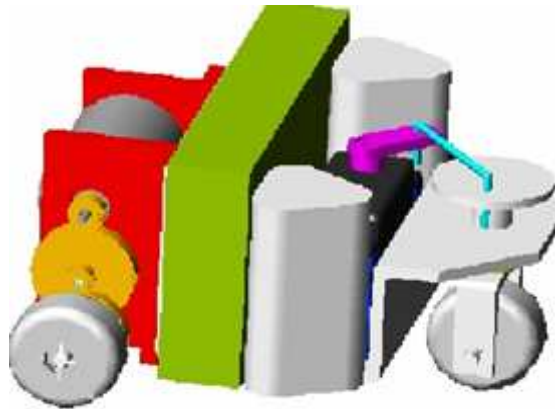


BUREAU D'ETUDES

GE II 1^{ere} année

Enseignants BE / GE1



Projet RAFIL

Robot Autonome FILoguidé

Module n°ERGE2



I. Introduction

1. Objectif
2. Mise en situation

II. Cahier des charges

1. Cycle de fonctionnement
2. Description du robot
3. Caractéristique opératoires et technologiques

III. Spécifications du système

1. Analyse de l'environnement
2. Spécifications Fonctionnelles

IV. Conception fonctionnelle

1. Délimitation du systeme
2. Première décomposition fonctionnelle
3. Raffinement de la fonction "Contrôle de la propulsion"
4. Raffinement de la fonction "Contrôle de la direction"

V. Étude Fonctionnelle

VI. Étude du schéma structurel de la carte puissance

1. Horloge
2. Interface moteur
3. Détection surintensité
4. Détection batterie faible
5. Régulation de tension 5V

VII. Etude des fonctions programmées

1. Sous-fonctions liées à la fonction « Gestion de la direction »
2. Sous-fonction liée à la fonction « Gestion de la propulsion »

VIII. Création de la carte électronique

IX. Tests, mesure et validation

X. Conclusion

XI. Annexes

1. Schéma structurel de la carte fille n°2 (Commande)
2. Schéma structurel de la carte fille n°3 (Infrarouge)
3. Schéma structurel de la carte mère (connecteurs d'entrées/sorties)
4. Implantation des différentes fonctions / Echanges des informations entre les cartes
5. Organisation de mon étude

XII. Webographie

I. Introduction

1. Objectif

Dans le premier projet les objectifs recherchés étaient les suivants :

- méthode de travail,
- utiliser les outils de CAO
- maîtriser les techniques de câblage
- effectuer les essais et test d'une carte imprimée.

Dans ce projet les étudiants devront en plus être capable de :

- rechercher des documentations techniques,
- d'effectuer des choix technologiques,
- de concevoir et valider une partie commande en VHDL.

Notre robot devra au final être capable de suivre un ligne droite sur une ligne de 1 mètre, puis d'être capable de suivre une série de virage sur 1 mètre et d'être capable d'effectuer de tour de circuit à la vitesse la plus rapide possible, enfin il devra être capable de signaler la surtension moteur a l'aide d'une LED.

2. Mise en situation

Le travail sera effectué en binôme.

L'organisation de l'étude s'effectue de la manière suivante :

	Activité	Travail à prévoir	sem
Présentation	Etude fonctionnelle		1
Fonctions	Calculs et <i>simulation</i>	Réalisation du schéma structurel	1-2
Kicad	Capture		3
	PCB	Nomenclature	3-4
Simulation VHDL		Analyse du fichier VHDL	3-4
Documentation	Rédaction	Rendre les typons avant les vacances	4
Fabrication	Perçage + câblage Programmation		5
Réalisation	câblage Programmation	Test des Robots	6
Tests	Mesures	Fiche d'essais	7
Essais sur piste	Test sur robot ou maquette	Fiche de test	6-8
Dossier	Analyse et dossier	Rendre le dossier complet	9

Ce tableau est prévisionnel. Les temps proposés sont indicatifs mais il est conseillé de s'en rapprocher pour mener le projet à son terme. Vous remplirez et donnerez le planning de votre étude sur les 9 séances (feuilles jointes à la fin de ce dossier).

La contrainte la plus importante reste la fabrication des cartes imprimées.

Documents utilisés :

- Notices techniques et notes d'applications disponibles sur :

Commun GEII\ge1\rafil\index.htm et www.geii.free.fr/rafil

II. Cahier des charges

Un véhicule ou robot filoguidé permet d'effectuer, sans intervention humaine, des travaux de manutentions dans une unité de production. Dans le cas présent, il s'agit d'effectuer un déplacement sur une surface plane après un ordre de commande manuelle, issu d'une télécommande infrarouge ou d'un poussoir.

Le robot suit principalement une ligne peinte (blanche sur fond noir) qui peut être droite ou avec des rayons de courbures élevés.

En l'absence de piste, le robot doit se déplacer en ligne droite avec la possibilité de détecter les obstacles. Dans ce cas, il doit s'arrêter.

1. Cycle de fonctionnement

Au repos, le robot est à l'arrêt et positionné sur une piste.

Il doit être capable de suivre la piste de bout en bout ou être au moins autonome pendant un temps supérieur à 10 minutes.

Pour un cycle de fonctionnement, on impose la séquence suivante :

- le robot est à l'arrêt.
- l'utilisateur active la mise en route au moyen d'une liaison infrarouge ou d'un poussoir.
- Le robot suit la piste et s'arrête sur ordre de l'utilisateur.

2. Description du robot

Le robot possède 3 roues. La roue avant est directrice et les deux roues arrières sont motrices.

Ces dernières sont entraînées en rotation au moyen d'un moteur électrique et d'un système mécanique de réduction de la vitesse par roues dentées.

La partie mécanique proposée a été voulue la plus simple possible. En effet, le châssis supporte uniquement le moto réducteur (ensemble moteur + réducteur mécanique) ainsi que le servomoteur (d'aéromodélisme) utilisé pour piloter la roue et fixer la direction du robot.

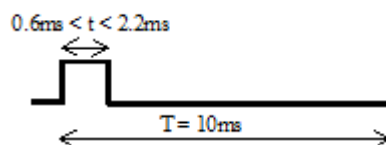
La commande et l'alimentation de l'ensemble est assurée par une carte-commande fixé sur le dessus du châssis.

3. Caractéristique opératoires et technologiques

Le servomoteur est commandé par un signal à Modulation de Largeur d'Impulsions (M.L.I) de période 10 ms.

L'angle de direction varie de -90° à $+90^\circ$ pour une largeur d'impulsion comprise entre 0.6 ms et 2.2 ms.

Le moteur d'entraînement est alimenté par un convertisseur continu-continu (pont en H). Le suivi de trajectoire est assuré par 3 capteurs placés en arrière de la roue directrice.



III. Spécifications du système

Il s'agit de réaliser tout ou partie de la carte de commande qui permet le déplacement autonome du robot.

1. Analyse de l'environnement

a) Les entités :

Les entités en relation avec le système à concevoir sont :

- l'utilisateur ou l'opérateur
- la partie mécanique du robot ou chariot qui inclut les roues motrices et le moteur d'alimentation, ainsi que la roue directrice et son servomoteur.
- Les capteurs de trajectoires.

b) Entrées, Sorties et Comportement des entités :

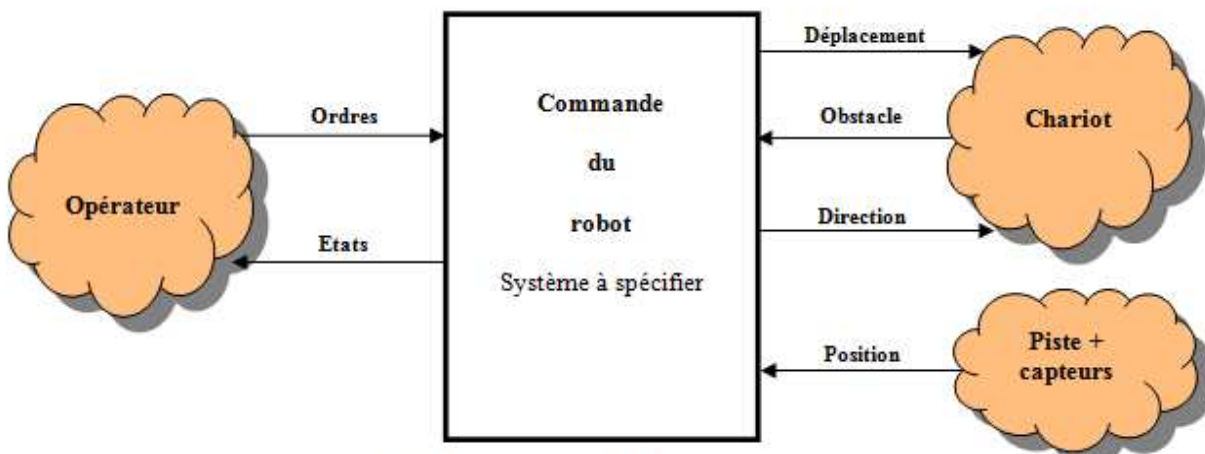


Figure 3-1 : Le système et les liens avec son environnement

L'opérateur :

- Il procède à la mise sous tension du système par une action manuelle Marche/Arrêt (M/A). Dans cette phase, le moteur de déplacement sera hors-tension, laissant le temps au robot d'orienter correctement son système de direction par rapport à la piste.
- Il procède également à la mise en route de la motorisation, une fois le robot placé sur la piste et orienté.
- Il est informé de l'état du système visuellement.

La partie opérative :

- Elle assure le déplacement et l'arrêt du robot à partir des roues motrices pilotées par un moteur continu au travers du convertisseur continu-continu. Le robot peut se déplacer en avant ou en arrière (Propulsion).
- Elle assure la direction du robot grâce à la roue directrice avant au travers de la commande du servomoteur (Direction) seulement lors du déplacement en marche avant.

La piste (capteurs de position relative à la piste) :

–Les capteurs fournissent une information logique en fonction de leur position relative par rapport aux bords de la piste.

2. Spécifications Fonctionnelles

Quelles sont les contraintes imposées au système? Performances attendues (électriques, temporelles) ou conditions d'utilisation.

On ne s'intéresse ici qu'à la carte de commande du robot dont la fonction principale est :

- Assurer en automatique le déplacement du chariot suivant une piste balisée.

a) Spécifications Opératoires :

Le déplacement doit se faire en détectant d'éventuels obstacles. La gestion des obstacles est assurée en interne par une détection de surintensité sur l'alimentation des roues motrices.

La suivi de trajectoire est assuré par 3 capteurs placés en arrière de la roue directrice.

La mise en route du moteur de déplacement se fait par une commande à distance (télécommande).

b) Spécifications Technologiques :

Le moteur d'entraînement est alimenté par un convertisseur continu-continu (pont en H).

La carte de commande intègre la gestion de la puissance des roues.

Le servomoteur est commandé par un signal à Modulation de Largeur d'Impulsions (M.L.I) de période 10 ms. L'angle de direction varie de -90° à $+90^\circ$ pour une largeur d'impulsion comprise entre 0.6 ms et 2.2 ms.

La gestion du cycle sera supportée par un composant programmable in-situ.

Des leds permettront de visualiser les états des commandes générées vers la commande des roues motrices et de la roue directrice.

L'alimentation du robot est faite en autonomie par batterie (6 éléments de 1.2Volt)

IV. Conception fonctionnelle

1. Délimitation du système

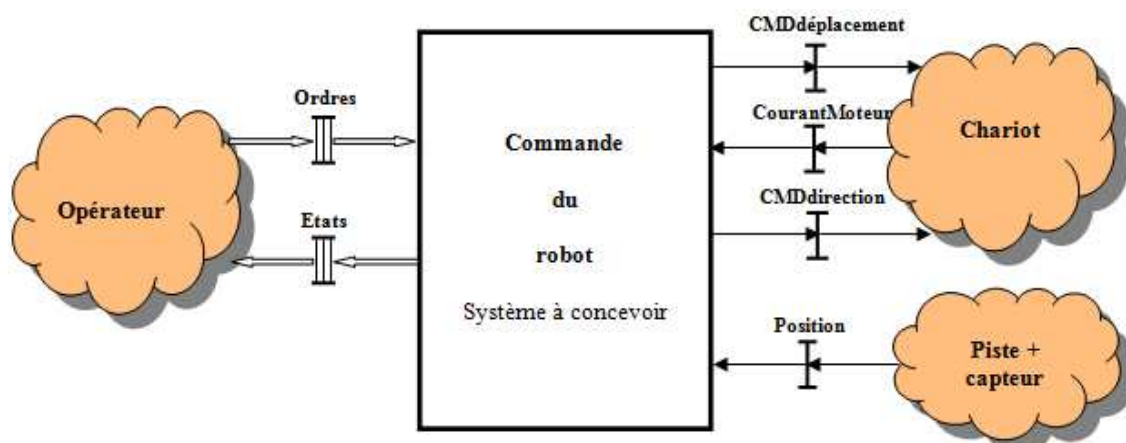


Figure 4.1 – Délimitation du système pour la commande du robot

2. Première décomposition fonctionnelle

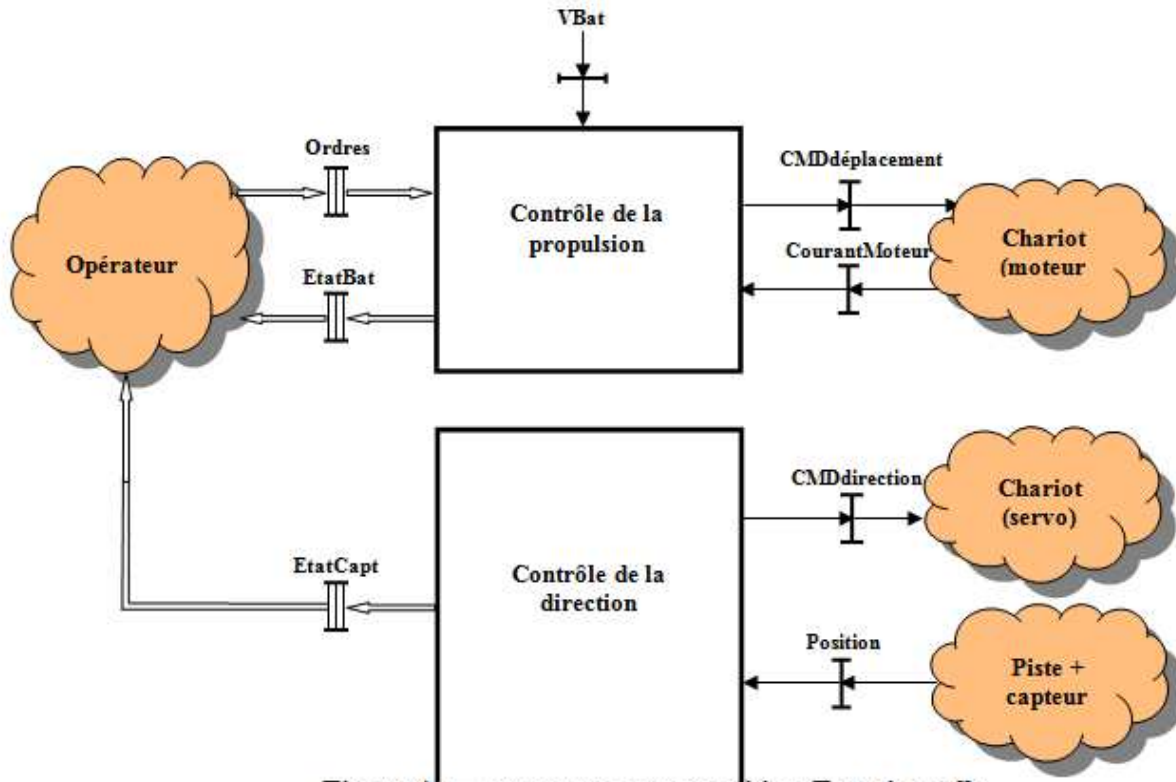


Figure 4.2 – Première Décomposition Fonctionnelle

Fonction : Contrôle de la propulsion

La fonction "Contrôle de la propulsion" permet de gérer la propulsion en marche avant ou arrière du robot en fonction des ordres de l'utilisateur.

Entrées : **Ordres** : Commandes marche avant ou arrière (inverseur ou commande Infrarouge)

CourantMoteur : Valeur analogique [0..Imax] image du courant moteur.

VBat : tension batterie

Sorties : **EtatBat** : visualisation de l'état de la batterie.

CMDdéplacement : la commande de déplacement ordonne la mise en marche avant ou arrière du chariot (à travers le moteur d'entraînement).

Fonction : Contrôle de la direction

La fonction "Contrôle de la direction" permet de gérer la direction du robot en fonction de la position sur la piste.

Entrées : **Position[2 .. 0]** : Valeur numérique codée sur 3 bits correspondant aux états logiques des 3 capteurs de position cg, cm et cd (capteurs **g**auche, **m**ilieu et **d**roit). Cette information donne la position du robot par rapport à la piste.

Sorties : **EtatCapt** : visualisation de la position du robot par rapport à la piste.

CMDdirection : la commande de direction donne une consigne au servomoteur avec un signal MLI (ou PWM). La relation entre la largeur d'impulsion et la direction est linéaire.

3. Raffinement de la fonction "Contrôle de la propulsion"

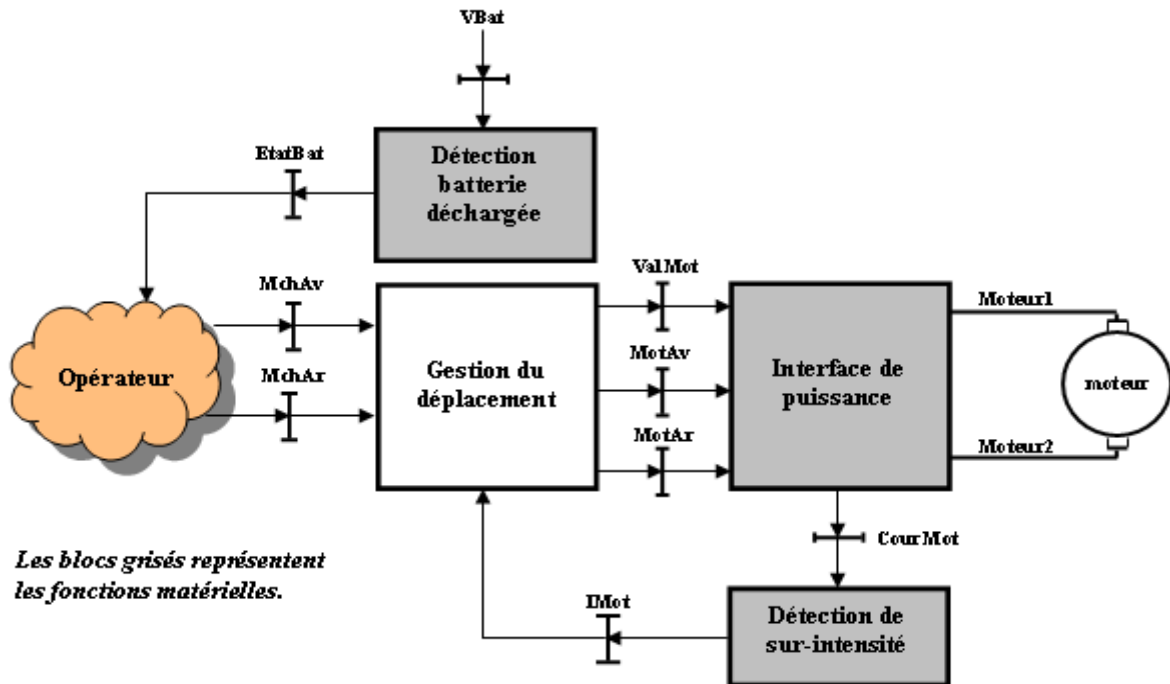


Figure 4.3

▪ Fonction : Détection de surintensité.

Entrée : **CourMot** : Valeur analogique [0..Imax] image du courant moteur.

Sortie : **Imot** : Information booléenne indiquant que le courant dans le moteur (CourMot) est supérieur à un seuil donné.

▪ Fonction : Détection Batterie déchargée

Entrée : **VBat**, : tension aux bornes de la batterie

Sortie : **EtatBat** : indication optique (LED) signalant à l'opérateur que la tension batterie VBat est inférieure au seuil critique de 6V.

▪ Fonction : Interface de puissance (Pont en H)

Elle assure l'alimentation du moteur pour les 2 sens de rotation avec la possibilité de faire varier la vitesse de rotation du moteur. Le choix de l'interface de puissance découle des contraintes technologiques (type de moteur et d'énergie disponible).

▪ Fonction : Gestion du déplacement

Entrées : **MchAr** et **MchAv** (inverseur ou télécommande Infrarouge): commandes actives au niveau bas non simultanées.

Imot : information booléenne de surintensité du moteur.

Sorties : **ValMot** : commande du moteur : un signal MLI (PWM) peut être utilisé pour diminuer la vitesse de rotation du moteur.

