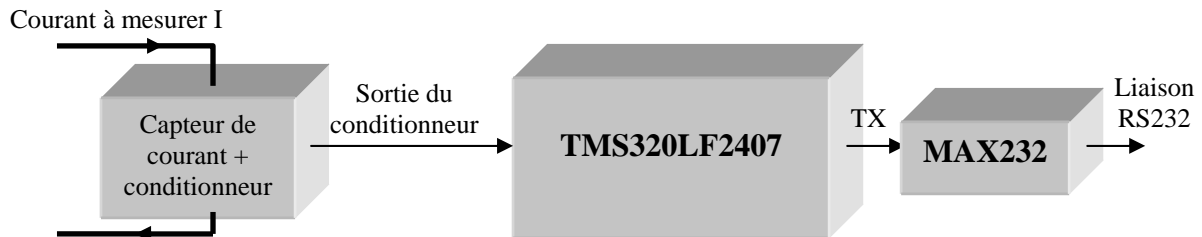


Partiel MC43

Exercice 1 : Acquisition sans repliement

L'exercice traite la programmation d'un système de mesure et de transfert série schématisé par la figure suivante. Il est composé d'un capteur de courant, de son conditionneur, d'un DSP contrôleur TMS320LF2407 et d'un driver MAX232 :

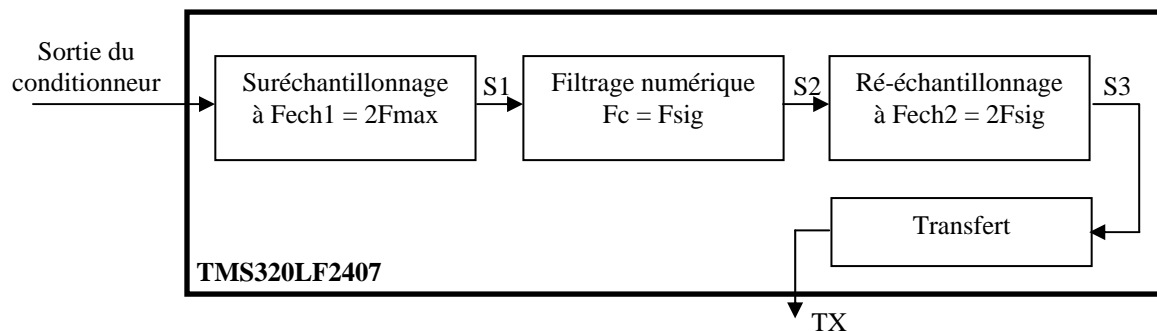


La tension délivrée en sortie du conditionneur est comprise entre 0 et 3,3 V. Elle est amplifiée de sorte à obtenir des résultats de conversion exprimant numériquement le courant I en milliampères.

Le courant à mesurer contient des composantes de fréquences différentes que l'on peut décomposer de la manière suivante :

- $0 < F < F_{sig}$: composantes utiles à mesurer
- $F_{sig} < F < F_{max}$: composantes indésirables induites par un convertisseur de puissance

Pour permettre d'adapter la mesure au choix de F_{sig} et à la valeur de F_{max} , on décide d'éviter le repliement du aux fréquences indésirables, par suréchantillonnage conformément au schéma de principe suivant :



Anti-repliement :

- I.1 Dans le cas où le signal analogique provenant du conditionneur serait directement échantillonné à $2F_{sig}$ sans filtre anti-repliement, représenter le spectre du signal échantillonné faisant apparaître le repliement et les fréquences F_{sig} et F_{max} .
- I.2 Dans le cas où on utiliserait un filtre anti-repliement, indiquer quelle devrait être sa fréquence.
- I.3 Représenter le spectre du signal échantillonné avec filtre anti-repliement.
- I.4 Dans le cas utilisant le filtrage numérique décrit en début d'énoncé, représenter les spectres des signaux numériques S1, S2 et S3. Montrer alors que le procédé évite le repliement.

