dans le cadre du projet ARES, qui a pour objectif de fournir un robot autonome, pour la coupe de robotique EUROBOT, qui s'est déroulée, cette année 2011, à la Ferté-Bernard du 01 au 04 Juin.

Le projet ARES se compose de sous projets encadrés par différents professeurs : une partie développement, où sont élaborés les programmes utiles au robot (communications, stratégie, etc.), une partie asservissement, une partie modélisation, qui a pour objectif de tester différentes stratégies et ainsi permettre de visualiser les avantages et les inconvénients de chacune.

Une dernière partie, non encadrée, intègre le projet : la vision.

Pour finir, le projet qui nous intéresse ici est la géolocalisation.



Partie II : La coupe de robotique 2011 : Chess up !

La coupe de robotique EUROBOT est un concours international ouvert aux équipes de jeunes amateurs. Le thème retenu cette année fut les échecs, mais un jeu d'échecs particulier : le but du jeu est de rapporter à son équipe le maximum de points. Pour ce faire, des pions sont disposés sur le terrain de jeu. Chaque pion rapporte des points. Des figures (reine et roi) sont aussi présentes. Le but est donc d'apposer les pions sur la couleur de sa zone de départ. Nous pouvons également empiler des pions et les surmonter d'une figure pour engranger davantage de points. Certaines zones (marquées d'un point noir) sont des zones de sécurité, zones desquelles aucun pion ne peut être déplacé suite à son dépôt (le robot adverse ne peut déplacer un pion mis par son concurrent dans cette case).

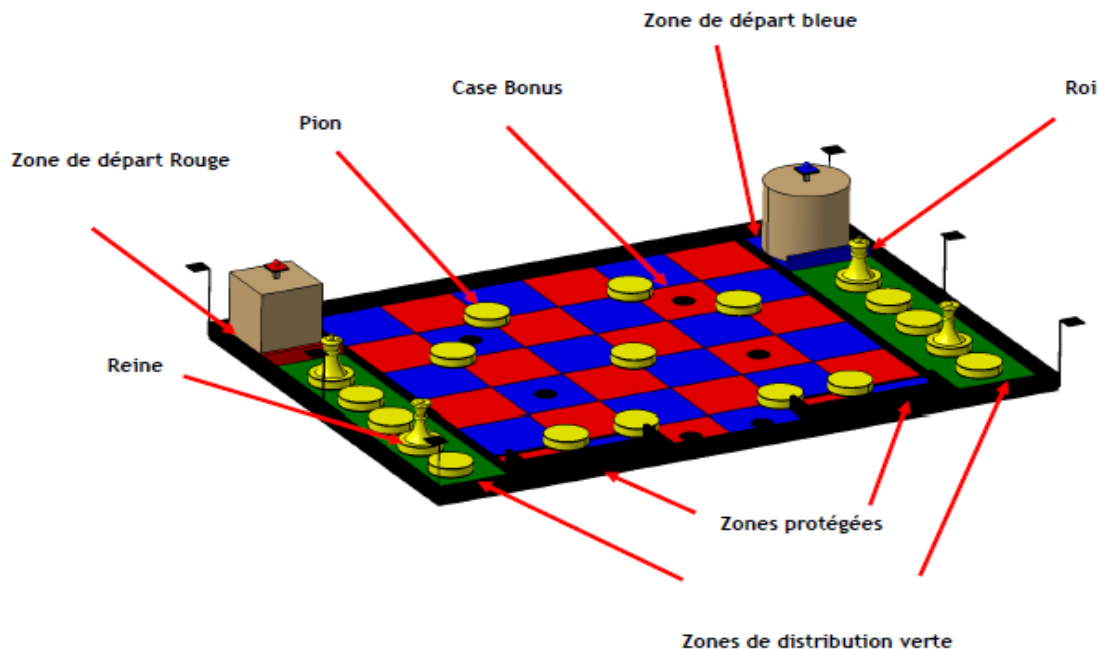


Figure 1 : terrain Chess up !

Partie III : Objectif

Nous allons donc, dans ce cadre, réaliser un système de géolocalisation du robot ARES, système qui devra donner une localisation précise (de l'ordre du centimètre) du robot, ainsi que du robot adverse ; cette dernière permettant de s'affranchir de capteurs de proximité et permettant une meilleure efficacité du robot en termes de trajectoire.

Pour satisfaire cet objectif, il nous faut prendre en compte les contraintes inhérentes à la coupe et au projet ARES (autres groupes de projets).

Partie IV : Cahiers des charges

Notre projet se doit de satisfaire deux cahiers des charges que l'on peut distinguer comme suit : cahier des charges spécifique à la coupe et cahier des charges spécifique au projet ARES.

Avant tout développement, notons que notre système de localisation se basera sur l'utilisation d'un système de balises apposées sur le pourtour du terrain et de balises embarquées dans le robot. Ce choix sera justifié par la suite, mais est nécessaire à la compréhension du cahier des charges.

1) *Cahier des charges spécifique à la coupe.*

Le système de localisation par balises doit satisfaire différentes contraintes :

a) Contraintes balises « terrain »

- 3 balises au maximum se situent sur le pourtour du terrain selon la disposition donnée figure 2.
- Les balises « terrain » peuvent être reliées entre elles par liaisons filaires.
- Les balises « terrain » se situent à 35 cm au-dessus du niveau de la zone de jeu (terrain).

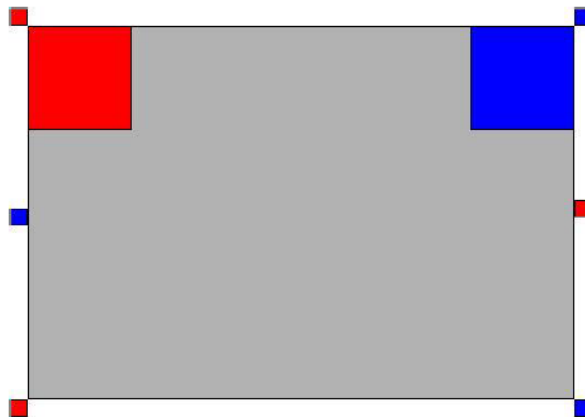


Figure 2 : disposition balises terrain et zones de départ robots.

A ces contraintes de positionnement des balises « terrain » s'ajoutent des contraintes de taille et de poids:

- Dimensions maximales des balises terrain : Lxlxh : 80x80x160xmm.
- Poids maximum : 300 g.

b) Contraintes balises embarquées

Balise embarquée robot adverse :

La principale contrainte vis-à-vis du robot adverse est de satisfaire des dimensions précises de balises : ces dimensions étant : Lxlxh : 80x80x80 mm. (cf. figure 3)

Balise embarquée robot ARES :

La contrainte concernant la balise embarquée pour notre robot est que le système doit se situer exclusivement sous la projection du mât d'accueil de la balise du concurrent, entre 35 et 43 cm de hauteur. Sa surface ne doit donc pas excéder les dimensions Lxl : 80x80 mm, comme nous pouvons le constater d'après la figure 3).

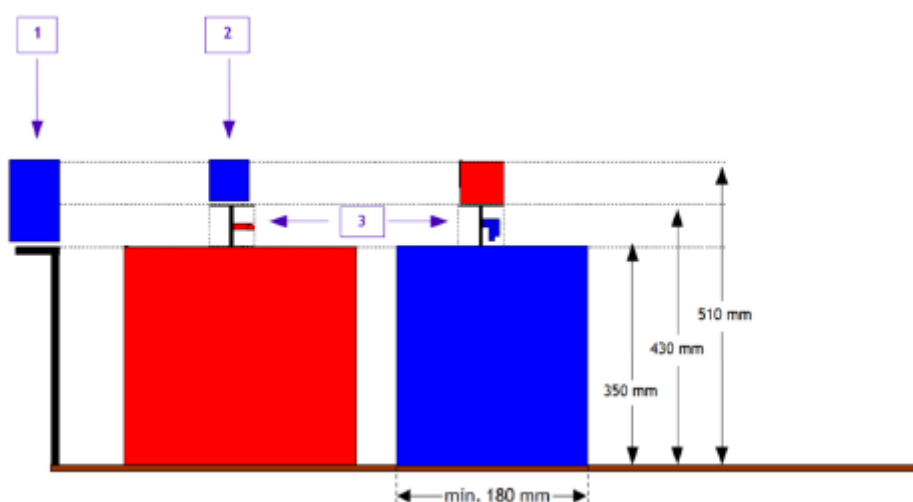


Figure 3 : Balises terrain et embarquée.

c) Contraintes technologiques

Chaque équipe est libre de choisir la technologie qu'elle souhaite employer lors de la coupe, mais certaines contraintes sont précisées :

Lors de la Coupe une forte lumière ambiante est présente (projecteurs), ce qui peut poser des problèmes pour les dispositifs infrarouges, qui risquent de saturer et ne plus fonctionner correctement.

L'organisation de la coupe utilise des dispositifs radio à haute fréquence (fréquence non communiquée, mais nous supposons que les fréquences utilisées sont celles habituellement utilisées pour l'émission radio : fréquences de l'ordre du MHz). Ces fréquences peuvent donc déranger la communication radio robot-balise des robots utilisant cette technologie.

Pour les équipes utilisant la localisation grâce aux lasers, seuls ceux de catégorie 1 et 2 sont admis, à la condition que le rayon du laser de catégorie 2 ne soit pas projeté en dehors de l'aire de jeu.

2) Cahier des charges spécifique au projet ARES

Le cahier des charges que doit satisfaire notre système de géolocalisation est le suivant :

- La rapidité et la précision des mesures. En effet, la position doit être connue suffisamment de fois (5 fois par seconde), et doit être suffisamment précise (de l'ordre du centimètre) pour que la partie asservissement puisse adapter la trajectoire en temps réel.
- Le prix de la technologie : nous ne pouvons nous permettre de commander du matériel trop onéreux, qui certes, apporterait une très bonne précision de mesures, mais qui n'entre pas dans le cadre des budgets qui nous sont alloués par l'ENSEA et ARES.
- Le système de géolocalisation se basera sur un système de balises terrain/embarquée, puisque le robot ARES possède déjà un système de localisation interne en la présence d'un système d'odométrie, système par ailleurs peu fiable car sensible aux perturbations extérieures (secousses, glissement).

