

RAPPORT DE PROJET

ALIMENTATION
CARTE MERE
COMMUNICATION

Samuel HUET

Coéquipiers

*Thomas COUTANT
Jean-Gabriel MASSICOT
Antoine VENIER*

Encadrants

*Aloïs MICLO
Steve NGUYEN*

Table des matières

INTRODUCTION	3
CAHIER DES CHARGES	4
PLANIFICATION	4
PREVISIONNEL	4
REEL	4
GESTION DE LA BATTERIE	5
LES BATTERIES LiPo	5
DIAGRAMMES FONCTIONNELS	6
SYSTEME DE CHARGE ET DECHARGE	8
ETUDE	8
DISTRIBUTION DE PUISSANCE	11
PUISSANCES REQUISES	11
LES REGULATEURS	11
INTERRUPTEUR	15
MISE EN ŒUVRE	16
CHOIX DE LA BATTERIE	23
PROBLEMES ET SOLUTIONS	23
CARTE MERE	24
DIAGRAMME FONCTIONNEL	24
NECESSITES	24
DIAGRAMME FONCTIONNEL	25
FONCTIONS	25
SIGNAUX	26
MICROCONTROLEUR	27
CHOIX	27
ENVIRONNEMENT	28
BOUTON RESET	28
FILTRAGE	28
QUARTZ	29
CONNECTION	30
CENTRALE INERTIELLE	30
CAPTEURS OPTIQUES	30
ESC	30
MODULE RADIO	30
REPARTITION DES I/O	31
NIVEAU DE BATTERIE	33
MISE EN ŒUVRE	34
DIAGRAMME STRUCTUREL	34
TOP LAYER	36
BOTTOM LAYER	37
TOP OVERLAY	38

BOTTOM OVERLAY	39
PLAN DE PERÇAGE	40
COMPOSANTS	41
BRANCHEMENTS	41
PROBLEMES ET RESOLUTIONS	42
PROGRAMMATEUR	42
SCHEMATIQUE	43
MISE EN ŒUVRE	43
PROBLEMES ET SOLUTIONS	45
CARTE D'EXTENSION	46
SCHEMATIQUE	46
MISE EN ŒUVRE	47
<u>PROGRAMME</u>	<u>48</u>
BATTERIE	48
ETUDE DE FONCTION ADC	48
MISE EN ŒUVRE	50
MODULE RADIO	50
ETUDE DE FONCTION SPI	51
MISE EN ŒUVRE	53
I.H.M	53
ETUDE DE FONCTION UART	54
MISE EN ŒUVRE	56
<u>CONCLUSION</u>	<u>57</u>
<u>ANNEXES</u>	<u>58</u>
CODE	58
LEXIQUE	58
SOURCES	59

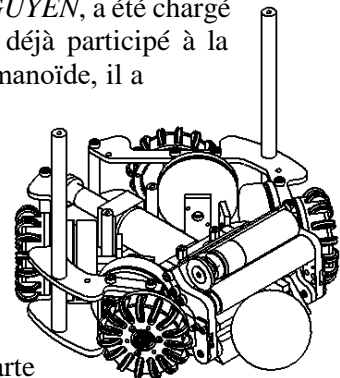
Introduction

Dans le cadre de notre projet de 3^{ème} année à Ynov – Aéronautique et Systèmes Embarqués (anciennement ESTEI), il nous a été demandé de concevoir un prototype de robot pouvant participer à la RobotCup, catégorie SSL (Small Size League). Pour ce faire, notre robot doit correspondre à un cahier des charges précis et les rôles ont été répartis au sein de 4 groupes de 4 personnes, chaque groupe ayant pour objectif d’avoir un robot fonctionnel à la fin du temps imparti.

La Robocup est une compétition internationale, conçue pour mettre en évidence les progrès en robotique et en intelligence artificielle. Elle se divise en plusieurs catégories allant des robots humanoïdes de différentes tailles, jusqu’au robots holonomes. Les robots possèdent un microcontrôleur tous liés à divers capteurs et actionneurs leur permettant de conserver la balle et de l’envoyer à un autre robot ou de marquer un but. Les règles sont précises et données sur le site officiel parmi lesquels on retrouve des problématiques de dimension, mais également de vitesse. Ainsi dans le cadre de notre prototype, le cahier des charges est légèrement amoindri afin de nous faciliter la tâche. Le robot final quant à lui devra respecter toutes les règles du jeu pour avoir la chance de participer à la Robocup 2020 à Bordeaux.

Un dossier nous a été remis, nous expliquant les problématiques à étudier, ainsi que certains points du cahier des charges, comme par exemple un budget de 500€. Un lien vers ce dossier est disponible dans l’annexe de ce document. De plus, un intervenant extérieur, *Steve NGUYEN*, a été chargé de nous superviser, et nous aiguiller tout au long de ce projet. Ayant déjà participé à la compétition, et l’ayant remporté à plusieurs reprises dans la catégorie humanoïde, il a su nous conseiller au mieux.

Dans mon groupe, les rôles ont été répartis ainsi : *Thomas COUTANT*, en charge de créer un système capable d’envoyer la balle, mais aussi de la conserver afin de dribler. *Jean-Gabriel MASSICOT*, dont le rôle est d’animer les moteurs grâce à des ESC. *Antoine VENIER*, chargé d’asservir les déplacements grâce à divers capteurs, et enfin *Samuel HUET*, désigné pour gérer la batterie, fournir les puissances nécessaires, assurer la cohésion des différentes parties au sein du robot grâce à une carte mère et assurer la communication radio.



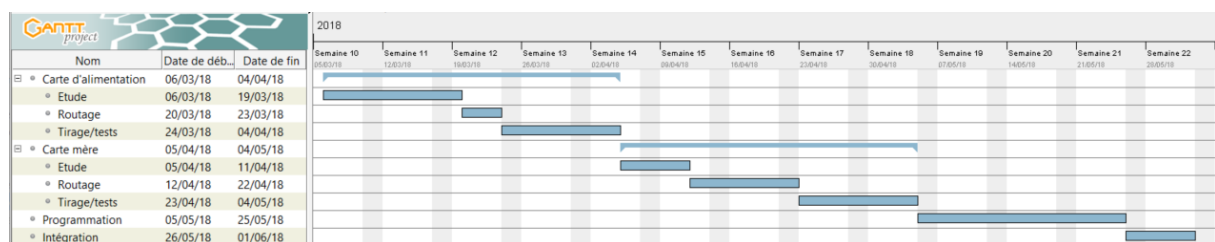
Cahier des charges

Critère	Imposé	Personnel
Famille de microcontrôleur	NXP	
Jeux d'instruction	CISC	
Dimension	25x25 cm	15x20 cm
Batterie	LiPo 3s	
Communication	Radio	
Module radio	nRF24L01	
Autonomie	De 20 à 35min	
Langage de programmation	C	
Logiciel de CAO	Altium Designer	
Budget maximum	500€	200€
I.H.M Niveau de batterie		> 3 précision
Temps de charge		5h max
Tension de charge		Entre 5v et 20v
Température de fonctionnement		Entre 0 et 40°C

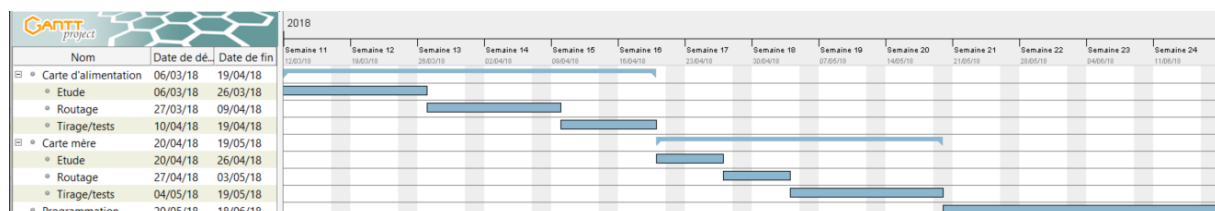
Planification

Afin de planifier les différentes étapes du projet, j'ai réalisé un diagramme de Gantt, montrant toutes les étapes allant de la recherche de solutions techniques jusqu'au débogage de la carte. Le planning prévisionnel a été réalisé dès les premiers jours du projet, et est à comparer avec le diagramme réel qui a été réalisé à la fin du projet. On peut constater qu'un retard a été pris dès le démarrage. En effet, ayant sous-estimé les problématiques liées à la carte d'alimentation, toutes les autres tâches en ont été affectés.

Prévisionnel



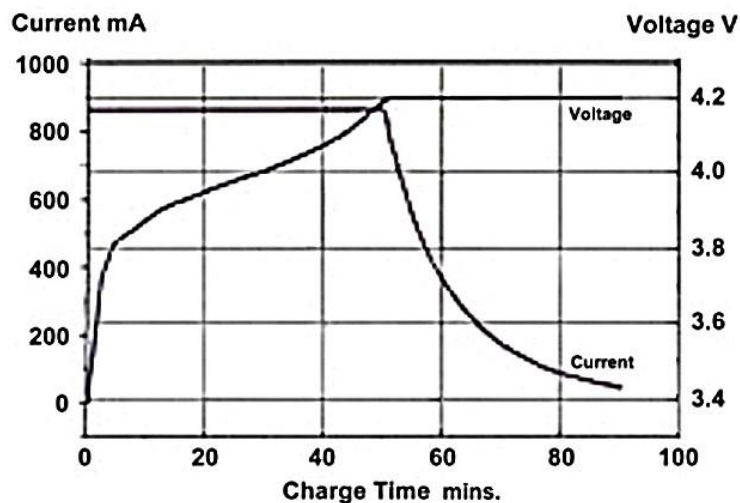
Réel



Gestion de la Batterie

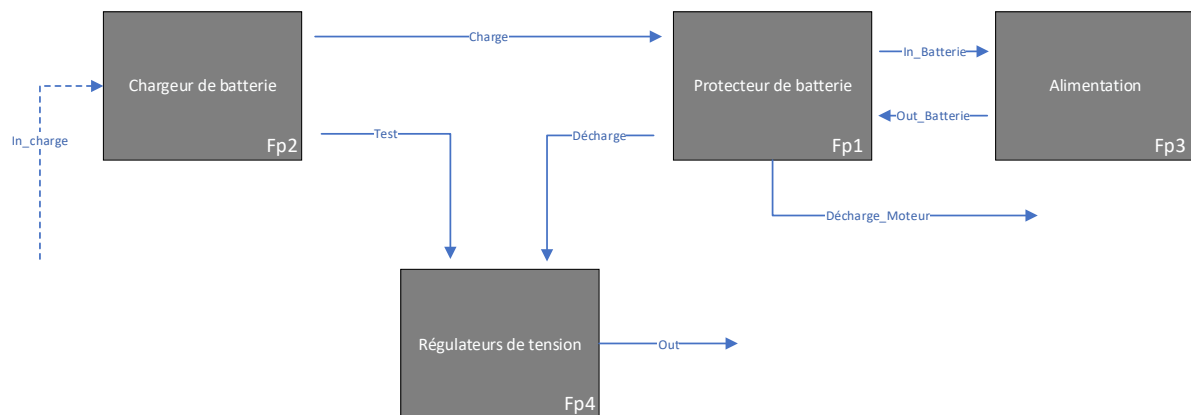
Les batteries LiPo

Les batteries Lithium-Polymère sont à ce jour les plus performantes s'agissant du rapport puissance/capacité. A performance équivalente, une batterie d'un autre type (MiMh, NiCd...) aura une masse plus importante. De plus, leur capacité de décharge fait partie des meilleures. Les batterie LiPo sont donc particulièrement adaptée au modélisme. La batterie imposée dans ce projet est une batterie 3S, qui indique qu'elle inclue 3 cellules en série. La capacité est variable mais tourne généralement autour de 2000 mA/h, ce qui veut dire que la batterie pourra fournir 2000mA pendant une heure, après quoi il faudra la recharger. Une décharge plus rapide est possible, et indiqué par le nombre de C. Par exemple, si une batterie a 20C, avec une capacité de 2000mA/h, elle pourra décharger 20C = 40 A, mais seulement pendant $60/20 = 3$ minutes. Chaque cellule à une tension maximum à ses bornes de 4.2V au maximum et 3.7V en typique. Notre batterie atteindra donc 12.6V max et 11.1V typique. Attention cependant à ne pas tomber en décharge profonde. En effet, chuter sous les 3V par cellule lors de la décharge lui fera perdre en performance voir la rendra inutilisable. Une batterie LiPo a une durée de vie d'environ 3 ans, qu'on l'utilise ou non. Passé ce temps, elle perdra en performance. Elle offre ses meilleures performances lors de ses 100 premier cycles charge/- décharge. Concernant la charge, les batteries à base de lithium demandent une tension constante. Attention à ne pas dépasser 4.2V par élément sous peine de voir des flammes apparaître. Pour la majorité des batteries de ce type, le courant ne doit pas dépasser 1C (certains modèles acceptent 3 voir 4C). Cependant, les chargeurs font en sorte de diminuer le courant quand la charge arrive à sa fin.

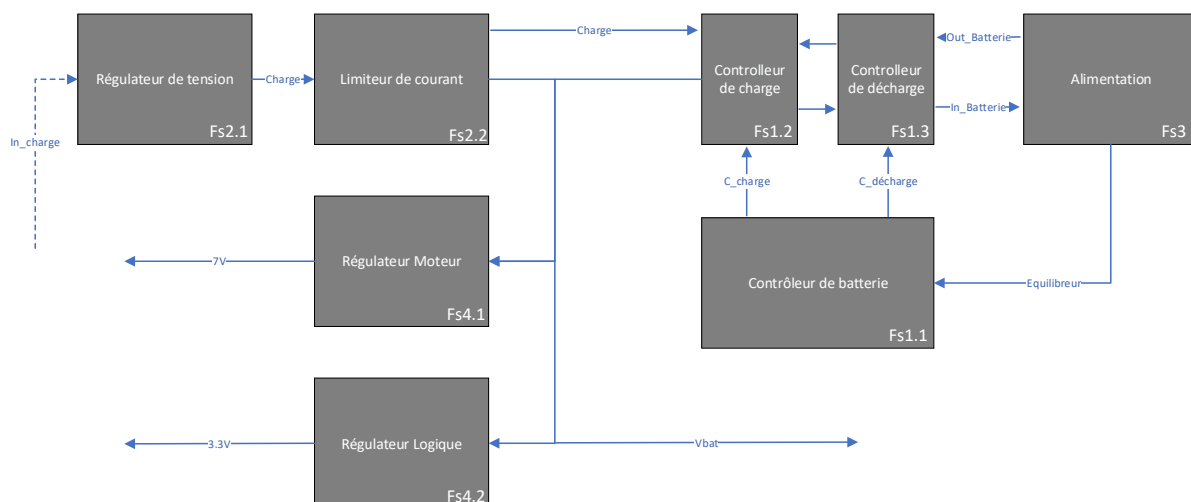


Diagrammes fonctionnels

Niveau 1



Niveau 2



Fonctions

<i>Fonction</i>	<i>Description</i>
<i>Fs1.1 – Contrôleur de batterie</i>	Contrôle la tension de chaque cellule composant la batterie et veille à ce qu'elle ne tombe ni en décharge profonde, ni en surcharge
<i>Fs1.2 – Contrôleur de charge</i>	Commandé par le Contrôleur de batterie, ce dispositif coupe la charge si nécessaire.
<i>Fs1.3 – Contrôleur de décharge</i>	Commandé par le Contrôleur de batterie, ce dispositif coupe la décharge si nécessaire.
<i>Fs2.1 – Régulateur de tension</i>	Adapte la tension d'entrée vers une tension capable de charger la batterie.
<i>Fs2.2 – Limiteur de courant</i>	Limite le courant entrant dans la batterie lors de la charge afin de ne pas l'endommager.
<i>Fs3 - Alimentation</i>	Batterie fournissant tension et courant lors de l'utilisation.
<i>Fs4.1 – Régulateur Moteurs</i>	Régule la tension de batterie en une tension permettant le fonctionnement d'un moteur, tout en laissant le courant nécessaire à ce dernier pur fonctionner.
<i>Fs4.2 – Régulateur Logique</i>	Régule la tension de batterie en une tension permettant le fonctionnement du dispositif logique.

