Objet Connecté
3 novembre 2024

Master 1 1eeea option Electronique

Moniteur Cardiaque

Élèves :
Prénom 1 Nom 1
Prénom 2 Nom 2
Prénom 3 Nom 3
...

Enseignants
O. Bernal
E. Peuch
H. Kaouach
H.C. Seat
G. Prigent
...

3 novembre 2024

Table des matières

1 La conversion analogique-numérique

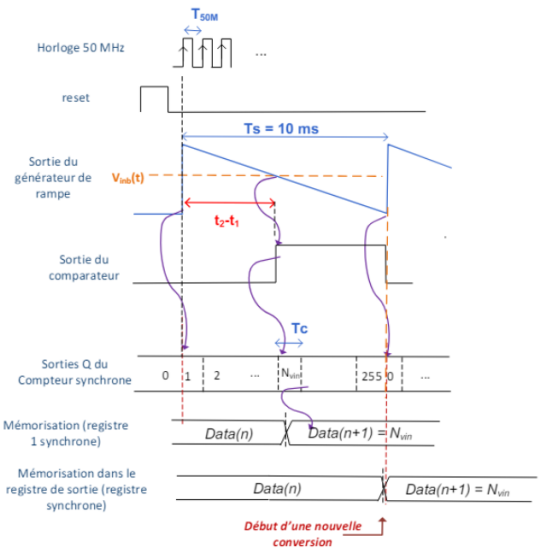

FIGURE 1 – chronogramme expliquant le pricipe de fonctionnement de la CAN

La partie précédente nous laisse avec un signal bien filtré, mais pour pouvoir bien l'exploiter et effectuer des tests statistiques sur ce dernier, il faut le convertir dans le domaine numérique. Dans notre application, nous optons pour une conversion avec une résolution de 8 bits.

Nous commençons tout d'abord par échantillonner notre signal d'entrée grâce à un ECB (échantillonnage et blocage), ce qui permet une discrétisation temporelle en rendant le signal constant sur des périodes de 10 ms. Ensuite, nous générons une rampe et effectuons un comptage de 0 à 256, établissant ainsi une relation de proportionnalité entre les tensions prises par la rampe et les nombres de conversion, ce qui permet une discrétisation de tension avec un pas de quantification de $q = \frac{V_{ech_{max}}}{256}$.

Pendant chaque période d'échantillonnage (cycle de conversion), nous prenons le nombre correspondant à l'intersection entre la rampe et le signal échantillonné comme étant notre donnée numérique. Nous verrons dans la suite les détails de fonctionnement de la CAN et la caractérisation de ses différents composants.

1.1 L'échantillonneur bloqueur

L'échantillonneur bloqueur permet de capturer et de maintenir une tension d'entrée à un instant précis, en utilisant un transistor MOS pour contrôler le chargement d'un condensateur.

Il est constitué d'un transistor MOS qui, lorsqu'il est commandé par un signal d'entrée, si $V_{gs} > V_{th}$,

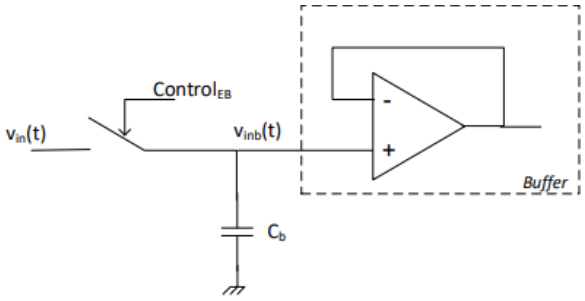

FIGURE 2 – Schéma de principe de l'échantillonneur bloqueur

se comporte comme un interrupteur fermé et donc la capacité se charge. Lorsque le signal d'entrée désactive le MOSFET ($V_{gs} < V_{th}$), il se comporte comme un interrupteur ouvert mais, grâce au buffer, la capacité conserve la tension mémorisée. On devra alors caractériser les composants le constituant, commençons par choisir un MOS adéquat pour notre application.

1.1.1 caracterisation des Mos

□

on a le choix entre deux transistors MOS : 2N7000 et ZVN4424A dont les caractéristiques sont illustrées dans le tableau suivant :

TABLE 1 – Spécifications des Transistor MOS

Paramètres	2N7000	ZVNL120A
$V_{GS_{Break}}$ (V) +-20	+-20	
$V_{GD_{Break}}$ (V) 60	200	
V_{th} (V)	2,1	1,5
$I_{ds_{max}}$ (A)	0.2	0,180
T_{on} (s)		
T_{off} (s)		

Afin de choisir le MOS le plus adéquat, il faut considérer certaines caractéristiques :

— Temps de commutation

— Impédance

— Phénomène d'injection de charges

Dans notre cas, le temps de commutation est de l'ordre du nanoseconde, ce qui est négligeable devant la période d'échantillonnage a priori 10 ms.