Partie V : Etude préliminaire

Afin de choisir la technologie qui sera la nôtre, puis compte tenu des différentes contraintes qui nous sont dévolues et enfin dans le souci de s'intéresser aux différentes technologies qui sont susceptibles d'être présentes à la coupe, nous allons nous intéresser dans un premier temps à l'état de l'art dans le domaine qui nous incombe : la géolocalisation par balises.

Différentes technologies sont utilisées lors de la coupe de robotique; nous ne retiendrons ici que celles qui sont le plus utilisées. Nous nous contenterons de décrire brièvement ces technologies, car de multiples variantes existent.

1) Technologies retenues

a) Ultrasons

Principe : Emission ultrasonique et mesure des distances (robot-balises). Une possibilité est de posséder des balises terrain qui vont être des réflecteurs à ultrasons. La balise embarquée émet donc un signal, qui, lorsqu'il lui sera retourné, permettra de connaître la distance robot/balises et ainsi après triangulation, la position. (figure 4)

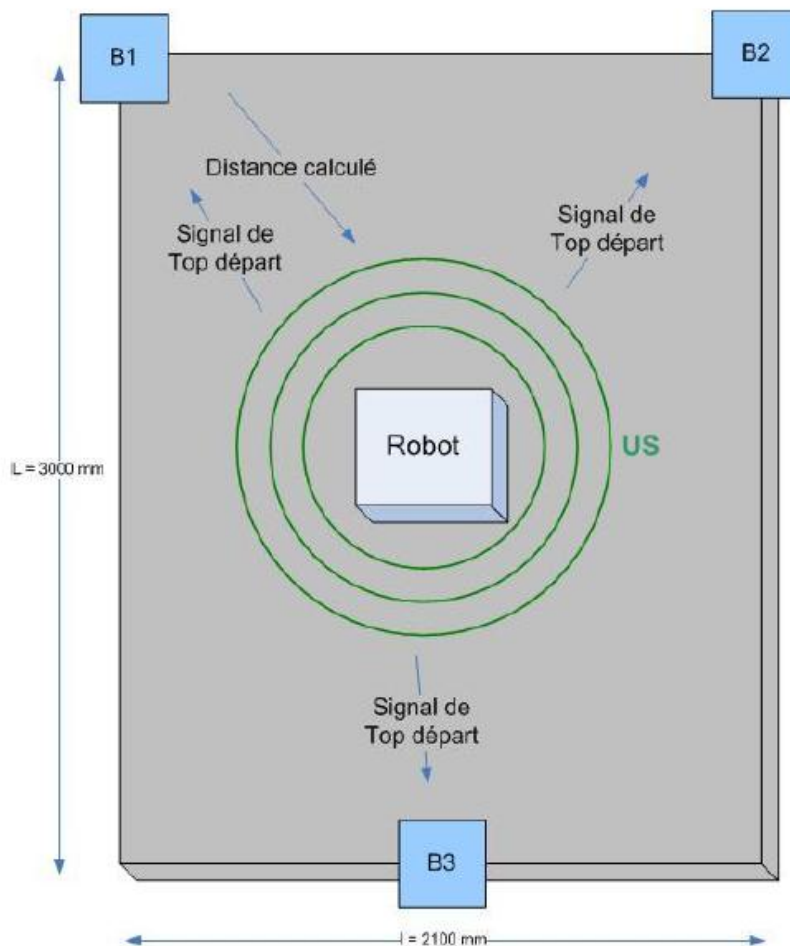


Figure 4 : Schéma de principe d'un système à ultrasons

Avantages/Inconvénients :

- Relativement simple de réalisation.
- Perturbations importantes si présence d'un autre système à ultrasons.
- Très sensible au bruit acoustique et vibrations (même hors de sa plage de fonctionnement).

b) Infrarouges

Principe : Emission Infrarouge et réception sur le robot pour une mesure des angles que fait le robot avec chaque balise. Les balises terrain émettent un signal infrarouge modulé qui après réception par la balise embarquée permet une détermination de la position par triangulation.

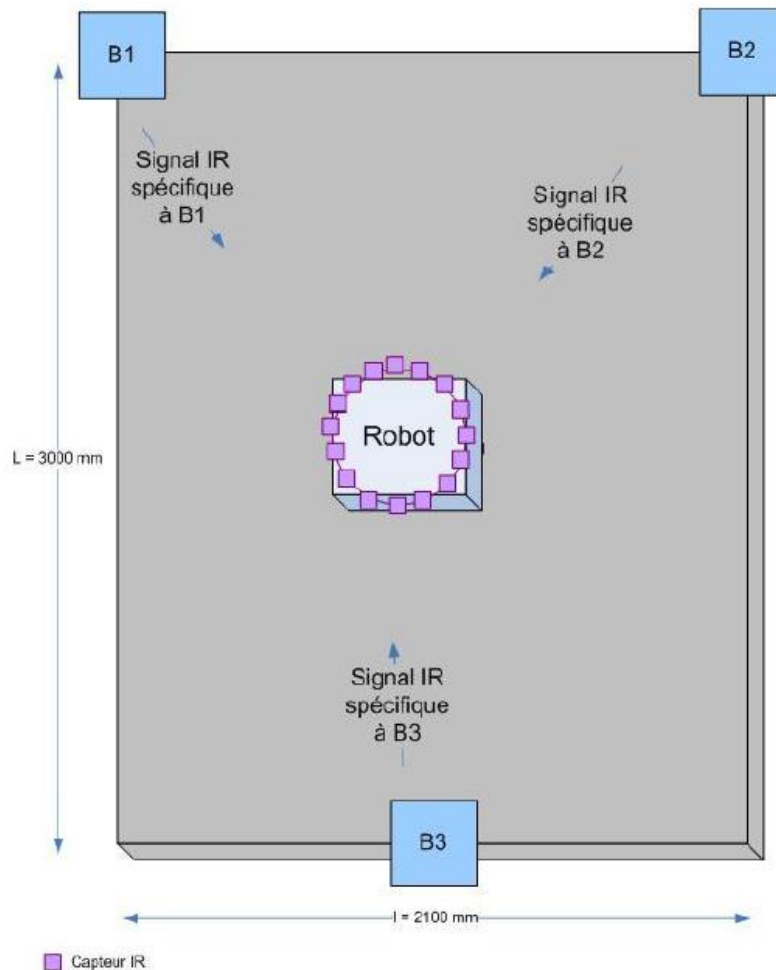


Figure 5 : Schéma de principe d'un système infrarouge.

Avantages/Inconvénients :

- Relativement simple de réalisation.
- Précis (ordre du centimètre)
- Perturbation possible si présence d'un autre système infrarouge (les deux systèmes s'aveuglent mutuellement ou se confondent)

c) Télémètre laser

Principe : Un laser se situe sur le robot et réalise des rotations sur 360° afin de balayer tout le terrain. Lorsqu'il rencontre une balise, il y a acquisition de la distance et possibilité d'acquérir l'angle entre les balises.

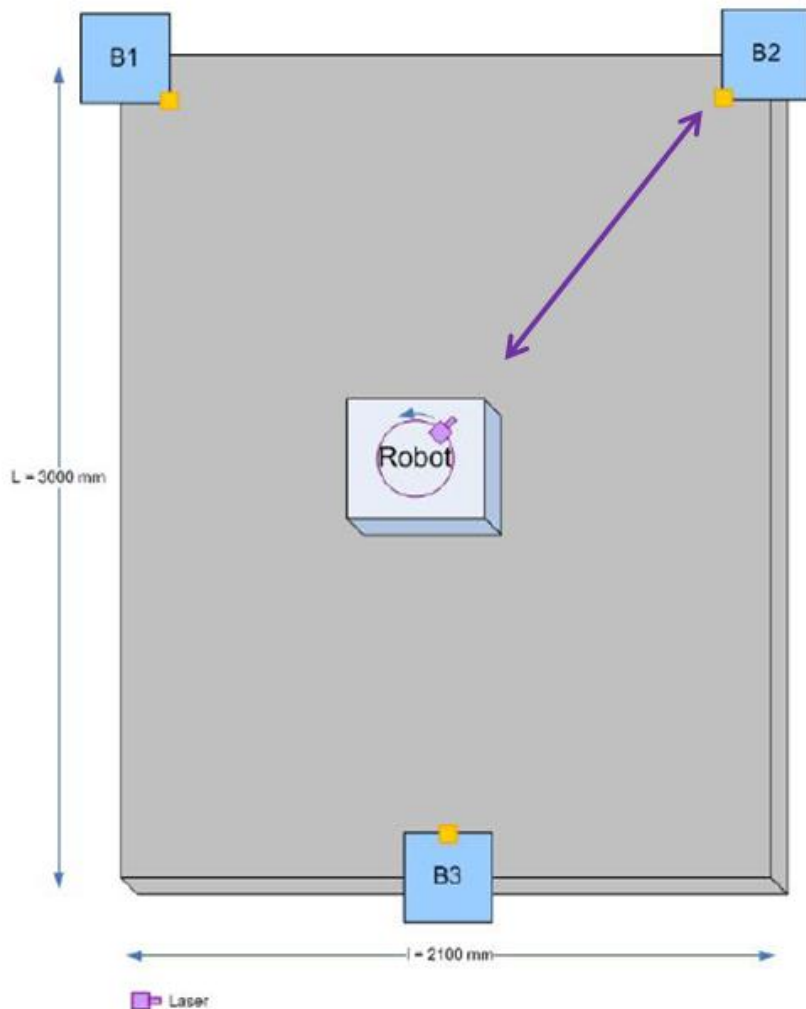


Figure 6 : Schéma de principe d'un système à télémètre

Avantages/Inconvénients :

- Extrêmement précis
- Mesure de l'angle et de la distance
- Relativement complexe (mécanique, électronique)
- Sensible à la stabilité du robot (indépendant de notre contrôle)
- Relativement onéreux
- Peu transposable sur le robot adverse (système volumineux et sensible au mouvement)

2) Choix de la technologie

Suite à de nombreuses discussions et entretiens, nous avons finalement décidé d'utiliser un système se basant sur l'utilisation de l'infrarouge, choix pris pour obtenir un compromis favorable entre précision, coût et capacité de réalisation dans les délais impartis et connaissances requises.

Principe :

Les 3 balises terrain seront des balises émettrices infrarouges (IR), qui enverront à tour de rôle, un signal que nous choisissons non modulé. Le choix du signal non modulé est pris en connaissance de causes : certes nous nous exposons à une saturation (car pas de contrôle automatique de gain possible) et un non-fonctionnement du système qui ne pourrait reconnaître le signal émis, mais un procédé sera mis en place (et décrit ci-après) afin d'essayer de remédier à ce problème.

