Réalisation d'un lecteur audio

But de la manipulation

L'objectif principal de cette manipulation est de réaliser un système numérique relativement complexe : un mini-lecteur audio

Pour ce faire, il sera nécessaire de comprendre le fonctionnement d'un convertisseur analogique-numérique ainsi que d'un convertisseur numérique-analogique

Prérequis

Avant d'entrer au laboratoire, il est conseillé de lire les chapitres suivants :

- Chapitre 5 : Entrées-sorties numériques
- Chapitre 4 : Timers

Il est encore conseillé de lire les sections 1 à 5 du complément « Programmation d'une carte à microcontrôleur »

Prédéterminations

Il vous est recommandé de répondre aux exercices du point 2

Objectifs

A la fin de ce laboratoire, vous devez être capable:

- De réaliser un programme sur un microcontrôleur
- D'expliquer le fonctionnement d'un convertisseur analogique-numérique
- De lier un cahier des charges aux périphériques d'un microcontrôleur.

1. Introduction à la conversion analogique-numérique

Introduction

Les microprocesseurs s'intègrent de plus en plus dans la réalisation de processus industriels (contrôle de la température, affichage d'une vitesse, systèmes d'alerte) ou grand public (systèmes audio, radioréveils,…).

Tous ces processus ont pour point commun qu'ils interagissent avec des grandeurs analogiques. Un processeur étant un être binaire, il est nécessaire de transformer les grandeurs physiques analogiques en nombres binaire et inversement.

La transformation du physique vers le numérique se fait en deux étapes :

- Transformation de la grandeur physique à mesurer (température, vitesse, …) en tension. C'est le rôle du capteur (thermocouple, accéléromètre,…)
- Conversion de cette tension en grandeur binaire, C'est le rôle du convertisseur analogique- numérique (CAN)

Le chemin inverse permet de transformer une grandeur numérique en une grandeur physique :

- Le convertisseur numérique-analogique (CNA) permet de transformer un nombre binaire en une tension
- L'actionneur transforme la tension en une autre grandeur permettant d'actionner un système (ex : four à induction, moteur électrique, vannes, …)

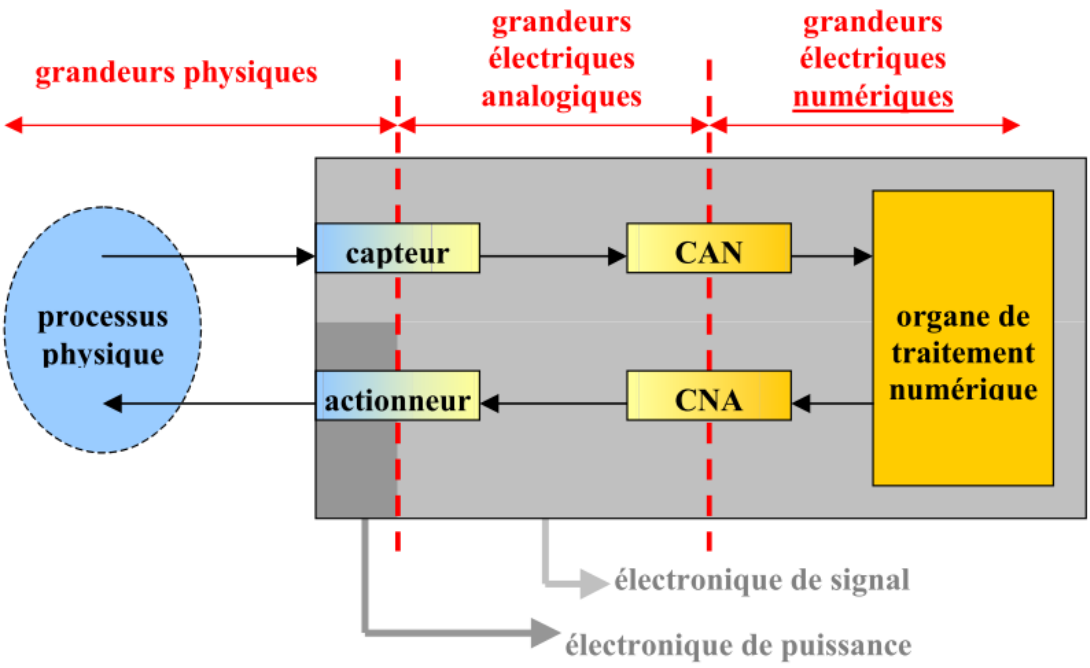


Figure 1 Architecture d'un processus

Conversion analogique – numérique

Le convertisseur analogique-numérique a pour rôle de transformer une tension analogique (évoluant de manière continue) en un nombre binaire codé sur un nombre de bits définis (8 ou 12 pour le dsPIC).