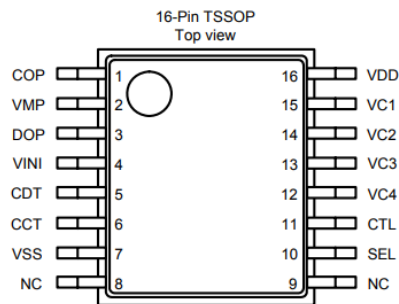
Signaux

<i>Signal</i>	<i>Description</i>
<i>In_Charge</i>	Tension apportée par une alimentation externe, visant à charger la batterie, régulée et limitée en courant par la suite.
<i>Charge</i>	Tension régulée à 15V, continue et limité en courant, dans le but de charger la batterie et éventuellement procéder à des tests.
<i>7V</i>	Tension continue de 7V, régulé depuis la tension de batterie. Limité en courant et visant à alimenter le moteur du dribbleur.
<i>3.3V</i>	Tension continue de 3.3V, régulé depuis la tension de batterie. Limité en courant et visant à alimenter le microcontrôleur et la partie logique.
<i>In_Batterie</i>	Tension de charge de la batterie.
<i>Out_Batterie</i>	Tension de sortie de la batterie.
<i>C_charge</i>	Tension logique, commandant la coupure ou la mise en route de la charge.
<i>C_décharge</i>	Tension logique, commandant la coupure ou la mise en route de la décharge.
<i>Equilibreur</i>	Tension analogique de chaque cellule composant la batterie afin d'en déterminer l'état.
<i>VBat</i>	Tension de batterie variant de 9V à 12.6V

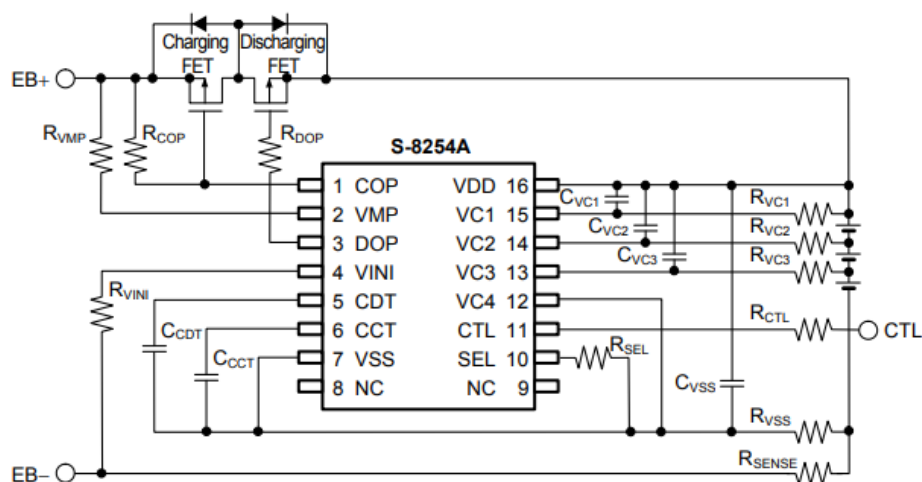
Système de Charge et Décharge

Etude

Afin de gérer la surcharge mais aussi pour éviter les décharges profondes, j'ai décidé d'utiliser le circuit intégré S-8254A, présent notamment sur les BMS disponibles sur internet (HX-3S-01). Ce composant gère donc les voltages extrêmes cellules par cellules pouvant aller de 3 V/cellules pour une décharge et de 4.28 V/cellules pour une surcharge, selon le modèle choisi : S-8254AAPFTTB-x. Pour ce faire, ce composant utilise des MOSFET afin de contrôler la charge et la décharge, qu'il faudra alors dimensionner en fonction des courants entrants et sortants. De plus, il faudra ajouter un régulateur capable de réguler en courant et en tension pour la charge. En effet, les batteries LiPo se chargent à tension/courant constants. Voici ce composant :



De plus, son application typique correspond totalement à nos attentes. J'utiliserai donc ce schéma dont les valeurs sont données par la datasheet. Cependant les MOSFET présents sur le HX-3S-01 ne sont pas capable de dissiper suffisamment de puissance, j'ai donc choisi de les remplacer par des IPB80P03P4L-04, qui ont une capacité à faire passer l'ampérage plus importante ainsi qu'un trou afin d'y appliquer un dissipateur.



Pour une consommation maximal estimé à 44A, une consommation de 11A en typique, et une résistance interne des transistors de 5mΩ, la formule suivante nous donnera sa puissance, à partir de laquelle nous pourrions estimer son augmentation de température.

$$P = I^2 \cdot R_{DSon} = 10W_{max} = 0.6W_{typ}$$

Avec un radiateur de 10°C/W sur chacun des transistors $R_{th} = 11.1 \text{ } ^\circ\text{C/W}$

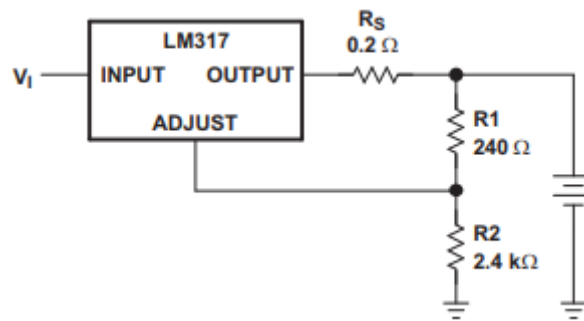
$$T_{max} = R_{th} \cdot P_{max} = +111^\circ\text{C}$$

$$T_{typ} = R_{th} \cdot P_{typ} = +6.6^\circ$$

Sachant que la puissance maximum sera atteinte dans de rares cas, cette température est correcte car elle se trouve tout de même en dessous de la température maximum du transistor qui est de 175°C. Il manquera alors uniquement au système de charge un régulateur tension/courant ainsi qu'un transformateur qui puisse se brancher sur secteur, afin de charger à courant/tension constants.

Régulateur de charge

Pour cela nous pouvons utiliser des LM317, capables de réguler une tension d'entrée de 15v et de limiter le courant à 1.5A.



Ainsi, un transformateur trouvable dans le commerce de 15v, pouvant fournir au moins 1 ampère suffira afin de recharger la batterie du robot, d'après les formules de la datasheet :

- $R_2 = \frac{V_{out}(R_1+1)}{1.25} = 10.090k\Omega$ si $R_1 = 1k\Omega$
- $R_s = \frac{1.25}{I_{out}} = 1.78\Omega$ pour un courant de sortie de 0.7A

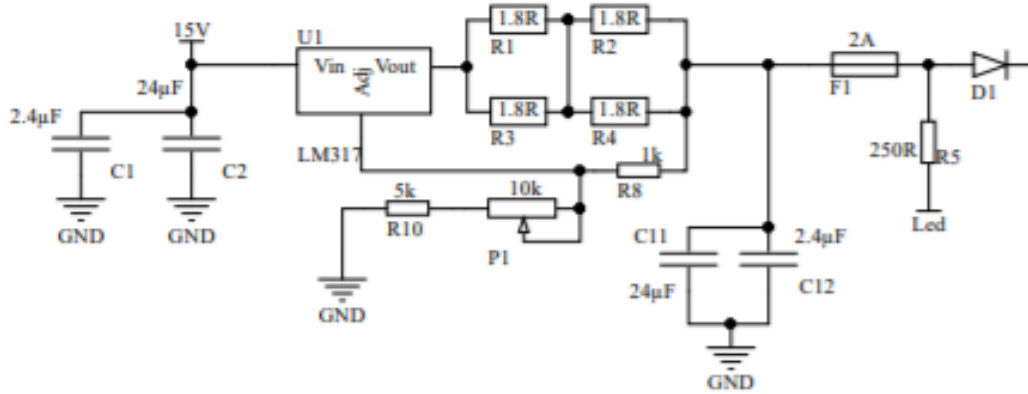
Pour une meilleure précision, on utilisera un potentiomètre à la place de R_2 . Concernant la dissipation, nous avons une puissance à dissiper de :

$$P = (V_{out} - V_{in}) \cdot I$$

$$P = (15 - 12.6) \cdot 0.7 = 1.68W$$

En prenant un boîtier TO-220, on obtient une résistance thermique de 37.9 °C/W. Le composant montera donc à une température de 103 °C qui est en deçà de la température maximum à 150 °C. Avec ce régulateur linéaire, nous avons un rendement de :

$$R = \frac{V_{out}}{V_{in}} = 84\%$$



Dissipateur

Concernant les deux transistor MOSFET dédiés à la charge et à la décharge, il serait intéressant de calculer la puissance qu'il leur sera nécessaire de dissiper. La puissance est donnée par la formule :

$$P_W = I^2 \cdot R_{DSon} = 8.3W$$

Pour des raisons de marge de sécurité, nous allons utiliser une puissance à 8.5W afin d'avoir une petite marge. Sans radiateur, on peut atteindre la température de 552°C. Inconcevable sachant que la température max du composant n'est que de 175°C. Il sera donc nécessaire d'y fixer un radiateur.

$$R_{th} = \frac{T_{max} - T_{ambient}}{P_{dissipée}} = 17.6 \text{ } ^\circ\text{C/W}$$

$$R_{th} = R_{jonction \rightarrow boîtier} + R_{boîtier \rightarrow radiateur} + R_{radiateur \rightarrow air}$$

$$R_{radiateur \rightarrow air} = \frac{T_{max} - T_{ambient}}{P_{dissipée}} - (R_{jonction \rightarrow boîtier} + R_{boîtier \rightarrow radiateur})$$

$$R_{radiateur \rightarrow air} < 15.4 \text{ } ^\circ\text{C/W}$$

Mais encore une fois, la prudence est de mise. En effet, les cartes étant censés être confinées au sein d'une coque, la température ambiante risque de monter. Il sera donc préférable d'utiliser un radiateur d'une dizaine de °C/W.

Distribution de Puissance

Puissances requises

	<i>Tension</i>	<i>Courant Max</i>	<i>Puissance Max</i>
<i>Moteurs</i>	Vbatterie	40A	504W
<i>Kicker</i>	Vbatterie	N.C.	N.C.
<i>Dribleur</i>	7V	4.41A	32W
<i>Photodiodes</i>	3.3V	100mA	0.33W
<i>Carte Mère</i>	3.3V	150mA	0.5W
<i>Capteurs Optiques</i>	3.3V	108mA	0.4W
<i>Capteurs à effet hall</i>	3.3V	100mA	0.5W
<i>Module Radio</i>	3.3V	12mA	0.04W
<i>Centrale Inertielle</i>	3.3V	3.9mA	12.87mW
<i>TOTAL</i>		44.98 A	

Les moteurs ainsi que le kicker seront branchés directement sur la batterie. En effet, ils demandent bien trop de puissance pour leur assigner un régulateur. Le dribleur quant à lui demande 1.5 ampère en nominal, lorsque le moteur atteint effectivement les 4.4A, cela signifie que quelque chose bloque le moteur. On peut alors imaginer un fusible thermique qui permettra d'en couper l'alimentation si un tel cas de figure arrivait.

Les régulateurs

Régulateur linéaire

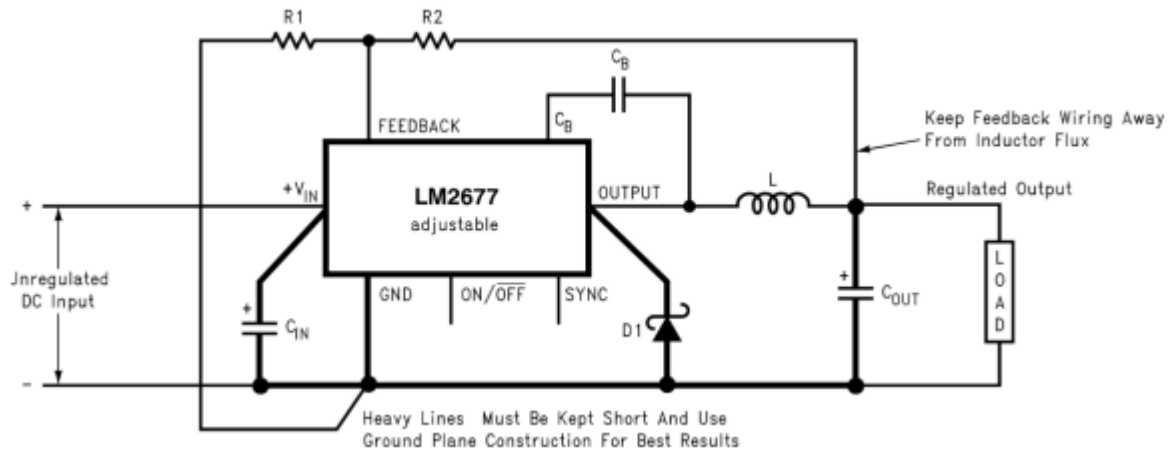
Le régulateur linéaire provoquera une chute de tension qui ajoutera systématiquement à la puissance à dissiper. Ils possèdent donc un très mauvais rendement et sont plutôt adaptés aux faibles puissances. Leur avantage majeur réside dans leur facilité de mise en œuvre : en effet, pour la majorité d'entre eux, deux résistances et deux condensateurs suffisent à son fonctionnement.

Régulateur à découpage

Le régulateur à découpage fonctionne grâce à une PWM appliqué à la tension d'entrée. Il prélèvera donc une fraction de la tension d'entrée. Cette technologie permet un rendement qui peut s'élever jusqu'à 95% et donc une dissipation moindre. Mais sa mise en œuvre est bien plus complexe par son environnement (inductance, condensateurs, résistances...). Ils peuvent donc s'avérer légèrement plus coûteux, il est aujourd'hui très répandu

Régulateur 7v

Ce régulateur doit être capable d'abaisser une tension comprise entre 9V et 12V, en 7V, ainsi que de faire passer 4.41A dans le pire des cas. Le LM2677 remplit ces critères.



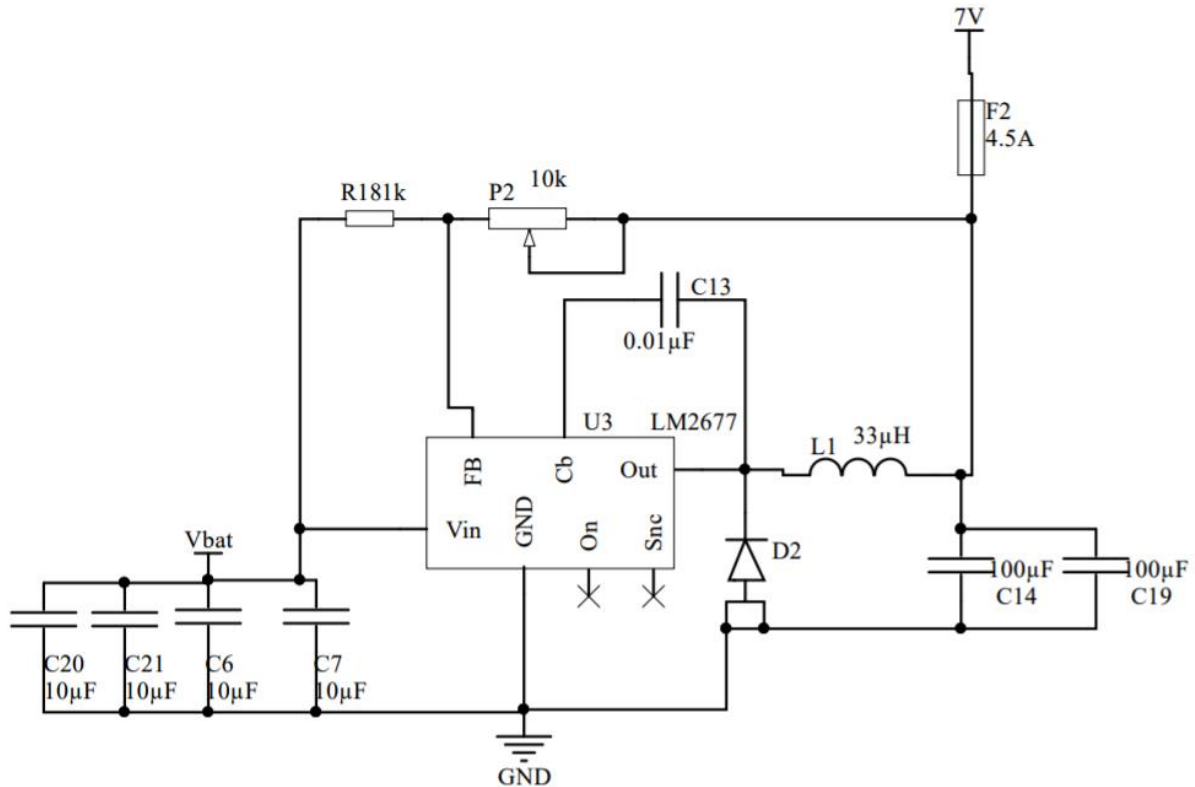
Ce composant possède une broche "feedback" qui permet de régler la tension de sortie grâce à un simple pont diviseur, tension de sortie donnée par la formule suivante :

$$V_{out} = 1.21 \left(1 + \frac{R_2}{R_1} \right)$$

$$\text{Si } R_1 = 1k\Omega, \text{ alors } R_2 = 4785\Omega$$

Il sera préférable de placer un potentiomètre à la place de R2 afin de pouvoir régler la tension avec plus de précision. La datasheet fournit également les valeurs des différents composants :

- $L = 33\mu H$
- $C_{in} = 4 \cdot 10\mu F$
- $C_{out} = 2 \cdot 100\mu F$
- $C_B = 0.01\mu F$
- $D_1 = MBRD835L$ (modèle recommandé)



Grâce aux graphiques fournis dans la datasheet, nous pouvons en déduire que le rendement se situera aux alentours des 87%, là où un régulateur linéaire nous aurait fourni au mieux 77%.