Les balises embarquées robot (concurrent et ARES) seront, dans un souci de réalisation et de charge de travail, identiques. Le système se veut donc complètement transposable et soumis aux contraintes des 2 types de balises pré-décrits.

Ces balises embarquées recevront le signal des balises émettrices et détermineront les angles entre chaque balise, afin de pouvoir donner, après triangulation, la position exacte d'une part du robot ARES et d'autre part du robot concurrent.

Comme les balises émettent à tour de rôle, une synchronisation entre la balise émettrice et la balise réceptrice est nécessaire. Pour cela, nous utiliserons un module X-Bee (fonctionnant sur le principe du Bluetooth), pour d'une part synchroniser les balises, et pour d'autre part que le robot concurrent nous envoie sa position via sa balise. Ces modules seront donc présents sur chaque balise embarquée et un seul de ces modules sera présent sur les balises terrain, puisque la synchronisation entre balises terrain se fera via un câble.

Les systèmes de maîtrise des différents ordres et organes de calcul seront des microcontrôleurs STM32, choix établi parce que ces derniers nous sont connus (ce qui évite une phase supplémentaire de recherche). De plus ces microprocesseurs ont la capacité suffisante (largement suffisante) pour réaliser ce que nous leur demandons de réaliser.

Le cycle de calcul d'une position du robot est le suivant :

- La balise embarquée réalise un scan du terrain afin d'acquérir le niveau du bruit d'infrarouge ambiant. Valeur qui sera utilisée pour déterminer la qualité d'une mesure. En effet, si la valeur de bruit est trop élevée, il risque d'y avoir saturation des détecteurs et donc erreurs du calcul de la position.
- Ensuite chaque balise terrain émet à tour de rôle un signal, qui est détecté par la balise embarquée.
- La balise embarquée détermine les différents angles entre les balises terrain.
- Avec l'acquisition des angles inter-balises, une triangulation est possible.

- Une vérification de la donnée envoyée aux autres organes du robot est effectuée, par comparaison avec la dernière valeur précédemment calculée.
- Le robot ARES reçoit la position du robot adverse qui aura effectué le même cycle de calcul.

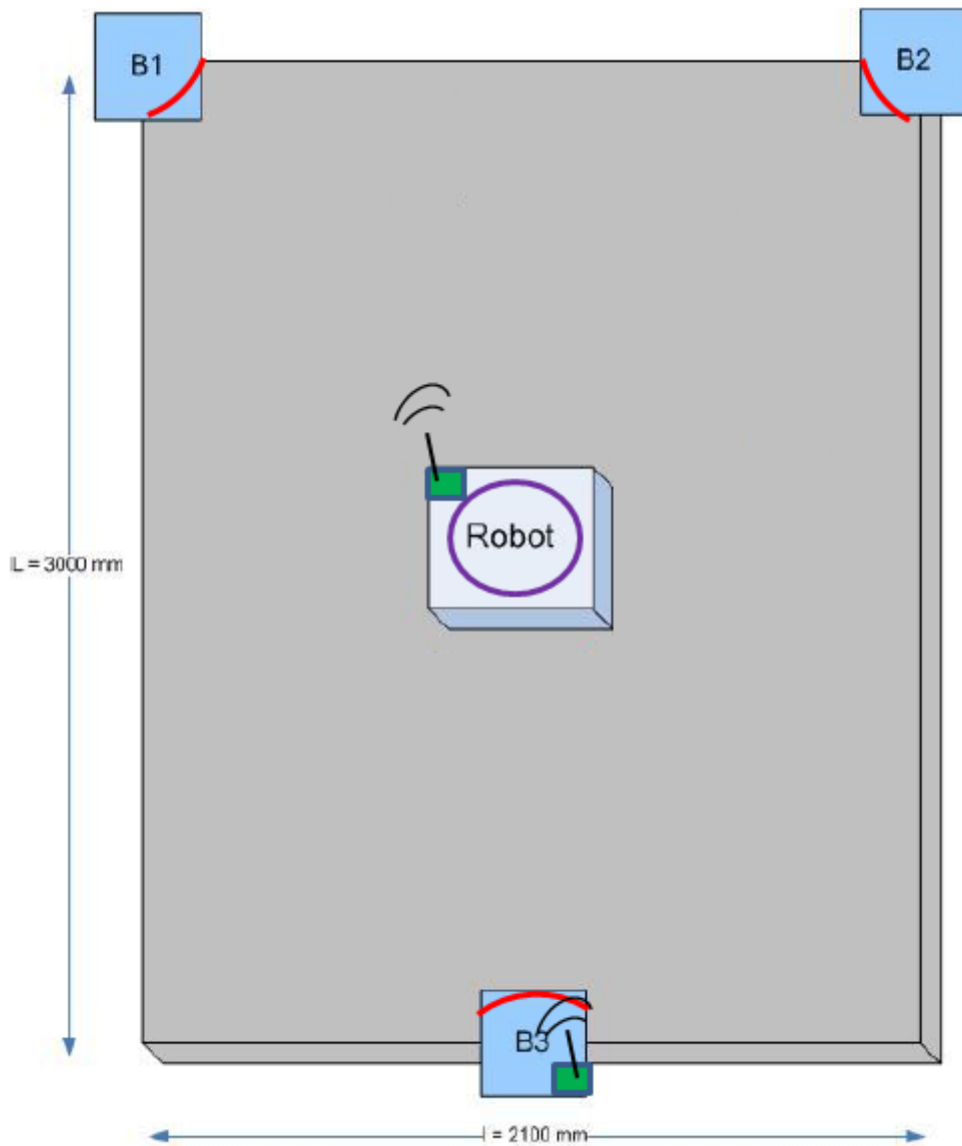


Figure 7 : Schéma de principe de notre système.

Partie VI : Conception hardware des balises

Choix des 2 composants essentiels (diode émettrice IR et photodiode réceptrice IR)

1) Choix des diodes IR

Les balises émettrices se doivent d'être des « phares » infrarouges qui vont émettre suffisamment de puissance pour que le détecteur intégré sur la balise réceptrice puisse détecter le signal émis.

Notre premier souci a donc été de déterminer quelle diode infrarouge nous utiliserions. Après de nombreux tests, nous nous sommes arrêtés sur le choix de la diode infrarouge suivante : une diode infrarouge GaAlAs de référence SFH 484.



Figure 8 : photodiode SFH484

Cette diode présente plusieurs propriétés intéressantes :

- Sa directivité (8° d'angle moitié). En effet, dans un souci de pouvoir envoyer un maximum de puissance et cela de façon directive, il nous faut focaliser l'émission, avec le compromis de multiplier le nombre de diodes d'émission.
- Sa puissance : cette diode a une intensité rayonnée de 80mW/Sr
- Nous pouvons permettre le passage d'une forte intensité dans cette photodiode (200 mA) pendant un très court instant (1 ms) afin d'augmenter l'intensité lumineuse émise.

En effet, dans un souci de forte émission, nous choisissons de pulser l'émission pour émettre au-dessus de la zone de fonctionnement habituel de la photodiode (100 mA).

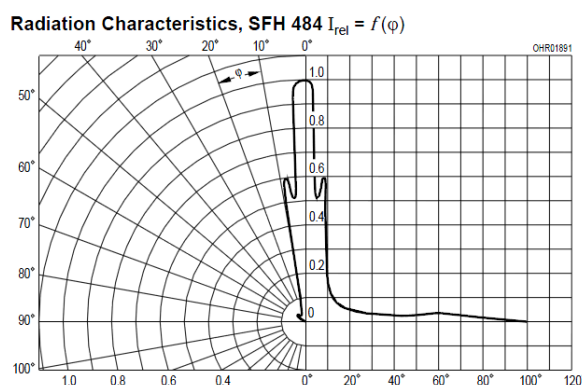


Figure 9 : Caractéristique intensité lumineuse-angle.

Forward Current
 $I_F = f(V_F)$, single pulse, $t_p = 20 \mu s$

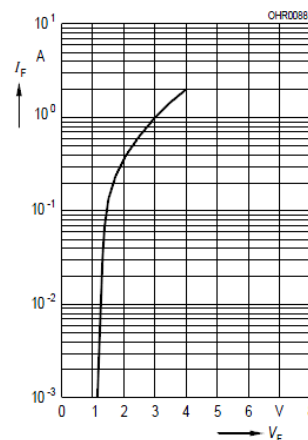


Figure 10 : Caractéristique tension-courant

2) Choix des photodiodes IR

Le choix de la photodiode réceptrice doit être pris en fonction de la diode IR choisie en émission. En effet, la diode IR émet à la longueur d'onde 880nm, il nous faut donc une photodiode fonctionnant à cette longueur d'onde ou très proche.

Ensuite, il faut que notre photorécepteur soit très directif afin que la détermination de l'angle, entre les balises, soit très précise. Ceci implique nécessairement une multiplication non négligeable des photodiodes sur la balise embarquée.

Après de nombreux tests, nous choisissons, la photodiode suivante : SFH213.



Figure 11 : photodiode SFH213.

Cette photodiode présente les caractéristiques requises pour le fonctionnement attendu. En effet, sa longueur d'onde de fonctionnement est de 850nm, ce qui est proche de la longueur d'émission. Les tests valideront que même s'il existe un écart entre la longueur d'onde d'émission et la longueur d'onde de fonctionnement du photorécepteur, la mesure des données est suffisamment correcte.

Cette photodiode présente aussi une bonne sensibilité : 0.62A/W.

Et ce récepteur présente un angle moitié de 10°, ce qui est suffisamment précis pour nos mesures.

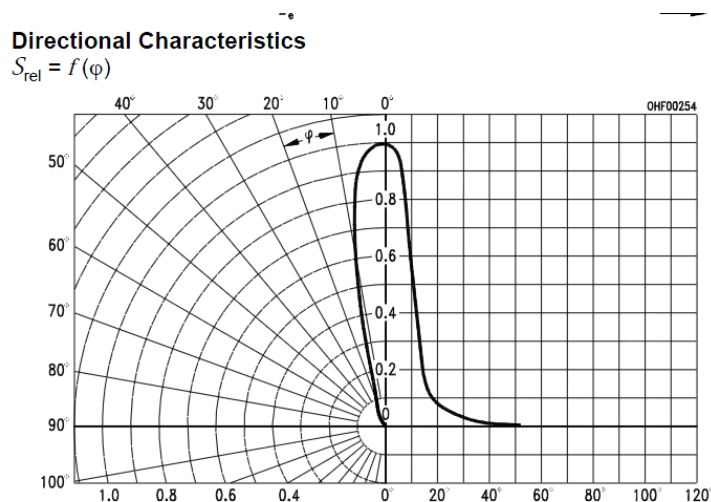


Figure 12 : caractéristique de directivité.