Pour ce faire, le convertisseur réalise une double quantification :

- Quantification dans le temps : c'est l'échantillonnage
- Quantification en niveau.
  Il faut noter qu'il est impossible de représenter parfaitement un signal analogique sur un nombre fini de bits.

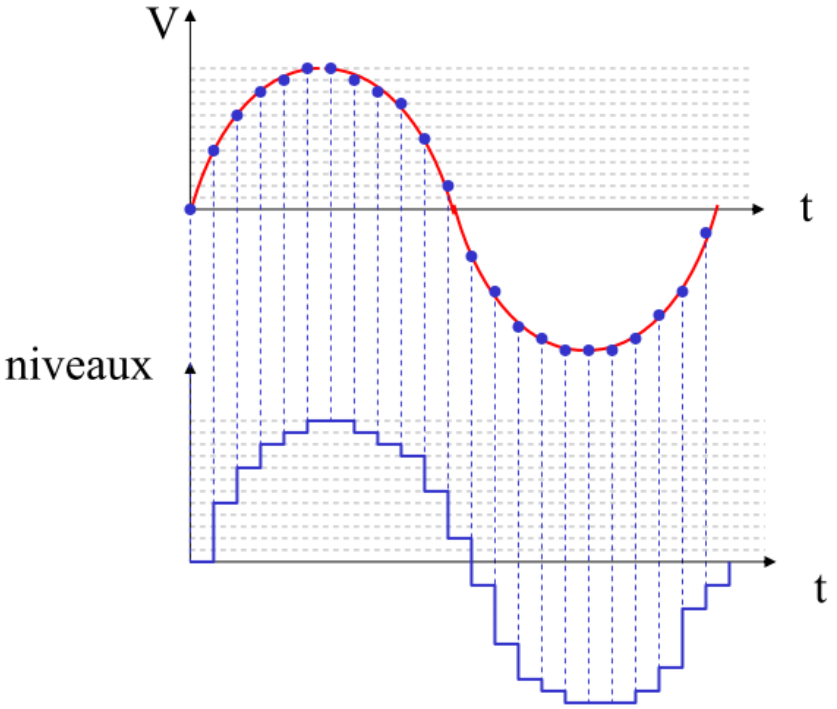
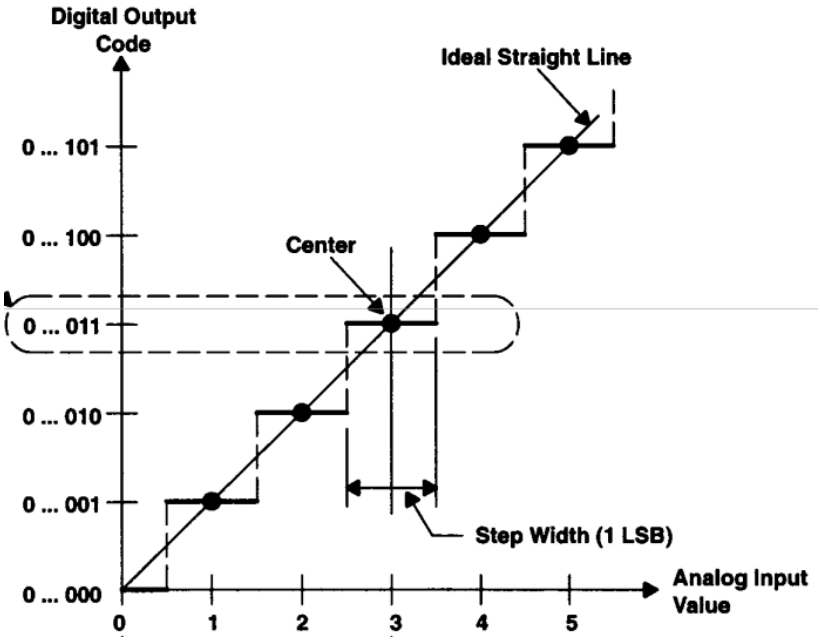