On trace alors sur PSPICE la résistance série équivalente de chacun des MOS proposés en fonction de la tension de commande V_{gs} grâce à un balayage en courant continu (DC sweep). en utilisant le circuit suivant

on obtient le trace suivant

On remarquera alors que le ZVNL120A a une résistance plus importante à l'état bloqué, ce qui permet une meilleure isolation de la capacité (peu d'in-

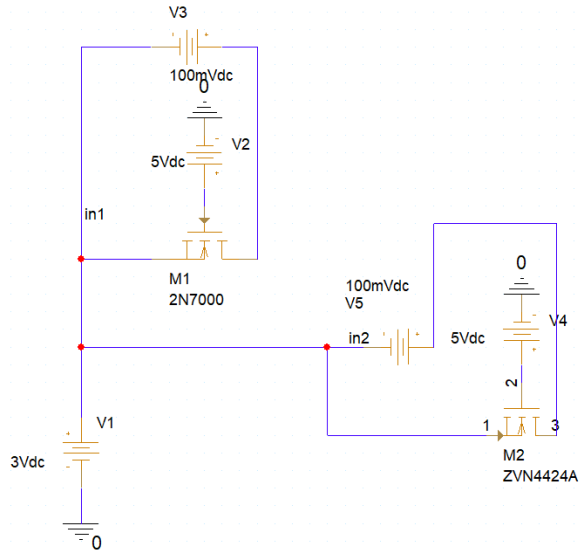

FIGURE 3 – R en fonction de V_{gs} en rouge 2N7000 et ZVNL120A en vert

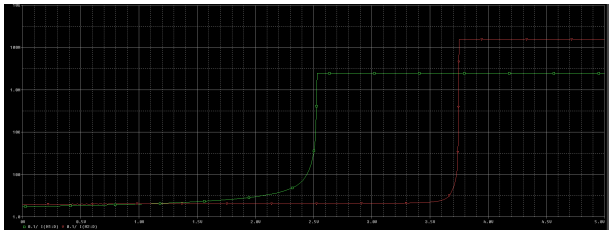

FIGURE 4 – R en fonction de V_{gs} en rouge 2N7000 et ZVNL120A en vert

jection de charges), mais aussi une résistance moins élevée à l'état passant, ce qui diminue la consommation et joue parfaitement le rôle d'un interrupteur. On prendra ce dernier pour la suite de notre conception.

Le premier schéma auquel on va penser est alors un NMOS en série avec une capacité. Cependant, le choix de lier directement le substrat avec la base (pour simplifier les calculs) pourrait causer des problèmes de non-linéarité à des tensions données. On utilisera alors deux NMOS en série pour éviter ce problème. On pourra mieux visualiser ce phénomène grâce au montage suivant.

On trace alors dans une même figure la tension échantillonnée par 2 NMOS en série (rouge), la tension d'entrée (bleu) et celle échantillonnée par un seul NMOS (vert).

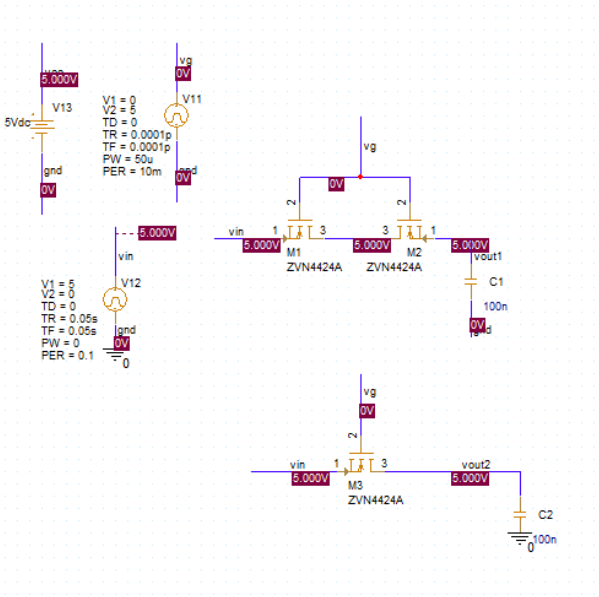

FIGURE 5 – Schéma avec deux NMOS en série

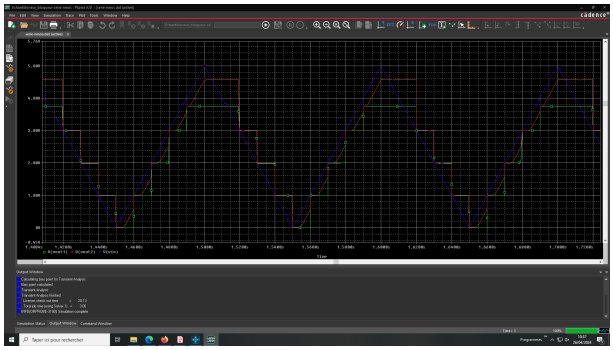

FIGURE 6 – Évolution de l'erreur d'injection en fonction de la tension d'entrée pour une capacité de C_H = XX pF

On remarquera alors qu'à partir d'une tension donnée, la fermeture de l'interrupteur ne charge pas le condensateur instantanément, et on observe que la tension de sortie est une image de la tension d'entrée (après division de tension). En effet, même lorsque $V_{gs} = 0\,V$, la liaison directe entre substrat et source se matérialise par une diode qui, lorsqu'elle est polarisée par une tension directe V_{in}, voit un passage de courant. Alors, le NMOS est court-circuité ($R_{on} \gg R_{srie}$ de cette diode).

Par contre, si on prend deux NMOS en série comme dans la figure, l'une de ces deux diodes sera toujours bloquée, empêchant tout court-circuit. Alors, on aura un échantillonnage correct.

Maintenant qu'on a caractérisé l'interrupteur, il nous reste à caractériser le condensateur. En effet, la valeur de ce dernier est déterminée par 2 contraintes :

1.1.2 caracterisation du codensateur

— L'injection de charge (soit liée aux NMOS, soit au condensateur lui-même)

— La fréquence d'échantillonnage de 100 Hz dans notre application

On commence par caractériser l'injection de charges. On peut mettre en valeur ce phénomène grâce au circuit suivant.

