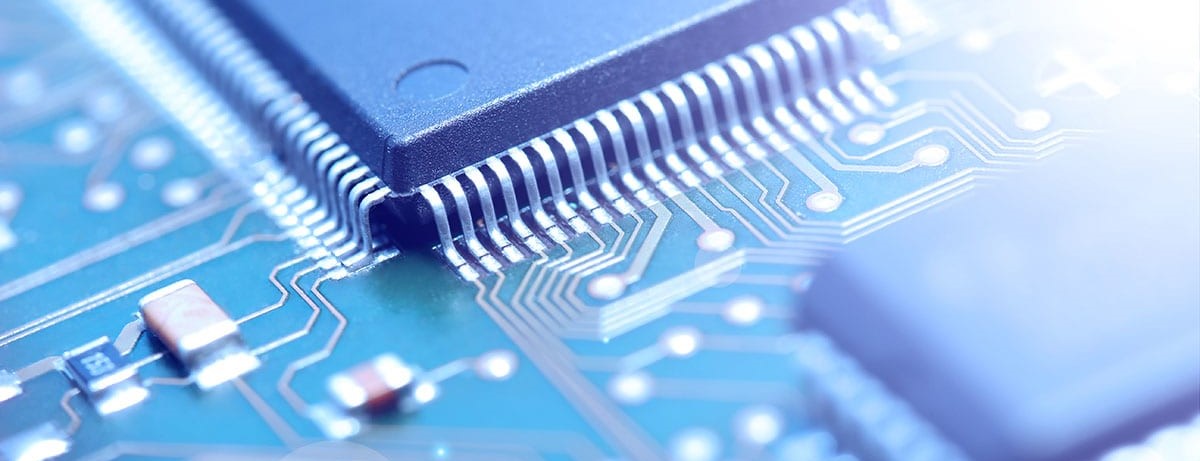


**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**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](#_Toc108564732)

[II. Convertisseur Analogique/Numérique ou CAN 4](#_Toc108564733)

[1. Préparation 4](#_Toc108564734)

[2. Initialisation du CAN 5](#_Toc108564735)

[3. Lecture de la valeur du CAN **Erreur ! Signet non défini.**](#_Toc108564736)

[III. Bus I2C et Convertisseur numérique/analogique (CNA) 5](#_Toc108564737)

[1. Préparation et initialisation du bus I2C 5](#_Toc108564738)

[2. Envoie sur Vumètre (cadran) 5](#_Toc108564739)

[IV. Affichage sur écran LCD 6](#_Toc108564740)

[1. Préparation 6](#_Toc108564741)

[2. Mise en route de l’écran LCD 6](#_Toc108564742)

[V. CONCLUSION 7](#_Toc108564743)

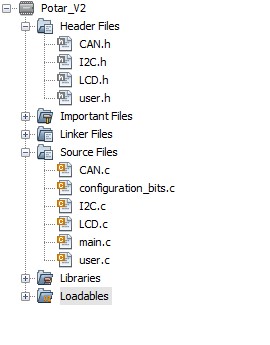
**TABLE DES FIGURES**

[**Figure 1 : Architecture du Projet** 3](#_Toc108564744)

[**Figure 2 : Schéma électrique pour utilisation du potentiomètre et CAN** 4](#_Toc108564745)

# 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**

# Convertisseur Analogique/Numérique ou CAN

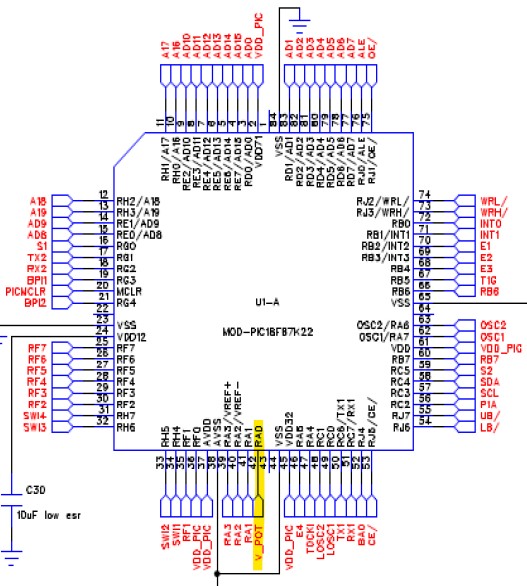
## Préparation

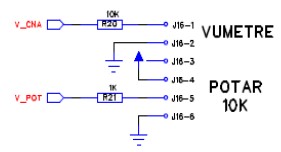
Le CAN utilisé est directement intégré à l’intérieur du microcontrôleur PIC18F87K22. Le but de notre CAN est de lire une tension sur un des ports du microcontrôleur, afin d’une connaitre la valeur. Notre CAN est sur du 12 bits. Il peut donc lire 4096 valeurs sur la plage GND 🡪 Vref. Nous obtenons ainsi les caractéristiques suivantes :

* **Tension de référence** Vref = 5V
* **Précision** : P = 5V / 4096 => quantum de 1.22 mV
* **Formule de convention entre ADC vers donnée en tension**: Vout = P \* CAN\_value

Ces éléments seront à programmer dans le code plus tard.

Nous pouvons lire sur le schéma électrique de la mallette que la sortie du potentiomètre est connectée sur la RA0 du microcontrôleur. D’après la datasheet du PIC18F, il s’agit du PORT A0.





**Figure 2 : Schéma électrique pour utilisation du potentiomètre et CAN**

## 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.

## Fonction de lecture de valeur du CAN

Nous appelons par la suite une fonction spécifiquement développé pour nous permettre de lire la donné 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.

# Bus I2C et Convertisseur numérique/analogique (CNA)

## 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.

## 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.

# Affichage sur écran LCD

## Préparation

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

## Mise en route de l’écran LCD

Nous avons d’abord adapté l’affichage sur l’écran à notre besoin actuelle. Nous souhaitions 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.

# 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.