

# Projet d'électronique III

Alexandre Carlessi

Thomas Denoréaz

Johan Berdat

3 juin 2013

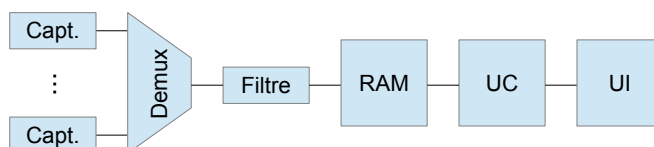
## Table des matières

<b>1</b>	<b>Introduction</b>	<b>1</b>
<b>2</b>	<b>Composants analogiques et échantillonnage</b>	<b>1</b>
<b>3</b>	<b>Unités de contrôle et de traitement</b>	<b>3</b>
<b>4</b>	<b>Interface utilisateur</b>	<b>4</b>
<b>5</b>	<b>Conclusion</b>	<b>4</b>

## 1 Introduction

L'objectif du projet est de développer un ordinateur de bord pour une voiture. Cela permet d'avoir une vision d'ensemble de la conception d'un système électronique.

L'automobile possède un certains nombre de capteurs (tachymètre, niveau d'essence...) qui relaient leur informations analogique jusqu'à l'unité de traitement. Le signal est échantillonné et stocké en mémoire, pour être analysé par l'unité de contrôle. Cette dernière prend les décisions qui s'imposent (avertissement et régulations) et envoie les informations à l'interface utilisateur.

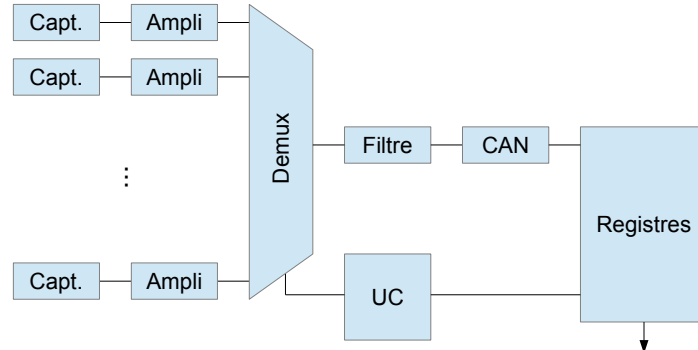


## 2 Composants analogiques et échantillonnage

On peut citer beaucoup grandeurs à mesurer dans une voiture.

- Tachymètres, pour mesurer la vitesse des roues et du moteur.
- Thermomètres, pour mesurer la température extérieure et du moteur.

- Baromètres, pour la pression de l'huile et autres fluides.



Des amplificateurs sont utilisés sur les capteurs pour permettre au signal analogique de parcourir les grandes distances qui les séparent du système central.

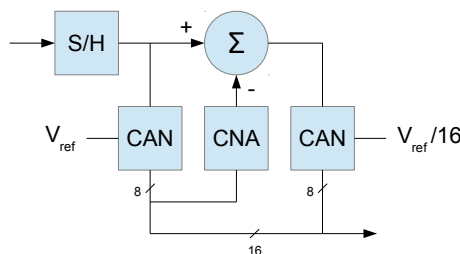
Pour économiser le nombre de composants utilisés, on n'utilise qu'un seul convertisseur analogique-numérique, après un démultiplexeur. Un contrôleur s'occupe d'échantillonner successivement les différents capteurs, pour mettre à jour les registres qui leur correspondent.

Ces informations ne requièrent pas une précision élevée. Nous avons donc choisi une fréquence d'échantillonnage pour chaque capteur de  $10\text{Hz}$ . Ainsi, si nous avons 5 capteurs, le contrôleur aura une fréquence de  $50\text{Hz}$ .