3) Balise émettrice

a) *Première approche*

La première idée développée, a été de concevoir, grâce à une série de transistors, l'alimentation des diodes infrarouges. En effet, nous utilisons des transistors NPN en commutation, transistors commandés par une tension au niveau de leur base et délivrant ainsi le courant nécessaire. Le signal de base serait donc le signal de commande provenant du STM32 placé sur une des balises terrain. Nous obtenons donc le schéma de principe suivant :

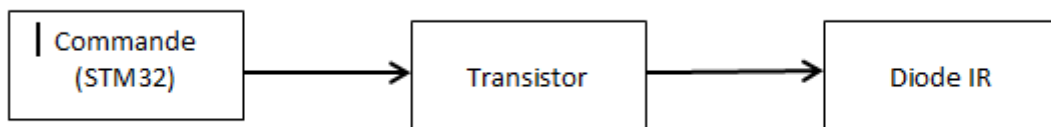
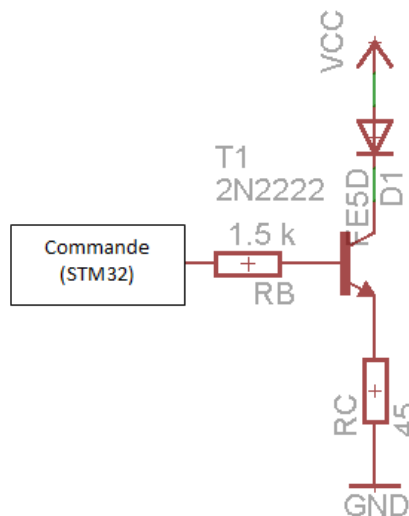


Figure 13 : Schéma de principe de l'émission.

Nous avons dans un premier temps utilisé des transistors NPN 2N2222, utilisés en commutation comme le montre le schéma suivant :



Les valeurs présentes sur le schéma précédent sont celles nécessaires à l'établissement dans la diode Ir d'un courant de 200 mA pour une tension d'alimentation de 15V et d'un gain du transistor β de 300.

Il est à noter que la résistance de collecteur est de 45 Ω , ce qui nécessite une résistance capable de résister à une puissance de 1.8W, ce qui est important ; on prendra donc garde à ne pas utiliser des résistances quart de watt, ou alors à créer un réseau de résistance en parallèle équivalent à une unique résistance de 45 Ω .

Notre première approche était donc celle-ci, mais pour avoir une bonne répartition de la puissance sur l'ensemble du terrain, il nous fallait, pour 2 des balises (balises quart de terrain), 7 diodes Ir et pour la balise demi-terrain, 14 diodes IR, soit un écart entre chaque diode de 12.8° , pour satisfaire aux caractéristiques de la diode.

Plusieurs problèmes se sont alors posés :

- Le nombre de photodiodes implique un nombre équivalent de transistors, donc un besoin en surface non négligeable et supérieur à la surface accordée par le cahier des charges (80x80 mm).
- L'utilisation d'un transistor Darlington suffirait à pallier à ce dernier problème, mais l'efficacité de ce type de dispositif est insuffisante pour le travail demandé (50% au maximum).

b) Seconde approche

Suite à des discussions avec un professeur d'analogique, nous avons découvert l'existence d'un composant réalisant ce que nous souhaitions et avec une efficacité de 97%. Ce composant est un driver de led (LDU7860S41000).



Figure 14 : Driver de Leds.

Il est capable de délivrer un courant de sortie allant jusqu'à 1 A, avec une tension d'alimentation variant de 7 à 60 V et une tension de commande TTL.

Ce composant fonctionne par le même principe que ce que nous voulions réaliser, c'est-à-dire qu'une tension de commande impose le caractère passant ou bloqué du courant et ainsi l'illumination.

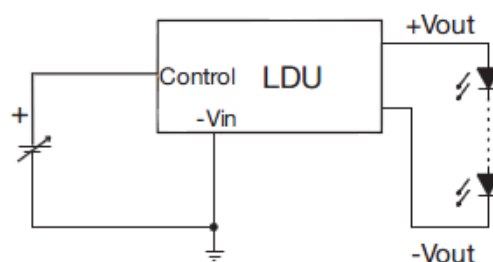


Figure 15 : Schéma de principe du montage émetteur.

c) Mise en œuvre

Une fois les différents composants choisis, il ne nous reste plus qu'à les assembler, comme le présentent les schémas des figures 16 et 17.

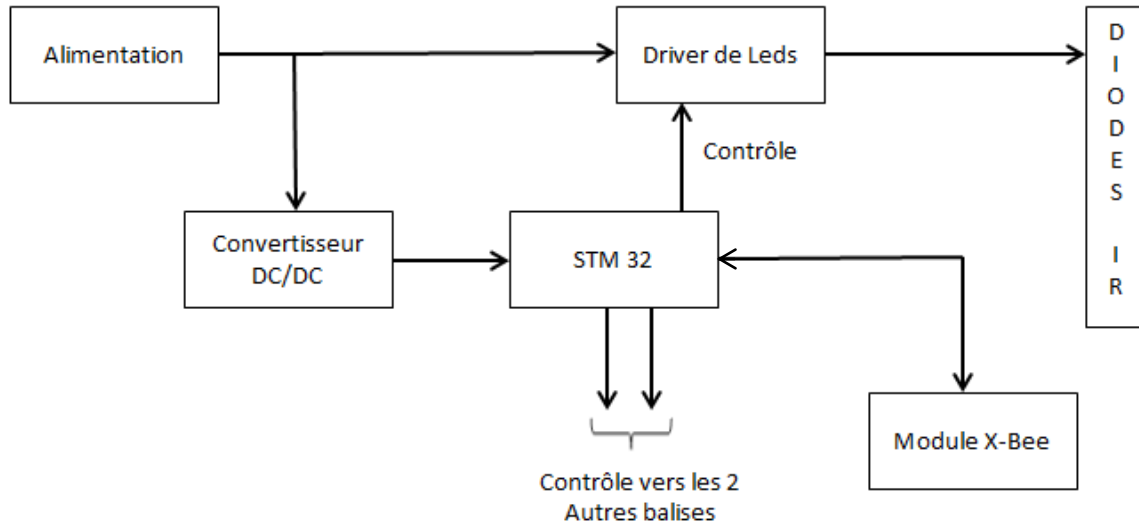


Figure 16 : Schéma de principe de la balise principale terrain.

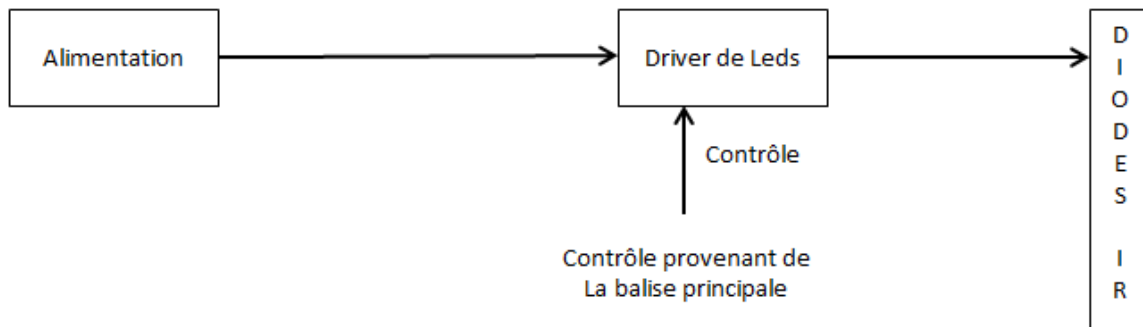


Figure 17 : Schéma de principe des balises secondaires terrain.

Nous pouvons voir que la balise principale comporte les différents éléments cités précédemment. En effet, l'alimentation va être connectée au driver de leds qui pour une valeur de 12V va permettre une sortie en courant de 200 mA, cette même alimentation va être convertie en une alimentation 3.3V, afin d'alimenter le STM32. Ce dernier va ordonner au driver de leds de commander l'alimentation en courant des leds et aussi envoyer au module X-Bee l'information de la balise en train d'émettre, module X-Bee qui enverra l'information à la balise réceptrice sur le robot ARES, et à celle sur le robot adverse.

Les balises secondaires ont le même rôle que la principale, à ceci près que le STM et le module X-Bee ne sont pas présents, car ces 2 balises secondaires sont reliées à l'aide d'un fil à la balise principale.

Vous pourrez trouver en annexe la schématique complète des balises terrain.

4) Balise réceptrice

a) *Première approche*

Les balises embarquées sont les balises réceptrices. Elles doivent donc, pour réaliser une bonne mesure de la position, connaître de manière précise les angles que font le robot avec les différentes balises, afin de réaliser la triangulation (calculs donnés en annexe).

Dans un souci de précision, nous décidons donc de réaliser une balise réceptrice munie de 32 photodiodes, soit un écart de 11.25° . Ce choix de 32 photodiodes est essentiellement dû au fait que le STM32 présente 16 entrées analogiques et 32 en est un multiple, mais nous reviendrons ultérieurement sur ce choix. De plus, cet écart est suffisant pour une bonne précision.

Le principe de fonctionnement des balises embarquées est le même pour les 2 balises, à savoir, la balise robot ARES et la balise robot concurrent. Ce choix a été fait pour faciliter la conception et le respect des délais de livraison.

L'anneau de photodiode reçoit donc le signal émis par l'une des balises. Suite à l'analyse de la réception grâce au STM32, on peut déterminer quelle photodiode est dans l'axe de la balise émettrice. La balise émettrice ayant aussi reçu l'information sur la balise émettrice (quelle balise émet).

Après que les 3 balises terrain aient émis leur signal, on peut déterminer par cette méthode, les 3 angles.