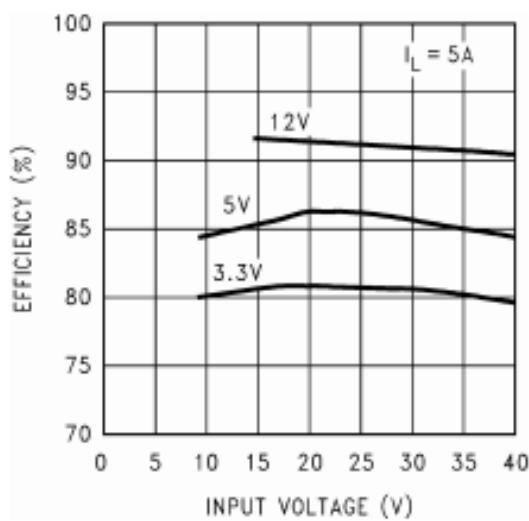


Figure 3. Efficiency vs Input Voltage

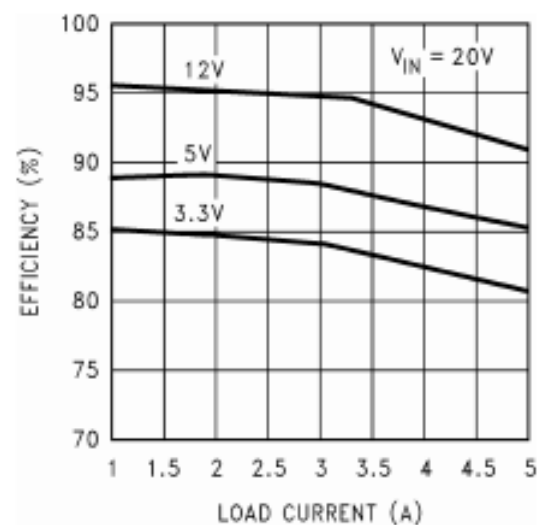


Figure 4. Efficiency vs I_{LOAD}

Concernant la puissance traversant les régulateurs à découpage, la formule suivante s'applique :

$$P = \frac{V_{out} \cdot I_{out}}{\text{Rendement}} - (V_{out} \cdot I_{out})$$

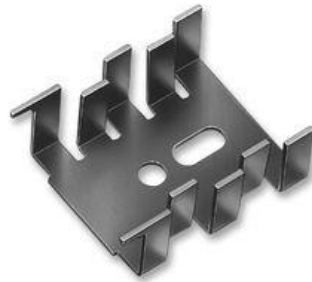
$$P = \frac{7 \cdot 4.41}{0.87} - (7 \cdot 4.41) = 4.6W$$

Sans dissipateur, et avec une puissance de 4.6W, ce composant peut atteindre les 325°C. Sa température maximum en fonctionnement est de 125°C, calculons alors le dissipateur nécessaire à son bon fonctionnement.

$$R_{radiateur \rightarrow air} = \frac{T_{max} - T_{ambient}}{P_{dissipée}} - (R_{jonction \rightarrow boitier} + R_{boitier \rightarrow radiateur})$$

$$R_{radiateur \rightarrow air} < 15.37 \text{ }^{\circ}\text{C/W}$$

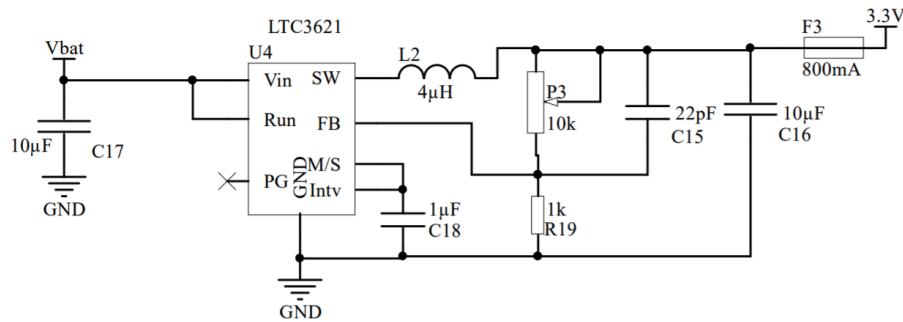
Etant donné que la puissance calculée est un maximum et n'arrivera qu'en de très rares cas, inutile de surdimensionner le dissipateur. Celui présent ci-contre est un dissipateur 15°C/W et un fusible sera rajouté si les 4 ampères sont dépassés.



Régulateur 3.3V

Il est donc nécessaire de réguler une tension de 3.3V depuis 11.1V, avec un courant de charge maximum de 578mA. Pour cela, nous pouvons utiliser un régulateur linéaire LTC3621 possédant une sortie fixe à 3.3V et un courant sortant maximum de 1A.

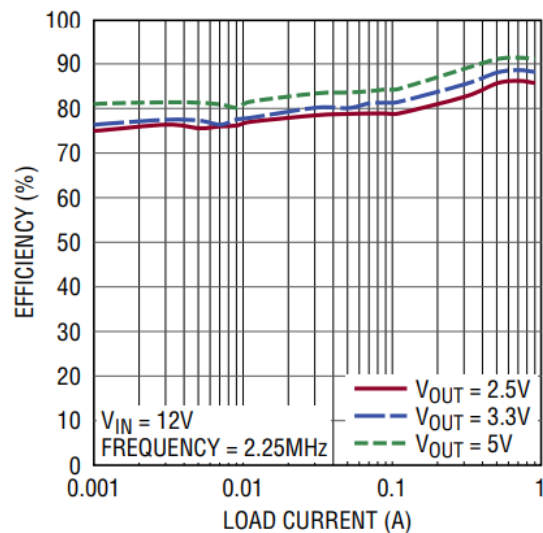
- $R_1 = 1k\Omega$
- $R_2 = (V_{out} - 0.6) \frac{R_1}{0.6} = 4.5k\Omega$
- $C_{FF} = 22pF$
- $C_B = 1\mu F$
- $C_{out} = 10\mu F$
- $C_{in} = 10\mu F$
- $L = \frac{V_{out}}{f \cdot \Delta I_{L(MAX)}} \left(1 - \frac{V_{out}}{V_{in(max)}} \right) = 4\mu H$



Voici son graphique de rendement. On peut voir qu'il reste relativement stable et haut quel que soit le courant tiré. Grâce à ce graphique, on peut déterminer la puissance à dissiper :

$$P = \frac{V_{out} \cdot I_{out}}{\text{Rendement}} - (V_{out} \cdot I_{out})$$

$$P = \frac{3.3 \cdot 0.578}{0.9} - (3.3 \cdot 0.578) = 0.21W$$



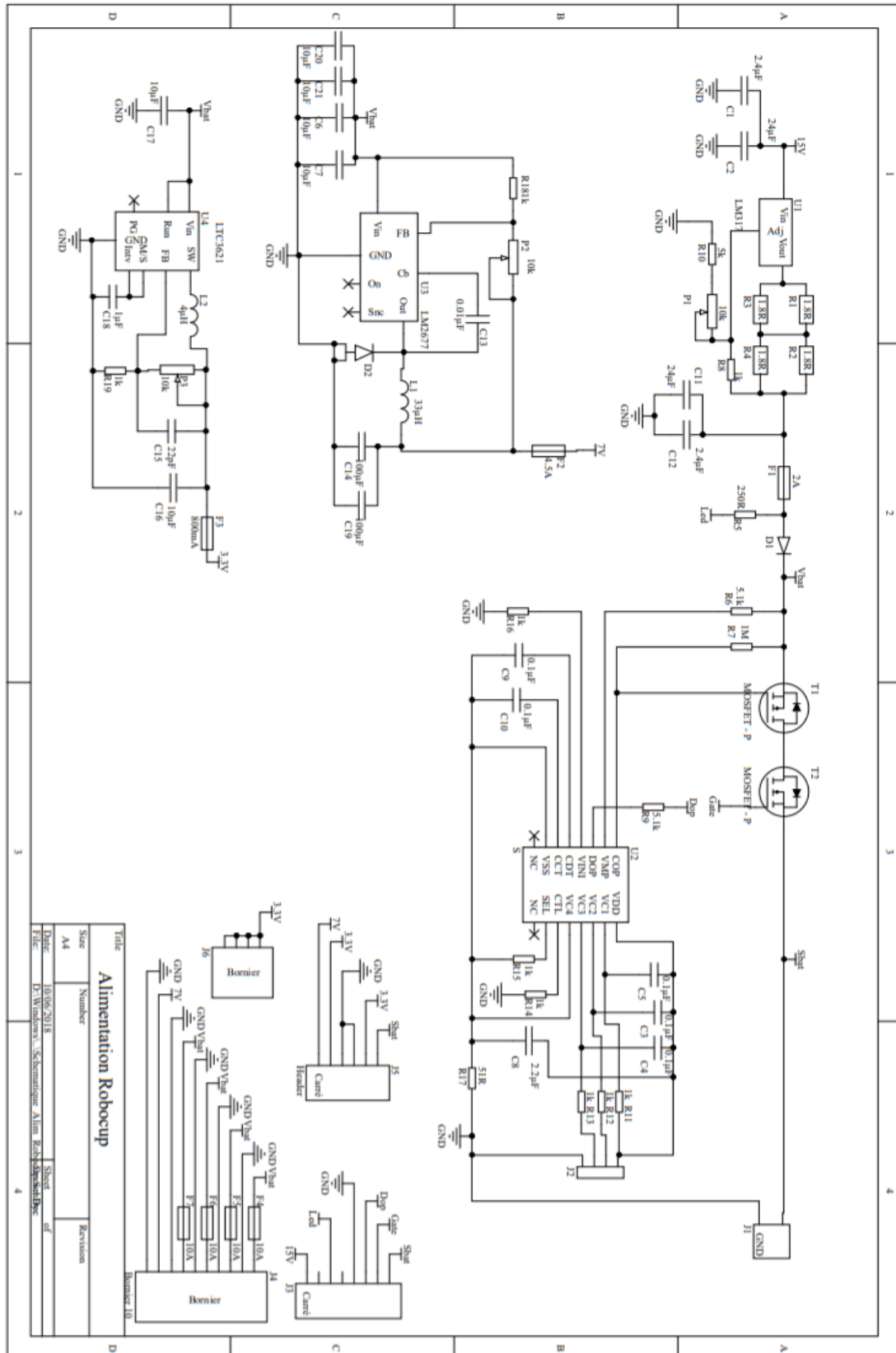
Cette faible puissance génèrera une augmentation de température de 8.4 °C, ce qui est très raisonnable et nous permettra de rester bien en deçà de la température maximum de 125 °C.

Interrupteur

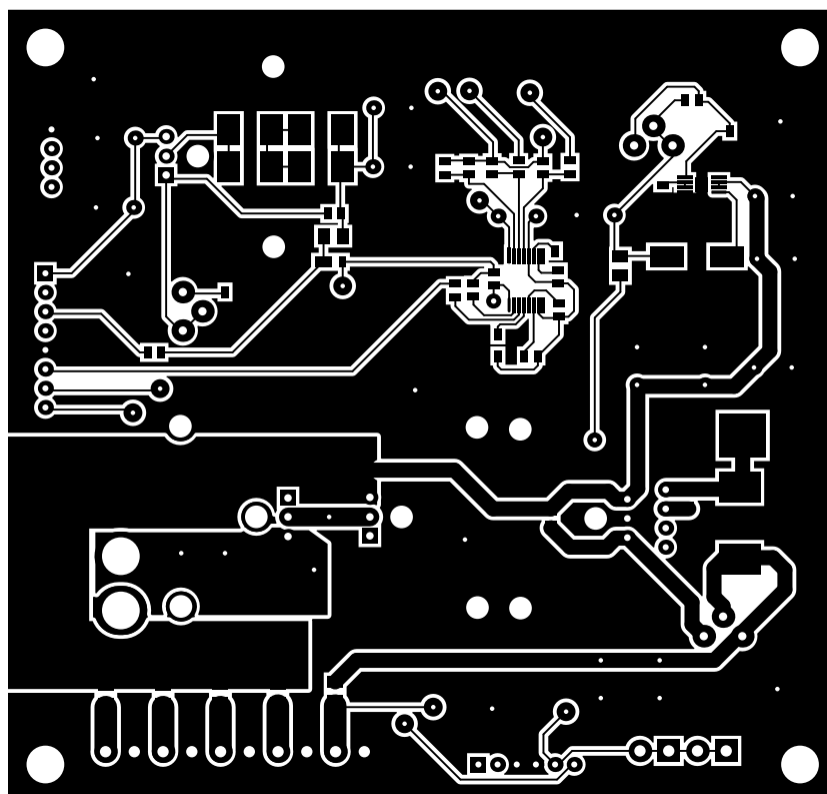
Un interrupteur sera placé sur la grille du MOSFET de décharge, permettant de pouvoir y amener au choix, la tension sortie du gestionnaire de batterie (Mode décharge sous condition que la batterie soit chargée) ou la tension de batterie (Mode éteint, plus aucun courant ne passe). Le BMS possède déjà une broche appelée CTL, permettant de passer d'un mode allumé ou éteint. Cependant cette broche ne permet plus la charge en mode éteint, c'est pourquoi j'ai choisi d'utiliser l'interrupteur de cette façon.

Mise en œuvre

Diagramme structurel

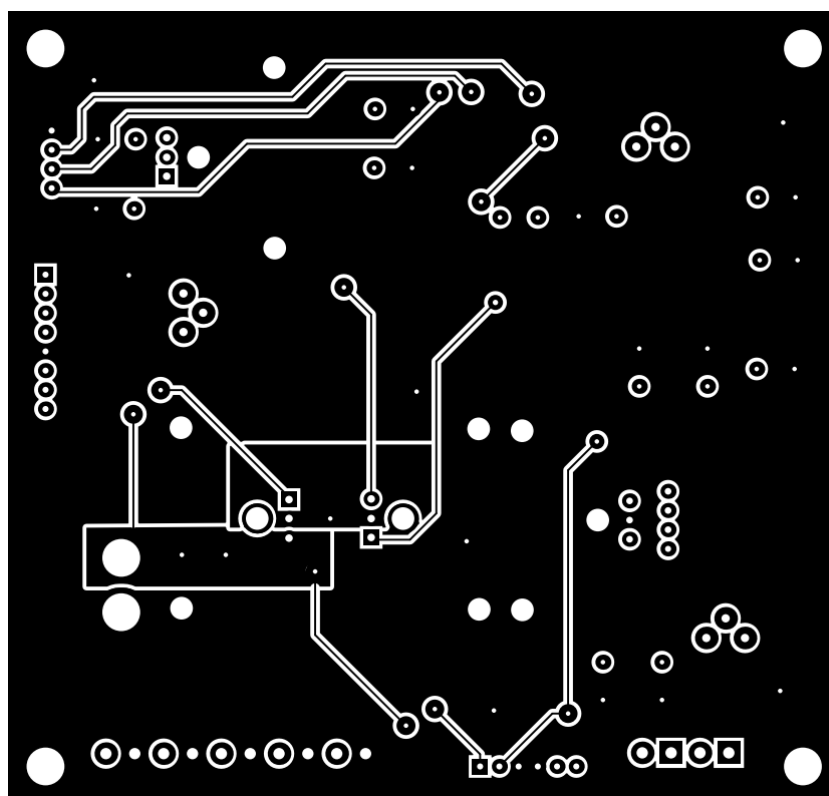


Top Layer



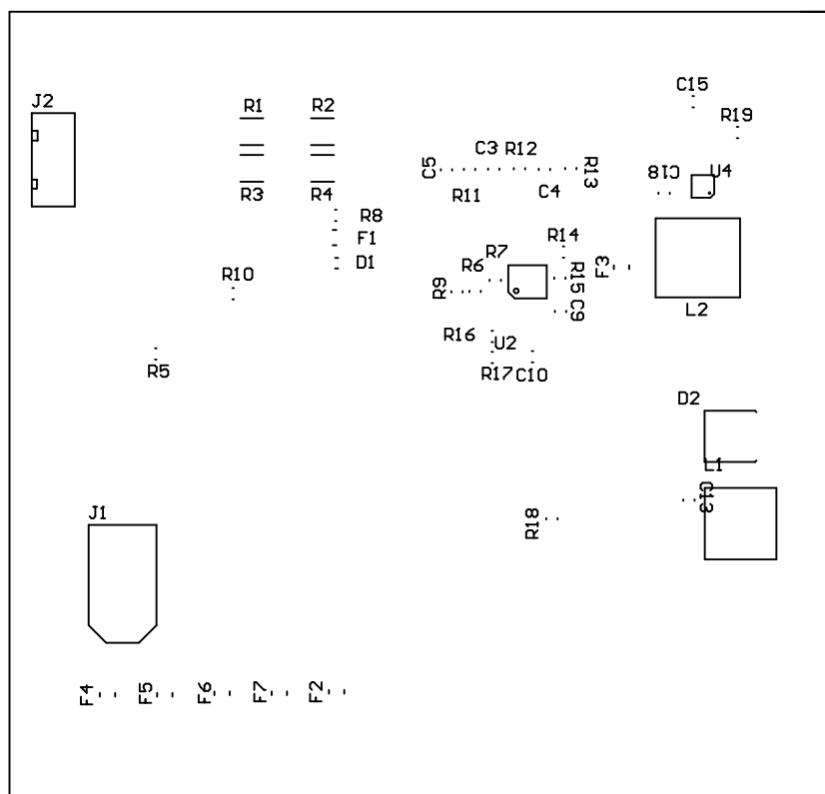
1 - Top Layer - Alimentation (1:1)