Principe d'échantillonnage et de filtrage :

L'échantillonnage à la fréquence $F_{ec1} = 2F_{max}$ est réalisé en utilisant le Timer3 qui déclenche les conversions analogiques numériques (voie0). Le convertisseur déclenche en fin de conversion le sous-programme d'interruption *interrupt void interADC(void)* qui réalise le calcul du filtre numérique de fréquence de coupure F_c . Le résultat donne le signal S2. Le ré-échantillonnage à $F_{ec2} = 2F_{sig}$ du signal S2 est simplement réalisé par sauvegarde d'un point toute les $N = F_{ec1}/F_{ec2}$ périodes d'échantillonnage (choisi entier), pour donner le signal désiré S3.

- I.5 Ecrire la fonction *init_Timer3(void)* initialisant le Timer3 pour $F_{ch1} = 100\text{kHz}$ ($F_{cpu} = 40\text{MHz}$).
- I.6 Ecrire la fonction *init_ADC(void)* initialisant le convertisseur analogique numérique.
- I.7 La fonction récurrente du filtre numérique est la suivante : $S2(n) = A \cdot S2(n-1) / B + S1(n) / B$. Ecrire la fonction d'interruption réalisant le calcul du filtre et donc du signal S2, en tenant compte du reste des divisions (A et B sont des constantes permettant de régler la fréquence de coupure du filtre).

Ré-échantillonnage et transfert sur la liaison série :

Le signal S2 est ré-échantillonné à la fréquence F_{ch2} pour donner le signal S3. On choisit $F_{sig} = 500\text{Hz}$, soit $F_{ch2} = 2F_{sig} = 1\text{kHz}$. Ainsi le ré-échantillonnage consiste à extraire un échantillon sur 100 du signal S2. Chaque échantillon du signal S3 est converti au format 8 bits (baisse de précision) puis transféré au format RS232 sur la liaison SCI (1 mot de 8 bits par échantillon).

- I.8 Rappeler quelles sont les vitesses de transmission maximales des normes RS232, RS423, RS422 et RS485.
- I.9 On choisit le mode de transmission RS232 suivant : données de 8 bits, pas de parité, 1 bit de STOP, vitesse de transmission de 19200 bauds. Ecrire en langage C la fonction *void init_SCI(void)* initialisant le module SCI ($F_{cpu} = 40\text{MHz}$).
- I.10 Proposer un schéma de connexion entre la sortie TX du DSP, le driver MAX232 et le connecteur DB9 RS232 en précisant le numéro des broches utilisées.
- I.11 Représenter la trame générée sur la sortie TX du DSP et sur la sortie RS232 du MAX232 dans le cas de la transmission de l'octet 0x27, en précisant les niveaux de tension.
- I.12 Ecrire en langage C la fonction *void envoi(char x)* réalisant la transmission d'un octet sur la liaison SCI.
- I.13 Donner la partie de code en langage C à ajouter dans la fonction d'interruption pour réaliser le ré-échantillonnage de S2 (calcul de S3) et la transmission de S3 sur la liaison SCI.
- I.14 Montrer que la vitesse de transmission de la liaison RS232 est supérieure au flux maximum des données.

Exercice 2 : Programmation du bus I2C

On étudie la programmation du contrôleur I2C PCA9564 interfacé dans l'espace mémoire du DSP TMS320LF2407 cadencé à 40MHz. Le bus I2C est connecté au composant I2C PCF8591 (convertisseur A/N et N/A). Les documentations des composants sont données dans le fascicule de cours.

1. Indiquer quels sont les registres du contrôleur PCA9564 et décrire leur rôle (2 lignes maximum par registre).
2. Indiquer comment agir sur ces registres pour envoyer une condition de START.
3. Indiquer comment agir sur ces registres pour envoyer une condition de STOP.
4. Indiquer comment configurer le contrôleur PCA9564 pour qu'il effectue un acquittement après réception d'une donnée.
5. La trame I2C suivante décrit la lecture sur le PCF8591 du résultat de conversion A/N. Ecrire en langage C le sous-programme *char ADC(void)* traduisant cette séquence.