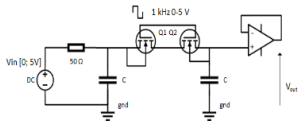

FIGURE 7 – Schéma de caractérisation de l'injection de charges

En effet, même en essayant de mémoriser la tension par un buffer, le condensateur se décharge par passage de courant. Pour caractériser cette décharge, on utilise deux capacités identiques qui servent comme des valeurs de référence, une sur l'entrée et l'autre sur la sortie. Alors, la quantité de charges déplacées entre le drain et la source est proportionnelle à la différence de tension entre l'entrée et la sortie.

$$\Delta V = V_{out} - V_{in} = \frac{Q_{out}}{C} - \frac{Q_{in}}{C} = \frac{\Delta Q}{C}$$

Or, ΔQ est indépendante de la valeur de C, le déplacement de charge dépend surtout des NMOS. D'autre part, pour notre CAN à 8 bits, on nous impose une erreur maximale de

$$\epsilon = \frac{LSB}{4} = \frac{0.5 \times V_{dd}}{255 \times 4} = 2.4\,mV$$

Or, puisque l'erreur est inversement proportionnelle à la valeur de C, on en déduit que

$$C_{min} = \frac{\Delta Q}{\epsilon}$$

on pourra ensuite calculer ΔQ grace au circuit imprime suivant avec C=100pF

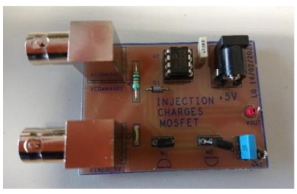

FIGURE 8 – Schéma de caractérisation de l'injection de charges

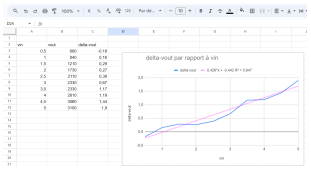

FIGURE 9 – Évolution de ΔV en fonction de V_{in}

On étudie alors l'évolution de ΔV en fonction de V_{in}. On trace ainsi la courbe suivante :

on procede a etuduer le meme phemomene sur psipce grace au shema suivant on obtient alors On

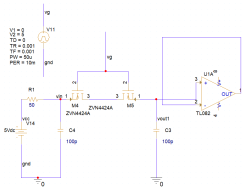

FIGURE 10 – montage PSPICE injection des charges

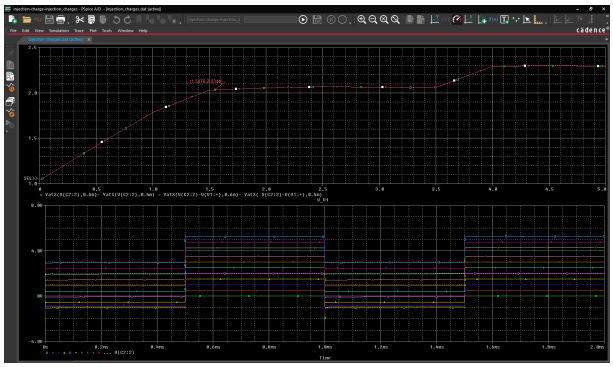

FIGURE 11 – Évolution de ΔV en fonction de V_{in} pspice

identifie le maximum de linéarité puisque la partie linéaire de l'erreur peut être tolérée en raison de sa linéarité. Par contre, on veut que ΔQ soit la plus petite possible, donc ΔV_{out} (non linéaire) doit être la plus petite possible.

On en déduit :

$$\Delta Q = \Delta V_{\max} \times 100\,\mathrm{pF}$$

Alors finalement, on en déduit :

$$C_{\min} = \frac{\Delta Q}{\epsilon} = \frac{100\,\mathrm{pF} \times \Delta V_{\max}}{2.4\,\mathrm{mV}}$$

La deuxième contrainte est la période d'échantillonnage de 10 ms. On désire avoir un condensateur complètement chargé dans $\frac{T_e}{256}$ puisque l'on réalise un CAN à 8 bits. Or, plus la valeur de C est grande, plus son temps de charge augmente. En effet, lorsque le NMOS est passant, on a un circuit équivalent RC série.($2R_onetC$)

Alors :

$$5\tau = 10R_{on}C \leq \frac{T_s}{256}$$

d'où

$$C_{\max} = \frac{T_s}{256 \times 10R_{on}}$$

On obtient alors

$$C_{\min} < C < C_{\max}$$

On prend alors la moyenne des deux extrêmes :

$$C = \frac{C_{\min} + C_{\max}}{2}$$

On choisit alors la valeur normalisée C correspondante.

On teste notre échantillonneur bloqueur avec un signal sinusoïdal de fréquence 2 Hz (fréquence typique des signaux PPG) et d'amplitude 500 mV.

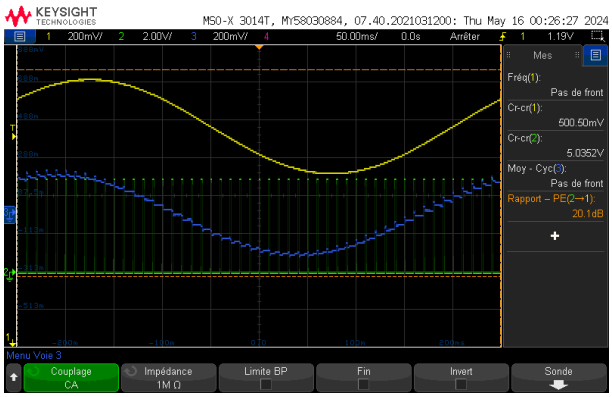

FIGURE 12 – Mesure mettant en évidence le principe d'échantillonnage à la fréquence 100 Hz avec un signal sinusoïdal en entrée de fréquence 2 Hz.

On remarque que le signal est alors correctement échantillonné, constant sur des périodes de 10 ms grâce au signal de commande (en vert) qui est de 0 V pendant 10 ms puis monte à 5 V en un instant négligeable devant T_s.

