



CAMPUS  
D'ENSEIGNEMENT SUPÉRIEUR  
ET DE FORMATION PROFESSIONNELLE

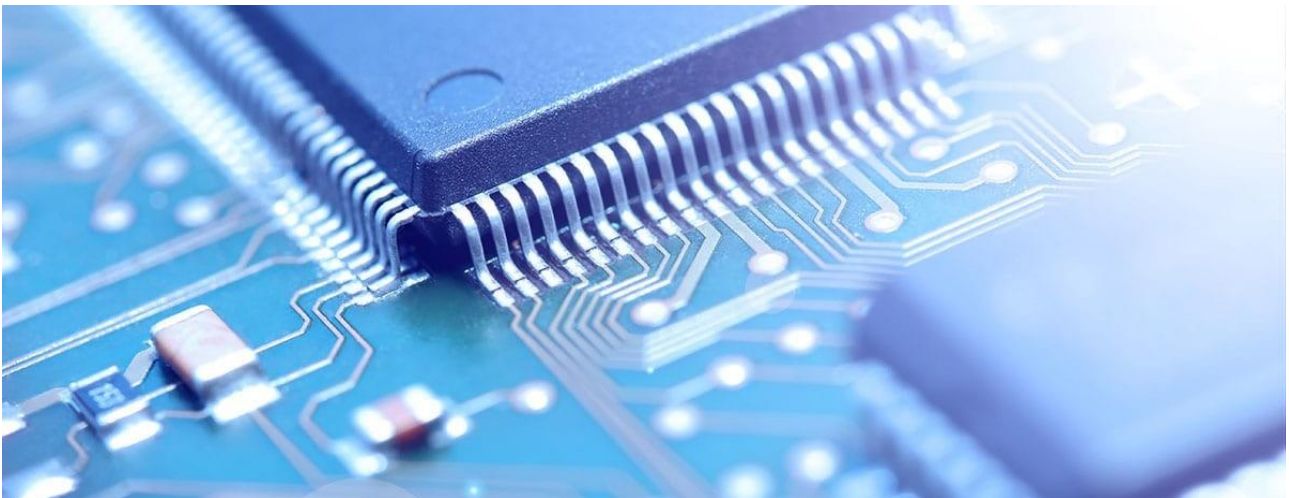
---

# TP MICROCONTROLEUR

## TP3 – Convertisseur analogique/numérique et numérique/analogique

---

Par Damien DUBOIS & Marion ESCOUTELOUP



CESI S3E – P1G1 – FIPA26

12 juillet 2022

## TABLE DES MATIERES

I.	INTRODUCTION .....	3
II.	Convertisseur Analogique/Numérique ou CAN.....	4
1.	Préparation.....	4
2.	Initialisation du CAN .....	5
3.	Lecture de la valeur du CAN .....	Erreur ! Signet non défini.
III.	Bus I2C et Convertisseur numérique/analogique (CNA) .....	5
1.	Préparation et initialisation du bus I2C .....	5
2.	Envoie sur Vumètre (cadran).....	5
IV.	Affichage sur écran LCD.....	6
1.	Préparation.....	6
2.	Mise en route de l'écran LCD .....	6
V.	CONCLUSION .....	7

## TABLE DES FIGURES

<i>Figure 1 : Architecture du Projet.....</i>	<i>3</i>
<i>Figure 2 : Schéma électrique pour utilisation du potentiomètre et CAN.....</i>	<i>4</i>

### I. INTRODUCTION

Le but de ce troisième et dernier TP de microcontrôleur est de mettre en œuvre plusieurs convertisseurs (CAN et CNA). Le convertisseur Analogique/numérique, ou CAN, a pour but de venir lire la tension aux bornes d'un potentiomètre et de convertir cette donnée en une valeur numérique (code binaire) lisible par le microcontrôleur (CAN interne). La seconde étape est de pouvoir rendre cette donnée visible par l'utilisateur via l'affichage sur un écran LCD, déjà utilisé dans les TP précédents, ainsi que sur un cadran. Pour afficher sur ce cadran, il nous sera donc nécessaire d'utiliser cette fois-ci un convertisseur numérique analogique, ou CNA, externe du microcontrôleur. Le CNA est contrôlé via un bus I2C.