Le schéma de principe en première approche de la balise réceptrice (balise robot) est le suivant :

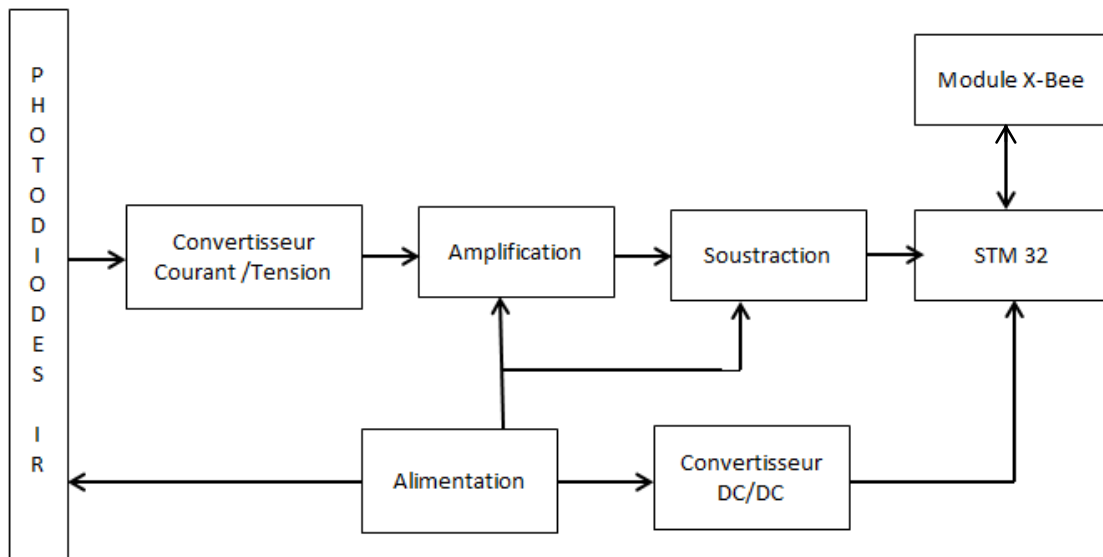


Figure 18 : Principe de fonctionnement du récepteur, première version.

Dans cette première version, les photodiodes sont polarisées en inverse, grâce à l'alimentation 12V des batteries du robot, polarisation qui permet une meilleure conduction du courant.

Lorsque les photodiodes reçoivent un signal, elles deviennent conductrices. Ce courant passe par un étage de conversion courant/tension. Cet étage est réalisé à l'aide d'un amplificateur opérationnel en montage trans-impédance (figure 19). L'étage d'amplification schématisé figure 18 est compris dans l'étage conversion. En effet, il suffit de choisir une valeur de résistance suffisante pour obtenir le gain désiré, car cette valeur correspond au gain de l'AO. Nous avons alors converti, puis amplifié le signal entrant. Une amplification est nécessaire en raison de l'ordre de grandeur du courant généré par la photodiode : une dizaine de nA. C'est pourquoi, pour notre fonctionnement, nous choisirons une valeur de résistance de 220k Ω , valeur qui nous permet de détecter encore une amplitude lorsque le robot est au barycentre des 3 balises, c'est-à-dire à environ 1.5m des balises. A cette distance on obtient une tension en sortie d'amplification d'environ 100mV, valeur suffisante pour le STM32.

Cette valeur de résistance permet aussi de ne pas saturer l'étage d'amplification lorsque l'on se trouve au plus près des balises, on obtient ainsi une amplitude de 7V, ce qui est en dessous de la saturation de l'amplificateur.

L'étage de soustraction est réalisé à l'aide d'un montage à amplificateurs opérationnels (figure 20). Cet étage était nécessaire car le STM32 ne possède que 16 entrées permettant la conversion analogique/numérique. C'est donc en majeure partie cette contrainte qui nous a décidés sur le choix du nombre de photodiodes.

La soustraction est effectuée entre 2 photodiodes opposées (2 photodiodes opposées ne pouvant être éclairées par la même source), ainsi le signal résultant donne un signal fonction de l'intensité lumineuse reçue moins la composante de bruit ambiant.

Nous traitons donc le signal par le STM32 et en déterminons l'angle et la position.

Ce système nécessite de nombreux AO, nous choisirons donc lors de la réalisation des composants CMS.

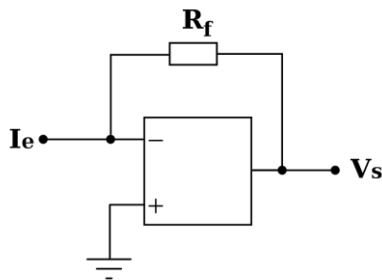


Figure 19 : Montage trans-impédance.

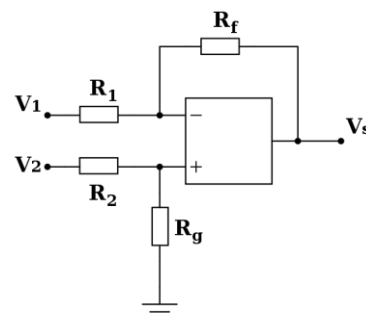


Figure 20 : montage soustracteur (dans notre cas $R_1 = R_2 = R_f = R_g$).

Malheureusement nous avons rencontré un problème avec ce principe de fonctionnement : nous nous sommes aperçus que le modèle de STM que nous possédions n'accepte pas d'entrée analogique négative, ce que nous donnait la sortie du soustracteur pour la moitié des photodiodes.

Nous avons donc modifié le fonctionnement.

b) Seconde approche

Face au problème rencontré, nous avons décidé d'utiliser les entrées analogiques du STM32 sans avoir recours à la soustraction. Il nous a donc fallu utiliser 2 STM 32 au lieu d'un seul. Le nouveau schéma de principe est le suivant :

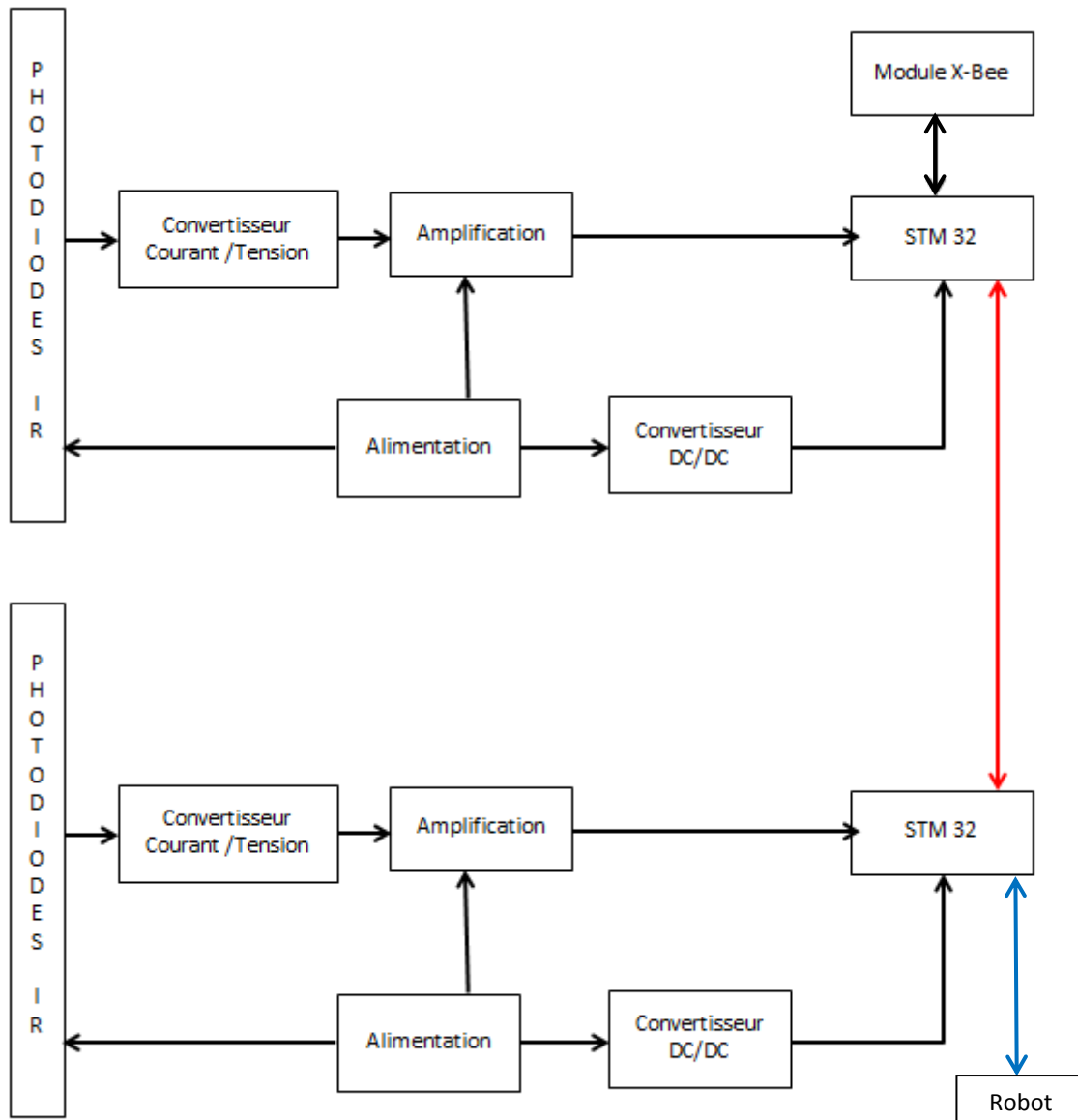


Figure 21 : Schéma de principe de la balise robot en l'absence de soustracteur.

La soustraction enlevée, 2 STM s'avéraient nécessaires, donc avec les contraintes de taille imposées, nous étions obligés de passer d'une carte à deux.

Le principe de fonctionnement est sensiblement le même que dans la première approche, à ceci près que la présence de 2 STM impose une synchronisation entre les 2 composants et l'utilisation d'un protocole de communication interne.

Le module X-Bee reçoit dans les 2 approches les informations provenant du X-Bee de la balise principale terrain, lui donnant d'une part l'identité de la balise émettrice et d'autre part la position du robot adverse, ceci pour le robot ARES. Le module X-Bee pour le robot concurrent reçoit l'information de la balise terrain et envoie sa position au robot ARES.

Les schématiques complètes des différentes cartes sont données en annexe.

Partie V : Programmation

L'ensemble des balises comprend en tout 7 cartes électroniques, et 5 STM32.

Les programmes des STM situés sur les balises de notre robot et sur les balises du robot adverse sont assez similaires, ainsi une globalité sera détaillée, et les différences entre chacun seront ensuite expliquées.

Quant au programme du STM sur la balise 180°, il est beaucoup plus simple et se comprendra au fur et à mesure.