1.2 Le convertisseur simple rampe

Le convertisseur simple rampe est une partie importante du CAN car en générant un signal linéaire sur une période d'échantillonnage, on saura sa valeur à n'importe quel instant au cours de cette période. En effectuant un comptage de 0 à 256 pendant T_S, chaque nombre correspondra à un intervalle de tension de la rampe. On s'intéresse donc au nombre correspondant à l'intersection entre la tension échantillonnée et la tension de la rampe. Si on stocke ce nombre, on aura stocké notre donnée numérique.

Intéressons-nous à la génération d'un tel signal. Pour l'instant, nous verrons comment stocker la donnée numérique plus tard.

1.2.1 Le générateur de rampe

On génère notre rampe grâce à un montage intégrateur avec l'amplificateur opérationnel MC, considéré ici comme idéal. Le schéma est donné par :

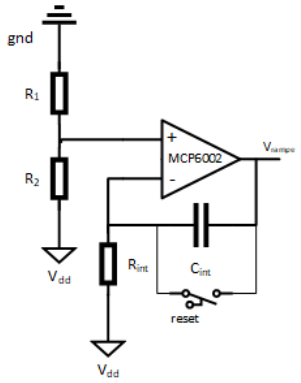

FIGURE 13 – Schéma de principe du générateur de rampe

Pour avoir une conversion correcte, on désire que la rampe balaie toutes les tensions prises par le signal échantillonné, ainsi chaque valeur du signal échantillonné aura un nombre correspondant.

$$0 < V_{\text{rampe}} < \frac{V_{\text{DD}}}{2}$$

Comme ici on veut une pente négative, on veut qu'à l'initialisation de la rampe (t=0 fermeture de l'interrupteur qui n'est d'autre que les deux NMOS commonde par tension), la tension soit maximale. Lorsque l'interrupteur est fermé, on a :

$$V^+ = V^- = V_{\text{rampe}}(0) = \frac{V_{dd}}{2}$$

$$\text{et} \quad V^+ = \frac{R_1}{R_1 + R_2} V_{dd}$$

$$\text{donc} \quad \frac{R_1}{R_1 + R_2} V_{dd} = \frac{V_{dd}}{2}$$

$$\text{d'où} \quad R_1 = R_2$$

On prend alors $R_1 = R_2 = 1\,k\Omega$. On s'est rendu compte qu'il est préférable de prendre une valeur plus élevée pour limiter le courant entrant dans l'ampli, mais la différence est négligeable étant donné que l'impédance d'entrée de l'ampli MC2308 est très élevée.

Ensuite, l'interrupteur s'ouvre pendant une durée de 10 ms et on revient au montage intégrateur classique.

En appliquant la loi des nœuds au point A, on obtient :

$$\frac{V^+ - V_{\text{DD}}}{R_{\text{int}}} + \frac{d(V_{\text{rampe}} - V^+)}{dt} = 0$$

Or $V^+ = \frac{V_{\text{DD}}}{2}$, on obtient alors :

$$\frac{d(V_{\text{rampe}})}{dt} = -\frac{V_{\text{DD}}}{2R_{\text{int}}C_{\text{int}}}$$

D'où par intégration :

$$V_{\text{rampe}}(t) = -\frac{V_{\text{DD}}}{2R_{\text{int}}C_{\text{int}}}t + \text{cst}$$

Or d'après la condition initiale, on a :

$$V_{\text{rampe}}(0) = \text{cst} = \frac{V_{\text{DD}}}{2}$$

D'où finalement :

$$V_{\text{rampe}}(t) = \frac{V_{\text{DD}}}{2} - \frac{V_{\text{DD}}}{2R_{\text{int}}C_{\text{int}}}t$$

On remarque que la pente de la rampe est imposée par les valeurs de R_{int} et C_{int}. On peut alors déterminer les valeurs de R_{int} et C_{int} convenables à notre application. On rappelle que la rampe doit balayer toutes les tensions prises par la tension échantillonnée. On désire alors qu'après une période d'échantillonnage, la tension rampe soit nulle :

$$V_{\text{rampe}}(T_s) = \frac{V_{\text{DD}}}{2} - \frac{V_{\text{DD}}}{2R_{\text{int}}C_{\text{int}}}T_s = 0$$

Ou encore :

$$1 - \frac{T_s}{R_{\text{int}}C_{\text{int}}} = 0$$

Alors :

$$T_s = R_{\text{int}}C_{\text{int}}$$

On prend alors C_int la plus grande disponible pour minimiser R_{int} et donc la consommation. On garde dans la suite le couple $(1\,\mu\text{F}, 1\,k\Omega)$. on pourra alors visualiser notre rampe pour une frequence de commande de 100hz On remarquera que la tension

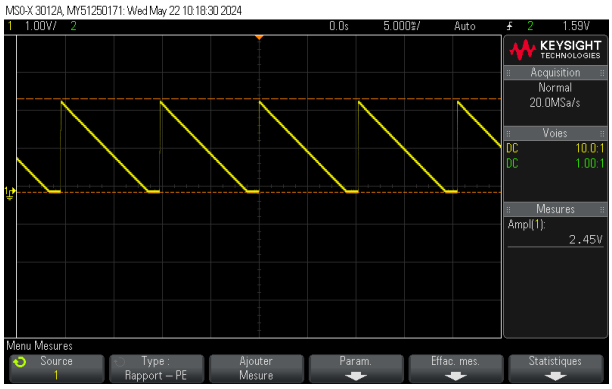

FIGURE 14 – Mise en évidence du fonctionnement du generateur rampe avec un signal carre de frequence 100hz et un temps montant de 20ns

va de $\frac{V_{\text{DD}}}{2}$ au moment de l'initialisation à 0 après une période de 10 ms. Le fonctionnement du générateur de rampe est validé.