MotAv et **MotAr** : rotation du moteur pour un déplacement en marche avant ou arrière.

Les 2 commandes ne doivent pas être actives simultanément.

4. Raffinement de la fonction "Contrôle de la direction"

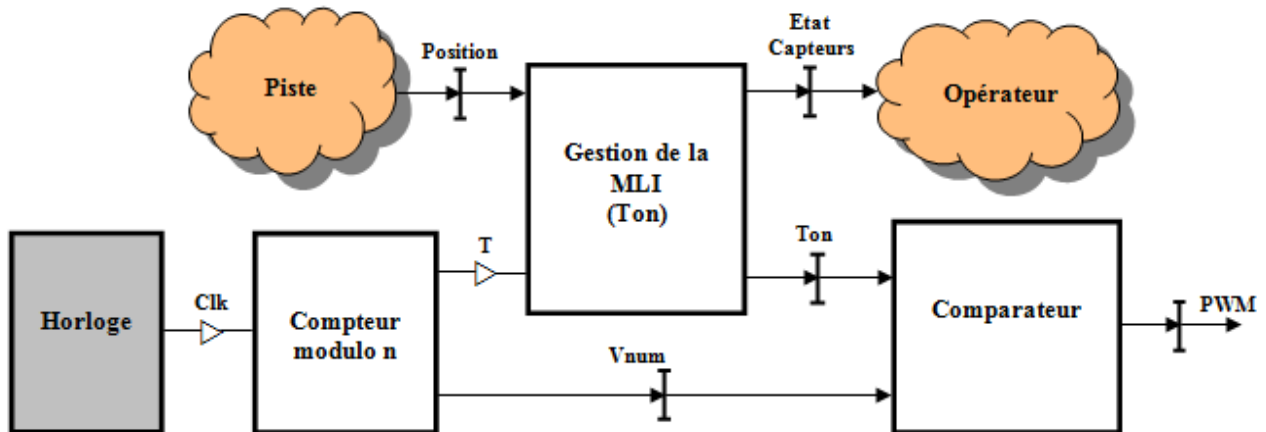


Figure 4.4

▪ Fonction : Gestion de la MLI

Entrées : **Position[2 .. 0]** : Valeur numérique codée sur 3 bits correspondant aux états logiques des 3 capteurs de position cg, cm et cd (capteurs **g**auche, **m**ilieu et **d**roit). Cette information donne la position du robot par rapport à la piste.

T (événement) : indicateur de dépassement produisant un signal dont la période est de 10ms.

Sorties : **Etats Capteurs** : indication optique (LEDs) signalant la position du robot par rapport à la piste.

Ton : valeur numérique image de la durée de l'impulsion à générer pour obtenir une direction souhaitée.

▪ Fonction : Comparateur

Entrées : **Ton** : valeur numérique image de la durée de l'impulsion à générer pour obtenir une direction souhaitée.

Vnum : valeur numérique évoluant en dents de scie.

Sortie : **PWM** signal de commande du servomoteur (0.6ms correspond à un angle de -90° et 2.2ms à un angle de 90° avec une période égale à 10ms).

▪ Fonction : Compteur modulo n

entrée : **Clk** : signal d'horloge dont la fréquence est de 102.4kHz.

Sorties : **T** (événement) : indicateur de dépassement produisant un signal dont la période est de 10ms.

Vnum : valeur numérique évoluant en dents de scie.

▪ Fonction : Horloge

Sortie : **Clk** : signal d'horloge dont la fréquence est de 102.4kHz.

V. Étude Fonctionnelle

a) Analyser le mode de fonctionnement du robot :

- *analyse de la fonction « Contrôle de la propulsion ».*

La fonction « Contrôle de la Propulsion » comprend :

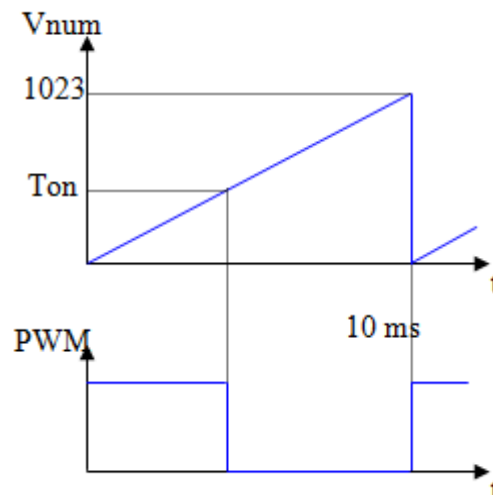
- un module de détection de surintensité du courant moteur qui permet de détecter la présence d'un obstacle
- un module de détection de batterie déchargée
- un module de gestion du déplacement qui permet de choisir le sens (avant ou arrière) du moteur, mais aussi de générer un signal MLI pour faire varier la vitesse de rotation du moteur
- et une interface de puissance constituée d'un pont en H choisis pas rapport aux contraintes technologiques

- *analyse de la fonction « Contrôle de la direction » (suivi de piste).*

La fonction « Contrôle de la Direction » comprend :

- Un module de gestion de la MLI qui génère une valeur numérique T_{on} image de la durée de l'impulsion à générer pour obtenir la direction souhaitée
- Un compteur modulo n dont le signal d'horloge est de fréquence 102.4kHz soit une période de $9.76\mu\text{s}$ or le signal de sortie V_{num} à une période de 10ms soit 1024 fois plus longue : on a donc un compteur modulo 1024
- Un comparateur qui compare la valeur de T_{on} et V_{num} pour générer un signal de rapport cyclique dont T_{on} est l'image.

Exemple:



Le robot intègre les 4 cartes électroniques suivantes :

- Carte fille n°3 : Carte InfraRouge (fonctionnement expliqué par l'enseignant)
- Carte fille n°2 : Carte de Commande (CPLD cypress)
- Carte fille n°1 : Carte Puissance (à réaliser)
- Carte mère : Carte accueillant les connecteurs d'entrées/sorties

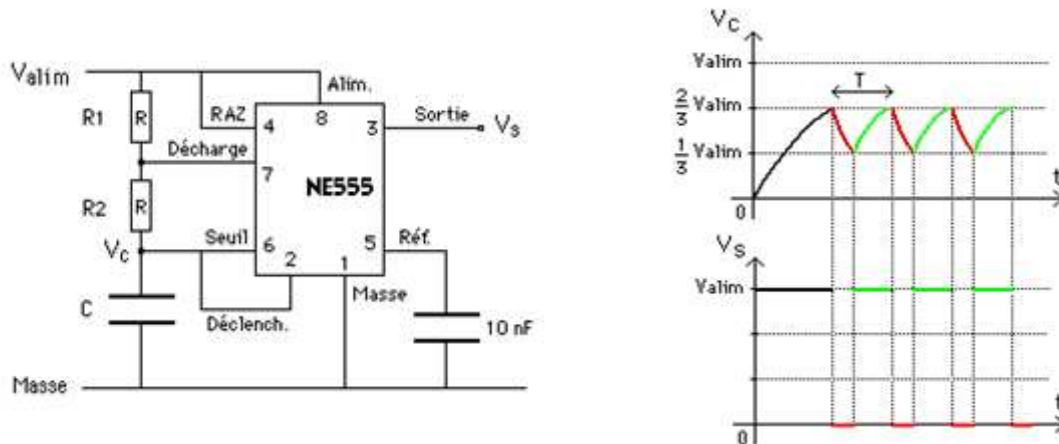
Les schémas structurels de la carte mère et des cartes filles n°2 & 3 sont fournis en annexe.

VI. Étude du schéma structurel de la carte puissance

1. Horloge

La fonction « horloge » doit produire un signal dont la fréquence est de 102,4kHz. On réalise cette fonction en utilisant un circuit intégré dédié : le NE555 en mode astable.

a) A partir de la documentation, décrire le fonctionnement de ce circuit en mode "astable".



Le condensateur est supposé initialement déchargé : $V_c = V_{seuil} = V_{decl} = 0 \text{ V}$, donc $V_s = V_{alim}$. Le condensateur se charge à travers $R_1 + R_2$ sous V_{alim} avec $V_c(0)$ nul selon $V_c = V_{alim} \cdot e^{-t/(R_1 + R_2)C}$, jusqu'à ce que $V_c (= V_{seuil} = V_{decl})$ atteigne $2/3 V_{alim}$.

La sortie passe à l'état bas et le transistor devient passant; le condensateur se décharge à travers R_2 selon $V_c = 2/3 V_{alim} \cdot e^{-t/(R_2)C}$ jusqu'à ce que $V_c (= V_{seuil} = V_{decl})$ atteigne $1/3 V_{alim}$.

La sortie passe à l'état haut et le transistor est bloqué; le condensateur se charge à travers $R_1 + R_2$ sous V_{alim} selon $V_c = V_{alim} \cdot (1 - 2/3 e^{-t/(R_1 + R_2)C})$ jusqu'à ce que V_c atteigne $2/3 V_{alim}$.

La sortie passe alors à l'état bas et le transistor est passant. Et le cycle recommence.

On observe une oscillation périodique V_s entre 0V et V_{alim} et de V_c entre $1/3 V_{alim}$ et $2/3 V_{alim}$.

Le système n'arrive pas à trouver un régime stable, on dit qu'il est astable.

(Chronogramme explicatif ci-dessus)

b) Déterminer les valeurs des composants associés. On utilisera un potentiomètre afin de pouvoir régler précisément la fréquence du signal.

Démarche des choix composants :

Nous voulons une fréquence de 102.4 kHz soit une période de 9,765625us or d'après la documentation $T = C \cdot \ln 2 \cdot (R_1 + 2R_2)$

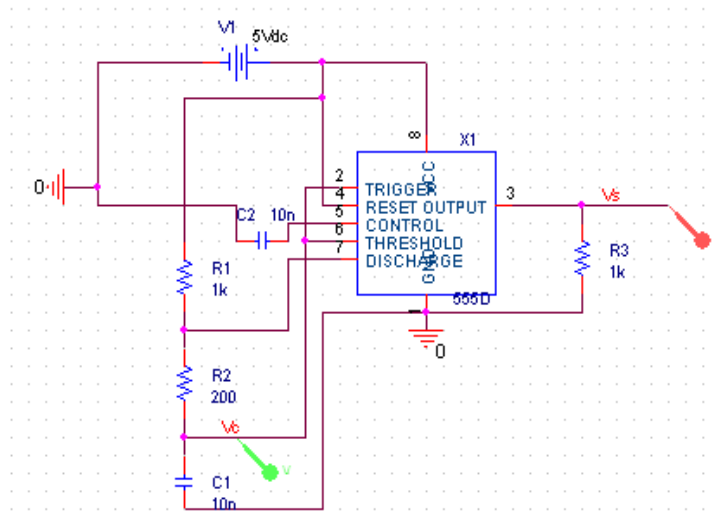
Donc $R_1 + 2R_2 = T / (C \cdot \ln 2) \approx (14 \cdot 10^{-6}) / C$, si on choisi $C = 1\text{nF}$ alors

$R_1 + 2R_2 = 14088,81876\Omega$, si on choisi $R_1 = 10\text{K}\Omega$ alors $R_2 = 2044,4\Omega$.

Pour R_2 on choisira une résistance de $1\text{k}\Omega$ suivi d'un potentiomètre de $1\text{k}\Omega$ pour permettre un réglage de la fréquence plus précis.

c) Simuler sous Orcad/Pspice le montage afin de valider le choix des composants.

Pour la réalisation, il faudra mettre le potentiomètre coté « Bottom » pour pouvoir facilement effectuer un réglage à l'aide d'un tournevis.

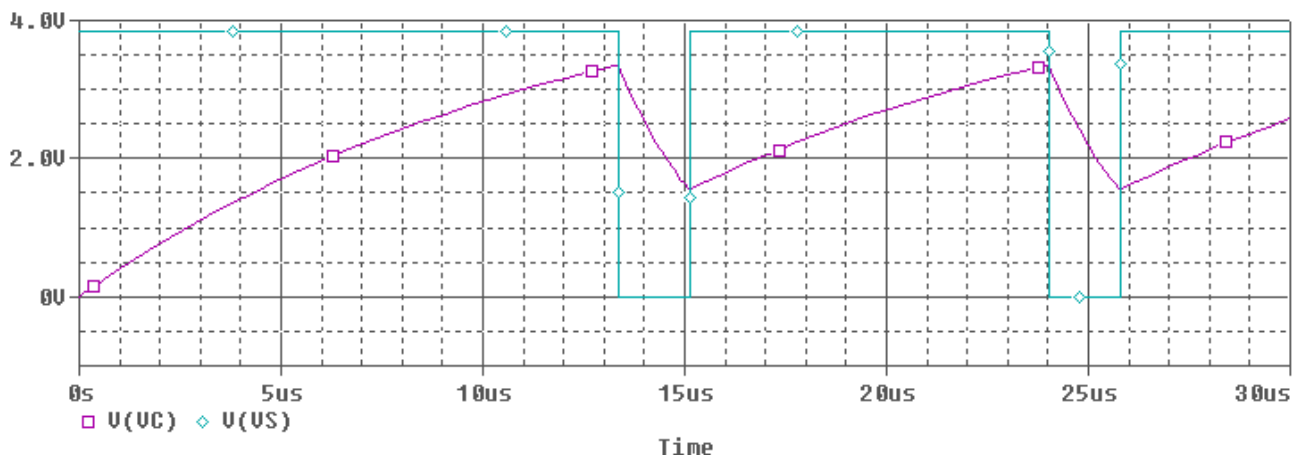


C2 est un condensateur imposé par la documentation constructeur.

R3 est la résistance de sortie du montage indispensable pour la simulation mais qui ne sera pas intégrée à la carte. (erreur rectifié par un via sur la carte).

Durant la simulation $C1(\text{réelle})=C1/10$ et $R_{\text{total}}(\text{réelle})=R_{\text{total}}*10$, car il manquait des condensateurs de 10nF, pour que la carte chauffe moins et pour mieux faire varier la fréquence.

Par simulation, on obtient les courbes suivantes :



Le schéma correspond à la documentation et aux valeurs souhaités donc le montage est validé.

2. Interface moteur

Les caractéristiques du moteur à aimants permanents utilisé sont les suivantes :

- tension d'alimentation : 6 à 15 Volts.
- réducteur dont le rapport est fixé par le choix des engrenages (autour de 1/16).
- vitesse à vide 12600t/mn sous 12 Volts
- Courant max 1,24A pour un couple de 70,6 gcm.

Pour ces moteurs de faible puissance, le pont en H peut être réalisé par un seul circuit prévu à cet effet. On propose d'utiliser le circuit intégré de référence L6202 (voir notice).

a) Analyser le fonctionnement interne de ce circuit.

Schéma interne du L6202 :

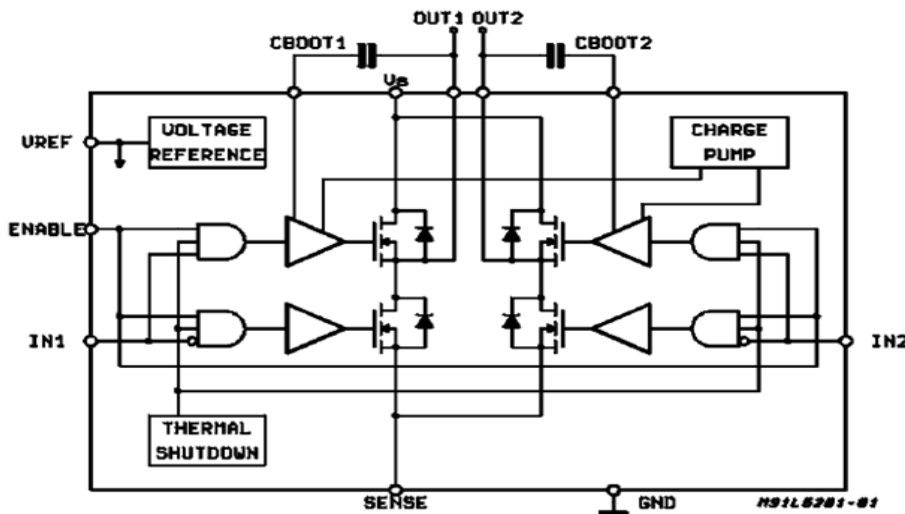
Le pont en H n'est validé que si le signal enable (Valmot) est au niveau haut, sinon il est impossible de commander le moteur, il est en régime libre.

Lorsque In1, qui équivaut à MrchAv, est au niveau haut et In2, qui équivaut à MrchAr, est au niveau bas, les transistors T1 et T4 sont passants alors que T2 et T3 sont bloqués, le moteur tourne donc en marche avant.

Lorsque In2 est au niveau haut et In1 est au niveau bas, les transistors T1 et T4 sont bloqués alors que T2 et T3 sont passants, le moteur tourne donc en marche arrière.

Lorsque In1 et In2 sont actifs simultanément, il y a utilisation du frein moteur.

Enfin, on observe la présence de diode de roue libre pour éviter la destruction par surtensions des transistors à cause de l'impédance du moteur.



b) Donner la table de vérité décrivant le fonctionnement de ce pont pour les 4 modes suivants : arrêt, rotation dans un sens (correspondant à la marche avant du robot, marche arrière et rotation en marche avant mais à vitesse réduite).

Table 1

Inputs			Output Mosfets (*)
	IN1	IN2	
$V_{EN} = H$	L	L	Sink 1, Sink 2
	L	H	Sink 1, Source 2
	H	L	Source 1, Sink 2
	H	H	Source 1, Source 2
$V_{EN} = L$	X	X	All transistors turned OFF

Note : Source 1 ⇔ Transistor en haut à gauche soit T1 passant

Source 2 ⇔ Transistor en haut à droite soit T2 passant

Sink 1 ⇔ Transistor en bas à gauche soit T3 passant

Sink 2 ⇔ Transistor en bas à droite soit T4 passant

L = Low H = High X = DON't care

(*) Numbers referred to INPUT1 or INPUT2 controlled output stages

On en déduit le fonctionnement suivant pour le moteur :

Si IN1=0 et IN2=0 : OUT1 et OUT2 sont à la masse ⇔ Moteur à l'arrêt (libre)

Si IN1=0 et IN2=1 : OUT1 est à la masse et OUT2 à V_s ⇔ Moteur dans le sens 2

Si IN1=1 et IN2=0 : OUT1 est à V_s et OUT2 à la masse ⇔ Moteur dans le sens 1

Si IN1=1 et IN2=1 : OUT1 et OUT2 sont à V_s ⇔ Moteur bloqué (frein)

c) Les signaux de commandes sont compatibles TTL : est-il possible de commander un moteur alimenté directement par la batterie du robot (tension comprise entre 7.8V et 12V)?.

Vs est l'entrée d'alimentation du pont en H (S comme Supply). On remarque que celle-ci n'est reliée à aucun organe de commande. Il est donc possible d'y connecter directement VALIM pour piloter le moteur. Par contre, IN1, IN2 et EN sont des entrées de commande qui doivent être pilotées par des signaux TTL (venants du Cypress).

La documentation constructeur indique que les signaux Vin et Ven doivent être compris entre -0.3 et 7V (absolute maximum rating). Or les signaux de commandes TTL sont compris entre 0 et 5V, donc la commande est possible.

De même, la documentation constructeur indique que la tension d'alimentation du moteur doit être comprise entre 6 et 15V. Or la tension de batterie est comprise entre 7.8 et 12V, donc la commande du moteur est possible.

Afin de protéger le circuit Cypress, il est demandé d'utiliser un circuit buffer pour la commande de ce pont en H. On pourra, par exemple, utiliser des portes logiques (non inverseuses) entre le CPLD et le L6202.

d) Quelle est la puissance dissipée dans le circuit L6202 si le courant dans le moteur est de 0.5A?

Le circuit L6002 dispose d'une broche "sense" utilisable pour mesurer le courant dans le moteur.

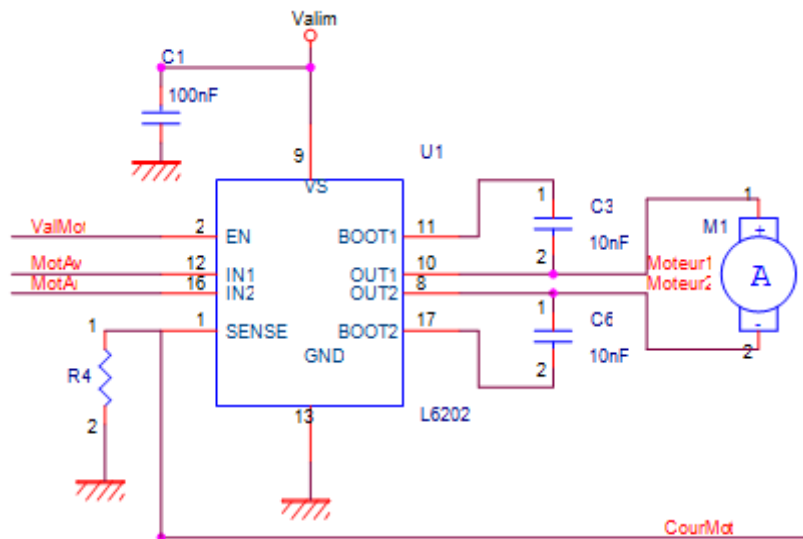


Figure 5-1

La puissance dissipée dans le circuit L6202 si la MCC consomme 0.5A, est de :

- 0.3Ω pour RDS on
- 0.5A pour la MCC

Or il y a 2 transistor saturés à chaque fois donc $P=2RI^2= (0.6*(0.5)^2)=0.15W$.

e) Déterminer la valeur de la résistance (R4 sur le schéma ci-dessus) à utiliser pour obtenir une tension image du courant dans le moteur.