1) Approche concrète

De manière très concrète, l'agencement de l'ensemble doit s'effectuer de la façon suivante : Les 4 STM des balises robot effectuent les 16 acquisitions correspondant aux photodiodes qui leur sont connectées.

Ensuite, la première balise terrain (appelée tower1 dans le programme) s'allume, et les 4 STM effectuent à nouveau 16 acquisitions.

De même pour les tower2 et tower3.

Notons que chacune des acquisitions est stockée dans un tableau différent, ainsi à la fin du cycle précédent, nous obtenons 4 tableaux de 16 acquisitions pour chaque STM.

A la fin de ce cycle, les données contenues dans chaque STM sont mises en forme, centralisées, et traitées pour obtenir les angles nécessaires à la triangulation.

Il est évident que les différents STM doivent être synchronisés régulièrement pour que chacun sache à tout moment quelle est la balise émettrice qui est éclairée. Pour cela des points de synchronisation sont placés à des endroits clefs dans le programme, et ceux-ci doivent être minutieusement pensés et organisés sous peine d'avoir un programme qui donne l'impression de fonctionner correctement alors qu'en réalité il n'en est rien.

2) Approche avancée

Pour aller plus loin dans le fonctionnement du programme et de la synchronisation entre les 5 STM, reprenons plus synthétiquement un cycle de calcul de position avec un schéma, et précisons la tâche de chaque STM, à quel moment, et sous quelle base de synchronisation.

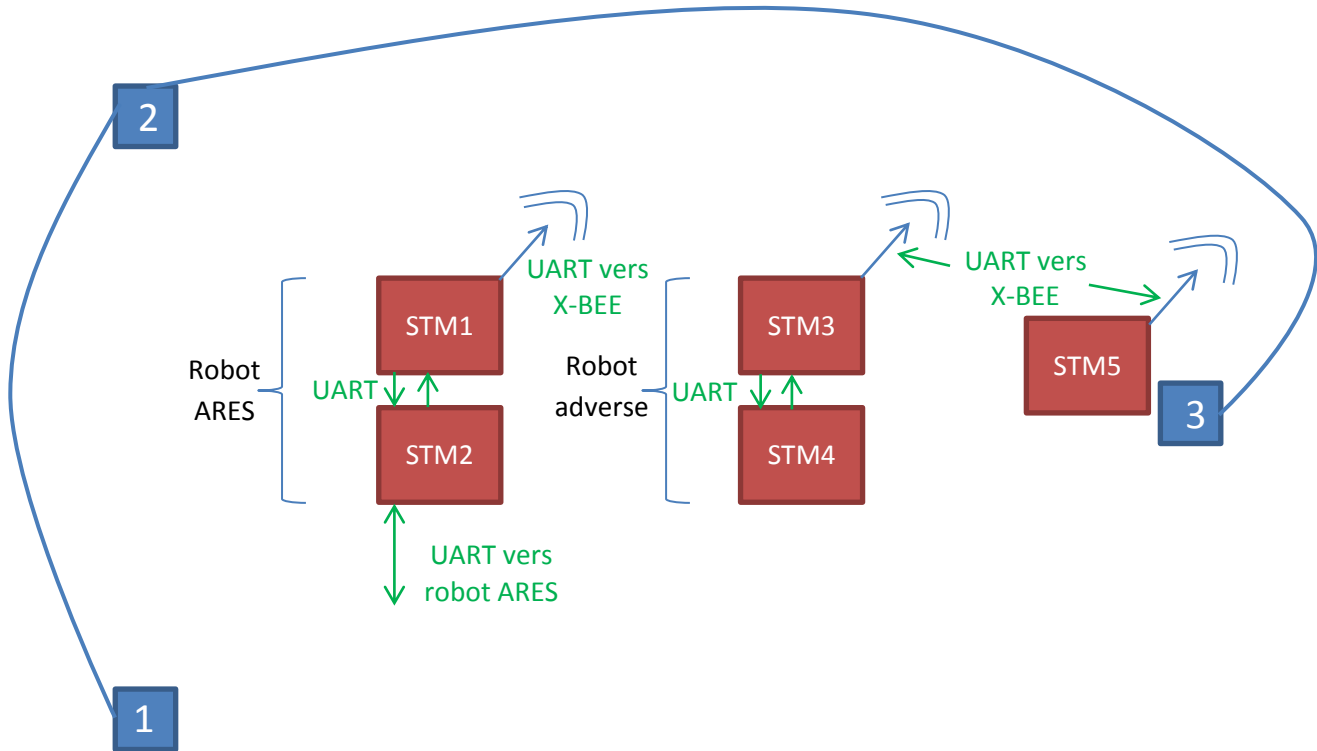


Figure 22 : Schéma des liaisons physiques entre les systèmes.

Notons que les X-Bee servent en réalité de fils virtuels entre les STM.

En effet, par défaut les X-Bee permettent de remplacer une liaison UART filaire par une liaison UART sans fil. Nous détaillerons en annexe les configurations à adopter pour que cela fonctionne à la plus grande vitesse possible.

Notons également qu'il aurait été possible de remplacer le STM5 (celui sur la balise 180° par le X-Bee seul, mais que pour cela il aurait fallu utiliser l'ADC du X-Bee, ainsi qu'une PWM en sortie (sans entrer dans le détail) mais qu'avec cette méthode il n'y aurait pas eu de retour d'information, et il n'y aurait aucune certitude que l'ordre d'allumage de telle ou telle balise ait bien été pris en compte.

Enfin dans le détail du programme nous parlerons de « bits de synchronisation » entre les STM. Nous avons en effet rajouté deux fils entre les STM 1 et 2 puis deux autres entre les STM 3 et 4 pour faciliter l'attente entre les STM.

Revenons au détail du programme :

Initialisation.

Les STM initialisent les différents ports, les différentes connexions UART, et surtout leur ADC.

Le STM sur la balise 180° éteint toutes les balises émettrices.

A la fin de cette initialisation, l'ADC est lancé, la première acquisition du bruit ambiant commence (pendant ce temps, les STM ne sont pas utilisés, seul les ADC acquièrent les données).

Interruption 0 de l'ADC.

Chaque STM doit faire les 16 acquisitions du bruit, puis attendre les autres. En effet nous devons attendre que chacun d'eux ait terminé d'acquérir les données pour allumer la première balise.

Ainsi, lorsque le STM1 a terminé ses acquisitions il le fait savoir au STM2 via le bit de synchronisation prévu à cet effet (en fait l'un des deux fils va du STM1 vers le STM2 et l'autre dans le sens inverse ; pour qu'un STM informe à un autre qu'il est arrivé au point de synchronisation, il inverse la valeur du bit c'est-à-dire que s'il était à « 1 » il le passe à « 0 » et vice versa).

Lorsque le STM4 a terminé ses acquisitions il le fait savoir au STM3, qui lui, lorsqu'il aura cette information et qu'il aura lui-même terminé, transmettra au STM1 (via l'UART connecté au module X-Bee) que la balise située sur le robot adverse a fini les acquisitions.

Enfin, le STM1, lorsqu'il a reçu l'intégralité de ces informations, et qu'il a lui-même terminé ses acquisitions, envoie l'ordre au STM5 (via l'UART connecté au X-Bee) d'allumer la balise numéro 1.

Le STM5 allume donc la balise 1 et dit aux STM1 et 3 via le X-Bee que la balise est allumée.

Les STM1 et STM3 vont à leur tour informer respectivement les STM2 et STM4 que la balise est allumée, et qu'ils peuvent lancer la nouvelle acquisition (par « informer », il faut comprendre que le bit de synchronisation changera d'état et que le programme suivra son cours normal).

A ce moment-là les STM 1, 2, 3 et 4 changent le tableau destiné à recevoir les acquisitions de leur ADC pour ne pas écraser les précédents résultats. Ceci est très important, car les données ne sont traitées qu'une fois les trois angles mesurés, et non au fur et à mesure.

Enfin, l'ADC est lancé, les acquisitions de tous les ADC commencent.

Interruption 1 et 2 de l'ADC.

Le principe est le même que l'interruption 0, excepté que l'on change de balise émettrice.

Interruption 3 de l'ADC.

Le STM1 attend que les autres STM aient fini leurs acquisitions de la même manière que les étapes précédentes, mais une fois les acquisitions terminées les choses se compliquent :

Les STM 1,2,3 et 4 soustraient l'acquisition 0 à chacune des acquisitions 1 à 3, cela signifie concrètement que l'on soustrait le bruit ambiant aux mesures, on ne garde donc dans les acquisitions que ce que l'on a envoyé via les balises émettrices.

Ensuite, chacun des STM détermine quelles sont les deux photodiodes qui ont reçu le plus de lumière pour chacune des acquisitions avec les balises allumées.

Ces deux valeurs, les STM 2 et 4 vont les envoyer via les UART respectivement aux STM 1 et 3. Attention, pour cette action, il faut penser au fait que l'ADC numérise les entrées analogiques sur 12 bits, et que les transmissions UART se font sur 8 bits, donc il faudra découper chacune de ces données en 2*6 bits ou 8 + 4 bits par exemple, et les reconstituer en réception. En plus de ces résultats, les STM 2 et 4 transmettent aux STM 1 et 3 quelles sont les photodiodes qui ont reçu ces valeurs.

Une fois que les STM 1 et 3 ont reçu toutes ces données ils disposent donc des 4 photodiodes qui ont reçu le plus de lumière et ce pour chacune des différentes acquisitions. Si tout s'est bien passé, ces 4 photodiodes devraient être côte à côte.

Enfin, les photodiodes étant disposées tous les $11,5^\circ$, on considère que la première photodiode correspond à 0° (en réalité 360° , nous verrons pourquoi), la seconde à $11,5^\circ$ etc ... Ainsi, pour les 4 photodiodes correspondant à l'acquisition balise 1 allumée, on détermine un angle en effectuant le barycentre de ces 4 photodiodes et pondéré par les valeurs des acquisitions correspondantes.

On fait de même pour les acquisitions effectuées balises 2 et 3 allumées, et on obtient donc 3 angles. La différence entre ces angles, modulo 2π , nous donne les angles entre les balises vues du robot.

Avec ces résultats et les calculs fournis en annexe, nous sommes capables de donner la position du robot.

Les STM 3 et 4 en ont fait de même, et déterminent la position du robot adverse. Cette dernière est envoyée au STM1, qui va à son tour envoyer ces positions au STM2 qui les enverra aux autres organes du robot via l'UART prévu à cet effet.