On remarquera cependant que le condensateur C_{int} se décharge environ 0.5 ms avant T_S. Cela s'explique par les valeurs réelles des composants prises (970 Ω par exemple au lieu de 1 kΩ exactement). Cette avance est en effet négligeable et ne cause pas de perte de résolution.

1.3 comparateur

Maintenant que nos deux composants analogiques fonctionnent correctement, nous désirons pouvoir convertir la tension échantillonnée en donnée numérique.

Pour ce faire, nous nous intéressons au moment de l'intersection de la rampe avec la tension échantillonnée. Le nombre correspondant à cet instant est notre donnée numérique, puisque ce nombre nous donne l'instant d'intersection et donc la tension échantillonnée grâce à l'équation de $V_{\text{rampe}}(t)$.

En effet, en considérant que le nombre 0 correspond à la tension $\frac{V_{\text{DD}}}{2}$ et 255 à 0, nous obtenons l'équation suivante permettant la reconversion d'un nombre en tension :

$$V_{\text{ech}} = \frac{V_{\text{DD}}}{2} - \frac{V_{\text{DD}}}{2}\frac{N}{255}$$

Avec N le nombre correspondant à l'intersection de la rampe et la tension échantillonnée.

Nous pourrons mettre en évidence cette intersection grâce à un amplificateur monté en comparateur avec V_{rampe} et V_{ech} comme entrées. Nous aurons alors en sortie :

$$V_{\text{comp}} = V_{\text{sat}}^{+} \quad \text{si} \quad V_{\text{rampe}} < V_{\text{ech}}$$

$$V_{\text{comp}} = V_{\text{sat}}^{-} \quad \text{si} \quad V_{\text{rampe}} > V_{\text{ech}}$$

Ici, nous prendrons $V_{\text{sat}}^{+} = 5\,\text{V}$ et $V_{\text{sat}}^{-} = 0\,\text{V}$ pour que V_{comp} puisse aussi être vu comme un signal logique : 1 logique lorsque $V_{\text{rampe}} < V_{\text{ech}}$ et 0 logique lorsque $V_{\text{rampe}} > V_{\text{ech}}$.

Ce signal sera alors le lien entre la partie analogique et numérique du CAN.

1.4 PCB et test de la partie annalogique de a CAN

Maintenant que nous avons caractérisé la partie analogique, On trace les pistes du circuit éléctronique (PCB) à partir du schéma électrique de notre carte. Afin de réaliser le placement et le routage de notre carte, on utilise le logiciel PCB Editor de Cadence.

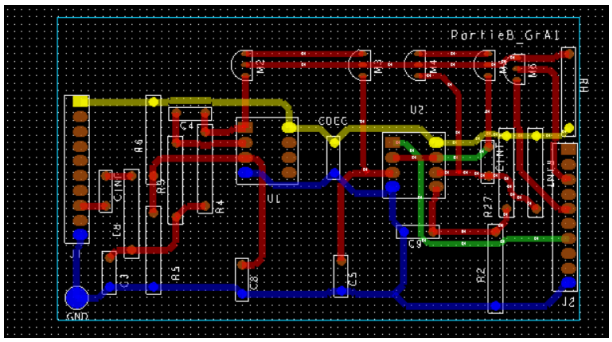

FIGURE 15 – PCB de la partie B

vérifions son fonctionnement final.

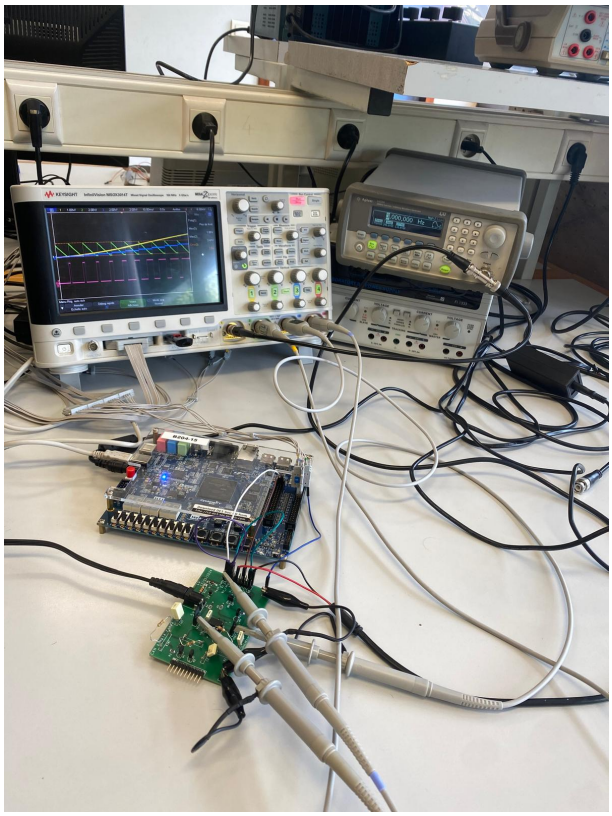

FIGURE 16 – Montage de test de la partie CAN

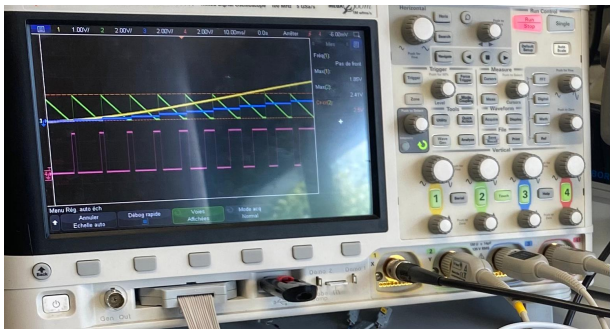

FIGURE 17 – Mise en évidence du fonctionnement de l'ADC avec le signal d'entrée sinusoidal à 1 Hz (jaune), la rampe (vert), le signal échantilloné (bleu) et la sortie du comparateur (rouge).