Bottom Layer



2 - Bottom Layer - Alimentation (1:1)

Top Overlay

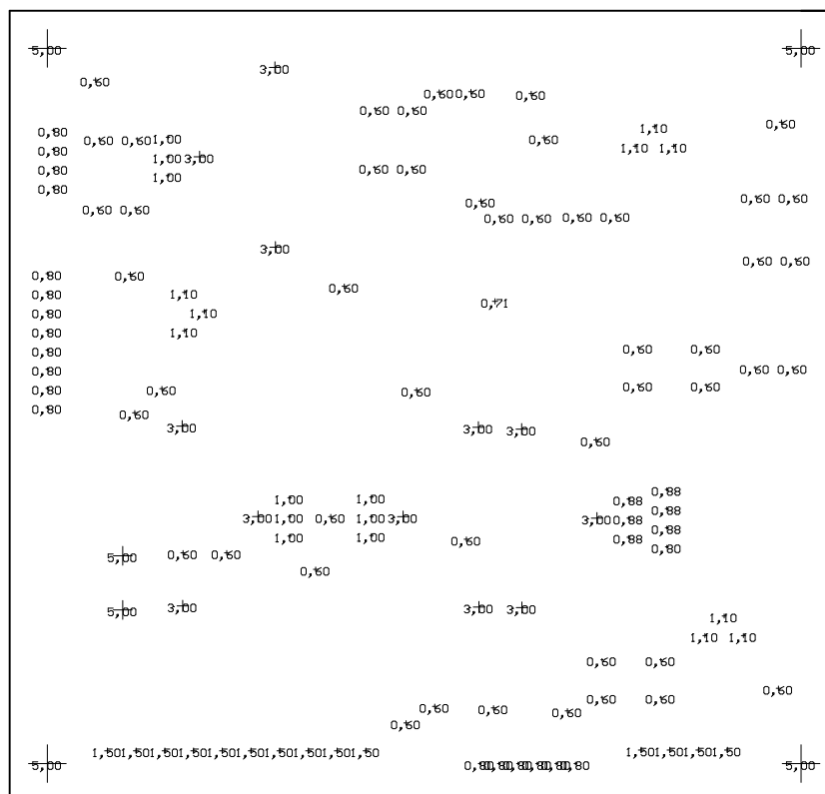


3 - Top Overlay - Alimentation (1:1)

[illegible]

20

Plan de perçage



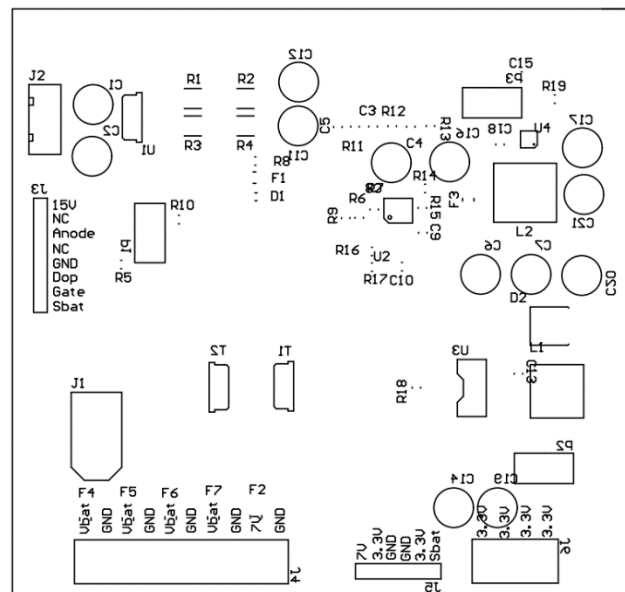
5 - Plan de perçage - Alimentation (1:1)

Composants

Nom du produit	Nom du fabricant	Référence du fabricant	Nom du fournisseur	Référence du fournisseur	Quantité	Prix unitaire H.T. (€)	Total H.T. (€)	Prix TTC (€)	Total TTC
Connecteur de décharge LiPo	Reely	4,0532E+12	Conrad	1373199 - 62	1	1,66 €	1,66 €	1,99 €	1,99 €
Connecteur d'équilibrage LiPo	Modelcraft	4,01614E+12	Conrad	207131 - 62	1	1,33 €	1,33 €	1,60 €	1,60 €
Fusible 1206 - 800mA	Bourns	5F-1206S080-2	Farnell	0291894	1	4,94 €	4,94 €	5,93 €	5,93 €
Régulateur à découpage	TI	0M2677T-ADJ	Farnell	0221960	1	4,61 €	4,61 €	5,53 €	5,53 €
Fusible 1206 - 10A	Littelfuse	0206SFH100F/24-2	Farnell	1832124	1	3,32 €	3,32 €	3,98 €	3,98 €
Bornier 8 voies	Weidmuller	070750000	Farnell	1866613	1	1,93 €	1,93 €	2,31 €	2,31 €
Self 33µH	Coilcraft	MSS1210-333MEB	Farnell	028833	1	1,24 €	1,24 €	1,49 €	1,49 €
Self 4µH	Coilcraft	MLC1245-402MLC	Farnell	0287952	1	1,24 €	1,24 €	1,49 €	1,49 €
P-MOSFET	Infineon	IPP80P03P4L04AKSA1	Farnell	2443406	1	1,20 €	1,20 €	1,44 €	1,44 €
Bornier 2 voies	Weidmuller	BL 5.08/2	Farnell	1131811	1	0,78 €	0,78 €	0,94 €	0,94 €
Gestionnaire de batterie	ABLIC	S-8254AAWFT-TB-G	Mouser	628-S-8254AAWFT-TB-G	2	1,61 €	3,22 €	1,93 €	3,86 €
Régulateur 3.3V	Linear Technology	0TC3621HMS8E-23.3#PBF	Farnell	0483292	1	4,93 €	4,93 €	5,92 €	5,92 €
Transfo 15V	XP Power	VET18US150C2-JA	RS Online	034-6928	1	13,65 €	13,65 €	16,38 €	16,38 €
Diode	ON	0M8RD83SLG	Farnell	9556249	1	0,46 €	0,46 €	0,55 €	0,55 €
Diode schottky	DIODES	0FLS240L-7	Farnell	0713897	1	0,34 €	0,34 €	0,41 €	0,41 €
Fusible 1206 - 1A	BEL	0685F1000-01	Farnell	2834835	1	0,21 €	0,21 €	0,26 €	0,26 €
Resistance 5.1k - 0805	Multicomp	MCWR08X5101FTL	Farnell	2447692	1	0,06 €	0,06 €	0,07 €	0,07 €
Dissipateur	AAVID THERMALLOY	063002B00000G	Farnell	2822613	5	0,67 €	3,35 €	0,81 €	4,03 €
Resistance 1.8R - 2512	Panasonic	ERJ1TRQF1R8U	Farnell	1717963	10	0,36 €	3,59 €	0,43 €	4,31 €
Régulateur à découpage	LINEAR	0TC3621HMS8E-25#PBF	Farnell	0483293	1	4,98 €	4,98 €	5,98 €	5,98 €
Fusible 1206 - 4.5A	Littelfuse	1206SF5450F/32-2	Farnell	0843834	1	3,20 €	3,20 €	3,84 €	3,84 €
Bornier 10voies	Multicomp	MC000024	Farnell	2007993	1	1,93 €	1,93 €	2,32 €	2,32 €
Bornier 4 voies	Phoenix Contact	1727036	Farnell	0704592	1	1,58 €	1,58 €	1,89 €	1,89 €
Resistance 1.5k	WALSIN	0WR08X152 JTL	Farnell	2670956	10	0,04 €	0,42 €	0,05 €	0,50 €
Header 2.54mm femelle 4 contact	Samtec	BCS-104-L-S-TE	Farnell	1667470	5	1,13 €	5,63 €	1,35 €	6,75 €
Batterie 5000mA/h 20C	Turnigy		HobbyKing	9067000276-0	1	17,36 €	17,36 €	20,83 €	20,83 €
Resistance 1M	WALSIN	0WR08X1004FTL	Farnell	2502674	10	0,04 €	0,39 €	0,05 €	0,47 €
TOTAL									105,06 €

Branchements

Afin de se repérer rapidement parmi les divers branchements disponibles sur cette carte, voici un plan des connexions, affichant en même temps le Top Overlay, le Bottom Overlay et le nom des connexions. Le connecteur J3 est donc un connecteur destiné à être lié à une carte d'extension. Le connecteur J4 délivre la puissance vers les moteurs et le dribleur. Quant aux connecteurs J5 et J6, ils sont destinés à alimenter la partie logique du système.



Choix de la batterie

Pour une consommation nominale d'environ 11A, nous pouvons utiliser ce calcul afin de déterminer la capacité de la batterie :

$$C = I \cdot t$$

Ainsi, pour un match de 20min nous aurons besoin d'une batterie de :

$$C = 11 \cdot 0.33 = 3630mAh$$

Et pour un match de 35 min :

$$C = 11 \cdot 0.58 = 6380mAh$$

Afin que le robot puisse tenir sur la durée, j'ai choisi une batterie de 5000mAh, qui lui permettra une autonomie de $t = C \cdot I = 5 \cdot 11 = 0.45$, équivalent à 27min.

Problèmes et solutions

Lors du premier branchement, le chargeur de batterie a semblé fonctionner parfaitement. Cependant, les deux régulateurs ne sortaient pas du tout la bonne tension. Pour le premier, censé convertir la tension de batterie en 7V, le problème venait en réalité d'une erreur de lecture dans la datasheet. En effet, la résistance R_{18} , utilisé pour le Feedback est destinée à être reliée à la masse et non pas à V_{IN} . De plus, le potentiomètre de récupération utilisée avait grillé, ce qui ne me permettait pas de régler la tension de sortie. Après résolution de ces deux problèmes, la tension de 7V sortait parfaitement et il a été possible d'alimenter le moteur du dribbleur. Concernant le régulateur 3.3V, le potentiomètre associé permettait de changer sa tension de sortie, mais aucunement d'atteindre la tension souhaitée, cette dernière ne descendant pas sous les 5V. Après analyse, il s'est avéré que le composant ne respectait pas les règles énoncées dans la datasheet, il a donc été nécessaire de le remplacer. Après l'avoir remplacé, il s'est avéré que le nouveau composant a exactement le même comportement que l'ancien. Après approfondissement, il s'agissait en réalité d'une erreur de commande. En effet, le composant existe en 3 version : 3.3V fix, 5V fix ou ajustable. Ce fut donc une simple erreur de commande de ma part.

Concernant la charge de la batterie, bien qu'étant très lente, elle a fonctionné. Cependant, suite à une erreur de ma part, un boulon est tombé sur la piste lorsque j'ai tenté de fixer le radiateur. Des étincelles ont alors apparues et j'ai été contraint de reprendre les pistes ayant sautées. Mais cela a grillé le BMS et il n'est plus en mesure de commander la décharge. Il est donc nécessaire de le changer.

Carte Mère

Diagramme fonctionnel

Nécessités

Afin de déterminer le nombre de broches et les fonctions nécessaires au microcontrôleur, j'ai réalisé un tableau regroupant les besoins dictés par chacune des personnes de mon groupe. Voici leur besoin selon chaque personne.

Thomas COUTANT, responsable de mise en place d'un dribleur et d'un kicker.

Thomas COUTANT

2 GPIOs
2 ADCs
1 PWM
1 Timer

Antoine VENIER, responsable de la programmation de la centrale inertielle, des capteurs optiques ainsi que de la mise en œuvre des déplacements.

Antoine VENIER

1 PC
1 SPI
2 Chip Selects
2 GPIOs

Jean-Gabriel MASSICOT, responsable de la fabrication des ESC, les GPIO et fonctions utilisé par ces derniers gèreront le mouvement des moteurs, les données suivantes seront donc à multiplier par 4, soit une connexion par moteur.

Jean-Gabriel MASSICOT

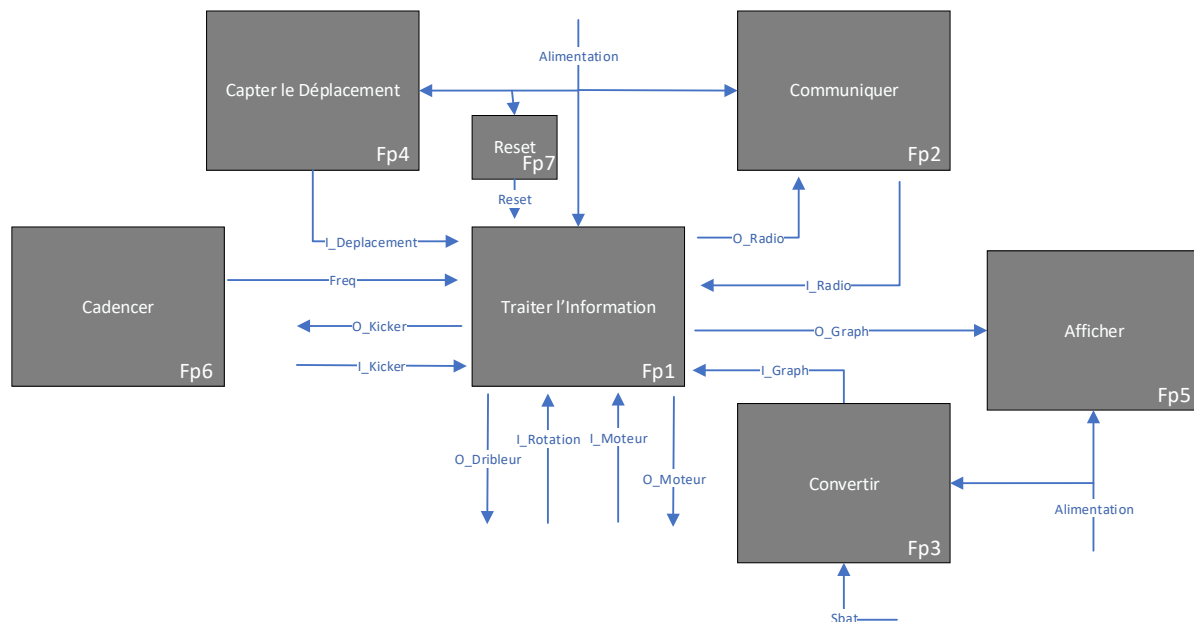
2 GPIOs
1 PWM
1 SPI
1 Chip Select

Et enfin, pour ma part, en plus de certaines broches réservées à la programmation, voici mes besoins concernant le module radio, et l'IHM.

Samuel HUET

Extal/Xtal
RESET
BKGD
1 ADC
5 GPIOs

Diagramme fonctionnel



Fonctions

Fonction	Description
<i>Fp1 – Traiter l'Information</i>	Traite toutes les entrées/sorties tels que les protocoles de communications tout en gérant l'ordre de chaque tâche préprogrammée.
<i>Fp2 - Communiquer</i>	Assure la communication radio avec l'IA commandant le robot.
<i>Fp3 - Convertir</i>	Convertit la tension de batterie en une tension compréhensible par le microcontrôleur.
<i>Fp4 – Capter le déplacement</i>	Informe le microcontrôleur des déplacements exacts effectués par le robot.
<i>Fp5 - Afficher</i>	Donne une information visuelle sur le niveau de batterie.
<i>Fp6 - Cadencer</i>	Cadence le microcontrôleur à une fréquence précise.
<i>Fp7 - Reset</i>	Permet la remise à zéro du programme en cas de besoin.

Signaux

<i>Signal</i>	<i>Description</i>
<i>Alimentation</i>	Tension de 3.3V, fixe et stabilisée visant à alimenter la partie logique.
<i>I_Déplacement</i>	Donnée logique, donnant des informations sur les déplacements x et y.
<i>O_kicker</i>	Ordre logique, ordonnant au kicker d'envoyer la balle.
<i>I_kicker</i>	Information de présence et de position de la balle,
<i>O_Dribleur</i>	Ordre logique mettant en marche le dribleur.
<i>I_Rotation</i>	Donnée logique, donnant des informations sur la rotation du robot ainsi que sur ses déplacements x et y.
<i>I_Moteur</i>	Information sur le sens de rotation et la vitesse du moteur servant à l'asservissement.
<i>O_Moteur</i>	Sorties permettant de chaque contrôler le moteur en vitesse et en position.
<i>I_Graph</i>	Entrée analogique.
<i>O_Graph</i>	Sortie logique permettant de contrôler des LEDs, affichant le niveau de batterie.
<i>I_Radio</i>	Entrée logique, transmettant les informations de déplacement émises depuis le PC
<i>O_Radio</i>	Sortie logique, envoyé au module radio, transmettant les informations sur la position, niveau de batterie du robot.
<i>Sbat</i>	Entrée analogique, tension de la batterie.
<i>Freq</i>	Fréquence de travail du microcontrôleur.
<i>Reset</i>	Tension logique ordonnant au microcontrôleur de revenir au début du programme.