Puis l'algorithme recommence.

Cette partie était un peu lourde à expliquer, mais si une personne souhaite reprendre le projet, le code sera très facile à comprendre avec ces explications.

On peut remarquer que finalement le code du STM 5 se contente de recevoir les ordres d'allumer les balises et de répondre « j'ai allumé les balises ». On peut tout à fait imaginer utiliser un composant comme un PIC pour effectuer cette mission. Si nous avons fait ce choix, c'était pour économiser le temps passé à apprendre à se servir correctement des PIC, mais aussi pour n'avoir que des STM dans le système et ainsi avoir moins de pièces de rechange à transporter en cas de casse (ce qui est évidemment arrivé).

Partie VI : Quelles sont les raisons qui ont fait que le projet n'ait pas abouti ? Qu'aurions-nous du faire autrement ?

La principale raison qui a fait que le projet n'ait pas pu être mené jusqu'au bout est le manque de temps, et cela provient de plusieurs sources.

La première grosse perte de temps que nous avons subie est que nous avons cherché à tout prix à fabriquer nous-même la conversion tension courant pour l'alimentation des diodes IR. Dans la théorie il suffit de mettre un transistor et deux résistances ; en réalité selon le seuil de la diode, le courant souhaité dans la diode, et le courant réel qui y passera, il y a un large gap.

S'il fallait refaire le projet, nous commanderions directement le driver à LED, ce qui serait beaucoup plus rapide.

Le choix des composants nous a pris un temps considérable ... Il a fallu créer un système d'émission pour chaque diode IR et un système de réception pour chaque photodiode, et ce afin de tester chaque combinaison de la dizaine de diodes/photodiodes que nous avons testée.

Ce travail étant fait, il serait bon, pour les éventuels successeurs, de ne pas recommencer les tests et prendre les diodes/photodiodes indiquées.

Nous n'avons pas détaillé les valeurs de ces tests (courants mesurés en sortie de photodiodes) pour ne pas prendre encore plus de temps.

Suite à ces pertes de temps, nous avons tardé à réaliser les tipons des cartes électroniques, et avons dû nous résigner à les faire à l'école sous peine de ne pas les avoir suffisamment tôt pour la coupe. Mais ce fut une erreur, car les cartes faites à l'école ne fonctionneront que sous peine d'avoir des pistes assez larges et assez espacées les unes des autres, ce qui n'était pas du tout notre cas en raison de la petite taille des cartes et de la quantité de composants à y mettre.

Nous avons donc fini les cartes le jour même de la coupe, et nous avons passé toute une journée à enlever les courts-circuits et réparer les coupe-circuits.

De plus, ces cartes comprennent beaucoup de via, ce qui en rend la réalisation encore plus hasardeuse.

Une grave erreur de notre part fut de croire que le STM pouvait numériser des signaux analogiques négatifs.

En fait, certaines versions du STM le peuvent (celles qui ont un plus grand nombre de pattes que le nôtre) et la documentation est commune aux différentes versions. Il aurait fallu, dès que nous avons eu l'idée, vérifier sur un schéma technique quelles étaient les pattes à câbler pour configurer correctement l'ADC ; à ce moment-là nous nous serions aperçu qu'il y avait cette différence entre les STM et nous serions partis sur autre chose.

Au final, cette erreur a été décelée 2 semaines avant la coupe. Il a donc fallu, dans ce temps, imaginer une façon de recevoir les 32 photodiodes avec 2 STM, refaire les tipons, et refaire une partie du programme.

Nous avons de ce fait passé les deux semaines très intenses à refaire ces parties.

Pour finir, c'étaient nos premières cartes électroniques, et nous n'avions jamais utilisé Eagle avant ... Nous avons alors malencontreusement inversé le sens de certains composants, ce qu'il a fallu régler avec des composants « volants » au-dessus des cartes. Même si tout le reste avait fonctionné, ce genre de câblage est source de bruit et n'aurait au final sûrement pas fonctionné.

Partie VII : Les points essentiels pour faire ce genre de projet

Ici nous allons lister les différents points clefs de tout le projet.

Si quelqu'un souhaite se lancer dans un projet du même genre et qu'il ne souhaite pas lire tout le compte rendu, alors la seule partie à lire serait celle-ci.

Si le budget vous le permet et si vous souhaitez asservir un moteur, utilisez un laser avec un miroir qui tourne, c'est la solution la plus efficace (si on exclut la vision par caméra). Sinon :

Utilisez le couple diodes/photodiodes que nous avons sélectionné.

Utilisez à tout prix un driver de LED.

Choisissez un microcontrôleur avec beaucoup de périphériques et surtout un ADC avec beaucoup d'entrées.

Faites faire des premières cartes vers janvier-février, cela vous apprendra suffisamment tôt à utiliser Eagle, et vous permettra de pouvoir prendre en compte les contraintes que vous devrez respecter sur les cartes électroniques.

Faites les parties drivers du programme au fur et à mesure : si vous savez que vous aurez besoin d'un ADC, alors codez-le et testez-le au plus vite, ça ne sera plus à faire et lorsqu'une partie de ce genre fonctionne, c'est à vie !

Enfin, dormez avec un papier et un crayon à côté ... ça sert de plus en plus lorsque l'échéance arrive...

Conclusion

Un projet inabouti ? Non !

Le travail que nous avons fait n'a jamais été fait auparavant au sein d'ARES. Si nous avons très détaillé ce compte rendu, et mis en annexe les programmes, cartes, et configurations que nous avons fait/utilisé, c'est parce qu'un système de géolocalisation est essentiel sur un robot performant. Il ne faut pas grand-chose pour faire aboutir notre système, et on espère que dès l'année prochaine un groupe de personnes utilisera notre travail pour réaliser un système fonctionnel.

Nous n'avons pas la prétention que le système soit parfait, ni qu'il ait le meilleur rapport qualité/prix, néanmoins par rapport à ce que l'on peut trouver sur internet, nous pouvons certifier qu'il n'y a pas de meilleure source que ce rapport pour avoir une bonne idée de ce que représente le projet en terme de temps et d'investissement.

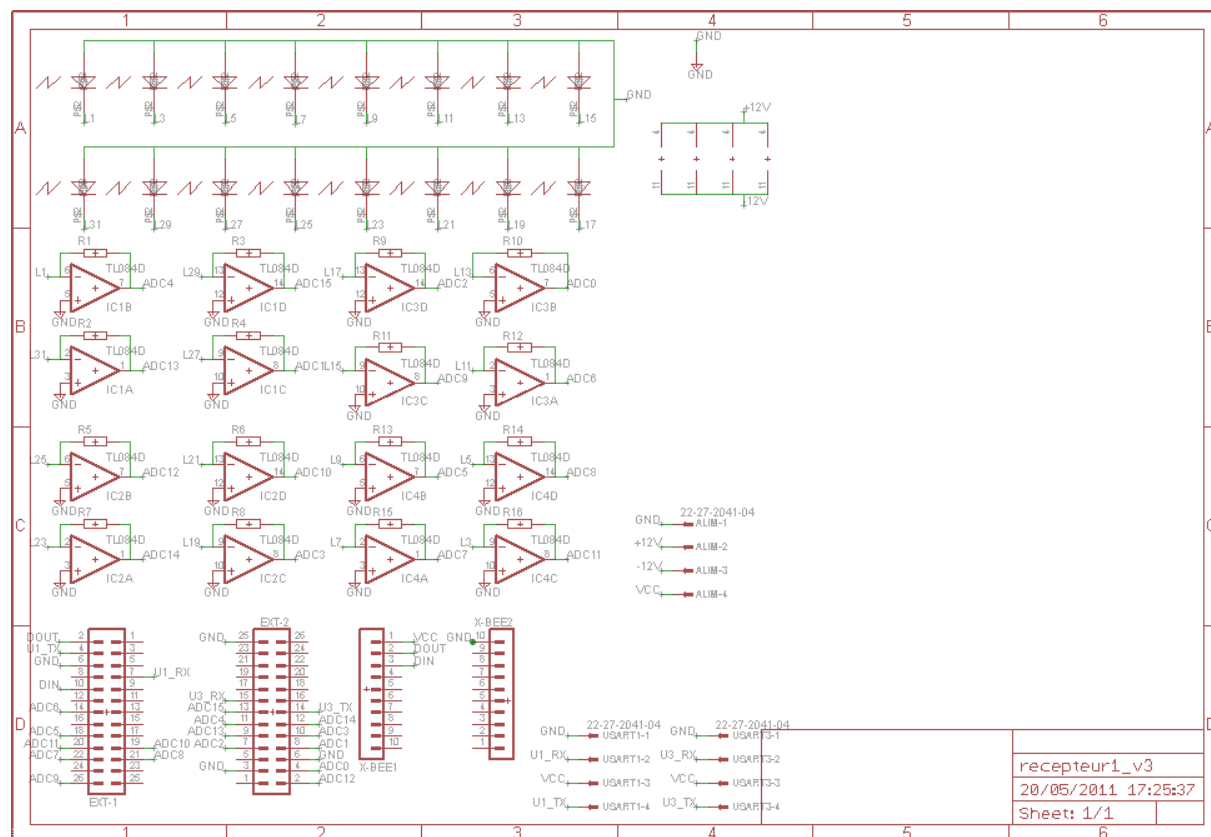
Si malgré cela ce rapport tombe dans l'oubli, il n'est pas exclu qu'un jour nous refassions les cartes et réessayons de les faire fonctionner.