Figure 2 Double quantifications

Le rendu d'un convertisseur numérique-analogique dépend donc essentiellement de deux paramètres :

- La fréquence d'échantillonnage
- La résolution, c'est-à-dire le nombre de bits N codant la grandeur convertie. La résolution peut également être définie électriquement comme étant la plus petite variation de tension détectable. Si le convertisseur a une plage de fonctionnement de 0V à 5V et convertit la grandeur sur 10 bits, la résolution est de $\frac{5V}{2^{10}} = 4.9mV$ (= plage d'entrée / étendue des codes binaires). Au final, à chaque code binaire sur N bits correspondra à une plage de tension d'entrée.
  .



Conversion numérique-analogique
Son principe est l'inverse du CAN : il transforme un nombre codé sur N bits en une tension comprise dans sa plage de sortie.
La tension obtenue est continue dans le temps (le convertisseur garde sa sortie constante tant qu'une nouvelle conversion n'est pas effectuée), mais elle est toujours quantifiée sur sa valeur car seules 2^N tensions différentes sont réalisables (chaque code binaire correspond à une tension).

2. Prédéterminations

1) Calculez la résolution du CAN se trouvant sur la carte Explorer16 sachant que la plage de tension d'entrée est de 0/3,3V et qu'il comporte 12bits
2) Déterminez l'espace mémoire nécessaire au stockage d'un fichier sonore de 3s, échantillonné à 11kHz, en sachant que chaque échantillon est stocké sur un octet.

3. Manipulation

1. Réalisation d'un voltmètre

Dans un premier, vous allez vous familiariser avec l'utilisation du convertisseur analogique-numérique à l'aide d'une application simple : un voltmètre mesurant la tension à la sortie d'un potentiomètre connecté à l'entrée AN0. Un retour visuel se fera en utilisant les LED comme bargraph (plus la tension est élevée, plus le nombre de LED allumées doit être élevé).
Le bargraph devra être rafraichi à une cadence de 1kHz

- Tracez le schéma bloc complet de votre système. Quels périphériques allez-vous utiliser ? Mettez en évidence les interfaces entre l'intérieur et l'extérieur de µC et indiquez clairement en encadrant les blocs internes au µC.
- Programmez l'ADC1 du dsPIC. Quelques lignes de code ont déjà été écrites pour mettre l'ADC dans un état de fonctionnement normal. Créez également

une base de temps à la bonne fréquence. Pour lier les deux, vous avez le choix entre plusieurs implémentations :

- Tester le bit de flag du timer pour lancer la conversion, et tester ensuite le bit de fin de conversion (IFS0bits.AD1IF)
- Automatiser un des deux tests (ou les deux) par le biais d'une interruption
- Configurer l'ADC de sorte à ce que le débordement du timer lance automatiquement la conversion. Ceci se fait en modifiant le champ AD1CON1bits.SSRC (cf. point 5.3 du guide de programmation)

NOTE : dans tous les cas, votre fonction ne peut pas être bloquante.

- Réalisez le traitement de l'information en provenance de l'ADC
- Parmi les choix d'implémentation, lesquels garantissent la période d'échantillonnage ?

2. Lecteur audio

On vous demande de programmer la carte de manière à ce qu'elle puisse jouer un fichier sonore sur un baffle qui sera connecté sur la sortie adéquate. Votre système devra comporter les fonctionnalités suivantes :

- Pouvoir émettre un son sur un baffle à l'aide du convertisseur numérique – analogique lorsqu'on appuie sur un bouton
- Lire un fichier sonore stocké sous forme de vecteur (variable *son*, dans son.h)
- Envoyer la commande au DAC externe. Ceci se fait via la fonction *ecrit_dac_signal()* , définie dans CNAserie.h
- Contrôler le volume du baffle à l'aide du potentiomètre
- indiquer le volume auquel vous vous trouvez.

Modifiez votre code pour tenir compte de toutes ces demandes. Au préalable, mettez à jour votre schéma bloc.