Microcontrôleur

Choix

Afin de contenter tous les besoins de chacune des personnes de mon groupe, j'ai choisi d'utiliser un Microcontrôleur MC9S08QE128, comportant 80 broches. Il possède donc de nombreuses fonctions que nous utiliserons complètement ou partiellement. Voici sa description globale :

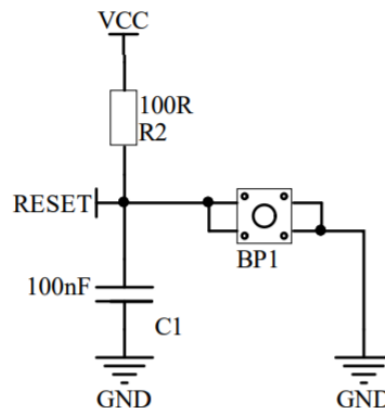
Fabriquant	NXP
Architecture	8 bit
Boîtier	LQFP-80
Série	S08QE
Fréquence d'horloge maximum	50.33MHz
Flash	128kB
RAM	8kB
Résolution ADC	12 bits
Nombre de voie de conversion ADC	24
Nombre d'entrée sortie (I/O)	70
Tension d'alimentation	1.8V à 3.6V
Interfaces	I ² C, SCI, SPI
Nombre de Timer	3

En plus de posséder de nombreux ports GPIO, il a également l'avantage de pouvoir être cadencé jusqu'à 50MHz grâce à une PLL interne.

Environnement

Bouton Reset

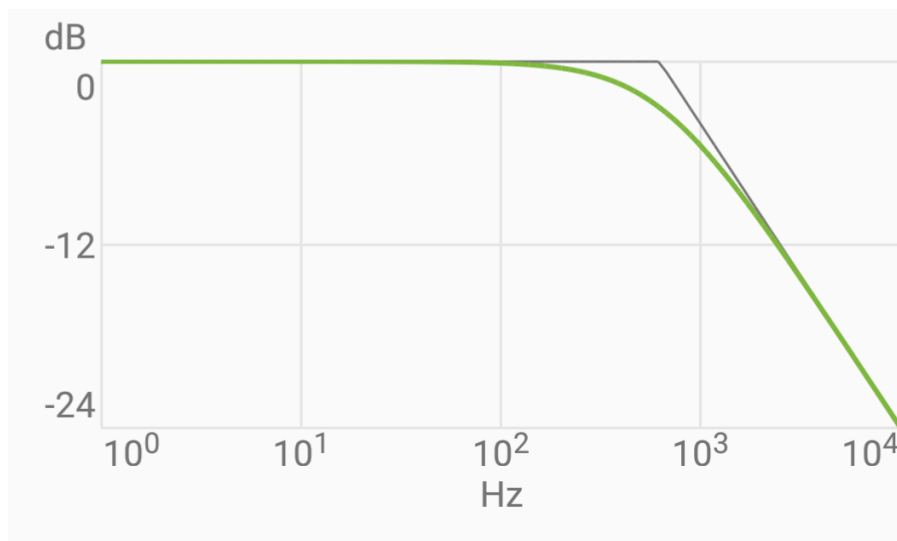
La broche RESET du Microcontrôleur fonctionne en inverse. Elle doit être à l'état haut. Une fois à l'état bas un certain laps de temps, le Microcontrôleur comprends alors qu'il doit revenir au début du programme. Ainsi, avec une résistance de 100Ω et un condensateur de 100nF , on atteint un état bas de $10\mu\text{s}$.



Filtrage

Afin de filtrer la tension d'entrée du microcontrôleur, un filtre passe-bas réalisé par deux condensateurs de découplage suffisent. J'en utiliserai donc un de $10\mu\text{F}$ ainsi qu'un de 100nF , qui devront être placés au plus près du composant afin d'être efficace.

En estimant la consommation maximum du microcontrôleur à 120mA , nous obtenons cette courbe : un filtre passe-bas à une fréquence de coupure 580Hz :

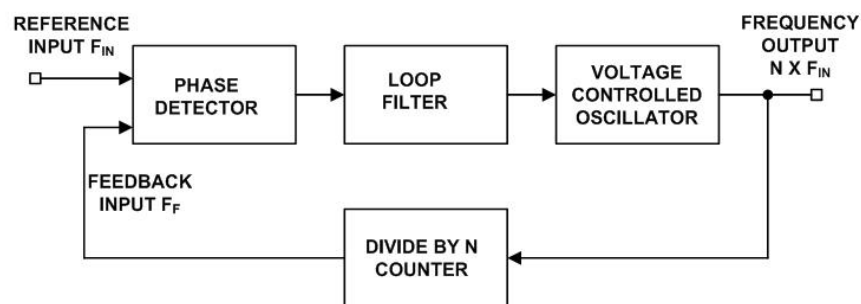


Quartz

Concernant la fréquence du Microcontrôleur, j'ai choisi d'utiliser une Quartz externe à 32.768 KHz. Par rapport à un oscillateur interne, il possède beaucoup d'avantage : le quartz est bien plus précis car moins sensible à la température ou à la pression (l'oscillateur interne étant composé d'un circuit RC). De plus, sa fréquence de 32.768 KHz est beaucoup utilisé notamment dans les montres car facilement divisible en secondes. De plus, une fréquence de 50MHz est atteignable en utilisant une PLL interne.

PLL

La PLL (Phase-locked loop), ou synthèse de fréquence en français, est un moyen très utilisé pour multiplier une fréquence. Elle est notamment utilisée dans la modulation afin de créer une porteuse. Il s'agit d'un asservissement particulier dont voici le schémas fonctionnel.



Ici nous l'utiliserons afin d'augmenter la fréquence de bus du microcontrôleur et ainsi accroître sa vitesse de traitement. Afin de réaliser ce genre d'opération, un oscillateur externe est préconisé. En effet, la multiplication de la fréquence via la PLL entrainera une grande imprécision en utilisant l'oscillateur interne composé d'un circuit RC.

Nous pouvons voir sur le tableau ci-dessous que la fréquence maximum (59.77MHz) peut être atteinte en utilisant un quartz de 32.768 kHz. Cependant, la fréquence maximum du microcontrôleur, inscrite sur la première page de la datasheet est de 50.33MHz. Nous atteindrons donc cette fréquence en utilisant ce quartz.

Ce réglage peut être réalisé par notre IDE (CodeWarrior). En rentrant simplement la fréquence souhaitée, il initialisera seul les registres de façon transparente pour nous.

▼ Clock settings		
▼ Internal clock		
Internal oscillator fre	32.768	32.768 kHz
▼ External clock	Enabled	
▼ Clock source		
Clock frequency [f]	0.032768	0.032768 MHz
Clock range	Low frequency	<31.25 kHz, 1 MHz>
► Low-power modes set		
Initialization interrupt prio	interrupts enabled	1
► CPU interrupts		
▼ Enabled speed modes		
▼ High speed mode		
High speed clock	External Clock	32.768 kHz
Internal bus clock	25.165824	25.165824 MHz; (50.331648...
Fixed frequency clock	0.016384	

Connection

Centrale Inertielle

La centrale inertielle servira à capter les mouvements en x et y. Sa programmation a été confié à *Antoine VENIER*, c'est pourquoi je lui ai demandé quelles broches je devais connecter au microcontrôleur. Ce module est donc composé de 8 pins parmi lesquelles : VCC, GND, SCL, SDA, XDA, AD0, INT. Son programme ne fonctionnant que grâce à l'I²C, il m'a conseillé de ne raccorder que les broches VCC, SDA, SCL et GND. Les autres pins resteront déconnectés mais seront tout de même utile au maintien physique du module. Elle s'alimente en 3.3V, donc aucun problème de compatibilité entre les niveaux logiques.

Capteurs Optiques

Les capteurs optiques, aussi capables de détecter les déplacement x, y du robot, sont indispensables car ils sont les seuls à pouvoir capter une rotation. Ils seront deux et le programme se servira des données de l'un comme point fixe afin de calculer la rotation du second, placé de façon excentré sur le robot. *Antoine VENIER* se chargeant également de la programmation de ce module, je me suis tourné vers lui afin de connaître les connections utiles au bon fonctionnement de son programme. Ils s'alimentent en 3.3V, donc aucun problème de compatibilité entre les niveaux logiques.

ESC

Afin de contrôler les moteurs brushless, *Jean-Gabriel MASSICOT* a réalisé des ESC maison (Electronic Speed Control). Afin de donner les ordres aux ESC, qui pourront par la suite être transmis aux moteurs, plusieurs types de sorties ont été utilisés. Par moteur, chaque ESC a besoin de 2 GPIO (sens et frein) et d'une PWM pour la vitesse. Ils s'alimentent en 3.3V, donc aucun problème de compatibilité entre les niveaux logiques.

Mais ce n'est pas tout, car un capteur angulaire a été placé au dos des moteurs afin d'asservir ces derniers en vitesse et en position. Il est donc nécessaire de récupérer les données de ces capteurs en SPI, ce qui rajoute 4 connections : MOSI, MISO, SCK, et CS.

Module Radio

Mettre en œuvre le module radio faisait partie de mon cahier des charges. Compte tenu du fait que le code récupéré (voir partie programmation) ne comportait pas d'interruption, j'ai choisi de ne pas l'implémenter malgré le fait qu'on retrouve une broche IRQ sur le module. On peut donc retrouver sur le module radio : MOSI, MISO, SCK, CS, CE (Chip enable), VCC, GND. Ce module s'alimente en 3.3V, ce qui ne donne aucun problème de compatibilité de niveau entre le microcontrôleur et le module.

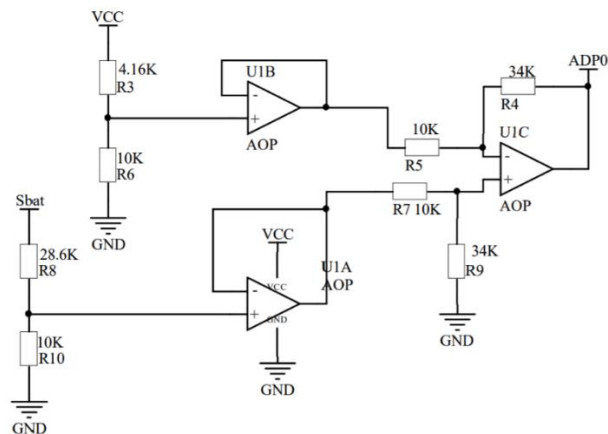
Répartition des I/O

NUMERO DE BROCHE	FONCTION UTILISE	PERSONNE CONCERNE	MODULE RACCORDE
1	MOSI2 (SPI)	Jean-Gabriel MASSICOT	Moteurs
2	SCLK2 (SPI)	Jean-Gabriel MASSICOT	Moteurs
3	N.C		
4	N.C		
5	N.C		
6	PTH4	Samuel HUET	Bargraph
7	N.C		
8	VDD	Réservé	
9	VDDAD	Réservé	
10	VREFH	Réservé	
11	VREFL	Réservé	
12	VSSAD	Réservé	
13	VSS	Réservé	
14	EXTAL	Réservé	
15	XTAL	Réservé	
16	PTH3	Samuel HUET	Bargraph
17	PTH2	Samuel HUET	Bargraph
18	PTH1	Samuel HUET	Bargraph
19	PTH0	Samuel HUET	Bargraph
20	N.C		
21	PTE5	Thomas COUTANT	LED témoin
22	N.C		
23	MISO1 (SPI)	Antoine VENIER / Samuel HUET	Capteur Optique / Module Radio
24	TPM3CH3	Jean-Gabriel MASSICOT	Moteurs
25	TPM3CH2	Jean-Gabriel MASSICOT	Moteurs
26	N.C		
27	N.C		
28	N.C		
29	N.C		
30	N.C		
31	PTJ5	Jean-Gabriel MASSICOT	Moteurs 2 (Chip Select)
32	PTJ4	Jean-Gabriel MASSICOT	Moteur 1 (Chip Select)
33	TPM3CH1	Jean-Gabriel MASSICOT	Moteur 2 (Speed)
34	TPM3CH0	Jean-Gabriel MASSICOT	Moteur 1 (Speed)
35	N.C		
36	N.C		
37	N.C		
38	N.C		
39	MOSI1 (SPI)	Antoine VENIER / Samuel HUET	Capteur Optique / Module Radio

40	SCLK1 (SPI)	Antoine VENIER / Samuel HUET	Capteur Optique / Module Radio
41	PTB1	Thomas COUTANT	I/O
42	PTB0	Thomas COUTANT	I/O
43	PTJ3	Antoine VENIER	I/O
44	PTJ2	Antoine VENIER	I/O
45	N.C		
46	ADP12	Thomas COUTANT	ADC
47	ADP9	Thomas COUTANT	ADC
48	N.C		
49	N.C		
50	VDD	Réservé	
51	VSS	Réservé	
52	N.C		
53	ADP10	Thomas COUTANT	ADC
54	PTJ1	Antoine VENIER	Capteur Optique 1 (Chip select)
55	PTJ0	Antoine VENIER	Capteur Optique 2 (Chip select)
56	N.C		
57	N.C		
58	MISO2 (SPI)	Jean-Gabriel MASSICOT	Moteurs
59	SDA (I²C)	Antoine VENIER	Central inertielle
60	SCL (I²C)	Antoine VENIER	Central inertielle
61	TPM2CH0	Thomas COUTANT	Kicker / dribleur
62	ADP0	Samuel HUET	ADC
63	PTC7	Jean-Gabriel MASSICOT	Moteur 4 (Chip select)
64	PTC6	Jean-Gabriel MASSICOT	Moteur 3 (Chip select)
65	PTG7	Jean-Gabriel MASSICOT	Moteur 4 (Brake)
66	PTG6	Jean-Gabriel MASSICOT	Moteur 3 (Brake)
67	PTG5	Jean-Gabriel MASSICOT	Moteur 2 (Brake)
68	PTG4	Jean-Gabriel MASSICOT	Moteur 1 (Brake)
69	N.C		
70	N.C		
71	PTG3	Jean-Gabriel MASSICOT	Moteur 4 (Direction)
72	PTG2	Jean-Gabriel MASSICOT	Moteur 3 (Direction)
73	PTG1	Jean-Gabriel MASSICOT	Moteur 2 (Direction)
74	PTG0	Jean-Gabriel MASSICOT	Moteur 1 (Direction)
75	N.C		
76	PTE0	Samuel HUET	Central inertielle (Chip select)
77	PTC5	Samuel HUET	Central inertielle (Chip enable)
78	TPM3CH4	Thomas COUTANT	Kicker / dribleur
79	RESET	Réservé	
80	BKGD	Réservé	

Niveau de Batterie

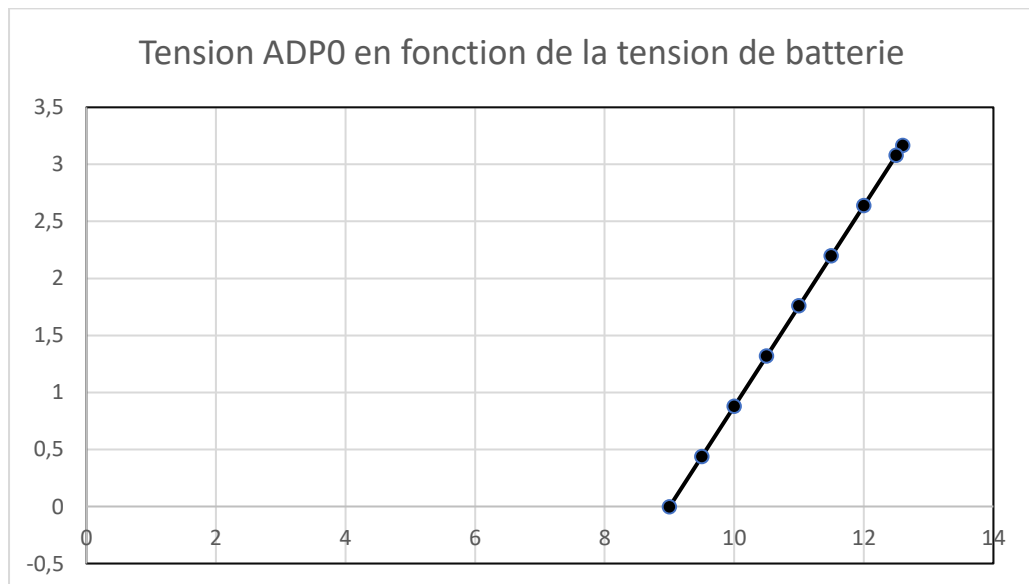
Afin de récupérer le niveau de batterie, compris entre 12.6v et 9v, sur le microcontrôleur, j'ai choisi de réaliser un montage, qui aura pour but de faire descendre cette tension entre 0 et 3.3V. Pour cela, j'ai réalisé un montage à base d'AOP.