Attention à la puissance dissipée dans cette résistance

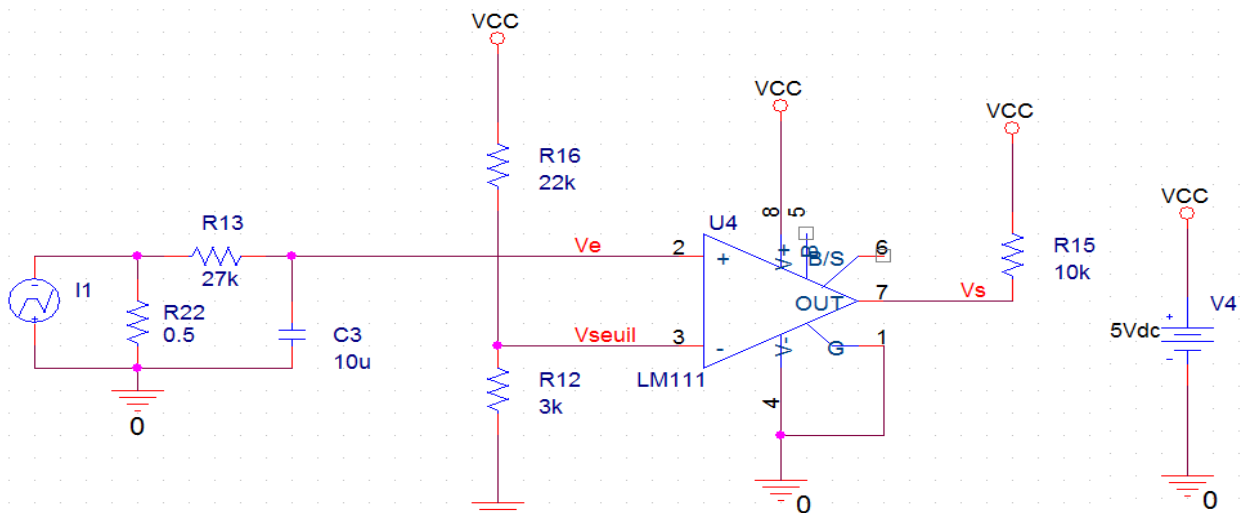
$P/I_{mot}^2 = 0,15/(0,25)^2 = 0,6\Omega$. On met 2 résistance de 1Ω en parallèle pour se rapprocher de 0,6.

3. Détection surintensité

a) Réaliser le schéma de détection de surintensité (comparateur). Une sélection d'AOP et de comparateurs vous est proposée sur le site internet du RAFIL.

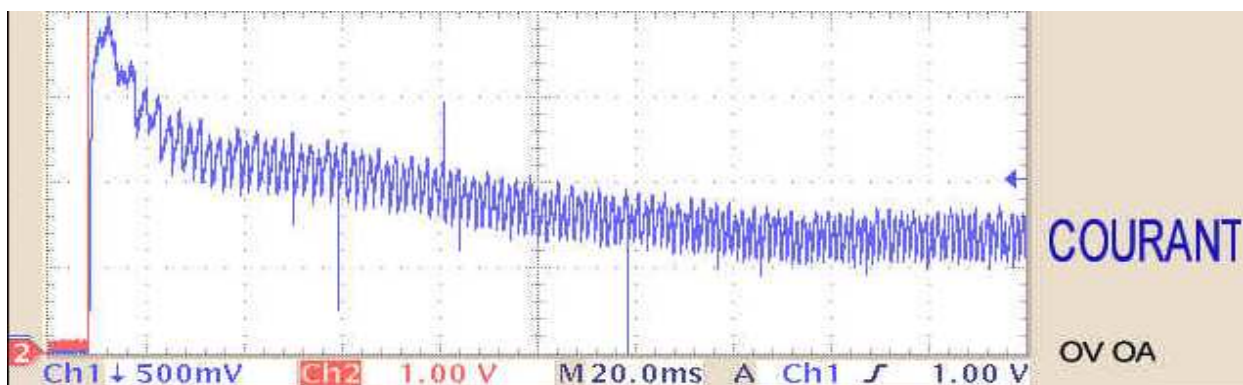
La structure mise en oeuvre pour la détection de surintensité est la suivante :

Il faudra précéder la structure de comparaison d'un circuit RC (filtre) pour ne pas prendre en compte l'impulsion du courant à chaque démarrage. (regarder les mesures du courant moteur sur le site Internet)



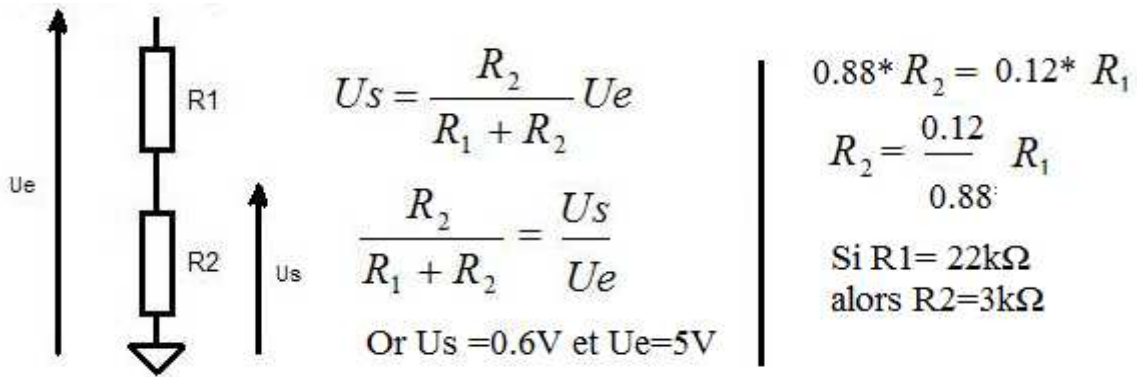
Ce montage est composé de trois ensembles :

- Le système de réglage du seuil de la détection de surintensité (réalisé à l'aide d'un potentiomètre). Cette partie du circuit fournira une tension de seuil VSEUIL.
- Le circuit RC (filtre) permettant de négliger l'impulsion de courant à chaque démarrage du moteur.
- L'AOP LM393 (pour la simulation nous avons choisi le LM111 car il était le seul semblable au LM393 simulable sur Pspice) permettant de réaliser la comparaison entre la Tension d'entrée VE (portant l'information de l'intensité du moteur) et la tension de Seuil (le choix de ce composant sera expliqué plus tard).



La sortie SENSE, associée à une résistance, permet de relever la tension image du courant circulant dans le moteur. Nous voulons détecter une surintensité, sur ce schéma nous voyons que l'image du courant du moteur normale est entre les 500mV et 560mV pour une résistance $R_4=1\Omega$. Lorsqu'il y a un surintensité nous voyons que l'image du courant du moteur est compris entre 1V et 2V. Or nous avons choisi une résistance $R_4=0,5\Omega$ donc on divise par 2 les résultats précédents, c'est pour cette raison que nous avons choisi un Vseuil compris entre 217mV et 600mV.

Calcul de $R_{12}=R_2$ et $R_{16}=R_1$:



Nous voyons également sur cette photo que le pique de démarrage dure à peu près 200ms. On choisi une constante de temps égale à : $\tau = RC = 270ms$ pour être sur de ne pas rencontrer le pique de démarrage. Si $R = 27k\Omega$ nous avons donc $C = 270 * 10^{-3} / 27 * 10^3 = 10 * 10^{-6} = 10\mu F$. La résistance R_{15} dite de « pull up » sert a limité le courant car le comparateur possède une sortie à collecteur ouvert, contrairement à l'AOP, nous l'avons donc mis a $10k\Omega$.

b) Donner les critères de choix retenu pour le comparateur à utiliser.

Pour ces moteurs à prix faible les tolérances sur les caractéristiques sont très larges.

D'autre part pour déterminer le couple, des essais sur la piste sont nécessaires.

Il faudra donc prévoir un réglage sur la détection de surintensité.

Suivant la solution, le signal généré (appelé Imoteur) sera actif au niveau haut ou au niveau bas. Il faudra en tenir compte pour la programmation du CPLD.

Le LM311 doit être alimenté par du +15V et du -15V , le TL082 doit être alimenté par du +18V et du -18V. On ne les choisira pas car il doivent être alimenté par des tensions négatif et que la batterie ne fournit pas une tension suffisante.

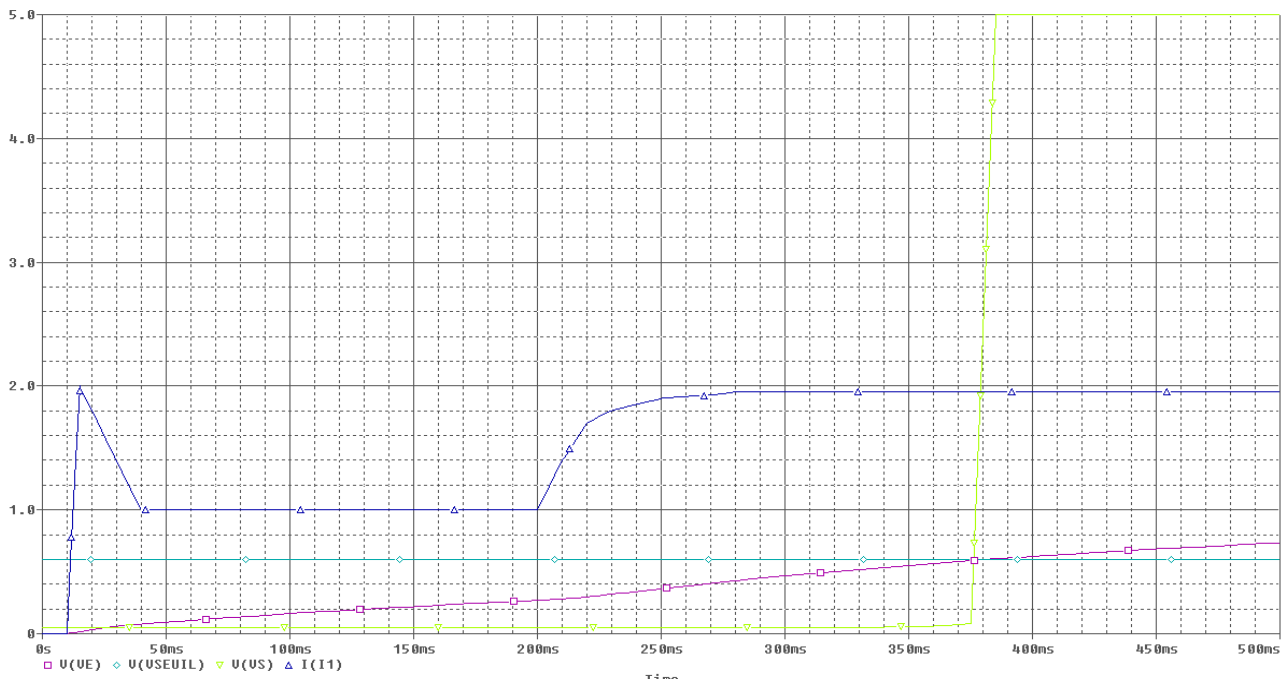
Il nous reste donc le LM358 et le LM393 qui doivent être alimenté par du 5V. Le LM393 est moins cher que le LM358. Un amplificateur opérationnel peut être utilisé en comparateur, mais ses performances seront en général moins bonnes qu'un vrai comparateur car il n'est pas optimisé pour cette application (temps de réaction plus lent si les écarts de tension sont faibles, etc).

Donc pour ce projet nous utiliserons le LM393 qui est un comparateur et non un Aop.

c) Simuler le montage proposé avec Pspice afin de vérifier la validité du montage.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
I(A)	0	0	2	1	1	1,4	1,7	1,8	1,9	1,95	1,97	1,99	2	2	2
T(ms)	0	10	15	40	200	210	220	230	250	280	300	350	400	440	500

Par simulation, on obtient les courbes suivantes :



I (en bleu), V_e (en rouge), V_{seuil} (en vert), et V_s (en noir)

Nous voyons que bien que nous ne détectons pas la surintensité au démarrage par contre la détection est un peu long après mais nous avons prévu ceci car il faut compter le poids du robot qui demandera plus d'intensité donc le montage est validé.

4. Détection batterie faible

La fonction « détection batterie faible » doit indiquer à l'opérateur que la batterie est déchargée à l'aide d'une DEL. Le seuil est fixé à 6V. Il faudra justifier l'origine de ce seuil.

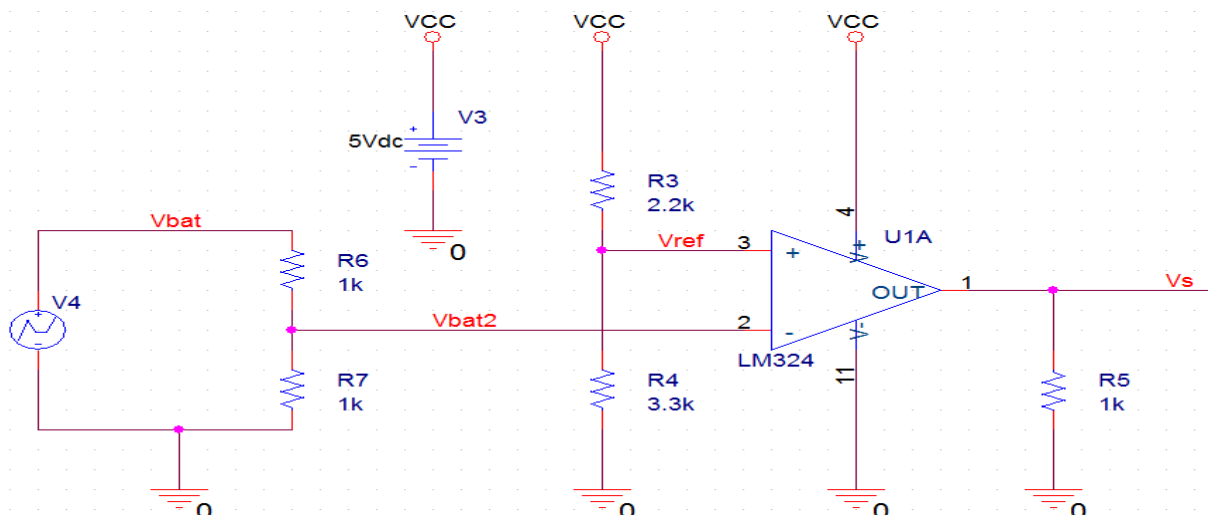
La batterie est composée de six éléments de 1.2V chacun. Le seuil de déchargement d'une batterie est de 1V par éléments, si la décharge est plus profonde, c'est-à-dire en dessous de 1V, on risque d'endommager la batterie. Donc le seuil minimum est de 6V

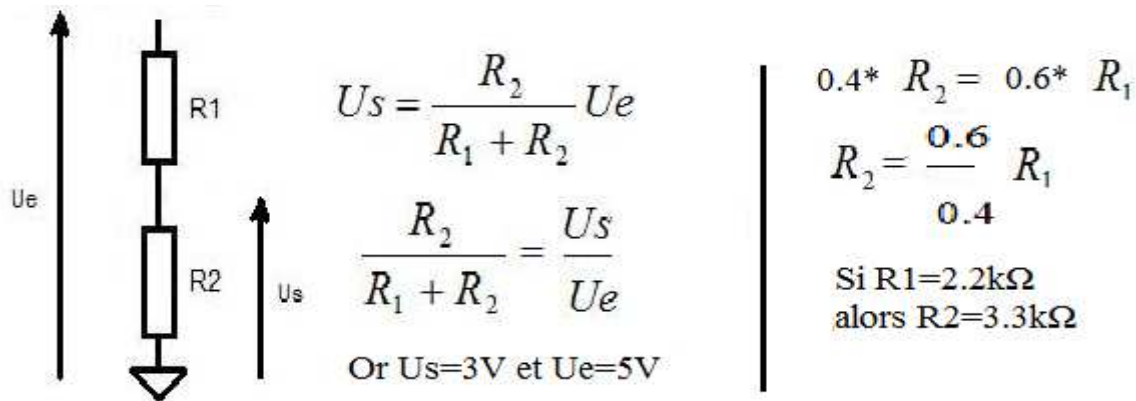
a) Réaliser le schéma de détection de batterie faible en justifiant le choix des composants. Une sélection d'AOP et de comparateurs vous est proposée sur le site internet du RAFIL.

Pour les mêmes raisons que ci-dessus nous utiliserons le LM393. De plus dans un boîtier il y a 2 comparateur.

La structure mise en oeuvre pour la détection de batterie faible est la suivante :

6V est supérieur à VCC nous avons donc mis un pont diviseur de tension pour diviser V_{bat} par 2. Ainsi la tension $0 > V_{ref} = 3V > VCC$.

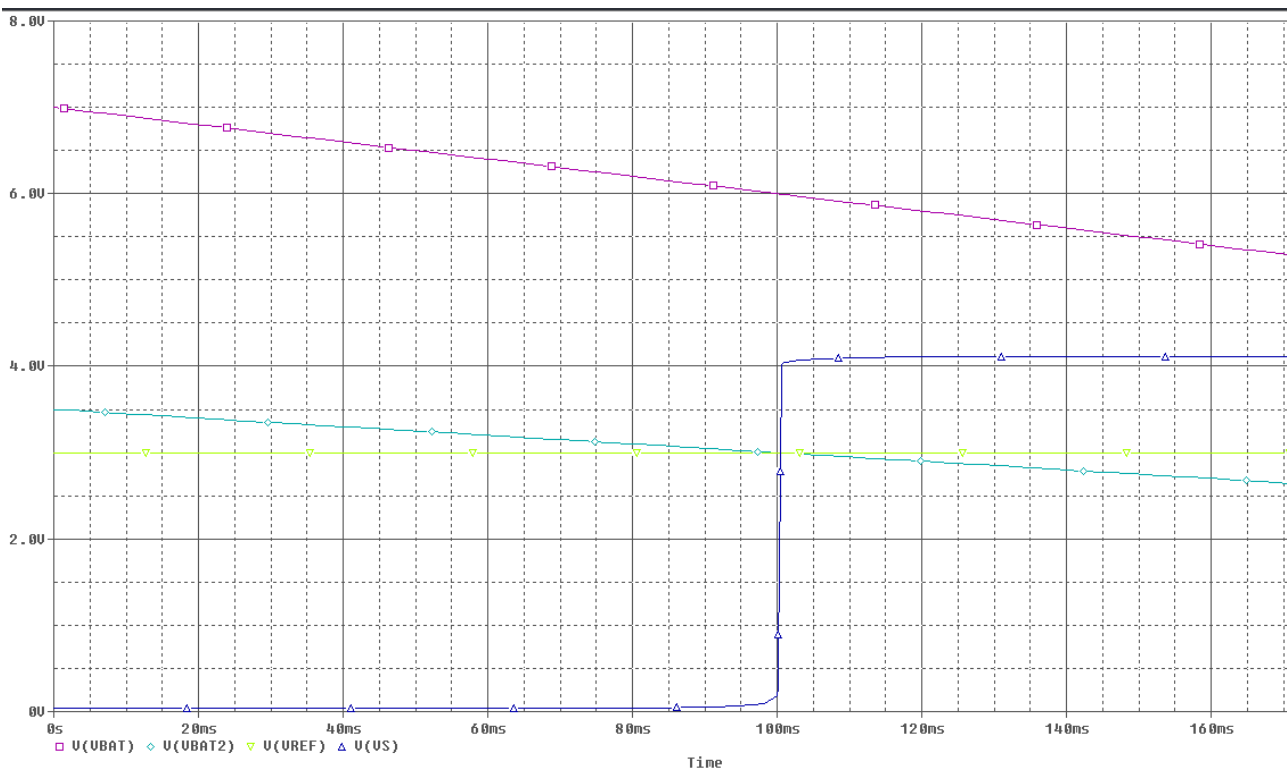




b) Simuler le montage proposé sous Orcad/Pspice afin de valider le montage.

On optimisera le schéma complet en utilisant le nombre minimal de circuits intégrés avec, par exemple, l'emploi d'un comparateur double.