3

On remarquera alors que, comme prévu, nous aurons un passage de V_{comp} de 0 V à 5 V à l'instant de l'intersection de V_{ech} et V_{rampe}. Nous garderons cela à l'esprit et nous essaierons de l'utiliser pour mettre en œuvre une logique de commande permettant la conversion numérique.

1.5 Compteur et logique de commande

pour la partie numerique on utiliseraa la carte DE1-soc avec le logiciel QUartus qu'on a eu l'opportunite de nous familiariser avec au cours du premier semestre. la machine logique a realiser est completement synchronise par un horloge de frequemce 50Mhz .

on commence deja par realiser notre comptage,notre CAN est a 8 bits donc on effectue un comptage de 0 a 256 en une periode de 10ms la periode de comptage est de $\frac{Ts}{256}$ donc sa frequence est $256 \times 100hz = 25,6khz$ On procède ensuite à la réalisation des fonctions numériques à l'aide du logiciel Quartus en réalisant le schéma bloc suivant :

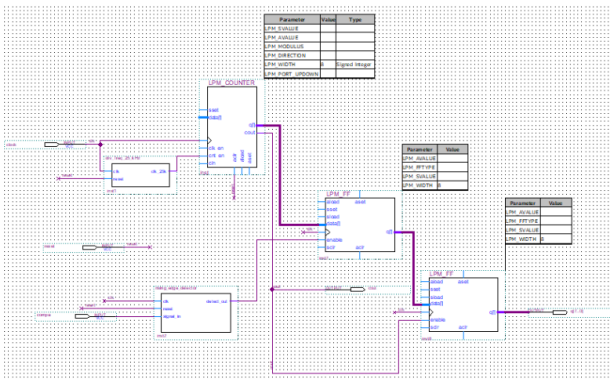

FIGURE 18 – schéma bloc en Quartus

ce schéma bloc est constitué de 5 blocs principaux : - une horloge de fréquence 50Mhz : qui

synchronise le système
- un diviseur de fréquence de 25 kHz : quin divise la fréquence de l'horloge par 2000 afin de synchroniser la fréquence d'entrée du compteur avec la fréquence de comptage ($f_c = \frac{1}{T_c} = 256 \times 100hz = 25,6khz$)

- un compteur de 8 bits : qui compte de 0 à 255

- un détecteur de front montant : qui génère une impulsion lorsqu'il détecte une transition de l'état bas à l'état haut dans le signal de comparateur.on opte a utiliser un detecteur de front montant pour eviter le probleme de rebond qui peut nuire a la fiabilite de notre CAN

- une première mémoire(qu'on pourra voir comme 8 bascules D paralel) lpm_{ff} : prend en entree les differents valeurs de comptage mais ne recopie en sortie que le nombre present sur l'entree quand le signal du comparateur passe à l'état haut(enable est relie la sortie du detecteur front montant de v_{comp}).

- une deuxième mémoire lpm_{ff} : elle est active à la fin de comptage (après une période T_s et pour une valeur de compteur ègale à 255) c'est à dire pour "cout = 1", elle mémorise la valeur recopiée sur la sortie de la première mémoire qui constitue notre donne numerique.

le fait d'attendre jusqu'a la fin d'une periode d'echontillonage pour transmettre la donne numerique a la sortie de CAN assure que lchaque donne correspond a une periode d'echontillonage precis on evite ainsi tout soucis de deformation du signal .

il est interresant de visualiser cout qui etant nulle pendant une periode d'echontillonage puis monte a 1 logique (3.3v) constitut un candidat parfait pour le signal de commande des interrupteur

on va alors utiliser Le signal "cout" afin de réinitialiser le générateur de rampe et l'échantillonneur-bloqueur afin de démarrer un nouveau cycle de conversion.

Il est à noter que la carte FPGA utilisée fonctionne en 3.3V mais accepte 5V. Pour cela, il est nécessaire de faire la conversion de 3.3V à 5V à l'aide d'un level-shifter :

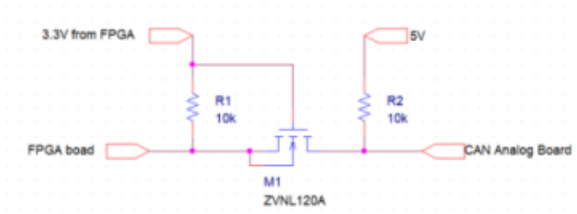

FIGURE 19 – Principe du level-shifter

Le level-shifter est constitué d'un transistor NMOS dont le drain est connecté au signal 5V par le biais d'une résistance R2, la grille est connéctée au signal 3.3V et le drain au signal de l'FPGA.Lorsque le signal de l'FPGA est bas (0V), le MOSFET est activé, connectant la sortie à 0V.En effet, $V_{GS} = V_G - V_s = 3,3 - 0 = 3,3V$ et donc le NMos est activé donc un canal lie le drain et la source d'où $V_s = 0V$. En revanche, lorsqu'il est haut (3.3V), le MOSFET est désactivé, donc le drain n'est plus connecté à la source permettant ainsi à la résistance de tirer la sortie à 5V.