Nous pouvons calculer quel sera la valeur d'ADP0 en fonction de la tension de batterie :

$$ADP0 = \frac{R4}{R5} \left(Sbat \frac{R10}{R8 + R10} - Vcc \frac{R6}{R3 + R6} \right)$$

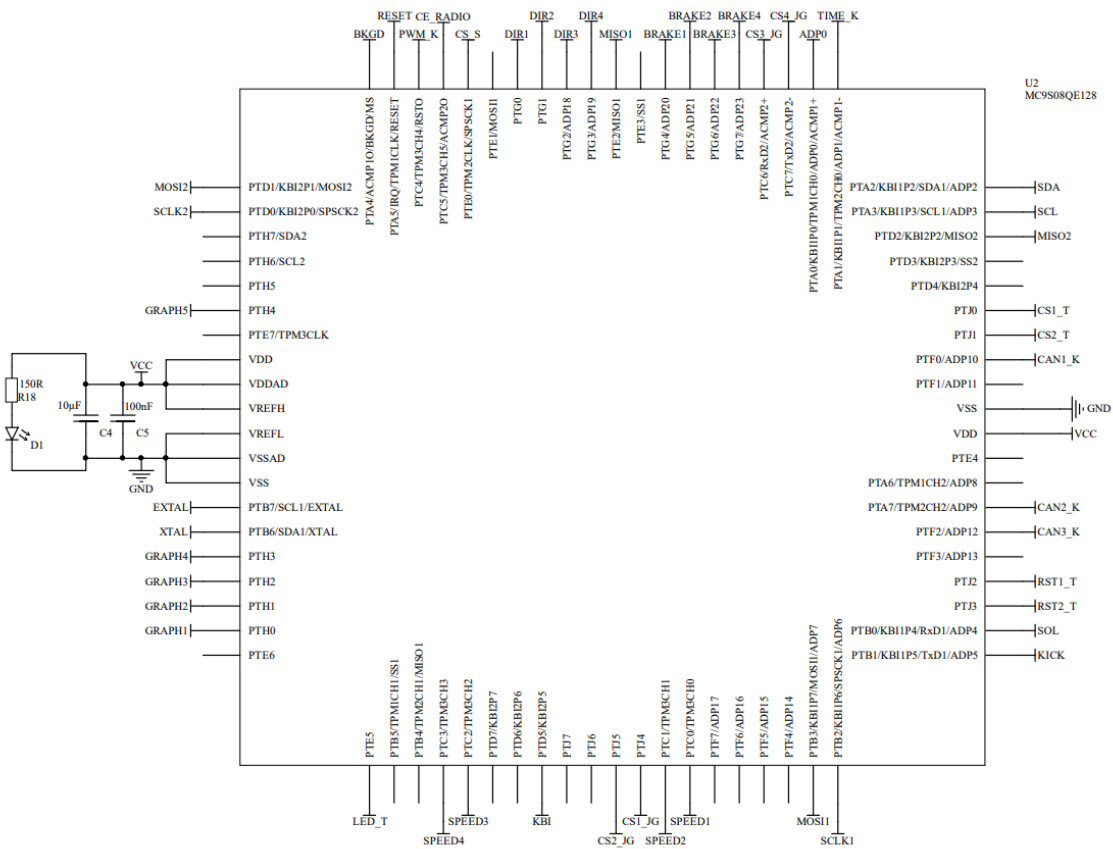
$$ADP0 = 0.88 Sbat - 7,922$$

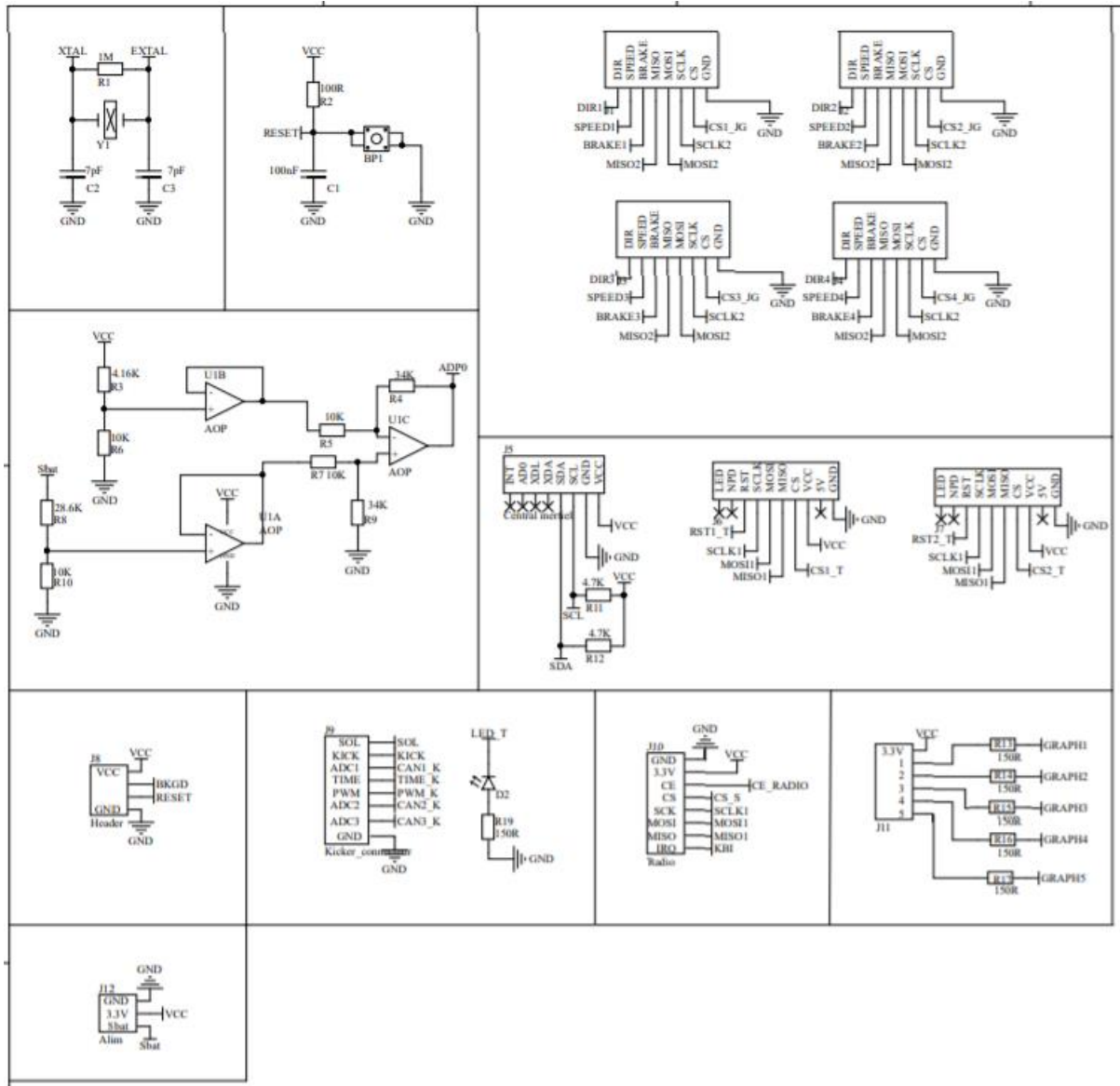


Nous pouvons voir sur le graphique ci-dessus que la tension de 3.3V ne sera pas dépassée, même lorsque la batterie atteindra sa tension maximum.

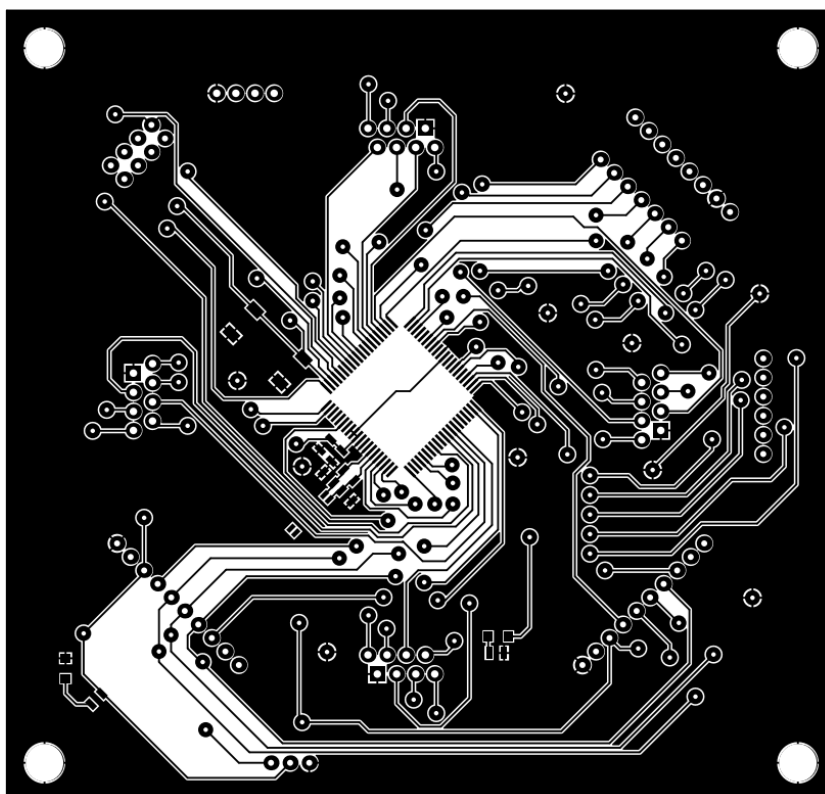
Mise en œuvre

Diagramme structurel



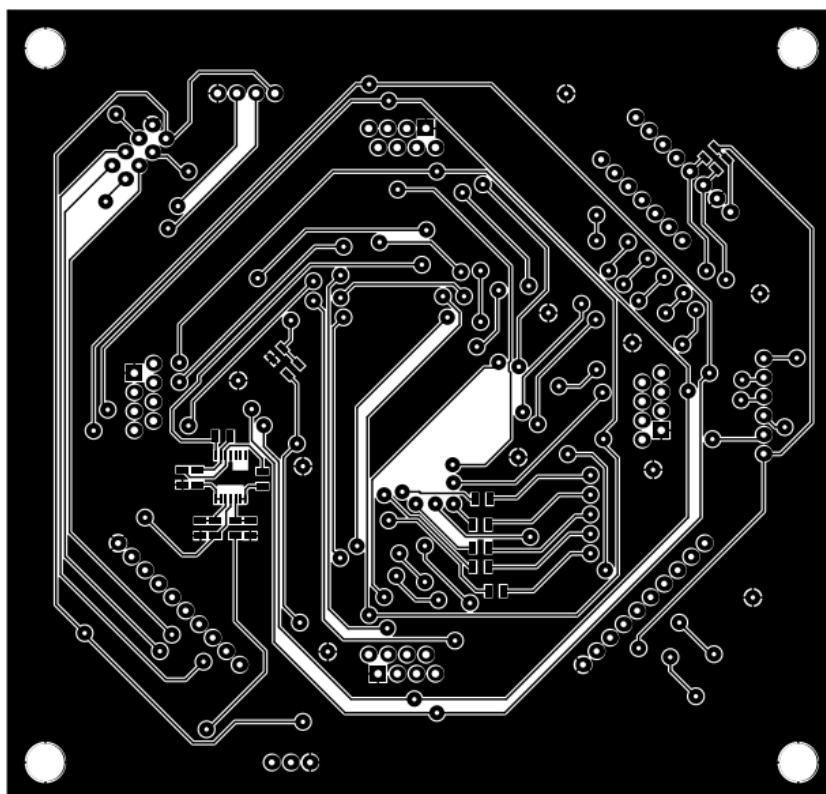


Top layer



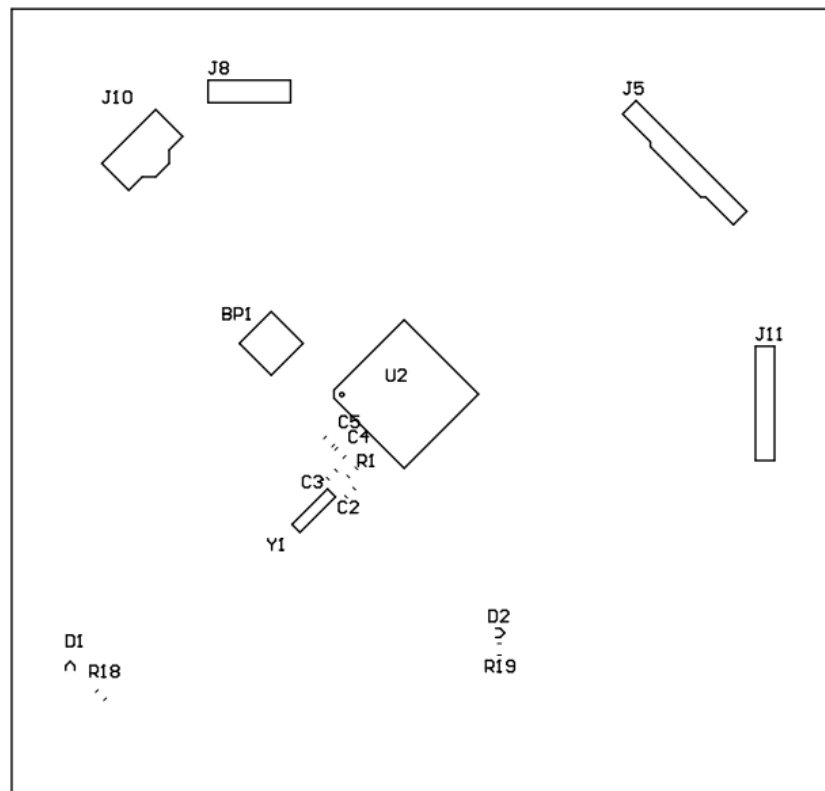
6 - Top Layer - Carte mère (1:1)

Bottom Layer



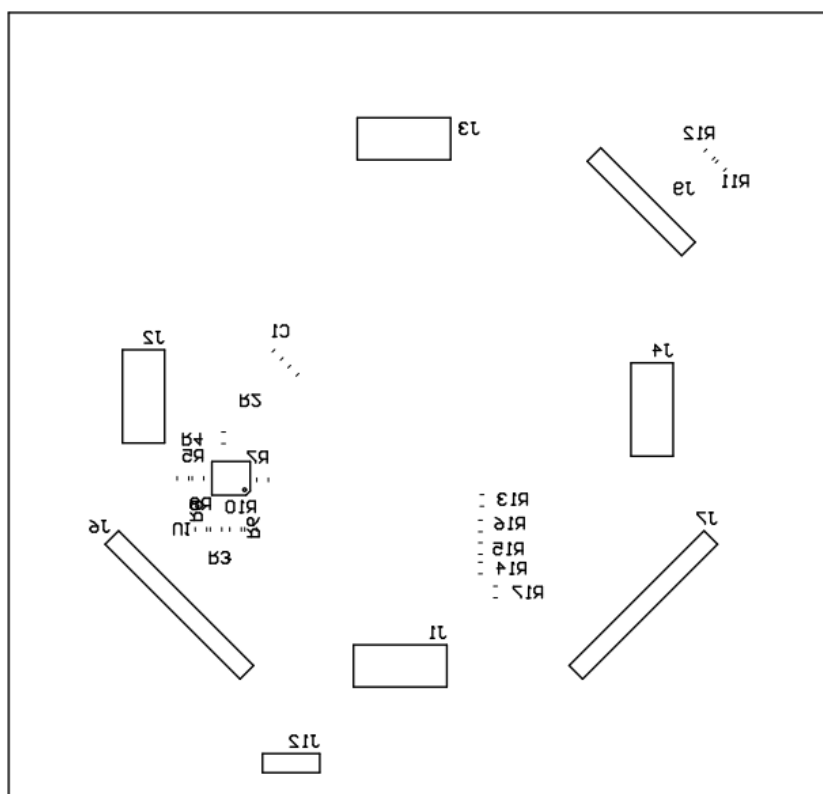
7 - Bottom Layer - Carte mère (1:1)