Par simulation, on obtient les courbes suivantes :



Vbat(en bleu), Vbat2 (en rouge), Vref (en vert), et Vs (en noir)

Nous voyons bien que la sortie passe à 1 lorsque Vbat < 6V donc notre montage est validé.

5. Régulation de tension 5V

Le robot est alimenté par une batterie 6 éléments de 7.2V/1.2Ah.

a) En utilisant la documentation sur les régulateurs linéaires, choisir un régulateur en expliquant sur quels critères ce choix a-t-il été effectué. Vous pouvez faire un choix entre un LM2940T-5, 78L05 et 7805 en les comparant dans un tableau.

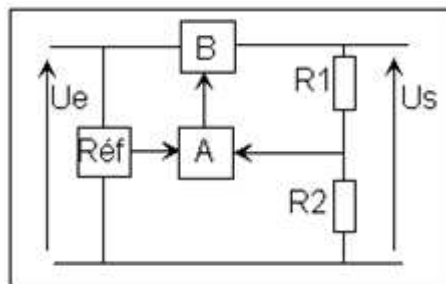
Nous devons mettre en œuvre un régulateur afin d'avoir une tension de 5V nécessaire au bon fonctionnement des composants. Nous avons le choix entre le LM2940T-5, le 78L05 et le 7805.

Nous avons choisi d'utiliser le composant LM2940T-5 car c'est le seul composant qui accepte une tension d'entrée minimale de 6V. Le dropout voltage correspond à la tension différentielle entre l'entrée et la sortie du composant nécessaire au bon fonctionnement du composant. La sortie, pour tous les composants est à 5V. Le dropout voltage du LM2940T-5 est typiquement de 0,5V et au maximum de 1V. On peut donc descendre à 6V sans craindre de dysfonctionnement. Par contre, les autres composants ont un dropout voltage d'au moins 1,7V. Il risque donc d'y avoir un mauvais fonctionnement si la tension batterie tombe à 6V, la sortie ne pourra probablement pas fournir du 5V. Pour ce qui est des autres paramètres, le composant LM2940T-5 répond au cahier des charges.

b) Analyser la structure interne de ces régulateurs pour comprendre leur fonctionnement en expliquant brièvement les fonctions suivantes :

Le régulateur de tension se comporte de la manière suivante :

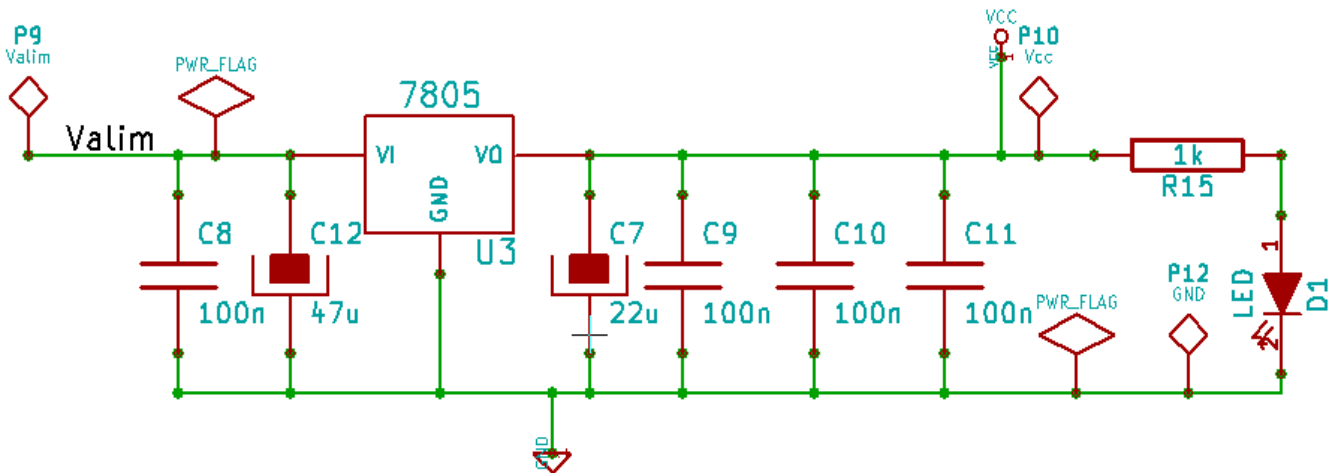
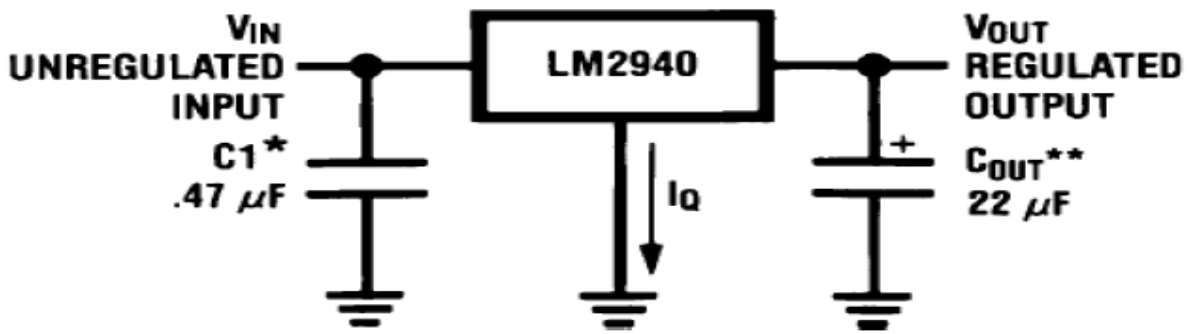
- Une source de référence (Réf) fournit une tension avec une résistance la plus grande possible en entrée
- Un amplificateur en mode comparateur (A) compare le tension de référence a un échantillon de le tension de sortie U_s pour agir sur l'organe de commande (B).
- L'organe de commande agit sur la tension de sortie suivant les instructions de l'amplificateur de comparaison
- Les résistances R1 et R2 permettent un échantillonnage de la tension de sortie



- *transistor ballast* : il permet de réguler la tension de sortie suivant les instructions donné par l'amplificateur de comparaison.
- *protection contre les surintensités* permet d'éviter la dégradation du régulateur dû à un courant trop élevé.
- *protection thermique* permet d'éviter la dégradation du régulateur du à une surcharge qui provoquerait une surtension donc un échauffement important.
- *Rendement du régulateur et dissipation.* Pour le LM2940T-5 il dissipe au maximum 20W et le 78L05 et le 7805 dissipent au maximum 16W

c) Faire le schéma structurel de cette alimentation avec protection contre les tensions inverses et découplage (voir note d'application).

Le montage est imposé avec des condensateurs de découplage. Un condensateur de 470nF en entrée permet d'absorber les perturbations sur l'alimentation. Un condensateur polarisé de 22nF en sortie permet de stabiliser le système. Le schéma structurel est donc le suivant :



Régulation de tension 5V

Pour protéger le circuit contre les tensions inverses on place une diode qui shunt le régulateur en cas d'inversion de tension.

d) Sous quelle forme part une partie de l'énergie en effectuant cette régulation ? Pour optimiser l'énergie de ce robot quel type de régulateur de tension faudrait-il plutôt utiliser ?

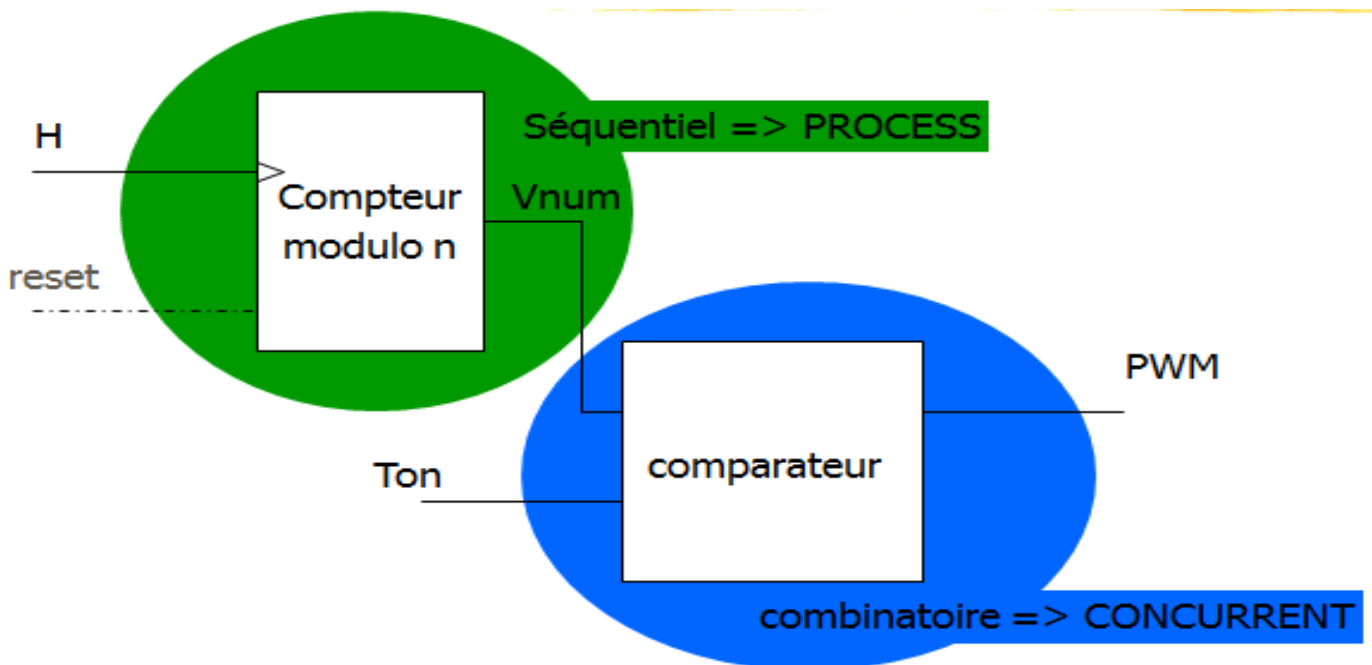
Les régulateur de tensions ont pour fonction de réguler la tension en sortie à une certaine valeur indépendante de la valeur de la tension d'entrée. Le courant, cependant, est le même en entrée qu'en sortie.

La forme sous laquelle part l'énergie effectuant la régulation est thermique, c'est pour quoi, pour certains montages, il est nécessaire de mettre un dissipateur thermique. La consommation en courant étant faible, il ne sera pas nécessaire dans notre cas d'ajouter de radiateur sur le composant.

Pour optimiser l'énergie du robot il faudrait utiliser un régulateur avec une faible chute de tension et qui travaillerait avec des tensions plus élevées car dans ce cas les pertes par effets joules serait moins grande et le rendement meilleurs.

VII. Etude des fonctions programmées

1. Sous-fonctions liées à la fonction « Gestion de la direction »

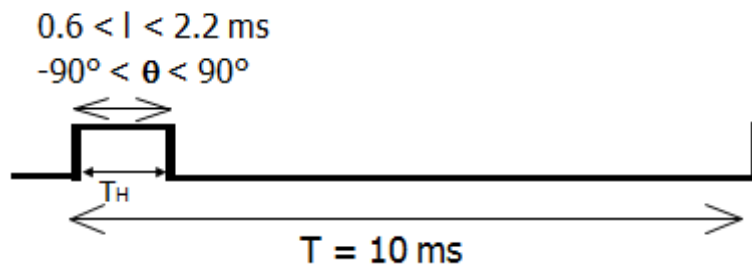


Il s'agit, dans un premier temps, de décrire les sous-fonctions « Compteur modulo n » et « Comparateur ». Ces deux fonctions permettent de générer un signal MLI sur la sortie PWM afin de commander le servomoteur. La largeur de l'impulsion du signal MLI sera fonction d'une consigne fixée par Ton.

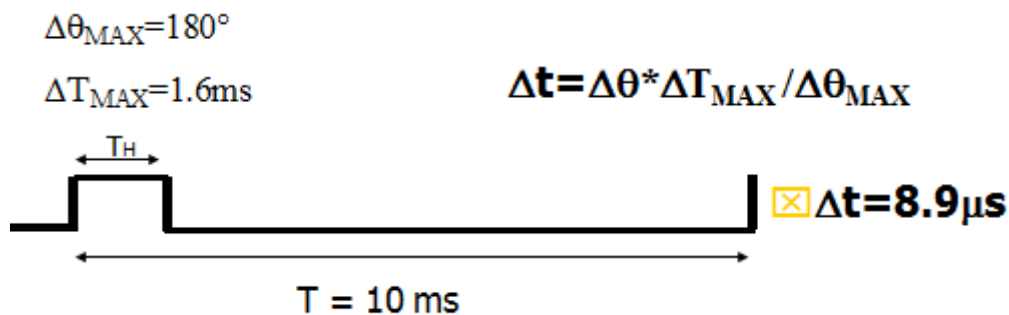
a) A partir de la présentation effectuée par l'enseignant, réaliser la description VHDL des sous-fonctions « Compteur modulo n » et « Comparateur ».

Nous allons réaliser un compteur 1024 et un comparateur qui va permettre de comparer Vnum et Ton (une valeur que l'on imposera après divers observation).

Le cahier des charges nous impose une résolution angulaire $\Delta\theta$ de 1° (la roue tourne par pas de 1°). Le servomoteur est commandé par le signal MLI suivant :



Nous devons donc déterminer la résolution temporelle Δt permettant de respecter la résolution angulaire souhaité :



Donc pour incrémenter la rotation de la roue de 1° il faut incrémenter T_h de Δt .

Il faut donc déterminer la valeur du coefficient N, qui permet de générer un signal (MLI) qui respecte la résolution angulaire et qui a une période $T = 10 \text{ ms}$, à partir d'une horloge de période Δt .

$$T = N * \Delta t \quad \text{N} = 1124$$

Cela revient à réaliser une division de fréquence, la division fréquence est réalisée par un compteur binaire : donner une valeur approchée de N, N_{aprox} en fonction d'une puissance de 2, tel que :

$$N_{aprox} = 2^n \approx N \quad \boxtimes N_{aprox} = 1024$$

De la valeur de N_{aprox} on en déduit la nouvelle fréquence d'horloge et la résolution angulaire qui en découle.

$$\Delta t_{aprox} = T / N_{aprox} \quad \left| \quad \Delta \theta_{aprox} = \Delta t_{aprox} * \Delta \theta_{MAX} / \Delta T_{MAX}$$

$$\boxtimes \Delta t_{aprox} = 9.76 \mu s \quad \left| \quad \boxtimes \Delta \theta_{aprox} = 1.09^\circ \quad \text{Soit} \quad \boxtimes F = 102.4 KHz$$

On utilise un Process car la valeur de Vnum change sur chaque front montant de Clk, le traitement se déroule donc dans le domaine séquentiel.

Texte en VHDL du compteur :

--Diviseur par 1024

Div_1024 : process(Clk) --Début du process compteur

begin

if(Clk'event and Clk='1') then --Attente front montant de Clk

 if (Rst='0') then

 Vnum<="0000000000"; --Remise à zéro synchrone

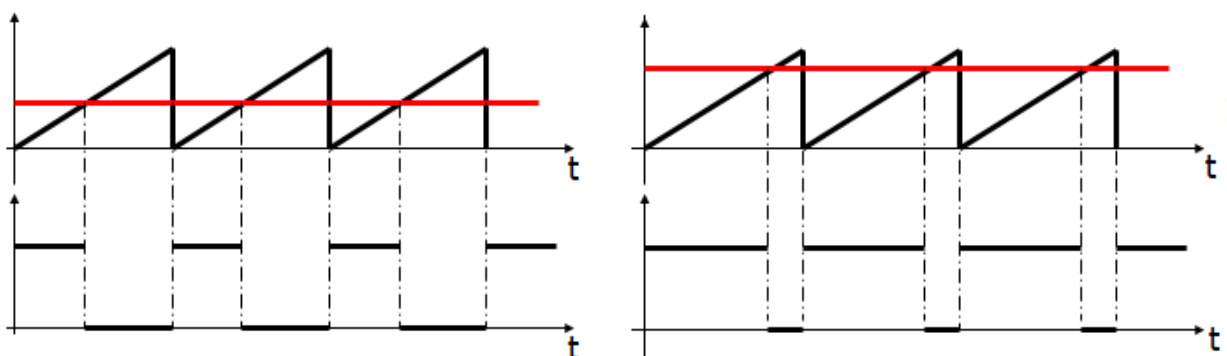
 else Vnum<=Vnum+1; --Incrémentation de Vnum à chaque front montant

 end if;

end if;

end process Div_1024; --Fin du process compteur

T<='1' when(Vnum=0) else '0'; --Impulsion de T toutes les 10ms



Comparaison d'un signal triangulaire périodique à une constante (consigne). Le temps haut (TH) dépend de la valeur de la constante.

On souhaite réaliser un comparateur entre la valeur Vnum et un signal Ton afin de réaliser le signal PWM commandant le servomoteur qui dirige le Rafil. En effet, celui-ci doit être au niveau haut entre 0,6 et 2.2 ms sur une période de 10 ms pour un fonctionnement correct du servomoteur de la direction du Rafil dans une mesure de plus ou moins 90° par rapport à la direction « tout droit ».

Cette comparaison s'effectue à tout instant, donc on place cette instruction dans le domaine

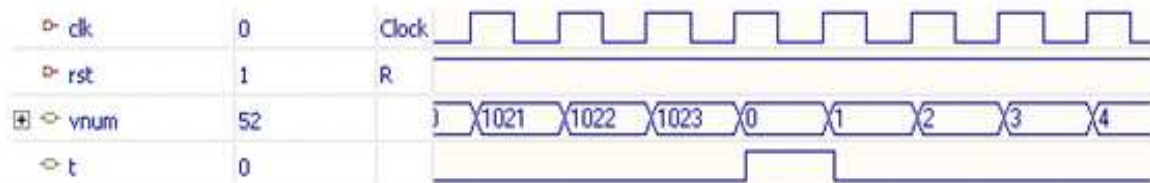
concurrent.

Texte en VHDL du comparateur :

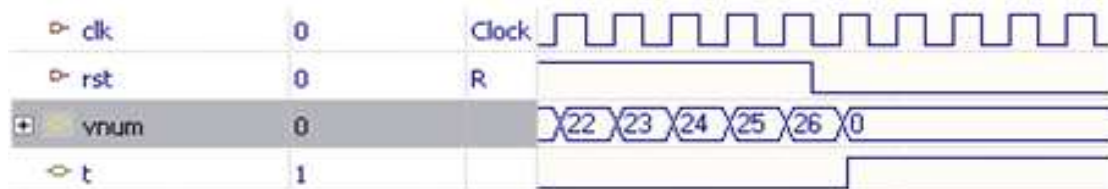
--Comparaison de Ton et Vnum

PWM<='1' when(Ton>=Vnum) else '0'; --Gestion du signal PWM de période 10ms

b) Effectuer la simulation. Afin de valider la solution, on testera différentes valeurs pour Ton et on vérifiera la largeur de l'impulsion du signal PWM pour chacune de ces valeurs.

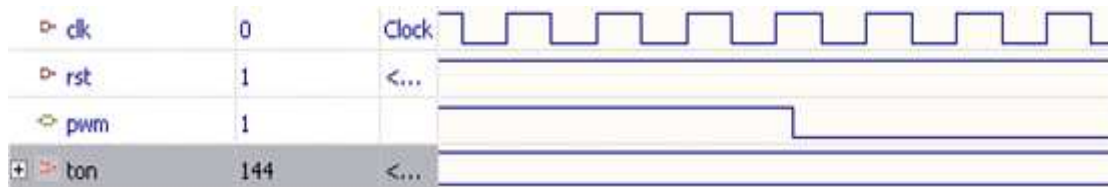


On obtient bien un compteur modulo 1024 et une impulsion sur T lorsque que Vnum =0.



On obtient bien une remise à zéro du signal Vnum lorsque le signal Rst est à zéro

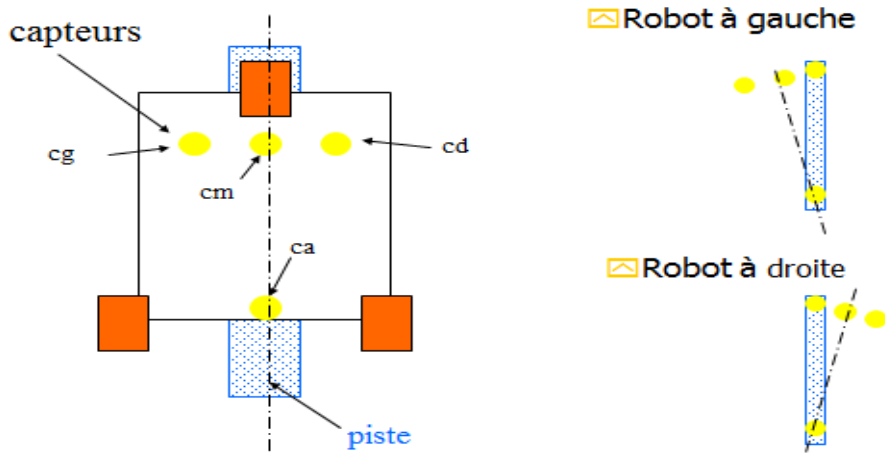
Donc le fonctionnement de ce compteur modulo 1024 est validé.