on valide d'abbord le fonctionnement du compteur

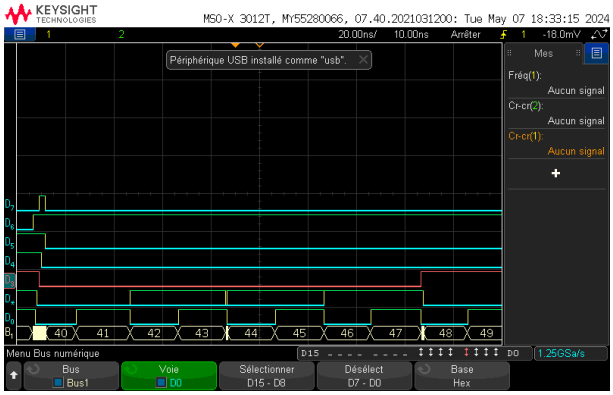

FIGURE 20 – Visualisation de la sortie du compteur

1.5.1 Le conversion numerique annalogique

Dans cette partie, on teste le générateur de rampe et le comparateur. En générant un signal d'entrée sinusoidal de fréquence 2Hz, on obtient les signaux suivant :

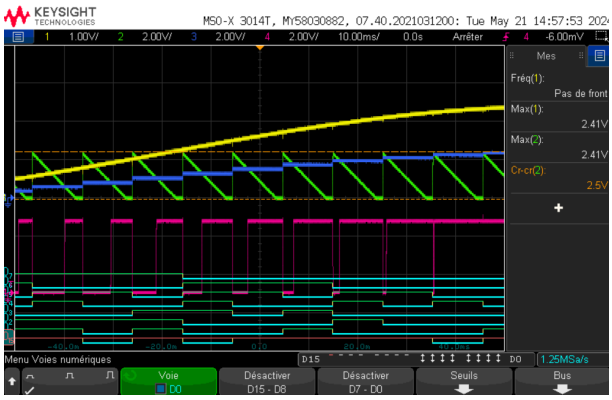

FIGURE 21 – Mise en évidence du fonctionnement de l'ADC avec le signal d'entrée sinusoidal à 2Hz (jaune), la rampe (vert), le signal échantilloné (bleu) et la sortie du comparateur (rouge).

Dans un premier lieu, on vérifie que la rampe obtenue satisfait à l'équation trouvée précédemment : $V_{\text{rampe}}(t) = -250 \cdot t + 2.5$ avec $V_{rampe}(0) = 2.5V$. Dans notre cas, par visualisation sur l'oscilloscope, l'amplitude maximale de la rampe est bien 2.5V . En vérifie aussi que lorsque le signl échantillonné dépasse le signal de rampe, le comparateur est à l'état haut, sinon il est à zéro. Ceci se reproduit sur chaque période de T_s. Dans un second temps, notre objectif est de reconstruire le signal analogique en utilisant le registre du CAN de la carte Arduino, en utilisant le programme fourni sur Moodle. On génère un signal sinusoïdal de fréquence 1 Hz en entrée et on visualise ensuite la sortie PWM de la carte Arduino sur l'oscilloscope. La PWM est un signal numérique qui simule une tension analogique en modulant la largeur d'impulsion d'un signal numérique. On utilise l'option de filtre passe-bas intégrée à l'oscilloscope afin d'éliminer les hautes fréquences indésirables du signal PWM.

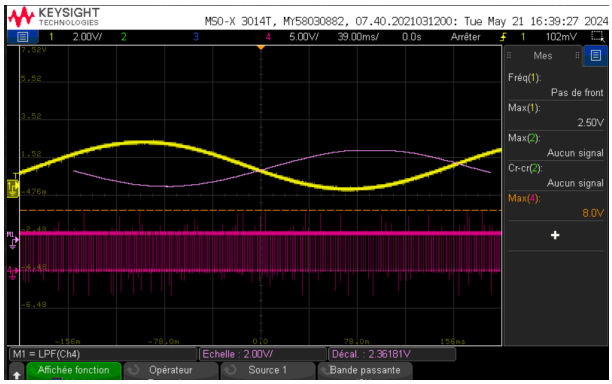

FIGURE 22 – Reconstruction du signal à l'aide d'Arduino

Ensuite, nous réaliserons une dernière validation de notre CAN grâce à Matlab. Nous injectons en entrée du CAN un signal triangulaire (1 V - 2 V) avec une fréquence de 50 mHz et une base de temps de 2,5 s sur l'oscilloscope.

Nous essaierons de le reconstruire à partir des bits de conversion grâce à Keysight Benchvue, en conservant toutes nos données de conversion sous forme de matrices Matlab.

Ainsi que le signal de commande *cout* qui permet d'associer une instantanée aux valeurs du registre du CAN.

Ensuite, en utilisant la proportionnalité entre le nombre et la tension précédemment établie, nous pourrons reconstruire le signal comme un signal analogique (à chaque instant correspond une tension). De toutes nos acquisitions Benchvue, nous choisirons des points d'acquisition maximale (500000 points).