Après le démultiplexeur, un filtre actif s'occupe de nettoyer le signal. On a opté pour un simple filtre passe-bas, afin de filtrer les fréquences supérieures à  $10\text{Hz}$ . Il serait bien sûr possible de cascader ce filtre pour améliorer la qualité (et éventuellement considérer d'autres types de filtres, tel le Sallen-Key de Butterworth d'ordre  $N$ ).

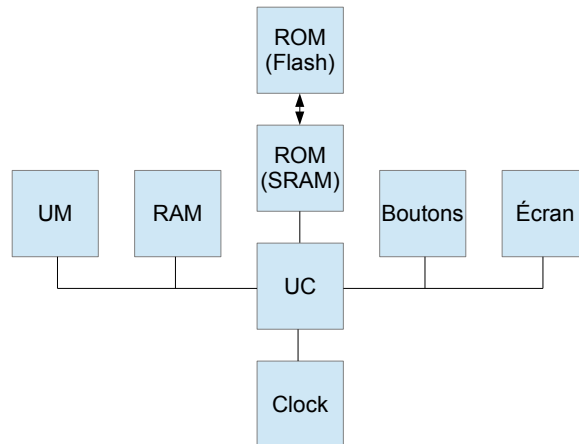
L'étape suivante consiste à convertir le signal analogique en signal numérique. Pour cela, on a à disposition plusieurs sortes de convertisseurs. Nous avons choisi la famille de convertisseurs A/N à approximations successives, étant adaptés pour un usage général. En effet, ils représentent un bon compromis précision, vitesse, prix.

Le seul désavantage de ce genre de convertisseur est le nombre limité de bits disponibles (8 à 14 bits). Pour corriger ce problème, il est possible de monter ces CAN en cascade, et ainsi augmenter la résolution. Nous estimons que 16 bits sont suffisants pour les grandeurs que nous utilisons.



Enfin, ce signal numérique est mémorisé dans des registres, qui pourront être lus par l'unité principale.

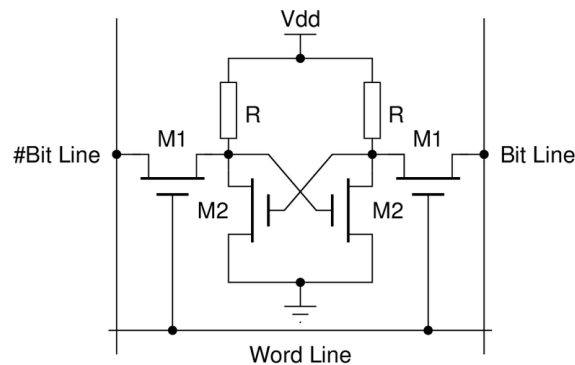
### 3 Unités de contrôle et de traitement



L'unité de contrôle (UC) est reliée à tout les autres composants, et se charge de coordonner les tâches.

Du côté de la mémoire, les différents registres des capteurs sont accessibles en lecture seule. Une mémoire SRAM nous paraît être un bon choix, rapide et robuste. Au vu de la taille minime nécessaire, son coût est acceptable.

À cela s'ajoute une autre mémoire vive qui contient les données nécessaires au système. De nouveau, nous utilisons une SRAM pour conserver les valeurs (compteur kilométrique, par exemple) lorsque le système est sous tension. Et lorsque le système est privé d'alimentation, une copie de la mémoire est sauvegardée sur une mémoire non-volatile, de type Flash. Cette décision est motivée par le nombre limité d'écriture sur les supports Flash. En effet, avec une fréquence d'écriture de  $10Hz$ , la mémoire Flash atteindrait rapidement ses limites de réécriture.



SRAM (source Wikipédia)

**TODO UC description du fonctionnement (algorithmes?)**

**TODO UC implémentation en VHDL**

**TODO HORLOGE description des choix et fonctionnalités**

**TODO HORLOGE implémentation en VHDL**

## **4 Interface utilisateur**

**TODO fonctionnalités (boutons, affichage...)**

**TODO implémentation en Java**

## **5 Conclusion**

**TODO wtf**