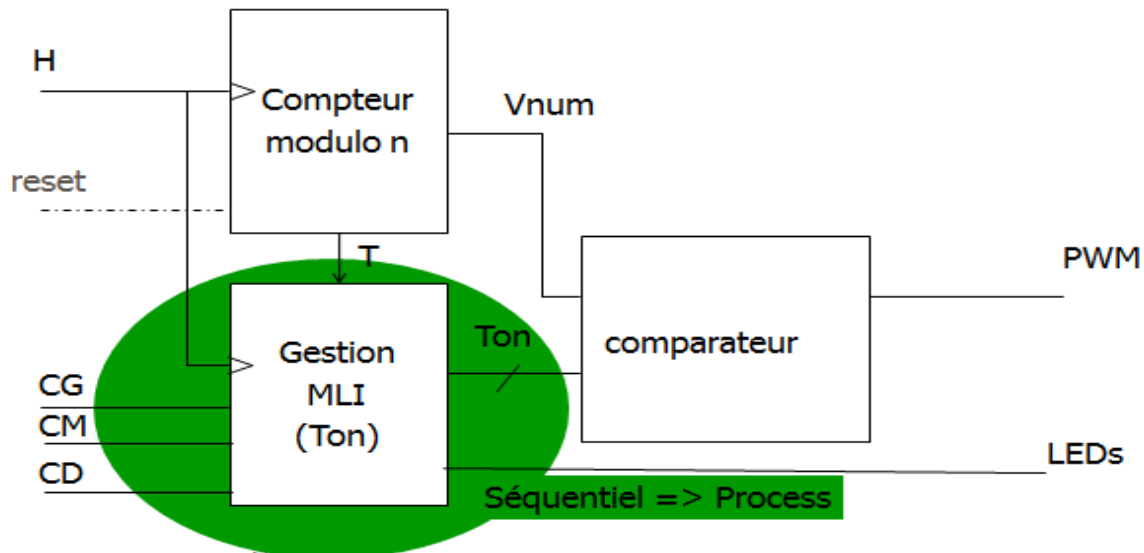


On obtient bien le signal Pwm souhaité, il est au niveau haut quand Vnum ≤ Ton et passe à zéro ensuite.

Donc le fonctionnement du comparateur associé au compteur modulo n est validé.

On veut maintenant décrire la sous-fonction « Gestion MLI » et prendre en compte les capteurs afin que la direction du RAFIL soit automatiquement contrôlée par sa position par rapport à la piste.





c) Compléter le tableau suivant afin de décrire le mode de fonctionnement du robot :

Position du robot par rapport à la piste	capteur gauche (cg)	capteur milieu (cm)	capteur droit (cd)	Direction souhaitée
aligné	1	0	1	tout droit
à droite	0	1	1	tourne à gauche
à gauche	1	1	0	tourne à droite
sorti de la piste	1	1	1	Mémoire la dir. précédente
intersection	0	0	0	tout droit

Attention : Si l'un des capteurs (cg, cm ou cd) est placé

- devant une zone noire, l'état logique récupéré est un '1'
- devant une zone blanche, l'état logique récupéré est un '0'.

On souhaite maintenant contrôler la valeur de Ton en fonction de la position du Rafil sur la piste, donc on choisit de moduler Ton en fonction des capteurs gauche (Cg), milieu (Cm) et droit (cd). Le mode de fonctionnement du robot doit être le suivant:

La gestion mli se déroule sur changement de valeur de position des capteurs ou si le capteur droit ou gauche est actif. Cette action se déroule donc sur front montant de T donc on utilise un Process pour que le traitement se déroule dans le domaine séquentiel. Nous réaliserons ce programme afin de respecter le tableau ci-dessus.

La formule pour obtenir la valeur de Ton, en décimal, en fonction du degré souhaité est :

$$Ton(\text{angle}) = \frac{\text{angle} * \left(\frac{1.4 - 0.6}{90} \right) + 1.4}{9.8 * 10^{-3}}$$

Pour chaque combinaison possible obtenue avec les capteurs de positions, on souhaite obtenir un angle particulier pour le servomoteur afin de suivre la ligne.

La valeur Ton choisie est une valeur de comparaison par rapport au compteur modulo 1024. Elle permet de régler le rapport cyclique de la MLI.

Les valeurs de Ton on donc était réglé précisément pour avoir la roue droit lorsque le robot est sur la ligne et que le robot suive bien la piste.

Texte en VHDL du MLI :

Gestion_MLI:process(Clk)

begin

if(clk' event and clk='1') then

if(T='1') then

-- aligner ou une intersection

if (((Cg='1')and(Cm='0')and(Cd='1'))or((Cg='0')and(Cm='0')and(Cd='0')))

then Ton<="10001111"; --Valeur de Ton pour aller tout droit(143)

end if;

-- à gauche // valeur de Ton pour aller à droite(167)

if ((Cg='0')and(Cm='1')and(Cd='1')) then Ton<="10100111";

end if;

--à droite // valeur de Ton pour aller à gauche(119)

if ((Cg='1')and(Cm='1')and(Cd='0')) then Ton<="01110111";

end if;

--sorti de piste // Valeur précédente de Ton pour faire revenir le robot sur la ligne blanche

if ((Cg='1')and(Cm='1')and(Cd='1')) then Ton<=ton;

if (Ton="00000000") then Ton<="10001111";

end if;

end if;

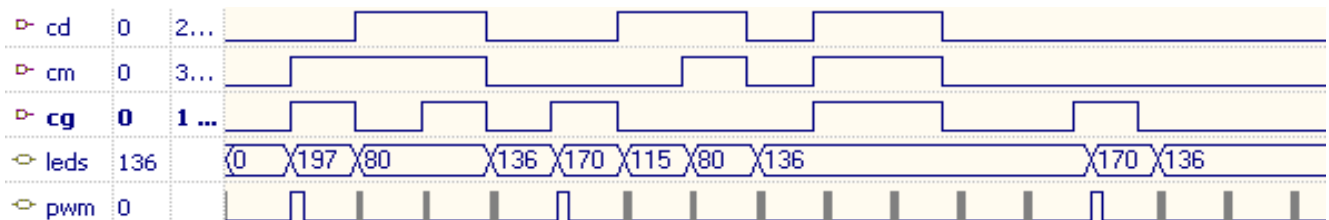
end if;

end if;

end process Gestion_MLI;

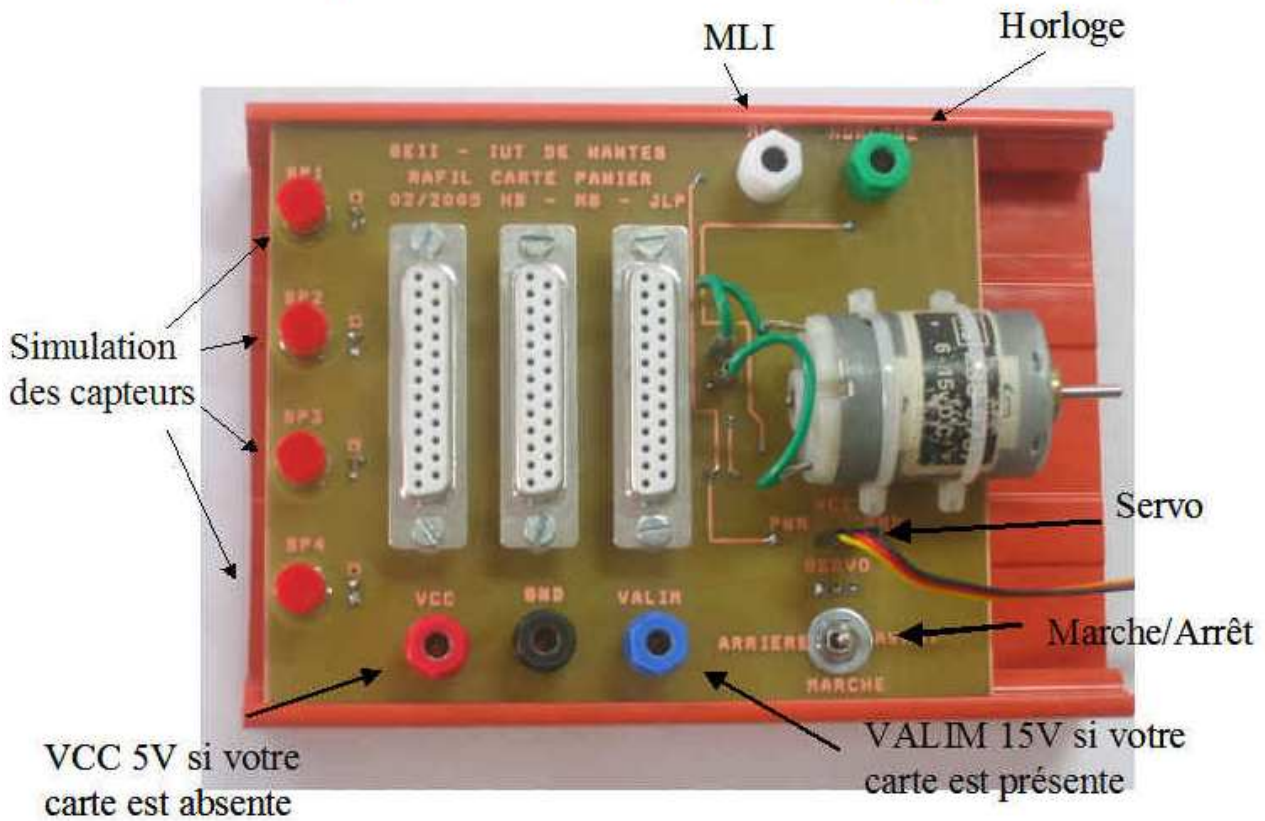
Leds<=Ton;

d) On vérifiera d'abord la validité de la solution par simulation et ensuite sur maquette (capteurs simulés et visualisation sur signal PWM).

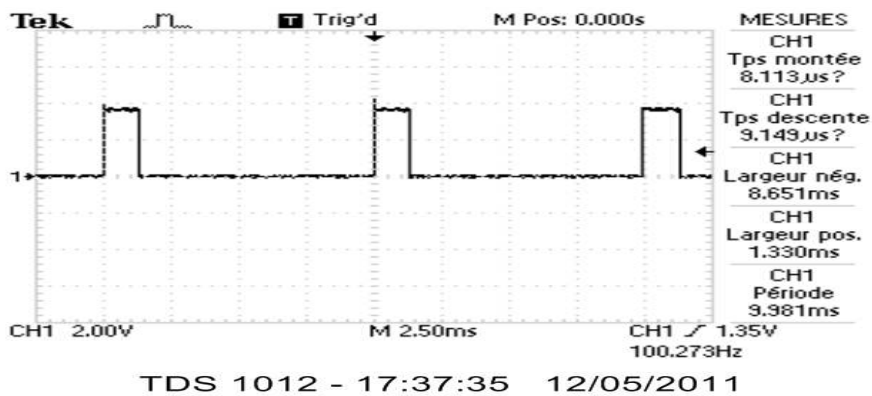


Mesure faite sur la carte de test :

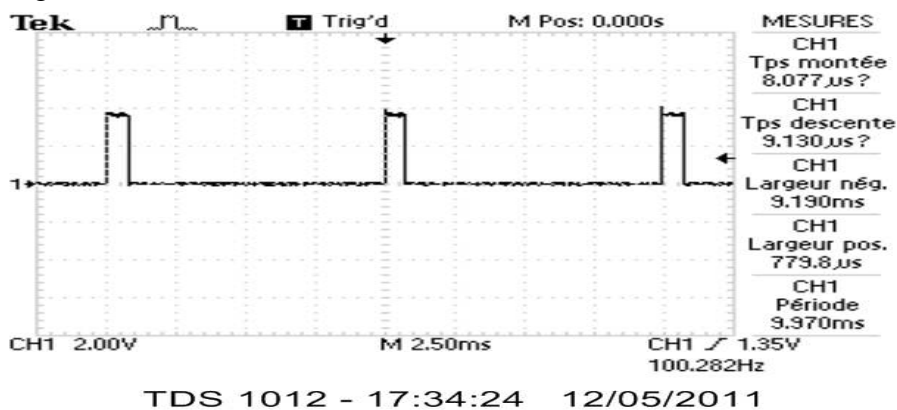
Carte test pour valider les descriptions VHDL



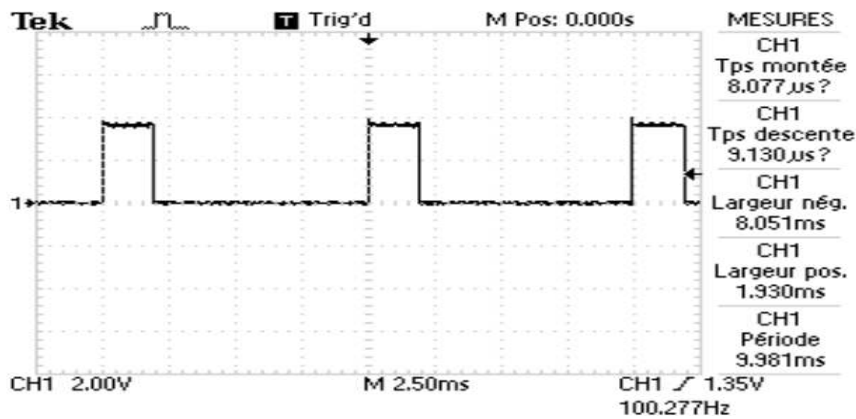
(Cg='0')and(Cm='0')and(Cd='0') et (Cg='1')and(Cm='0')and(Cd='1')



(Cg='0')and(Cm='1')and(Cd='1')

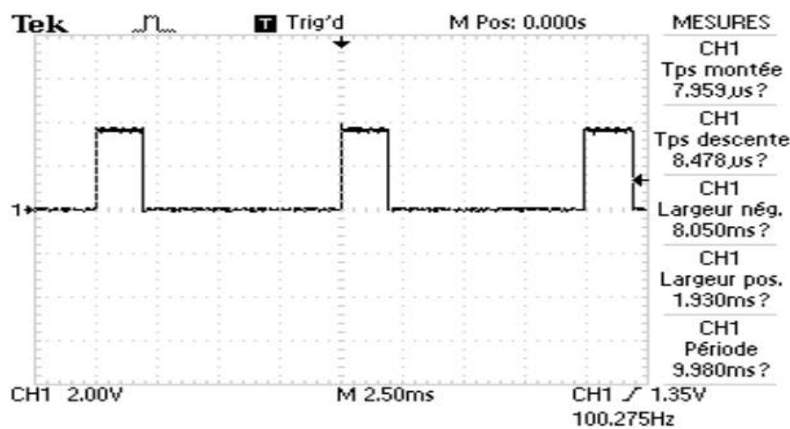


$(C_g='1') \text{and}(C_m='1') \text{and}(C_d='0')$



TDS 1012 - 17:31:10 12/05/2011

$(C_g='1') \text{and}(C_m='1') \text{and}(C_d='1')$



TDS 1012 - 17:44:12 12/05/2011

2. Sous-fonction liée à la fonction « Gestion de la propulsion »

Les entrées MchAv et MchAr sont réservées pour pouvoir commander à distance le déplacement en marche avant ou marche arrière du robot. Ces commandes sont bistables : un nouvel appui sur le bouton utilisé arrête le robot.

Exemple :

Appui sur la touche **Programme+** de la télécommande ☞ le robot avance

Nouvel appui sur la touche **Programme+** de la télécommande ☞ le robot s'arrête

D'autre part, pour éviter d'endommager le mécanisme, il faut que le robot puisse s'arrêter s'il rencontre un obstacle. Ensuite l'opérateur devra l'arrêter.

Pour la détection de l'obstacle, on utilise la détection de surintensité. Ceci évite de positionner un certain nombre de fin de courses autour du robot.

a) Modifier le fichier VHDL pour intégrer cette fonction "Gestion du déplacement".

On vérifiera d'abord la validité de la solution par simulation et ensuite sur maquette.

--Gestion du sens de déplacement

deplacement:process(Clk)

begin

if(clk' event and clk='1') then

case etat is

when Init=>

--étape initiale: Le robot Rafil est à l'arrêt et il est en attente d'une commande de marche avant

if(MchAv='0')and(MchAr='1')then

etat<=Avance;

--étape initiale: Le robot Rafil est à l'arrêt et il est en attente d'une commande de marche arrière

elsif(MchAv='1')and(MchAr='0')then

etat<=Recul;

end if;

when Avance=>

--étape Avance: Le robot Rafil est en marche avant et il rencontre un obstacle (surintensité du moteur)

if(Imot='1')then

etat<=Arret;

--étape Avance: Le robot Rafil est en marche avant et il est en attente d'une commande d'arrêt

elsif(MchAv='1')then

etat<=Init; --Retour à l'étape initial suite à un

arrêt normal

end if;

when Recule=>

--étape Recule: Le robot Rafil est en marche arrière et il rencontre un obstacle (surintensité du moteur)

if(Imot='1')then

etat<=Arret;

--étape Recule: Le robot Rafil est en marche arrière et il est en attente d'une commande d'arrêt

elsif(MchAr='1')then

etat<=Init; --Retour à l'étape initial suite à un

arrêt normal

end if;

when Arret=>

--étape Arrêt:Le robot Rafil est à l'arrêt et il est en attente d'une commande d'arrêt

if(MchAv='1')and(MchAr='1')then

etat<=Init; --Retour à l'étape initial suite à

une surintensité moteur

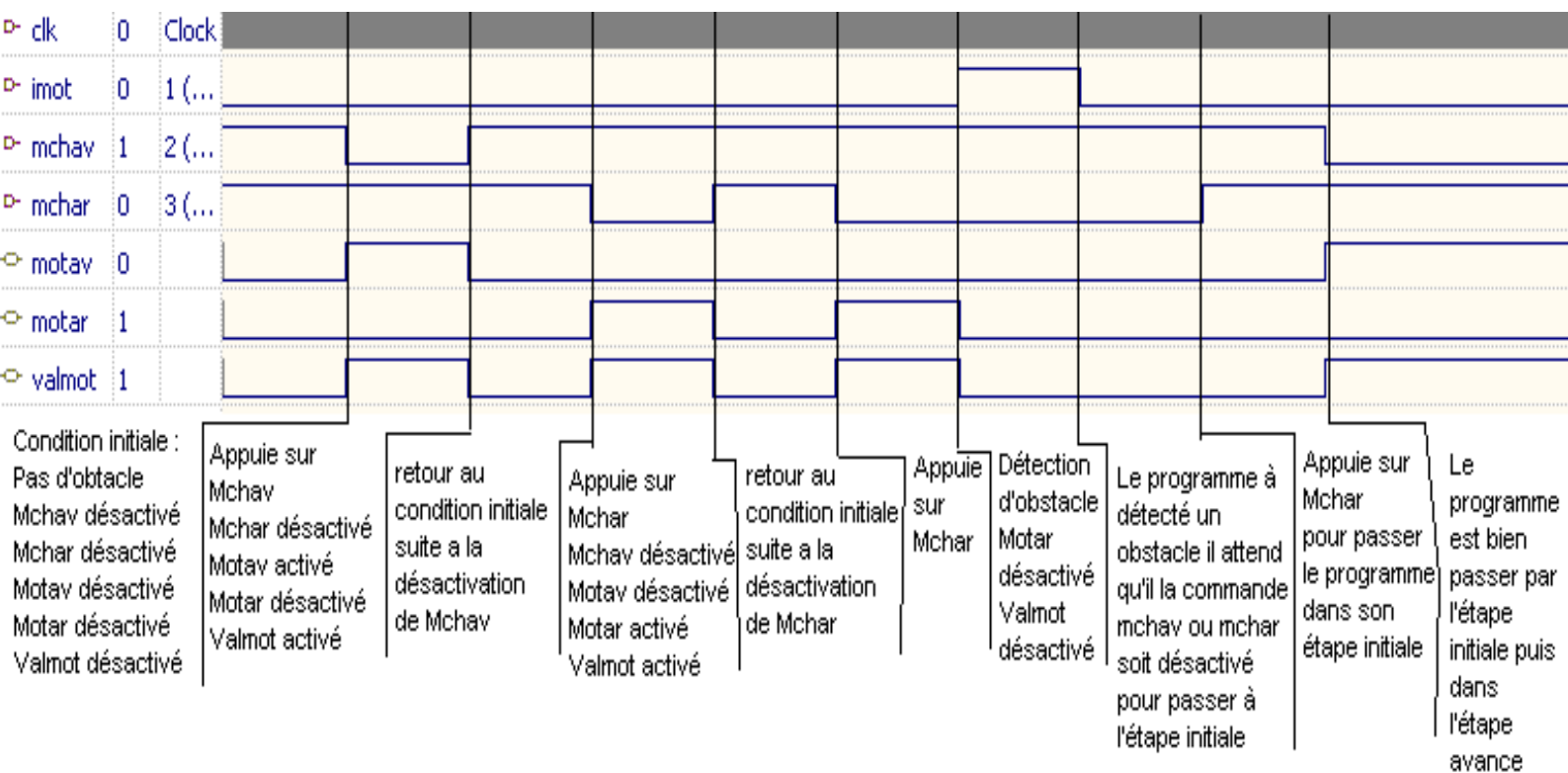
```

end if;
end case;
end if;
end process deplacement;

MotAv<='1'when(etat=Avance) else --Envoi du sens avant au moteur
'0';
MotAr<='1'when(etat=Recule) else --Envoi du sens arriere au moteur
'0';
Valmot<='1'when(etat=Recule or etat=Avance) else
'0';

```

Les boutons MchAv et MchAr sont en logique inverser. Lorsqu'on détecte une surintensité, grâce au signal booléen Imot, il faut réaliser un verrouillage sur le moteur et le mettre à l'arrêt. Pour déverrouiller la commande des moteurs, il faut mettre aux niveaux hauts l'état des signaux MchAv et MchAr.



b) Il est fort probable que le robot se déplace trop rapidement.