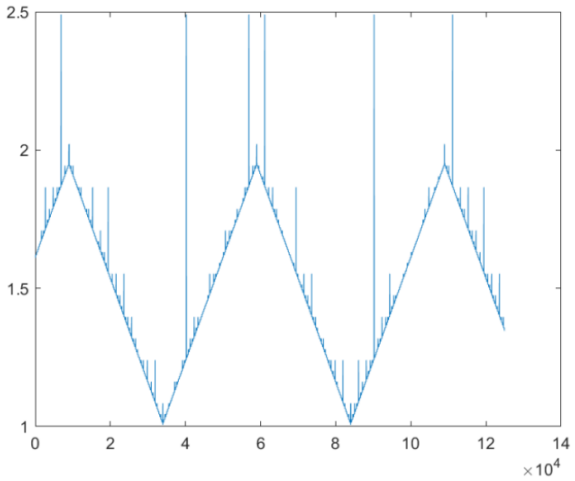

FIGURE 23 – Reconstitution d'un signal triangulaire à l'aide de Matlab

Il nous faut désormais caractériser l'INL (Integral Non-Linearity) et le DNL (Differential Non-Linearity) de notre CAN grâce à l'application Matlab disponible sur Moodle (*ADC-carac.mlapp*). Nous pourrons directement les tracer.

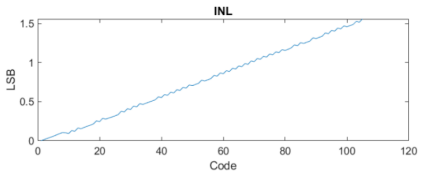

FIGURE 24 – INL de notre CAN

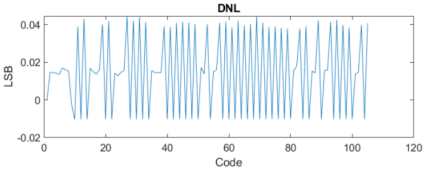

FIGURE 25 – DNL de notre CAN

Pour les deux figures, l'ordonnée représente l'erreur (entre la tension convertie et la tension originale) en multiples de LSB $\frac{2.5V}{255} = 9.8mV$ et l'abscisse représente le nombre de conversion (la donnée numérique). On remarque que l'INL évolue linéairement en fonction du nombre de conversions, donc plus la tension à convertir est grande, plus l'erreur augmentera.

La DNL (Differential Non-Linearity) représente l'erreur de linéarité différentielle et est définie par $V(N) - V_{N-1} - q$, où q est le pas de quantification de 9.8 mV. Dans le cas idéal, on veut avoir $V(N) - V(N-1) = q$. Cette erreur découle des variations du pas de quantification. Elle est considérée acceptable tant qu'elle reste dans entre -1 à 1 LSB, ce qui est largement respecté dans notre cas.

L'erreur associée au pas de quantification provient d'une pondération inexacte des tensions attribuées à chaque bit, entraînant une non-linéarité. Cette pondération inexacte résulte d'un déséquilibre des composants.

Avant de passer à l'acquisition du signal PPG réel, nous pouvons caractériser notre CAN grâce à un tableau résumant l'étude précédente.

1.5.2 Acquisition finale

Nous relions alors la chaîne complète photodiode+ampli+filtre+CAN et nous réalisons l'acquisition du signal réel sur Matlab comme précédemment fait avec la tension triangulaire.

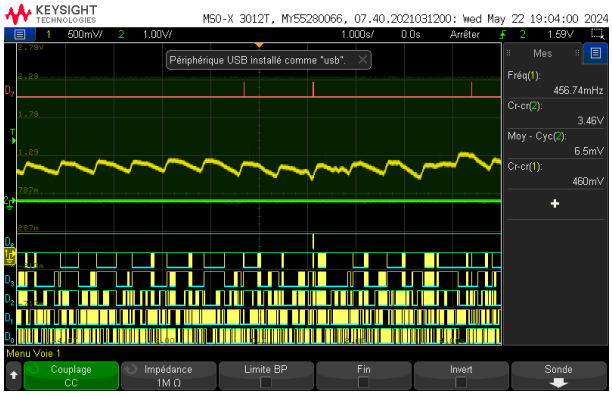

FIGURE 26 – Test d'acquisition en jaune le signal PCB filtré et en vert *cout* et les bits de conversion

Nous réalisons l'acquisition de ce signal (nous faisons apparaître sur Benchvue uniquement les bits et *cout*) et en utilisant (*ADC-carac.mlapp*), nous obtenons la reconstruction suivante de ce signal.

on remarque alors que la conversion numerique annalogique se ffait correctement et est fidele a la tension annalogique d'entree a quelque pics de bruits pres on devra alors filtrer se signal numerique encore plus dans le domaine du numerique

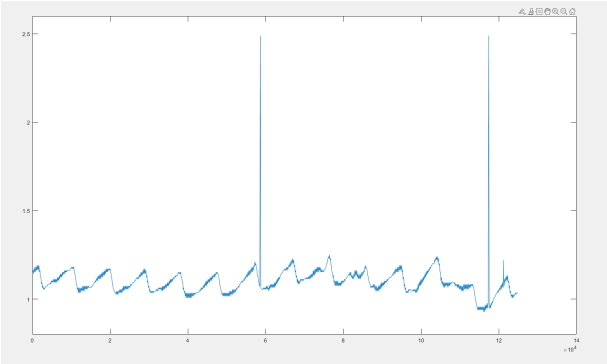

FIGURE 27 – Reconstruction du signal réel à partir des bits de conversion

pour pouvoir l'exploiter plus facilement dans la partie TNS

on procede de la meme maniere jusqu'a obtenir un signal typique PPG permettant une exploitation numerique plus facile on a alors choisi le signal suivant pour l'etude de la partie TNS

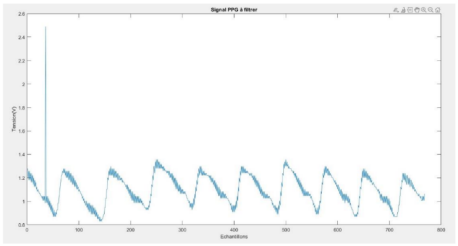

FIGURE 28 – Reconstruction du signal réel à partir des bits de conversion