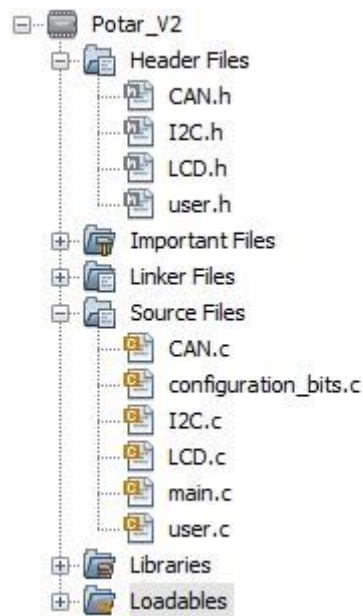


Figure 1 : Architecture du Projet



### 2. Initialisation du CAN

Nous avons commencé le développement du programme du CAN par la fonction d'initialisation de celui-ci sur le port utilisé. Cette fonction permet de mettre le CAN en fonctionnement avec les paramètres souhaités. Tout d'abord mettre la Pin à mesurer en mode analogique, choisir le nombre de coups d'horloge nécessaire à la conversion, justification à droite sur les 16 bits disponibles et la fréquence d'horloge pour cadencer le fonctionnement.

### 3. Fonction de lecture de valeur du CAN

Nous appelons par la suite une fonction spécifiquement développée pour nous permettre de lire la donnée du CAN.

Cette fonction fonctionne comme suivant :

- 1 : Lancement de la lecture de la donnée par le CAN
- 2 : Lecture registre ADRESH (octet de poids fort, bits 15 → 8)
- 3 : Lecture registre ADRESL (octets de poids faible, bits 7 → 0)
- 4 : Mise en forme des données sur 16 bits (concaténation de ADRESH & ADRESL)
- Et pour finir, la fonction renvoie la donnée mesurée.

## III. Bus I2C et Convertisseur numérique/analogique (CNA)

### 1. Préparation et initialisation du bus I2C

L'ensemble des codes permettant l'utilisation du CNA nous ont été directement donné lors des séances de cours et TP. Nous n'avons fait que les insérer dans notre programme.

### 2. Envoie sur Vumètre (cadran)

La mise en route du vumètre s'est activé automatiquement lorsque notre programme pour le CNA a été implémenté. En effet, celui-ci est directement relié à la sortie du CNA.

## IV. Affichage sur écran LCD

### 1. Préparation

Pour cette partie, nous avons réutiliser les éléments programmé lors du TP 2. Ainsi, les explications sont semblables.

### 2. Mise en route de l'écran LCD

Nous avons d'abord adapté l'affichage sur l'écran à notre besoin actuelle. Nous souhaitons afficher la tension avec 4 chiffres significatives, dont 3 après virgules. Nous avons ainsi réaliser une conversion entre nos chiffres un à un vers le code ASCII correspondant. Cette conversion est relativement simple pour les chiffres, il suffit d'y ajouter le code 0x30. Par exemple, pour afficher un '8' :  $0x30 + 0x08$ .

Pour afficher dans le bon ordre ces chiffres, nous avons réalisé des tampons :

- unité = nombre /1000                      → tampon = nombre%1000;
- dixième = tampon /100                    → tampon = nombre %100;
- centième = tampon/10;
- millième = tampon%10;

Puis nous paramètre le code ASCII de la virgule pour pouvoir l'afficher aussi.

## **V. CONCLUSION**

Lors de ce TP, nous avons pu développer un programme utilisant des convertisseur du type CNA et CAN. Nous avons validé le fonctionnement étapes par étapes en effectuant du debug en pas à pas et du test fonctionnel sur la mallette de TP.

Après quelques mises au point et conseils pédagogiques nous validons le fonctionnement.

L'ensemble du programme développé est fonctionnel sur la mallette misent à notre disposition.