Comment peut-on diminuer la vitesse de rotation du moteur avec la même tension d'alimentation?

Valider votre solution pour une vitesse 2 fois plus faible.

Pour déboguer mon programme j'ai choisie une logique plus compréhensible pour moi mais qui ne correspond pas vraiment au cahier des charges (c'est ce programme qui a été vérifier car je n'ai pas eu le temps de régler le problème avant la structure précédente qui bugger avec la variation vitesse).

Lorsqu'on détecte une surintensité, grâce au signal booléen Imot, il faut réaliser un verrouillage sur le moteur et le mettre à l'arrêt. Pour déverrouiller la commande des moteurs, il faut mettre aux niveaux hauts l'état des signaux MchAv et MchAr. Pour réaliser ce verrouillage nous ferons appels à une constante booléen « Valid » alors que dans la version précédente le programme se verrouillait lorsqu'il était dans arrêt.

--Gestion du sens de déplacement

```
deplacement:process(Clk)
```

```
begin
```

```
  if(clk' event and clk='1') then
```

```
    --on regarde les Etats des signaux MchAV et MchAR
```

```
    --le robot est à l'arrêt
```

```
    if(MchAv='1' and MchAr='1') then
```

```
      MotAv<='0';
```

```
      MotAr<='0';
```

```
    end if;
```

```
    if (Valid='0') then -- verrouillage de la surintensité
```

```
      --le robot est à l'arrêt et l'on appuie sur un bouton
```

```
      --marche avant enclenché
```

```
      if(MchAv='0' and MchAr='1' and MotAv='0' and MotAr='0')then
```

```
        MotAv<='1';
```

```
        MotAr<='0';
```

```
      end if;
```

```
      --marche arrière enclenché
```

```
      if(MchAv='1' and MchAr='0' and MotAv='0' and MotAr='0')then
```

```
        MotAv<='0';
```

```
        MotAr<='1';
```

```
      end if;
```

```
      --le robot est en marche est on ne touche à aucun bouton
```

```
      --le robot continue d'aller en avant
```

```
      if(MchAv='0' and MchAr='0' and MchAv='1' and MotAr='0')then
```

```
        MotAv<='1';
```

```
        MotAr<='0';
```

```
      end if;
```

```
      --le robot continue d'aller en arrière
```

```
      if(MchAv='0' and MchAr='0' and MotAv='0' and MotAr='1')then
```

```
        MotAv<='0';
```

```
        MotAr<='1';
```

```
      end if;
```

```

--on appuie sur les 2 boutons même temps arrêt
if(MchAv='1' and MchAr='1')then
    MotAv<='0';
    MotAr<='0';
end if;
end if;
end process deplacement;

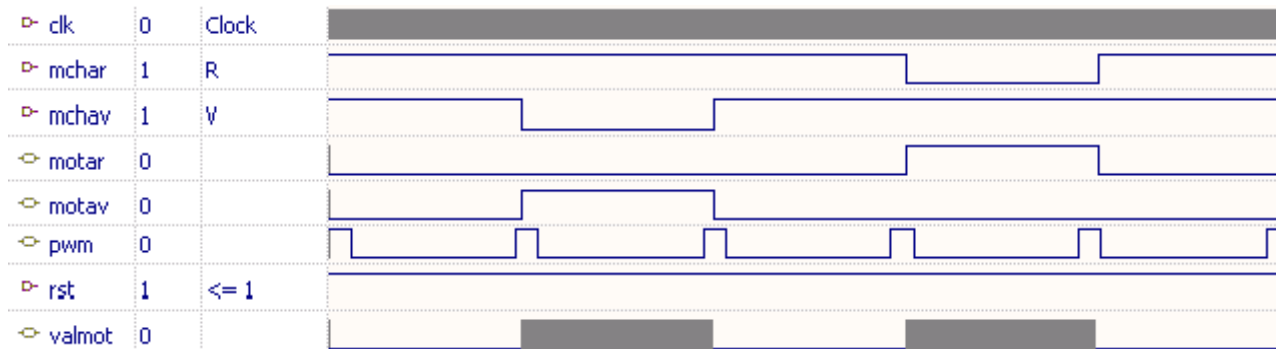
--compteur pour diminuer la vitesse du moteur
process(Clk)
BEGIN
    if(clk' event and clk='1') then
        Comp_mod_16<=Comp_mod_16+1;
    end if;
end process;

Valid<='1' when (Imot='1')else --vérouillage des moteurs
    '0' when (MchAv='1' and MchAr='1'); --dévérrouillage des moteurs

-- Gestion de la vitesse
-- vitesse élevé car le robot doit aller tout droit
vitesse<="11" when (((Cg='1')and(Cm='0')and(Cd='1'))or((Cg='0')and(Cm='0')and(Cd='0'))))else
-- vitesse moyenne car le robot tourne soit à droite soit à gauche
"10" when (((Cg='0')and(Cm='1')and(Cd='1'))or((Cg='1')and(Cm='1')and(Cd='0'))))else
-- vitesse lente car le robot est sorti de la piste
"01" when ((Cg='1')and(Cm='1')and(Cd='1'))else
"00"; -- arrêt

-- autorisation pour une vitesse élevée
Valmot<='1'when((MotAv='1' or MotAr='1') and vitesse<="11" and Comp_mod_16>"0001") else
-- autorisation pour une vitesse moyenne
'1'when((MotAv='1' or MotAr='1') and vitesse<="10" and Comp_mod_16>"0010") else
-- autorisation pour une vitesse lente
'1'when((MotAv='1' or MotAr='1') and vitesse<="01" and Comp_mod_16>"0111") else
'0'; -- pas d'autorisation
end archvhdl;

```



L'entrée EN (ENable) peut être utilisée pour faire varier la vitesse du moteur. Si EN=1, on autorise le fonctionnement. Si EN=0, tous les transistors sont bloqués (arrêt du moteur). En alternant rapidement les deux états sur EN (grâce à un signal MLI) on obtient une vitesse du moteur plus faible, en fonction du rapport cyclique de la MLI. Donc pour avoir une vitesse de 50% avec la même tension d'alimentation il faut écrire :

```
Valmot<='1'when((MotAv='1' or MotAr='1') and Comp_mod_16>"1000") else
    '0';
```

Mais nous voulons que notre robot aille le plus vite possible c'est pour cela que nous avons choisie de rouler à 93,75% pour aller tout droit, dans les virage à 87,5% et 56,25% lorsque le robot qui la piste.

Mon entête est :

```
-- Titre : Vhdl pour le robot rafil
-- Nom : RICARD Alexandre
-- Groupe : B
-- Etablissement : IUT de Nantes, Departement GEII
```

```
-- Definition des bliblioteques
```

```
library ieee;
use ieee.std_logic_1164.all;
USE work.STD_ARITH.all;
```

```
--Definiton des entrées/sorties
```

```
entity vhdl is
```

```
    port( Clk:in std_logic;
          Rst:in std_logic;
          Cd,Cm,Cg:in std_logic;
          MchAv,MchAr,Imot:in std_logic;
          PWM:out std_logic;
          Valmot,MotAv,MotAr:out std_logic;
          Valid:buffer std_logic;    --besoin seulement pour le deuxième cas
          Leds: out std_logic_vector(7 downto 0));
```


Attribute pin_numbers of vhdl: entity is --Attribution des numéros entrées/sorties

```
"Clk:13 "&
"Cg:4 "&
"Cd:2 "&
"Cm:3 "&
"PWM:42 "&
"rst:14 "&
"MchAv:6 "&
"MchAr:8 "&
"Imot:33 "&
"MotAv:37 "&
"MotAr:36 "&
"ValMot:40 "&
"Leds(7):30 Leds(6):29 Leds(5):28 Leds(4):26 Leds(3):25 Leds(2):24 Leds(1):21
Leds(0):20";
```

end vhdl;

architecture archvhdl of vhdl is

-- Signaux intermediaires facilitant la description par table de verite

signal T: std_logic; -- indicateur de la durée 10 ms

signal Vnum: std_logic_vector(9 downto 0); -- valeur numérique évoluant en dent de scie

signal Ton: std_logic_vector (7 downto 0); -- direction souhaité

signal vitesse: std_logic_vector (1 downto 0); -- vitesse du robot

signal Comp_mod_16: std_logic_vector (3 downto 0); --compteur pour diminuer la vitesse du
moteur

-- Mise en place des états que pourra prendre notre robot Rafil

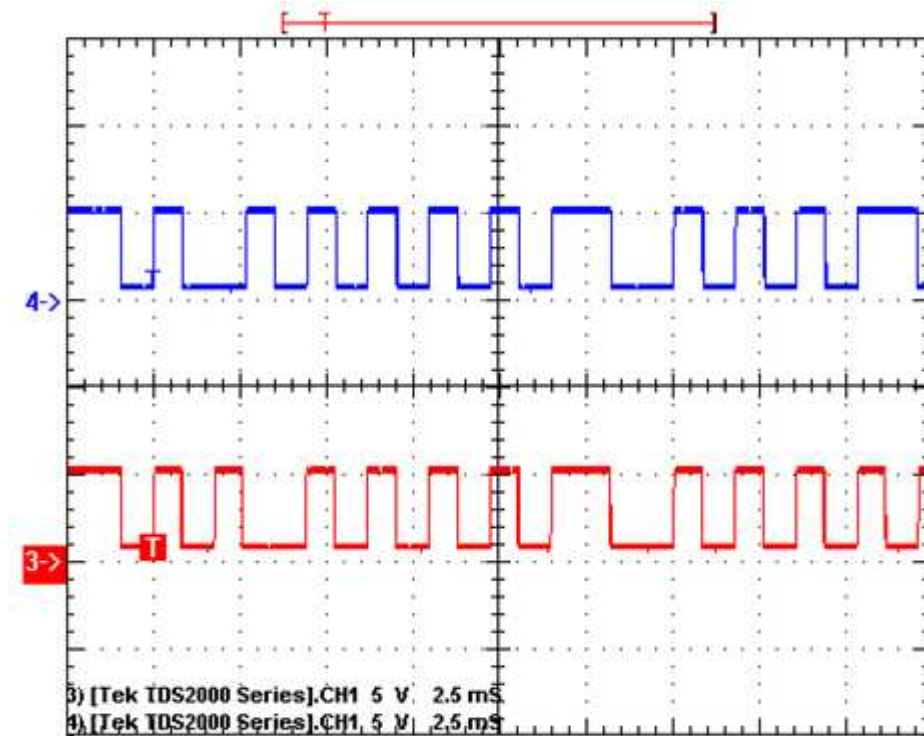
--type etats is(Init,Avance,Recul,Arret);

--signal etat:etats;

c) Si le temps le permet, analyse de la trame RC5

En mode monocoup, relever les deux trames RC5 qui permettent de démarrer le robot.

Décoder la trame à l'aide de la documentation en ligne



Une trame en code RC5 se compose d'une suite de 14 bits et sa construction est la suivante :

- - 2 bits de départ
- - 1 bit de basculement
- - 5 bits d'adressage du système
- - 6 bits d'instruction

Les 2 bits de départ sont utiles pour ajuster le niveau de la commande automatique du gain AGC dans le circuit intégré de réception.

Le bit de basculement indique une nouvelle transmission de données.

Sa valeur change à chaque nouvelle activation d'une touche afin de distinguer une nouvelle pression d'une pression continue sur la même touche.

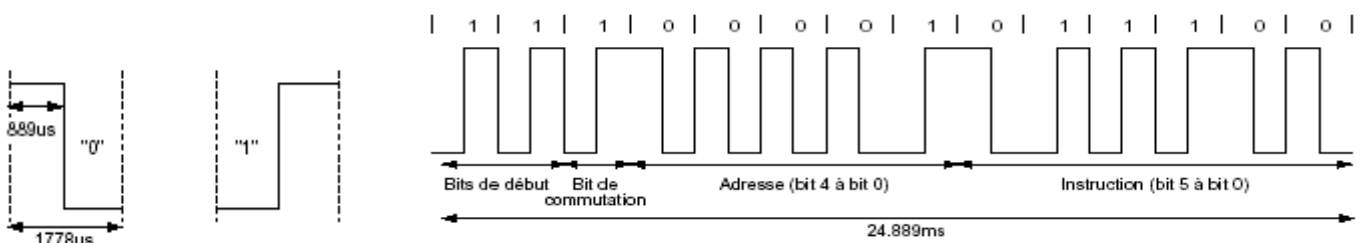
Les 5 bits suivants déterminent l'adresse du dispositif devant réagir à la commande.

Nous avons donc $2^5 = 32$ groupes d'adressage.

L'instruction destinée à l'appareil est codée dans les 6 derniers bits.

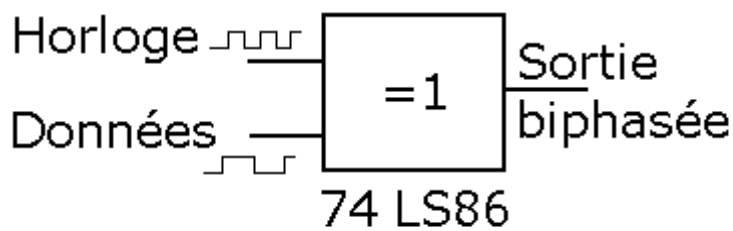
Ce qui donne $2^6 = 64$ instructions.

Chaque bit ayant une longueur de 1,778 ms, une trame rc5 dure alors $14 \times 1,778 = 24,892$ ms.

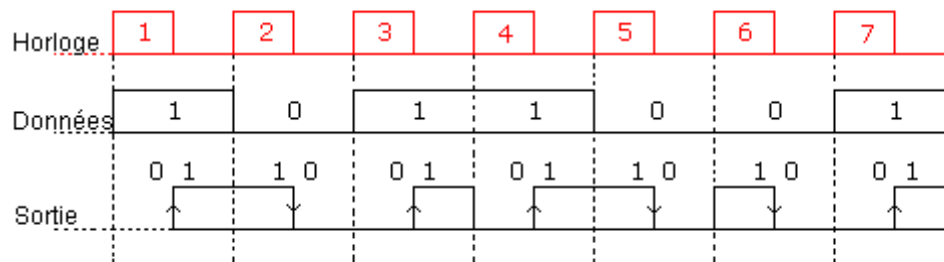


Les bits du code RC5 sont codés en biphase (codage Manchester), c'est à-dire qu'un bit est composé de 2 demi-bits alternés. La combinaison bas/haut caractérise un bit positionné à 1 et la combinaison haut/bas un bit remis à 0.

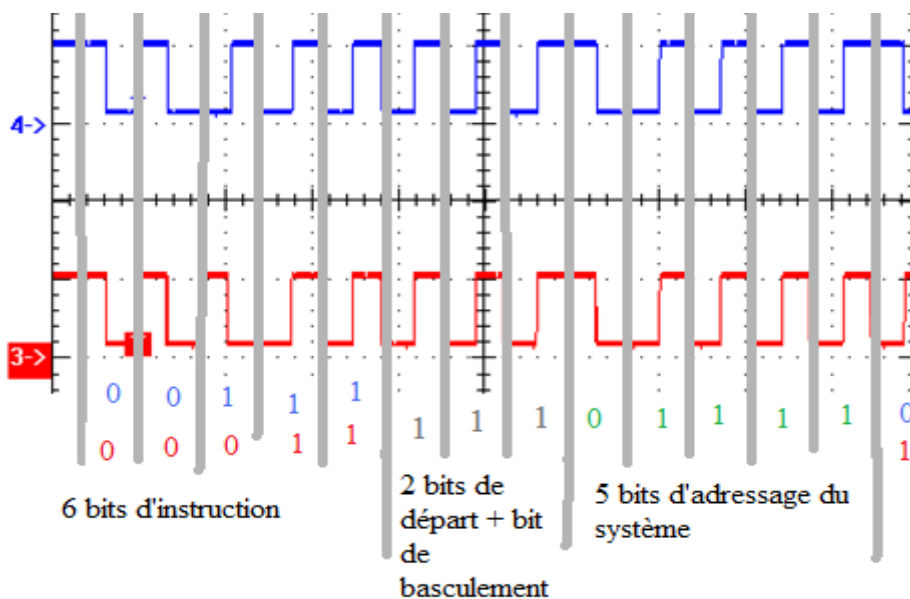
Le schéma électronique du générateur de code biphasé est représenté ci-dessous :



et le chronogramme du codage biphasé est disponible sur l'image suivante :



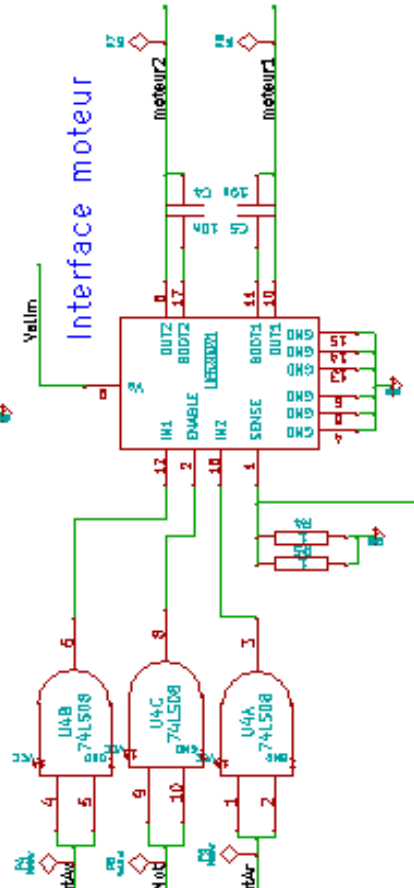
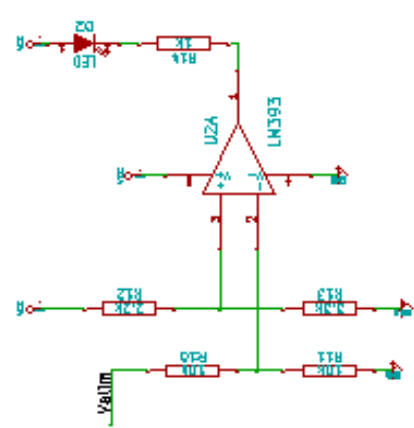
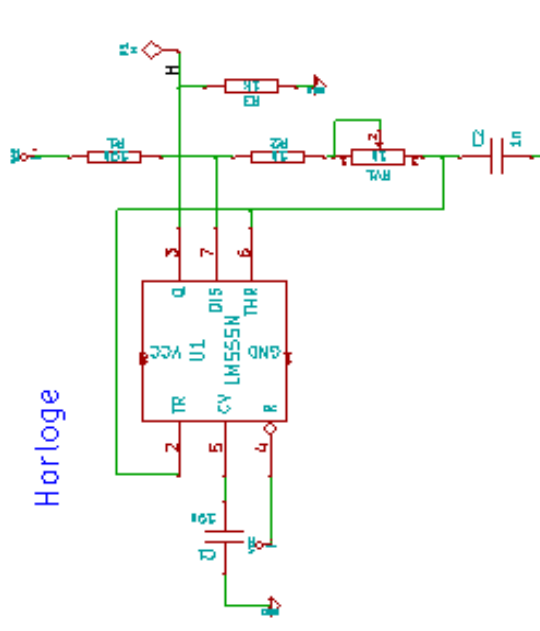
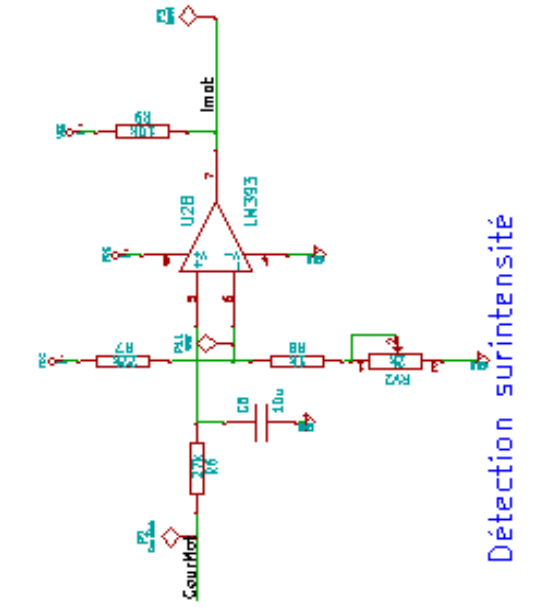
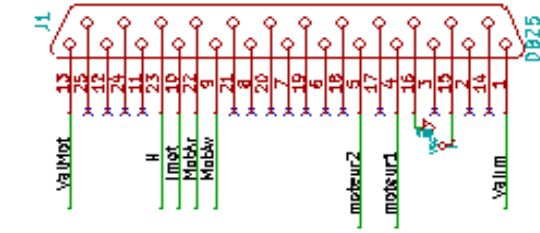
Sur la capture d'écran on voit 25ms donc on voit bien toute une trame. On remarque que le signal envoyé est :