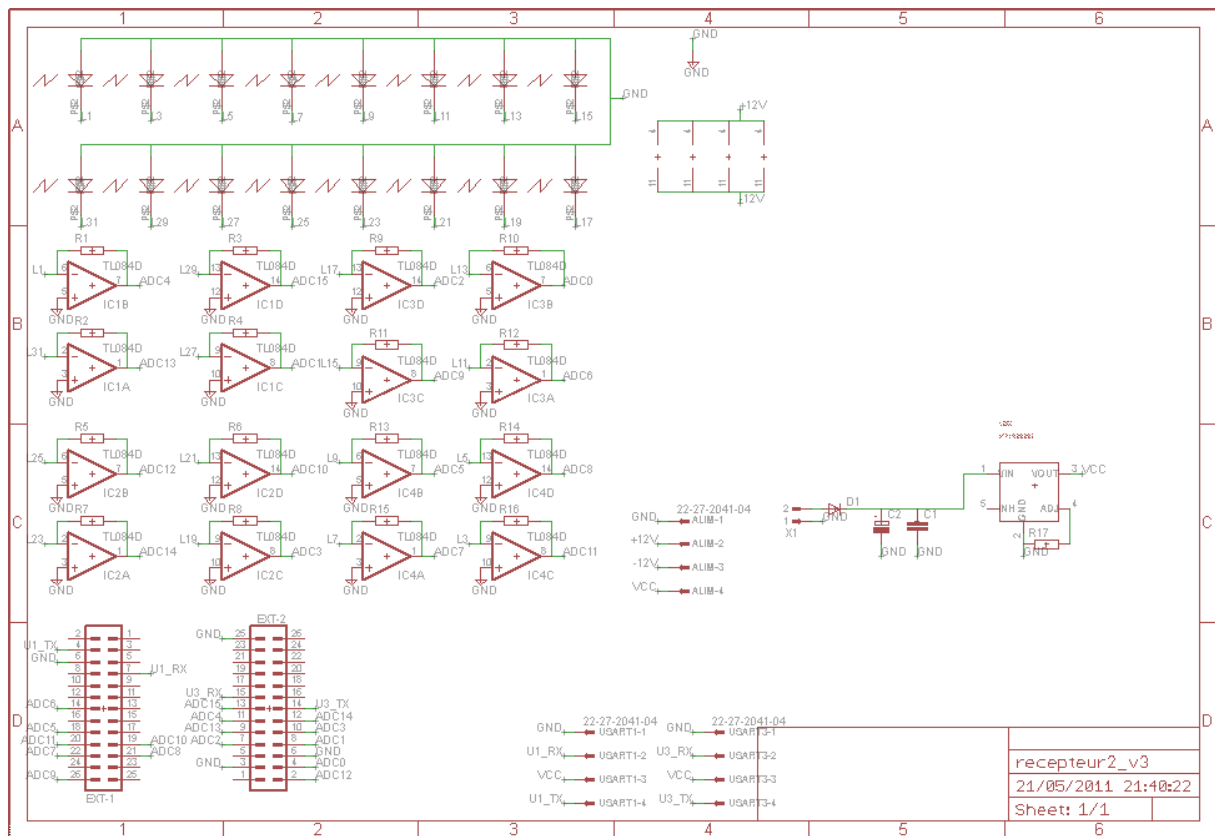
Pour rester plus classiques, nous concluons que ce projet nous a permis de visiter un large domaine de l'électronique, et de rendre visite à beaucoup de spécialistes à l'ENSEA. Nous avons désormais une idée concrète de ce que représente l'élaboration de A à Z d'un projet tel que nous pourrions faire en entreprise.

Enfin, même si maintenant nous sommes sûrs et certains de ne plus jamais vouloir faire de tipons et ne plus jamais manipuler de transistors bruts de notre vie, nous avons passé de très bons moments à réaliser ce projet, et la diversité du travail à accomplir ne nous a jamais fait ressentir de sentiment d'ennui.

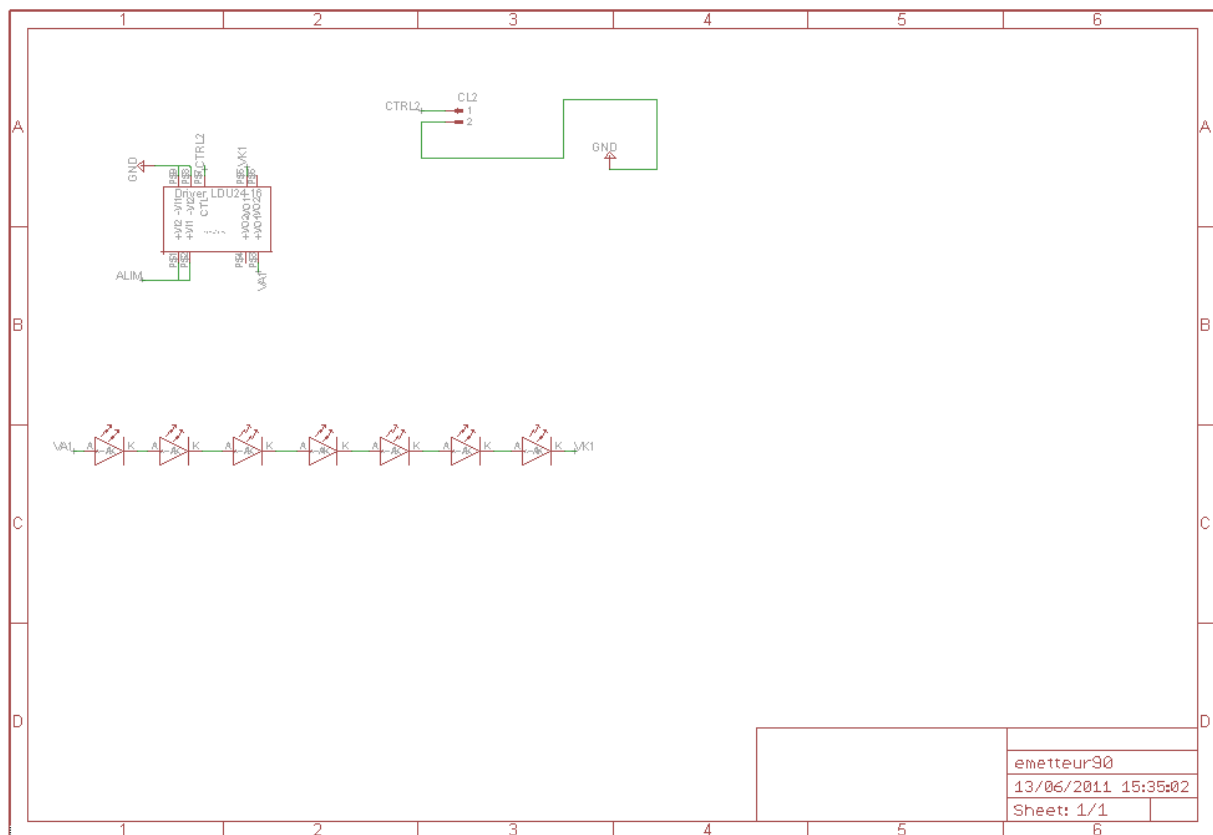
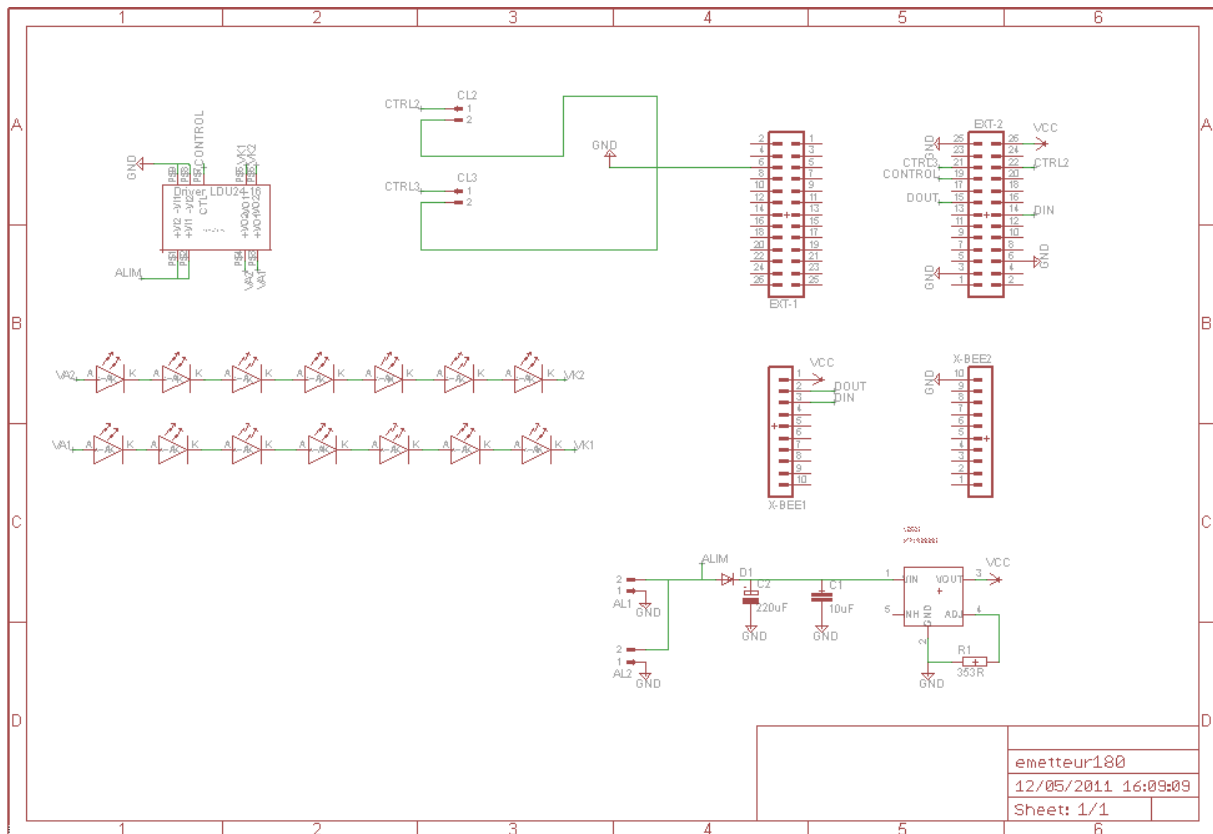
ANNEXE

Annexe : Schématique cartes réception (balises embarquées)





Annexe : Schématique cartes émission (balises terrain)



Annexe : Configuration des X-Bee

Nous n'avons pas pu tester la communication entre 3 X-Bee, donc nous ne mettons ici que ce dont nous soyons sur des résultats, c'est-à-dire la configuration de 2 X-Bee pour qu'ils communiquent ensemble sans être gênés par d'éventuels autres X-Bee.

Ces configurations sont à appliquer avec un utilitaire comme X-CTU.

X-Bee sur le robot ARES :

```
+++  
atre  
atdl1432  
atmy2341  
atbd7  
atwr  
atcn
```

X-Bee sur la balise 180° :

```
+++  
atre  
atmy1432  
atdl2341  
atbd7  
atwr  
atcn
```

+++ (sans retour à la ligne) permet d'entrer dans le mode superviseur

atre = paramètres par défaut

atmy = mon adresse

atdl = adresse de destination

atbd = vitesse de communication

atwr = enregistrer les paramètres

atcn = quitter le mode superviseur