Top Overlay



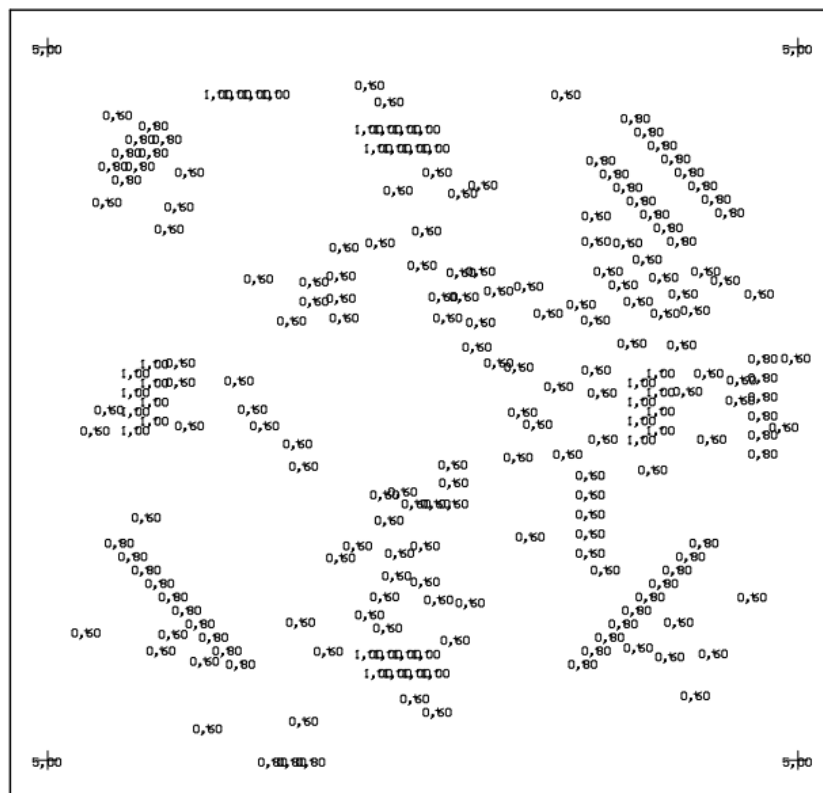
8 - Top Overlay - Carte mère (1:1)

Bottom Overlay



9 - Bottom Overlay - Carte mère (1:1)

Plan de perçage



10 - Plan de perçage - Carte mère (1:1)

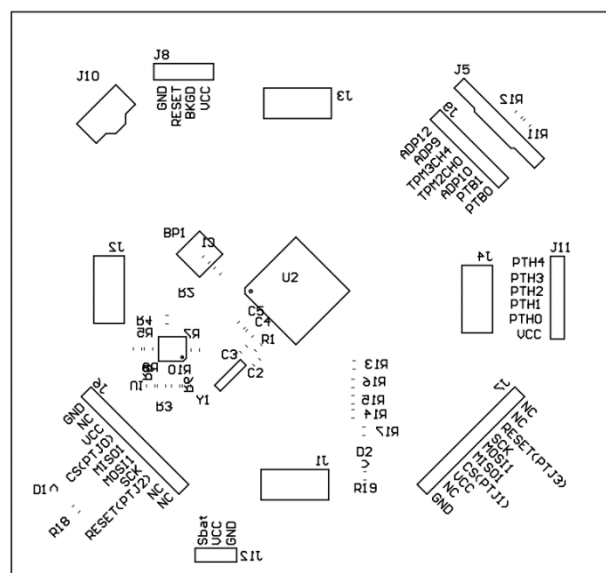
Composants

Les composants déjà commandés lors de la commande précédente ou réutilisés ne seront pas cités et comptés dans le tableau ci-dessous.

[illegible]

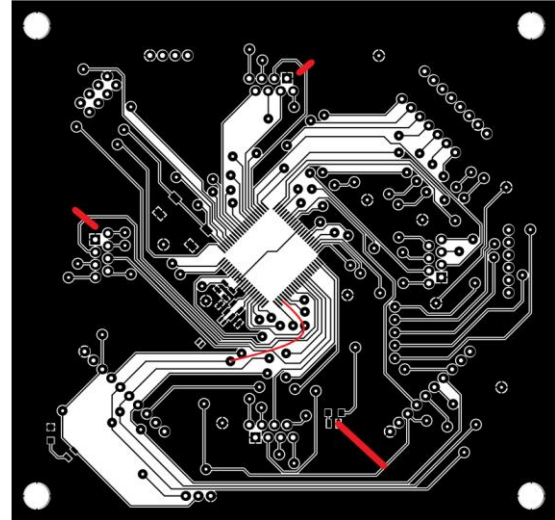
Branchements

Afin de se repérer rapidement sur les connecteurs, et en particulier sur une carte complexe comme celle-ci, voici un plan des connections externes. Le Bottom Overlay et le Top Overlay sont affichés en même temps, ainsi que les connections des headers J8, J9, J11, J7, J12 et J6. Les connecteurs J1, J2, J3 et J4 sont de type micro-matchs et les connecteurs J5 et J10 sont respectivement la centrale inertielle et le module radio, tous deux placés en direction du microcontrôleur.



Problèmes et Résolutions

Le tirage de la carte a permis la mise en évidence de plusieurs plans de masse isolés. On peut voir sur l'image ci-contre leur résolution. Mais la carte a un second défaut, en effet les broches d'un même fonction SPI, même si elles disposent de différentes configurations de sortie sur le microcontrôleur, vont par groupe. Ainsi, même si la broche du module radio était bien relié à MOSI1, elle ne l'était pas au bon. La reprise est visible sur la photo ci-contre également. Une fois ces reprises effectuées, et la piste MOSI1 originel coupée, la carte mère a semblé fonctionner parfaitement.



Programmateur

Pour des raisons de place mais aussi de sécurité (sur le projet de l'année dernière, il est arrivé plusieurs fois que le programmeur se flash mal, dépendant de l'ordinateur utilisé. Il devient alors impossible de le flasher à nouveau.), j'ai choisi de réaliser un programmeur externe. Il aura pour avantage de prendre peu de place, d'être remplaçable et réutilisable sur différents projets.

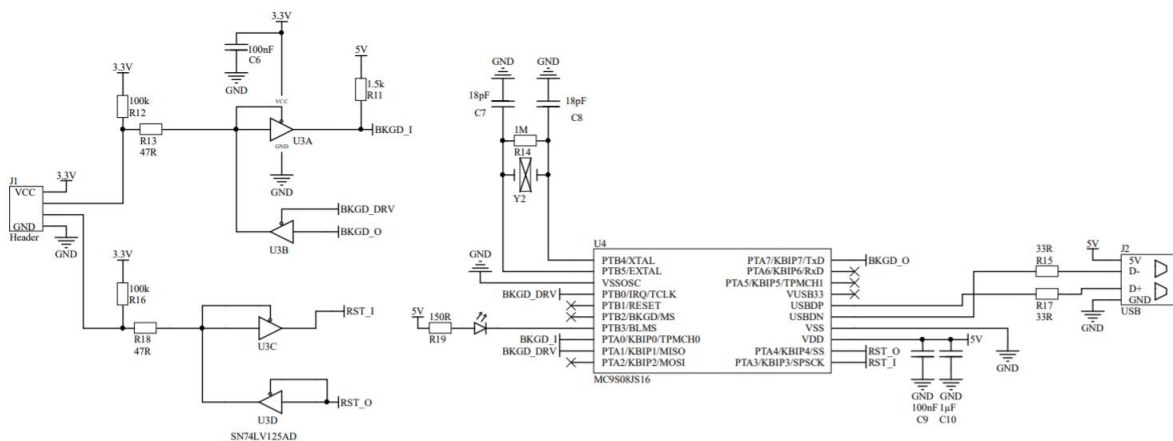
Pour cela je suis allé sur le site <http://usbdm.sourceforge.net>. Ce site regroupe les différents programmeurs et mises en œuvre pour différents microcontrôleurs, ainsi que leurs spécifications.

Notre microcontrôleur étant de la gamme HCS08, on peut voir sur ce tableau ci-dessus que seul l'USBDM_JS16CWJ nous permet le support de notre contrôleur sans surplus. Ce programmeur sera donc utilisé pour notre carte. Ce site fournit également un GitHub hébergeant schématique, routage et overlay.

Description	Current Design	Device Support					Features	
		RS08	HCS08, HCS12, CFV1	Kinetics	CFV2, CFV3, CFV4*	DSC*	Target Power	Serial Port
USBDM_IB16 (Unsupported)	No		X					
USBDM_CF_IMxxCLD	No	X	X	JTAG	X	X	X	
USBDM_SER_JS16CWJ	No		X					X
USBDM_CF_SER_JS16CWJ	No		X	JTAG	X	X		
USBDM_IMxxCLC	No	X	X				X	
USBDM_IMxxCLD	No	X	X				X	
USBDM_JS16CWJ	No		X					
USBDM_SWD_JS16CWJ	Yes		X	SWD				
USBDM_SWD_SER_JS16CWJ	Yes		X	SWD			X	X
USBDM_CF_JS16CWJ	Yes			JTAG	X	X		
USBDM_CF_IMxxCLD_V3 Also supports Tower systems.	Yes	X	X	JTAG	X	X	X	X
USBDM_OpenSDA Custom OpenSDA (FRDM-KL25Z)	Yes			SWD				X

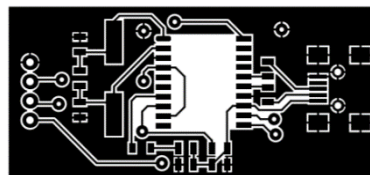
Schématique

Sur la schématique fournie par le site, on peut identifier des parties qui ne seraient pas pertinentes dans le cadre de notre projet. Par exemple, le régulateur 3.3v est à exclure car déjà présent sur la carte d'alimentation. En procédant ainsi, on peut dégrossir la carte programmeur et garder l'essentiel, à savoir le programmeur et son environnement (quartz, usb), et le buffer servant à adapter la tension entre le 5v de l'USB et le 3.3v du microcontrôleur. Voici alors la schématique qui sera utilisée.



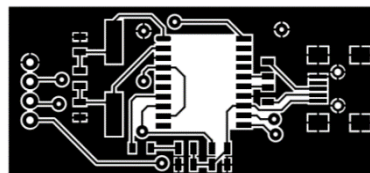
Mise en œuvre

Top Layer



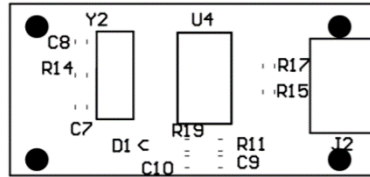
11 - Top layer - Programmeur (1:1)

Bottom Layer



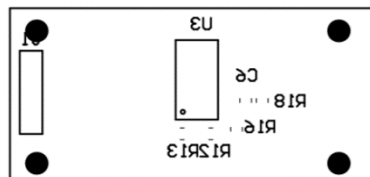
12 - Bottom Layer - Programmeur (1:1)

Top Overlay



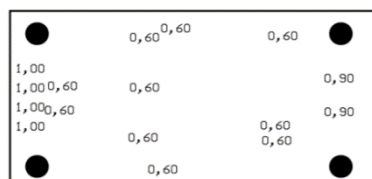
13 - Top Overlay - Programmateur (1:1)

Bottom Overlay



14 - Bottom Overlay - Programmeur (1:1)

Plan de Perçage



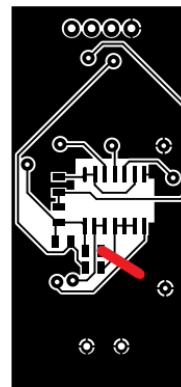
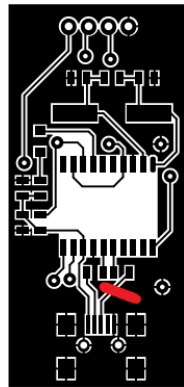
15 - Plan de perçage - Programmeur (1:1)

Composants

Nom du produit	Nom du fabricant	Référence du fabricant	Nom du fournisseur	Référence du fournisseur	Quantité	Prix unitaire H.T. (€)	Total H.T. (€)	Prix TTC (€)	Total TTC (€)
Connecteur micro USB type B	AMP - TE	1724025-1	Farnell	1724060	1	0,72 €	0,72 €	0,86 €	0,86 €
Quartz 12MHz	Abracon	ABLS-12.000MHZ-B2-T	Farnell	1652551	1	0,21 €	0,21 €	0,25 €	0,25 €
Buffer	TI	SN74LV125AD	Farnell	9592504	5	0,25 €	1,25 €	1,30 €	1,51 €
Resistance 100k	Multicomp	MCSCR08X104 JTL	Farnell	2074341	10	0,01 €	0,07 €	0,01 €	0,09 €
Resistance 47R	Panasonic	ERJ6RED47R0V	Farnell	2380581	10	0,06 €	0,61 €	0,07 €	0,73 €
Resistance 33R	Multicomp	MCMR08X330 JTL	Farnell	2073740	10	0,01 €	0,07 €	0,01 €	0,08 €
							TOTAL		3,51

Problèmes et Solutions

Hormis deux connections de masses isolées, cette carte programmeur n'a relevé aucun problème. La résolution est visible sur l'image ci-dessous.

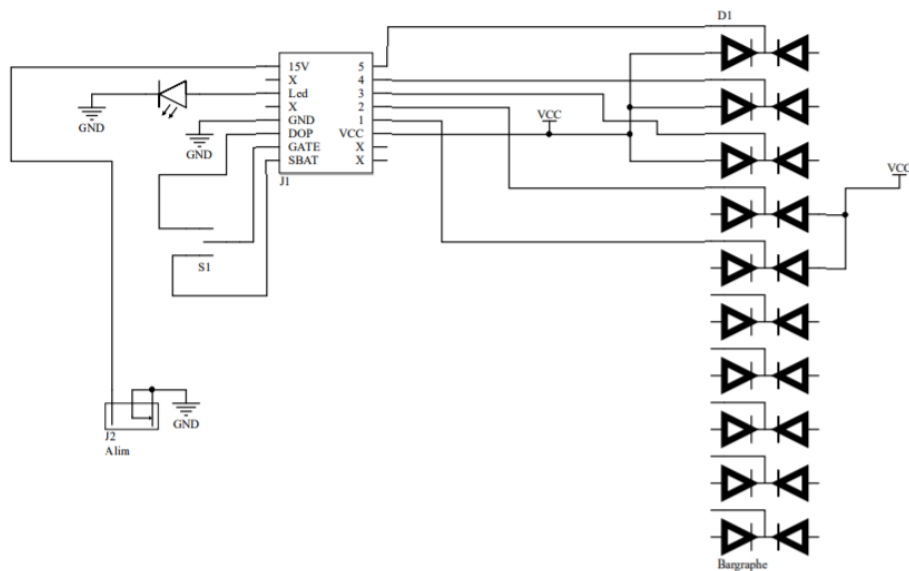


Carte d'extension

Dans le but de placer le connecteur de charge, le bargraph et les différentes IHM sur la carcasse du robot, j'ai choisi de réaliser une carte d'extension comportant :

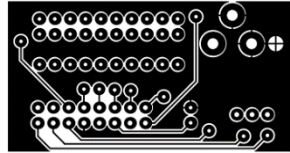
- Un connecteur de charge
- Un interrupteur On/Off
- Un témoin de charge
- Un bargraph affichant le niveau de batterie

Schématique

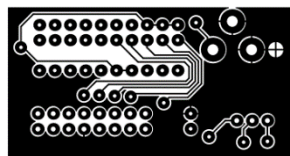


Mise en Œuvre

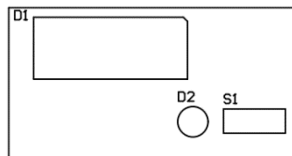
Top Layer



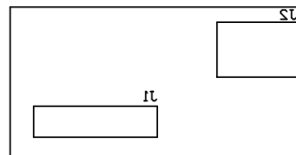
Bottom Layer



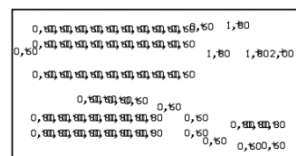
Top Overlay



Bottom Overlay



Plan de perçage



Composants

Nom du produit	Nom du fabricant	Référence du fabricant	Nom du fournisseur	Référence du fournisseur	Quantité	Prix unitaire H.T. (€)	Total H.T (€)	Prix TTC (€)	Total TTC
BarGraph	Kingbright	BC-10EGWA	Farnell	2290326	1	2,75 €	2,75 €	3,30 €	3,30 €
Interrupteur	NKK	SS12SDP2	Farnell	1524325	1	1,36 €	1,36 €	1,63 €	1,63 €
Connecteur d'alimentation	Cliff	BC681478	Farnell	2450496	1	1,97 €	1,97 €	2,36 €	2,36 €
								TOTAL	7,29 €

Programme

Afin de travailler en groupe sur le programme, nous avons utilisé un gestionnaire de version : Git. Cet outil permet de créer différentes branches au sein d'un même projet. Ainsi, chaque membre du groupe a pu travailler son programme sans devoir continuellement ajouter des modifications au projet final. De plus, ce programme permet de combiner tous les fichiers .c et .h au sein d'un programme final qui sera à porter sur le robot.

Batterie

L'image du niveau de batterie, ramené sur une plage de tension compris entre 0 et 3.3v, est liée sur la broche ADP0 du microcontrôleur. J'ai donc choisi de concevoir une fonction qui, avec le numéro du Channel en paramètre, retournera la valeur de l'ADC.

Etude de fonction ADC

ADCSC1

COCO		ADC0	ADCRH				
0		0	0	0	0	0	0

COCO (flag) : A 1, la conversion est complétée. RàZ par écriture de ADCSC1 ou lecture de ADCRL.

ADC0 : A 1, conversion continue après activation de la fonction.

ADCH : Sélection du channel, à 00000, channel 0 sélectionné.