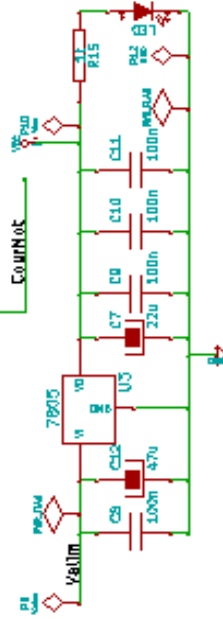
Pour analyser ces deux trames RC5 codées en codage Manchester, nous repérons d'abord les 2 bits de départ et le bit de basculement qui ne change pas selon l'appareille ou l'instruction on doit donc repérer cette suite de bits 010101. Une fois les 2 bits de départ et le bit de basculement repérés nous pouvons savoir les 5 bits d'adressage. Nous remarquons que les 5 bits sont identiques pour la marche avant et arrière ce qui confirme notre analyse car on envoie à la même adresse, seule les 6 derniers bits d'instructions diffèrent. Il nous reste plus qu'à repérer les 6 bits d'instruction, les 5 derniers sont situés avant les 2 bits de départ car nous avons mal positionné notre trame et le premier est situé après les 5 bits d'adressage système. Le message envoyé signifie :

- Donné pour l'adresse : 15 cette adresse est réservée
- Donné pour la marche avant(4) : 7
- Donné pour la marche arrière(3) : 35

VIII. Création de la carte électronique



Détection batterie faible



Régulation de tension 5V

RICARD Alexandre	
Fich. Carte puissance.sch	
Sheet /	
Title: Carte de puissance	
Size: A4	Date: 21 mai 2011
Rev: 1	Id: 1/1
KICad E.D.A. eschema (2010-05-05 BZR 2356)-stable	

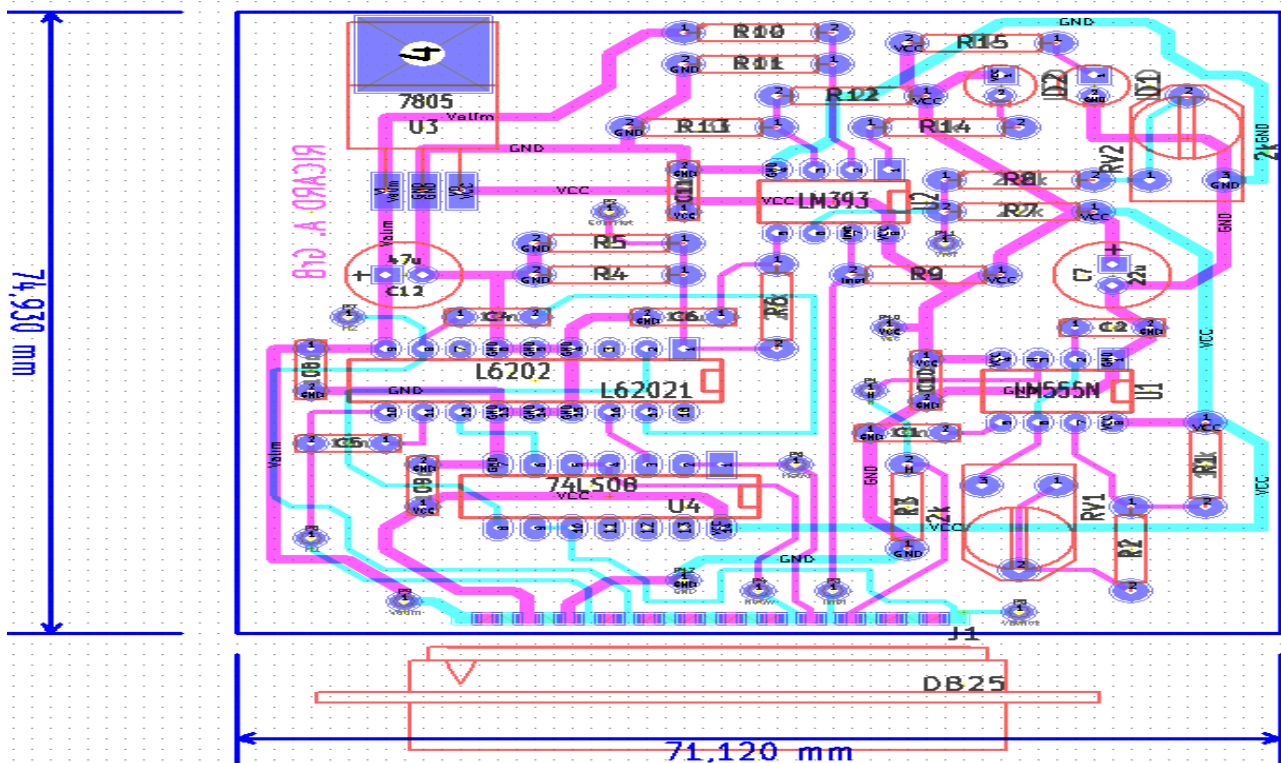
Nous avons d'abord créer le schéma structurel sur Kicad d'après les recherches effectuer précédemment. Pour protéger le CPLD en cas de défaillance du L6202, on a ajouté des portes logiques entre les deux composants sur les broches « ValMot », « MotAv » et « MotAr ». Le brochage du connecteur DB25 est imposé : il faut **impérativement** le respecter pour pouvoir faire les essais en fin de projet. Nous avons vérifiés les règles électriques (E.R.C.), puis nous avons généré la netliste.

1	C1 - 10n : C2	13	D2 - LED : LEDV	25	P10 - Vcc : PINTST	37	R10 - 10k : R4
2	C2 - 1r : C2	14	J1 - DB25 : DB25_CI_ETUD	26	P11 - Vref : PINTST	38	R11 - 10k : R4
3	C4 - 10n : C2	15	L62021 - L6202 : 18DIP-ELL300	27	P12 - GND : PINTST	39	R12 - 2.2k : R4
4	C5 - 10n : C2	16	P1 - H : PINTST	28	R1 - 10k : R4	40	R13 - 3.3k : R4
5	C6 - 10u : C2	17	P2 - CourMot : PINTST	29	R2 - 1k : R4	41	R14 - 1k : R4
6	C7 - 22u : C1VB	18	P3 - Imot : PINTST	30	R3 - 1k : R4	42	R15 - 1k : R4
7	C8 - 100n : C2	19	P4 - MotAv : PINTST	31	R4 - 1 : R4	43	RV1 - 1k : RV2X4
8	C9 - 100n : C2	20	P5 - ValMot : PINTST	32	R5 - 1 : R4	44	RV2 - 2k : RV2X4
9	C10 - 100n : C2	21	P6 - MotAr : PINTST	33	R6 - 27k : R4	45	U1 - LM555N : 8DIP-ELL300
10	C11 - 100n : C2	22	P7 - M2 : PINTST	34	R7 - 22k : R4	46	U2 - LM393 : 8DIP-ELL300
11	C12 - 47u : C1VB	23	P8 - M1 : PINTST	35	R8 - 200 : R4	47	U3 - 7805 : LM78XX
12	D1 - LED : LEDV	24	P9 - Valim : PINTST	36	R9 - 10k : R4	48	U4 - 74LS08 : 14DIP-ELL300

Enfin nous avons réalisé le typons. Pour ceci, il faut d'abord commencer par créer les dimension de la carte qui est imposée (identique à la carte supportant le CPLD : RAFIL-CY, hauteurs max 79mm, largeur: 73mm). Nous avons une largeur de 71,120mm et une hauteur de 74,930mm donc le cahier des charges est respecté. Ensuite nous devons placer les supports de CI (Circuits Intégrés). Puis les condensateurs de découplages allant avec ces CI. Ensuite, il faut placer le reste des composants pour obtenir une carte facile à router.

Puis, vient le routage a proprement parlé : relier les composants un à un en créant des pistes électriques sur les deux faces (les circuits en bleu coté composants, les circuits en rose coté cuivre) de la carte.

Le routage final est présenté ci-dessous. Les tracés plus épais correspondent à la tension d'alimentation ainsi qu'à la masse. Les autres tracés correspondent aux différentes fonctions effectués par la carte. Nous n'avons pas fait de plan masse pour éviter les liaisons entre les piste dû à la mauvaise impression du typons et de ainsi débbugger plus facilement notre carte.



Après avoir effectué le routage, il faut imprimer un typon (impression des deux faces de la carte sur un papier transparent). Ces négatifs sont attachés entre eux et synchronisés au niveau des perçages. On pose le dispositif sur la carte à imprimée ayant une surface sensible aux UV. La partie qui est insolé se détache dans un bain de solvants, puis on met la carte avec les surfaces a détaché dans un révélateur qui va enlever le cuivre non nécessaire sur la carte. Enfin on passe la carte dans une solution qui enlève le produit non insolé. On passe la carte gravé dans un bain d'étain pour protéger la carte de l'oxydation.

On vérifie qu'aucune piste ne se touche et qu'elles soient bien continues avec un multimètre. On perce les trous pour la mise en place des composants en faisant attention au diamètre des pattes des composants (1,2mm pour les potentiomètres, les points test, les bornes de U3 et les 4 trous des cotés, les autres trou seront percé en 0.8mm).

Nous soudons ensuite nos composants du plus petit au plus grand en fonction la nomclature situé ci-dessous.

Ceci fait, la carte est maintenant terminée ! Il va donc falloir procéder aux différents tests de la carte.

Quantité	Référence	Valeur	Désignation
	Condensateurs		
1	C7	22uF	Electrolytique
1	C12	47uF	Electrolytique
1	C6	10uF	Plastique
1	C2	1nF	Plastique
3	C1, C4, C5	10nF	Plastique
4	C8, C9, C10, C11	100nF	Plastique
	Résistances		
2	R4, R5	1Ω	Couche carbone
4	R2, R3, R8, R14, R15	1kΩ	Couche carbone
1	R12	2.2kΩ	Couche carbone
1	R13	3.3kΩ	Couche carbone
4	R1, R9, R10, R11	10kΩ	Couche carbone
1	R7	22kΩ	Couche carbone
1	R6	27kΩ	Couche carbone
	Résistances variables		
1	RV1	1kΩ	Monotour horizontal
1	RV2	2kΩ	Monotour horizontal
	Semi-conducteur		
1	D1	Verte	DEL Ø2
1	D2	Rouge	DEL Ø2
	Circuits intégrés		
1	U4	74LS08	4 portes AND à deux entrées
1	U2	LM393	2 comparateurs
1	U1	LM555N	Multivibrateur autonome astable
1	U3	LM2940T-5	Régulateur intégré linéaire
	Connecteurs / Supports		
12	Points test		Barrette sécable
1	L6202		Support CI tulipe / 16b
2	U1, U2		Support CI tulipe / 14b
1	U4		Support CI tulipe / 8b
1	U3		Régulateur de tension
2	DB25		Connecteur

IX. Tests, mesure et validation

Pour tester le fonctionnement du circuit, on adopte une démarche pas à pas. Chaque test devra correspondre aux attentes théoriques vues dans l'analyse structurale. Note : on teste la continuité des pistes avant d'implanter les composants.

Afin de réaliser les différents tests, il faut se munir d'un voltmètre, d'un oscilloscope relié à des pointes de touches, d'une plaque de test, d'une alimentation stabilisée et d'un robot rafil.

Tout d'abord, à l'aide du voltmètre réglé en mode test diode, il faut vérifier la continuité des pistes.

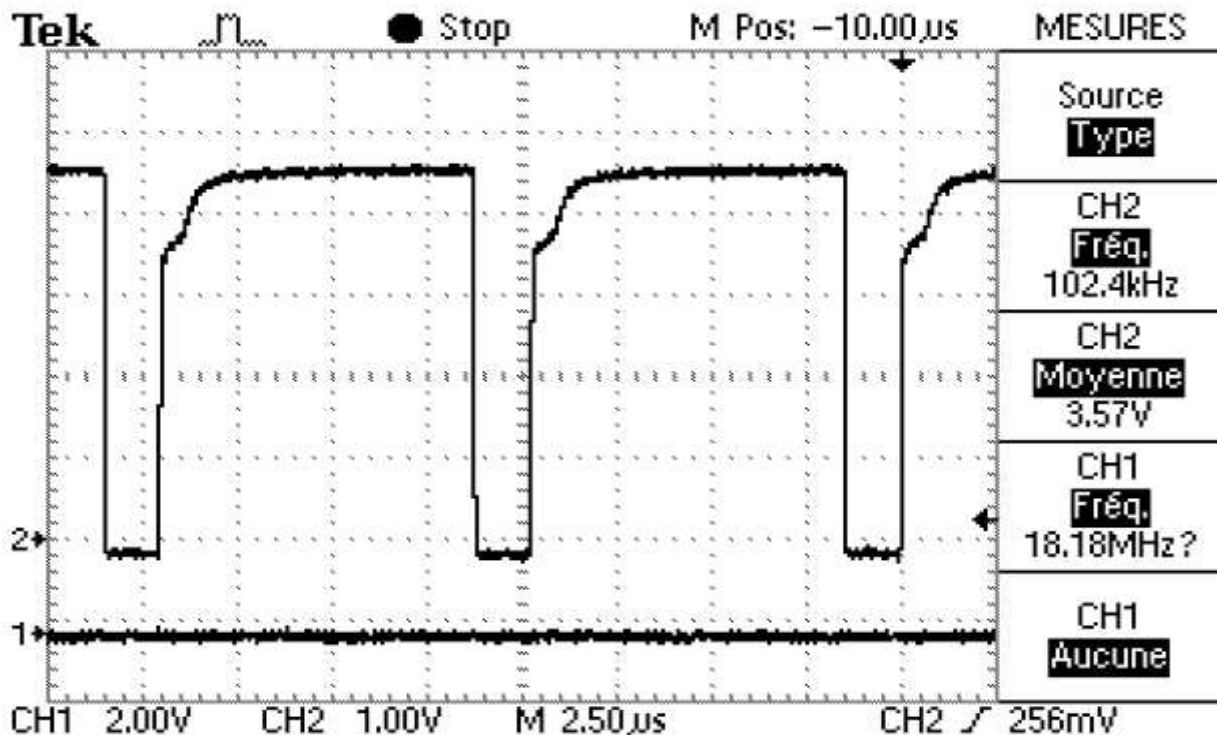
Puis, On met la carte sous tension en continu entre 6V et 15V, sans aucun circuit intégré (excepté le régulateur). Cette tension correspond à VBAT(6V et 12V). On vérifie alors la présence de cette tension au point test, au pin 9 du L6202 et au LM2940T-5. On vérifie aussi la présence de la masse à des points importants de la carte tels que le pin 2 du LM2940T-5 ainsi qu'aux masses des circuits intégrés. A ce niveau, l'alimentation est correctement établie.

On observe alors le fonctionnement du régulateur linéaire intégré LM2940T-5. Sur le pin 3, le composant doit fournir une tension constante de 5V pour tout VBAT € [6 ; 12]. On obtient VCC = 4,98V. Le composant fonctionne donc correctement et le cahier des charges est respecté.

On vérifie enfin que VCC alimente bien les circuits intégrés et les autres points importants de la carte. La fonction alimentation est alors validée.

Une fois ceci fait, il faut placer le NE555 sur son support, et vérifier à l'aide de l'oscilloscope le signal carré, ainsi que la fréquence obtenue qui doit correspondre à celle demandées.

On détermine alors la plage de fréquence disponible : $88\text{kHz} < H < 119\text{kHz}$. On règle H à 102,4kHz pour tout le reste du travail. Le signal H est donc le signal ci-après :

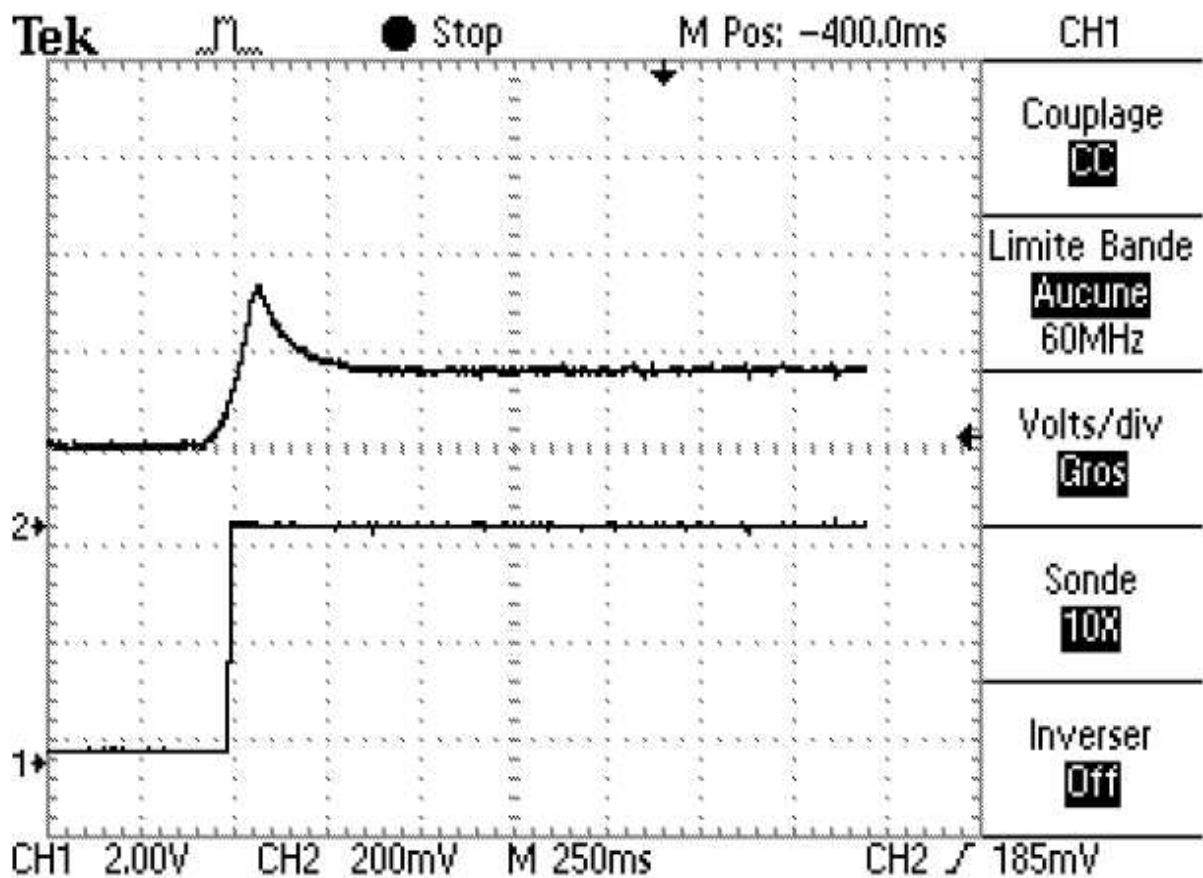


Le prochain test est celui de la détection de batterie faible, pour cela, il faut, à l'aide de l'alimentation stabilisée, générer une tension inférieure à 6V. Il faut regarder si la LED s'allume dès que le seuil est <6V et s'éteint lorsque le seuil est >6V (dans notre cas la diode est allumée quand la tension est supérieure à 6V et s'éteint lorsqu'elle est inférieure à 6V, cela est dû à un inversement de V+ et de V-).

Le test suivant est celui de la MchAv, la MchAr, de Valmot, du MotAv et du MotAr. Pour cela nous vérifions que lorsque MchAv est actif MotAv est à 1 et lorsqu'elle est inactive MotAv est à 0 sur les pins 4 et 5 du 74LS08. Nous faisons la même vérification avec MchAr et MotAr sur les pins 1 et 2 du 74LS08. Nous vérifions ensuite sur les pins 9 et 10 du 74LS08, Valmot est bien à 1 lorsque MchAv ou MchAr sont activés.

Une fois cela fait nous vérifions que les informations sur les pins 12 (MotAv/IN1), 2 (Valmot/Enable) et 16 (MotAr/IN2) sont correctes. Si toutes ces informations sont correctes alors lorsque MchAv est actif nous devons voir le moteur tourner dans un sens et lorsque MchAr est actif le moteur tourne dans l'autre sens.

Le dernier test est celui de la détection de surintensité, on utilise pour cela la maquette de test. On met en marche le moteur, on le bloque à la main et on vérifie que la sortie du comparateur (pin 7 ou point test « Imot ») bascule à 1 à ce moment. Dès que l'on relâche le moteur, la sortie doit repasser à 0. On règle la position de RV2 pour pouvoir observer ce phénomène. Le fonctionnement est correct, conforme au cahier des charges, la fonction est validée. Nous regardons aussi la patte 5 pour voir le courant moteur et ainsi régler la patte 6 pour détecter la surintensité.



Enfin, tester le fonctionnement global de la carte ainsi que le programme en la positionnant sur un robot et en conditions de tests réels, afin de bien tester les différentes fonctions. Notre robot suit bien la ligne ovale pendant plusieurs minutes lorsque l'on appuie sur MchAv, le robot recule bien lorsque on appuie sur MchAr et le robot s'arrête après quelque ms une fois qu'il a rencontré un obstacle.

Le test final est réalisé en 4 temps :

- Donner un algorithme et un programme de calibrage qui permet, en supposant que les robots soit tous réglés mécaniquement, d'incrémenter ou décrémente de un le pas angulaire du robot à chaque front sur un des capteurs.
- le robot doit suivre une ligne blanche rectiligne sur une distance de 1 mètres.
- le robot doit suivre une ligne blanche en forme ovale.
- Le robot doit s'arrêter lorsqu'il rencontre un obstacle.

Extension possible de l'application si le temps le permet :

- Remplacer le RAFIL par une 907 qui roule à 30km/h et adapter le programme pour faire suivre la piste à la voiture à la plus grande vitesse possible.

Nous avons validé notre carte et notre programme avec les 4 temps du test final mais nous n'avons pas eu le temps de faire l'extension.

X. Conclusion

Ce projet a été difficile à réaliser, tant que sur le programme VHDL que sur la création de la carte. Les quelques problèmes rencontrés ont été contournés. Mais une fois réalisée, cette carte va servir à « piloter » le RAFIL, cela va lui permettre de réaliser toutes les fonctions précédemment expliquées.

Lors de la création de cette carte, nous avons eu quelques problèmes :

- Lors de la préparation, il a fallu déterminer des valeurs de résistances. Il s'est avéré que quelques-unes de ces valeurs étaient fausses, il a donc fallu recalculer.
- Il a fallu décâbler des composants sur la carte, il y a donc quelque paté d'étain
- Peu de robots à disposition afin de tester le programme VHDL.
- Pas assez de cartes tests pour tous les étudiants.
- Temps très court, pas le temps de tout finir.
- Connecteur mal centré donc la carte est un peu trop grande d'un côté.

Malgré tout ces problèmes, l'essentiel a été fait, tout a été validé, nous avons appris beaucoup de choses, notamment lors des phases de résolution des problèmes et nous avons pu utiliser les connaissances apprises précédemment. Nous avons réussi à faire face aux différents problèmes rencontrés afin de valider le fonctionnement général du robot.

XII. Webographie

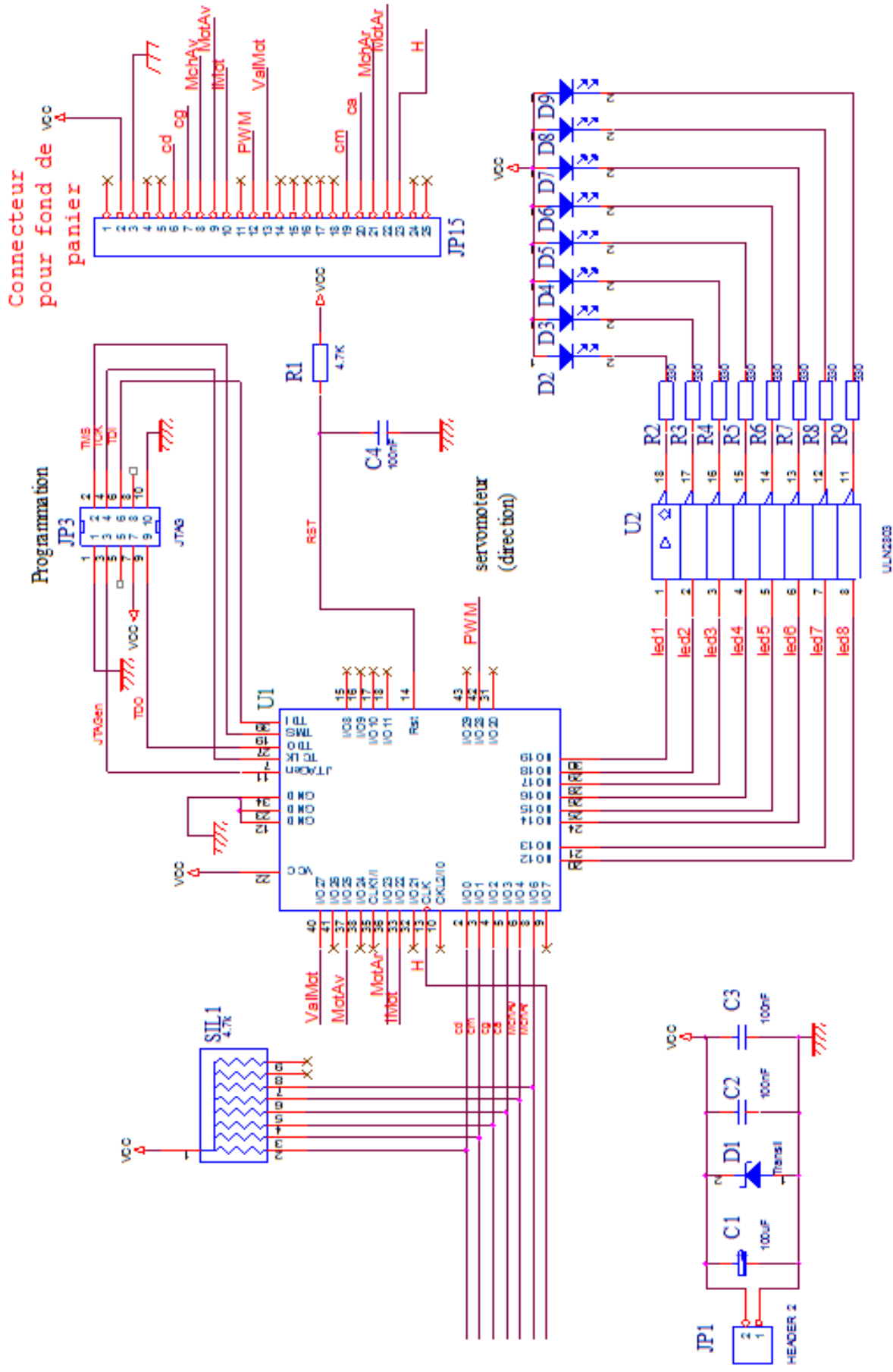
<http://www.electronique-radioamateur.fr/elec/schema/diviseur-tension.php>

Site sur le réseau expliquant les différentes fonctions du robot rafil

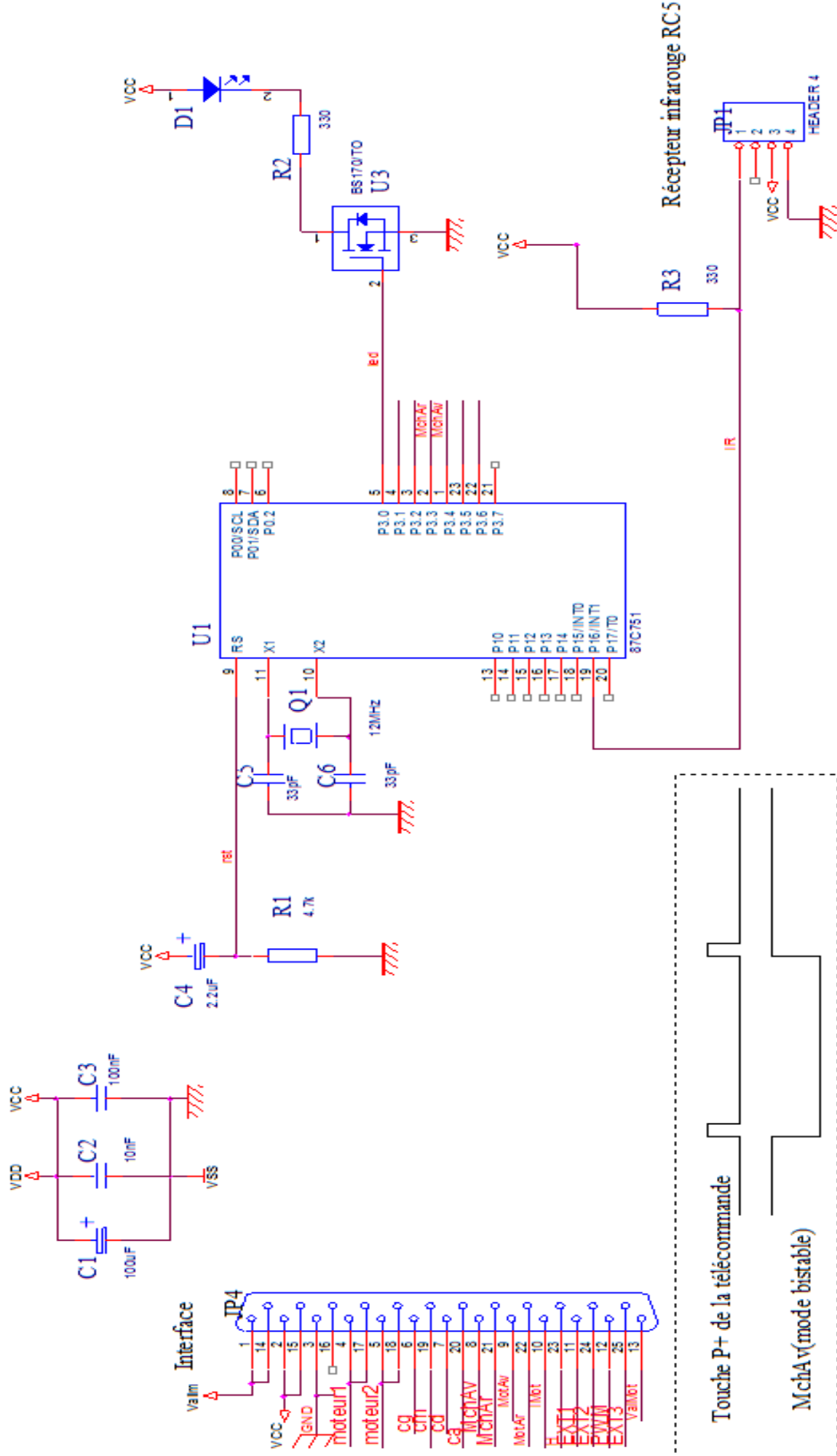
Documentation des différents composants sur le réseau

XI. Annexes

1. Schéma structurel de la carte fille n°2 (Commande)



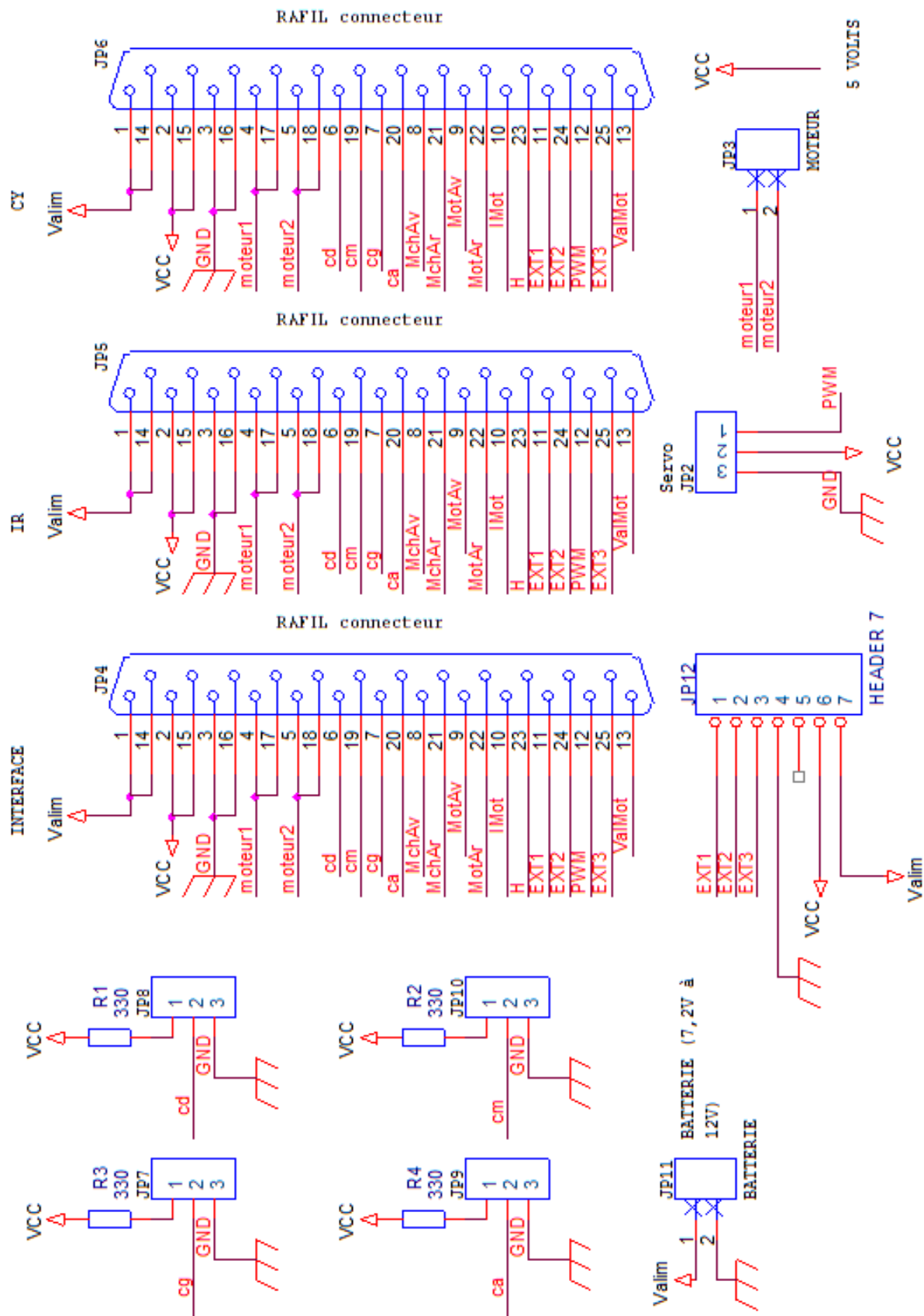
2. Schéma structurel de la carte fille n°3 (Infrarouge)



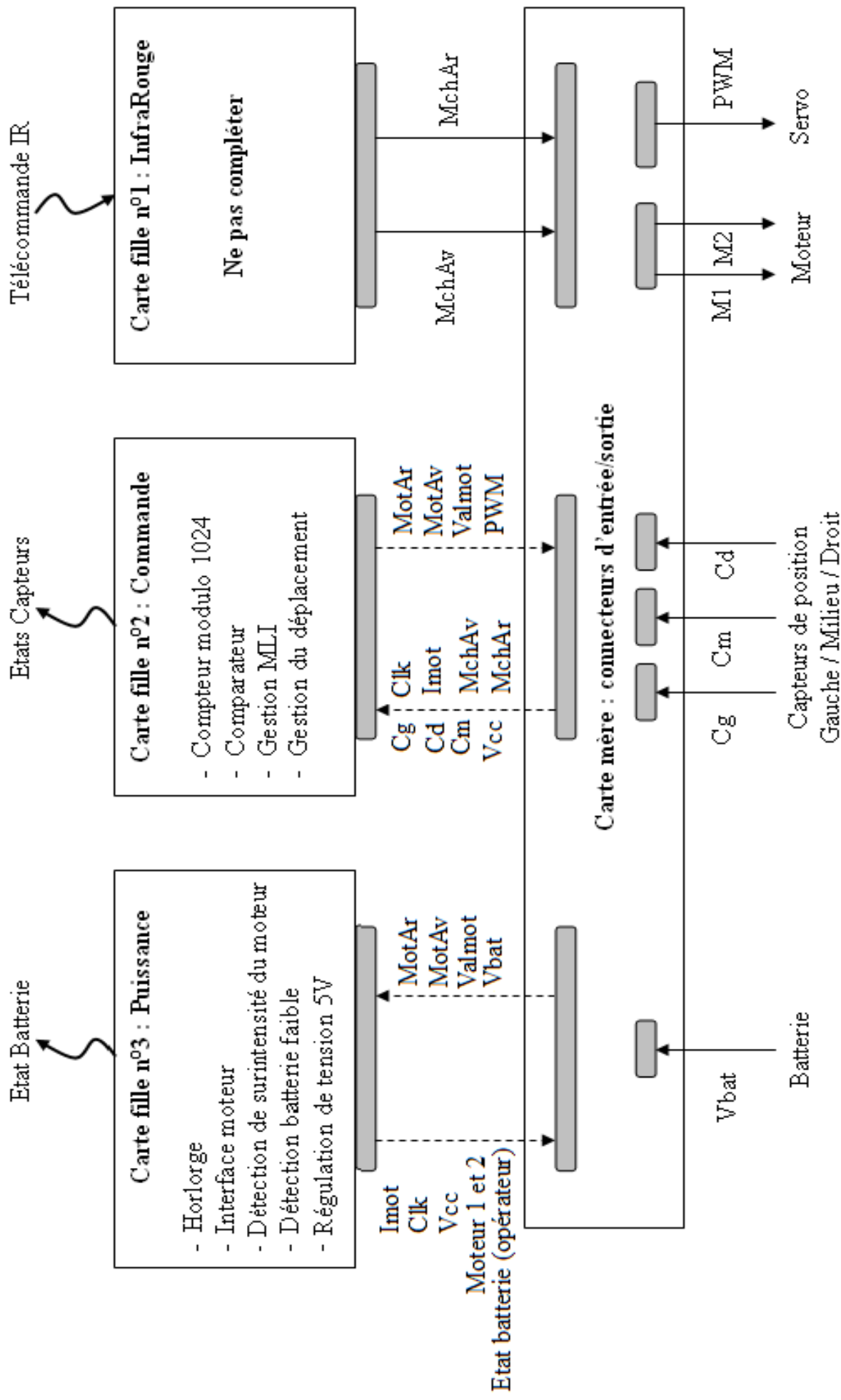
Touche P+ de la télécommande

MchAV(mode bistable)




3. Schéma structurel de la carte mère (connecteurs d'entrées/sorties)



4. Implantation des différentes fonctions / Echanges des informations entre les cartes



5. Organisation de mon étude

Séance	Travail effectué	Travail à réaliser à la séance suivante	Signature de l'enseignant à chaque séance
1	ABS	ABS	
2	Analyser les différentes fonctions que composera la carte et les testers sur Pspice.	Faire la carte sur Kicad et la finir pour la séance prochaine pour pouvoir imprimer le typon.	
3	Kicad OK. Typons imprimer.	Souder les composants sur la carte puissance et la tester.	
4	Toute les pistes sont bonne. Tout les composants souder. Horloge vérifier, détection batterie faible vérifier.	Finir de tester la carte puissance et la faire valider. Ecrire le programme du compteur 1024. Ecrire le programme de la gestion de PWN.	
5	Carte valider. Compteur 1024 et gestion de PWN fait.	Ecrire le programme Gestion_MLI	
6	Structure de Gestion_MLI faite. Mise en place des valeurs Ton théorique.	Trouver les bonnes valeur de Ton pour que le robot fonctionne correctement.	
7	Ton pour aller tout droit, à gauche et à droite trouver.	Tester la marche avant et la marche arrière. Tester la réduction de vitesse	
8	Marche avant et arrière OK. Modulation de la vitesse Ok.	Voir si le robot suis bien l'ovale. Voir si le robot s'arrête lorsqu'il y a une surintensité. Faire vérifier le bon fonctionnement par un professeur.	
9	Le robot suis l'ovale, il s'arrête lors d'un surintensité. Le programme à donc été valider.	Finir le compte rendu. Rendre le compte rendu	