Les bits non utilisés sont laissés dans leur état par défaut.

ADCRL

ADCRL							

Contient la valeur convertie.

ADCCFG

ADLPC	ADIV		ADLSMP	MODE		ADICLK	
0	0	0	0	0	0	0	0

ADLPC : A 0, configuration en mode rapide.

ADIV : A 00, pas de division de la fréquence de bus pour la fonction ADC.

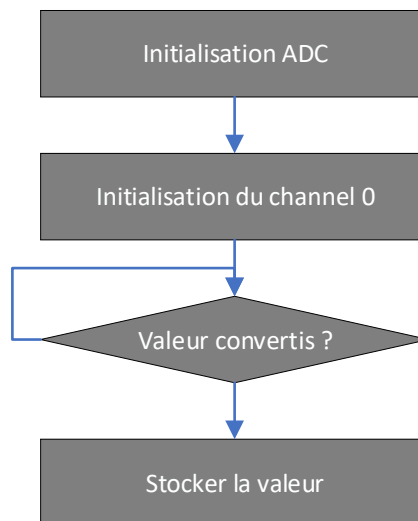
ADLSMP : A 0, conversion rapide.

MODE : A 00, conversion sur 8 bits.

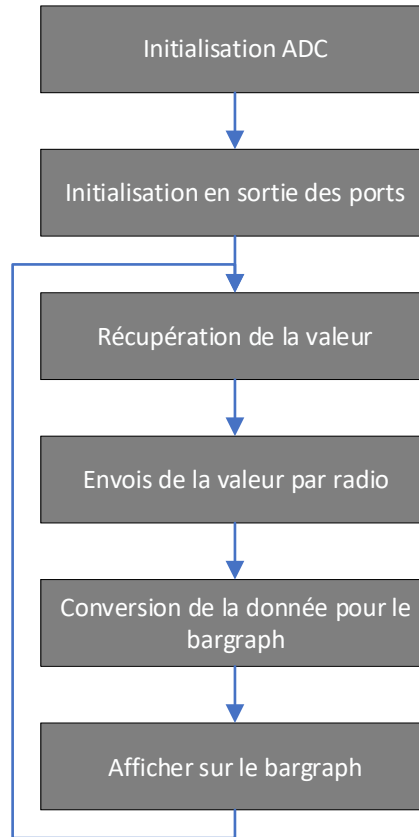
ADICLK : A 00, sélection de la fréquence de bus pour la fonction ADC.

Les registres non étudiés seront laissés dans leur état par défaut.

Organigramme



Mise en œuvre



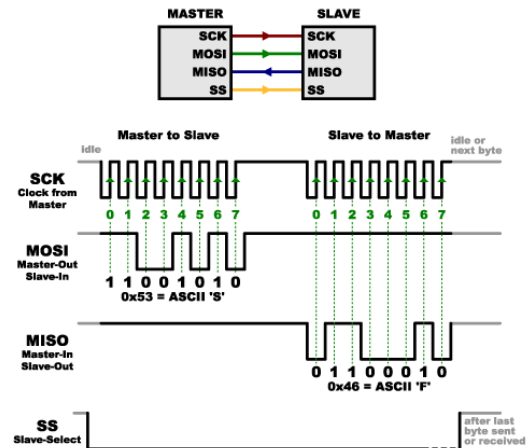
Module Radio

Pour éviter une longue étude du module radio, j'ai utilisé un code récupéré sur github, spécifiquement réalisé pour faire fonctionner le nRF24l01+ qui servira de couche d'abstraction logiciel. Le microcontrôleur utilisé par l'auteur étant différent, j'ai simplement eu à adapter la fonction `spi_transfer()` en utilisant les fonctions du microcontrôleur et modifier les fonctions permettant de gérer le Chip Select et le Chip Enable.

Il est important de rappeler que le module radio est câblé sur la fonction SPI1, que sa fréquence d'horloge maximum est de 8MHz mais que la fréquence maximum des capteurs optiques est de 2 MHz. La fréquence de bus du microcontrôleur quant à elle est de 25MHz.

Etude de fonction SPI

La fonction SPI étant basée sur le concept de registre à décalage, une procédure est à suivre si l'on veut établir une communication complète entre deux systèmes. Dans un premier lieu, en supposant qu'on veuille questionner le registre 0x53, nous devons envoyer ce registre via le bus de communication. Le capteur nous renverra une valeur, particulière ou non, mais en aucun cas en lien avec notre questionnement. Pour obtenir notre réponse, une deuxième interrogation doit être faite. C'est alors que notre capteur nous renverra la donnée souhaitée. Cela vient du fait que la transmission d'information par décalage de registre ne laisse pas suffisamment de temps au capteur pour récupérer la valeur souhaitée.



SPI1C1

	SPE		MSTR	CPOL	CPHA	SSOE	LSBFE
	1		1	0	0	0	0

SPE : A 1, active la fonction SPI.

MSTR : A 1, configure le microcontrôleur en tant que maître.

CPOL : A 0, configure le repos de l'horloge à l'état bas.

CPHA : A 0, configure l'échantillonnage sur le front du milieu.

SSOE : A 0, configure le chip select en mode manuel.

LSBFE : A 0, préconfigure la SPI en MSB first.

Les bits non utilisés sont laissés dans leur état par défaut.

SPI1C2

			MODFEN				
			0				

MODFEN : A 0, configure le chip select en mode manuel.

Les bits non utilisés sont laissés dans leur état par défaut.

SPI1BR

					SPR		
					0	1	1

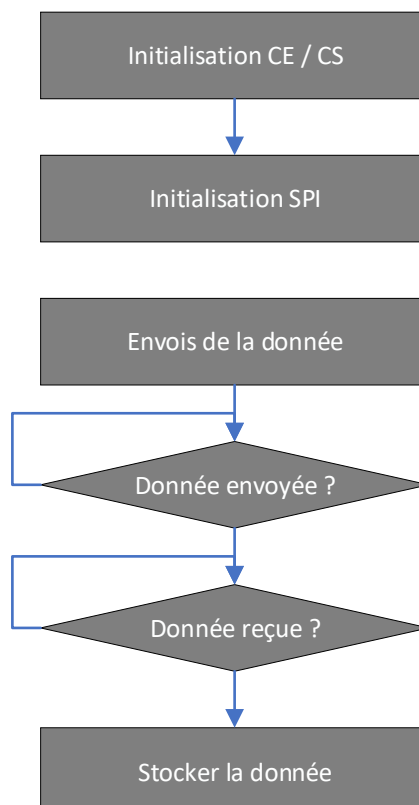
SPR : A 001, divise l'horloge interne par 16, atteignant une fréquence d'horloge de 1.5 MHz.

Les bits non utilisés sont laissés dans leur état par défaut.

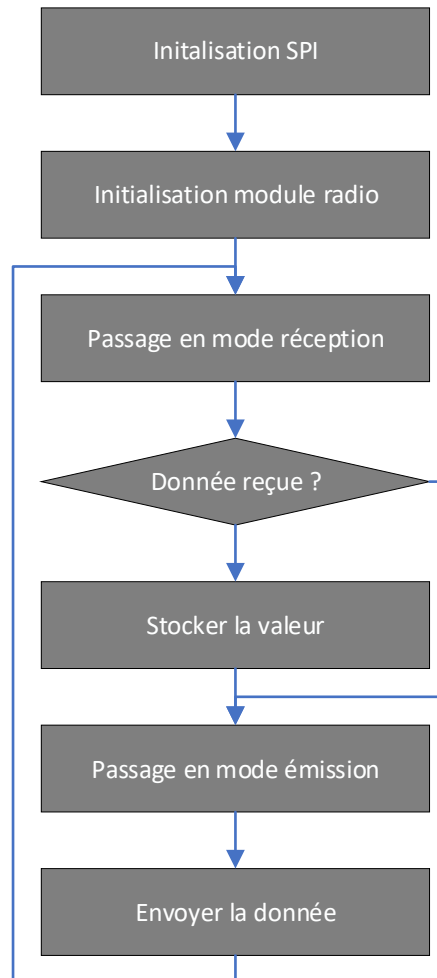
Les registres non étudiés seront laissés dans leur état par défaut.

Organigramme

L'organigramme ci-dessous montre la façon de procéder pour réaliser une seule émission, et une seule réception. Il faudra alors appeler deux fois la fonction pour récupérer une valeur.



Mise en œuvre



I.H.M

L'IHM du robot se fera sur le PC, via un bus de communication (UART). Pour ce faire, j'ai utilisé les cartes d'évaluations présentes à l'école, qui comportent un transceiver UART/USB. Il est donc nécessaire de créer une fonction UART à 9600 bauds afin d'établir la communication entre les deux appareils.

Etude de fonction UART

SCIBDH/L

			SBR				
			0	0	0	0	0
SBR							
0	0	0	1	1	0	1	1

SBR : A 11011, $BaudRate = \frac{F_{bus}}{16 \cdot SBR} = 9709.04 \text{ bits/s}$, donc une erreur de 1.12%.

Les bits non utilisés sont laissés dans leur état par défaut.

SCIC2

				TE	RE		
				1	1		

TE : A 1, transmission activée.

RE : A 1, réception activée.

Les bits non utilisés sont laissés dans leur état par défaut.

SCIS1

TDRE		RDRF					

TDRE (flag) : A 1, le registre d'émission est vide. RàZ par lecture de SCIS1 lorsque TDRE est à 1, suivis d'une écriture de SCID.

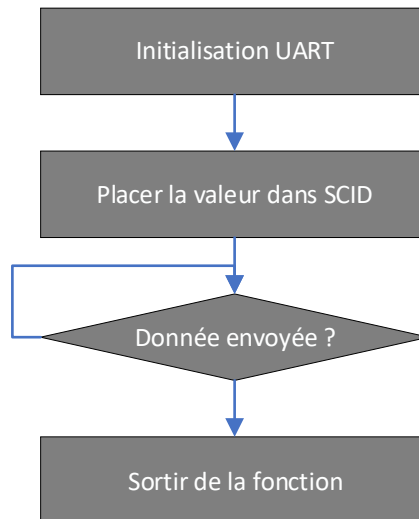
RDRF (flag) : A 1, le registre de réception est plein. RàZ par lecture de SCIS1 lorsque RDRF est à 1, suivis d'une lecture de SCID.

Les bits non utilisés sont laissés dans leur état par défaut.

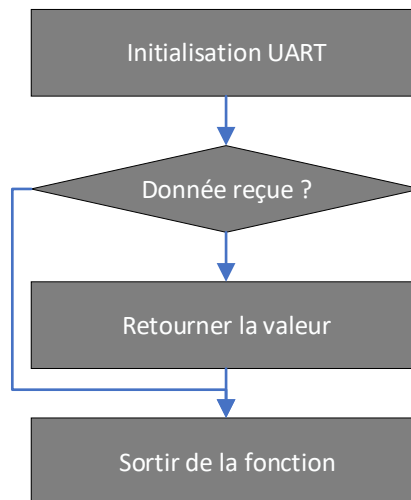
Les registres non étudiés seront laissés dans leur état par défaut.

Organigrammes

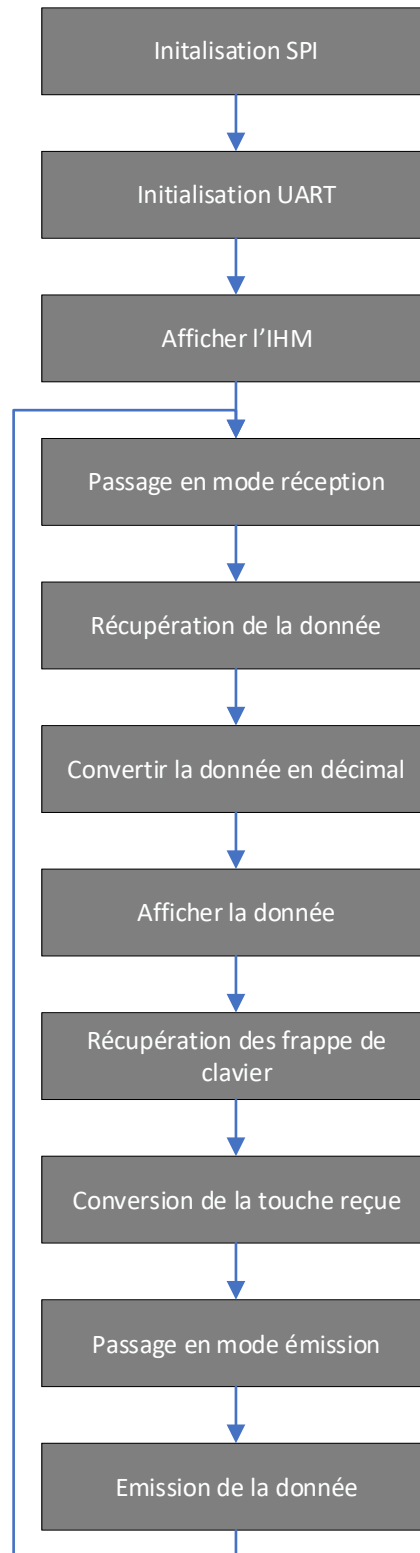
Emission



Réception



Mise en œuvre



Conclusion

Cette expérience m'a beaucoup apportée, autant en termes de conception technique que de travail d'équipe. En effet, étant dans un petit groupe de 4 personnes, les informations ont pu voyager rapidement, ce qui permet un travail d'équipe plus efficace. De plus, tous les membres du groupe ont eu pour objectif majeur la mise en place d'un robot fonctionnel, ce qui a beaucoup incité l'entraide selon les compétences privilégiées de chacun. Cependant le robot n'a, au jour où je rédige ce rapport, pas pu fonctionner. Les parties étant toutes importantes en termes de quantité de travail, j'ai personnellement manqué de temps notamment en termes de communication radio duplex et surtout de l'intégration de toutes les parties.

Cependant il nous a été laissé comme choix de pouvoir continuer ce projet sur les années à venir, et ainsi de développer une IA pouvant contrôler chaque robot, mais également de développer à nouveau chacune de nos parties avec une expérience nouvellement acquise afin de respecter chacune des règles de la Robocup et ainsi y participer. Je tiens à remercier *Steve NGUYEN* ainsi que *Aloïs MICLO* et *Pierre AUBRY* pour leur aide et leurs conseils tout au long de ce projet.

Si ce projet était à refaire, je réfléchirais davantage aux placements des composants et afin de faire tenir les deux cartes principales au sein d'une seule sur toute la largeur du robot, économisant ainsi beaucoup de place et évitant les connecteurs. Concernant ces derniers, bien que les headers utilisés restent très pratiques dans le cadre d'un prototype, ils sont sujet à énormément d'erreurs de branchements. Des nappes avec détrompeurs sont alors à préconiser.

Annexes

Code

Pour permettre une lisibilité maximale des codes sources, à la fois celui interne au robot mais également le code permettant d'afficher les informations du robot, voici un QR code, à flasher avec une application dédiée depuis votre smartphone. J'ai également joint sur ce lien les différents rapports de chacune des personnes composant mon groupe, ainsi que le dossier qui nous a été remis au début du projet comprenant le cahier des charges et les différentes problématiques. Je tiens à préciser que ce QR code est également cliquable afin de faciliter la lecture sur un support numérique.



Lexique

ADC : Analog to Digital Converter, aussi connu sous le nom de CAN (Convertisseur analogique numérique) en français.

BMS : Battery Management System. Contrôle les cellules d'une batterie afin de ne pas dépasser le seuil de charge ou de décharge.

CAO : Conception Assistée par Ordinateur. Dans notre cas, design hardware. Mais cette abréviation est également applicable à la conception 3D et à bien d'autres domaines.

ESC : Electronic speed control. Module permettant de contrôler les moteurs brushless.

ESTEI : École Supérieure des Technologies Électronique, Informatique.

GPIO : General Purpose Input/Output. Broche d'entrée/sortie logiciel.

IDE : Integrated Development Environment. Aussi connu sous le l'abréviation EDI (Environnement de Développement Intégré) en français.

IHM : Interface Homme/Machine. Regroupe tous les systèmes permettant de connaître l'état d'une machine.

PC : Inter-Integrated Circuit. Protocole de communication synchrone bifilaire.

LiPo : Lithium-Polymère. Technologie de batterie.

MOSFET : Metal Oxide Semiconductor Field Effect Transistor. Transistor à effet de champ.

PLL : Phase-locked loop.

PWM : Pulse Width Modulation. Modulation de la largeur d'impulsion.

SPI : Serial Peripheral Interface. Bus de communication synchrone quadri filaire.

UART : Universal Asynchronous Receiver Transmitter. Bus de communication asynchrone bifilaire.

Sources

Programmateurs	http://usbdm.sourceforge.net/USBDM_V4.12/html/index.html
GitHub programmation HCS08	https://github.com/podonoghue/usbdm-hcs08/tree/master/Hardware
Couche d'abstraction logiciel nRF24L01+	https://github.com/kehribar/nrf24L01_plus
Informations sur les différentes batteries	https://www.ni-cd.net/