

# SERIE N°6 (TS) TP N°1



## PRESENTATION GENERALE

Partie abordée ou système support:

**VOLTMETRE NUMERIQUE**

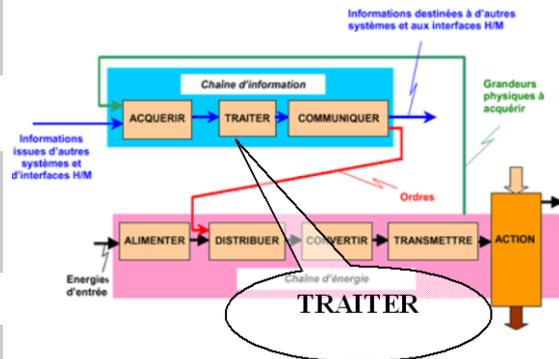
Intitulé du TP

**MISE EN ŒUVRE D'UN CONVERTISSEUR A/N**

Durée du TP

**2 H**

Axe(s) mis en œuvre par le TP :



## DONNEES PEDAGOGIQUES

Centre d'intérêt :

**CI.10 : TRAITEMENT DE L'INFORMATION (thème I11).**

Compétences attendues :

**Modifier la spécification comportementale à l'aide d'un éditeur, générer automatiquement le programme.**

Savoirs et Savoir-faire associés :

**B4 – Traiter l'information**

Pré-requis :

**Cours sur le CAN.**

## DONNEES TECHNIQUES

Environnement matériel et logiciel nécessaire :

**Un ordinateur muni des logiciels Code Vision et AVR studio.  
Un carte de développement STK500, une carte LCD, une carte d'interfce analogique.**

Documents à utiliser :

**Documentation des logiciels Code Vision et AVR studio.  
Listing du programme en langage C.**

**Introduction :** Lors de l'étude qui va suivre, vous allez mettre en œuvre un voltmètre numérique permettant des mesures dans la plage 0-5v. Ce voltmètre utilise comme unité de traitement le microcontrôleur ATMEGA163 associé à un afficheur à cristaux liquides et à une carte d'interface pour le prélèvement de la tension. En plus de ses capacités de calcul on exploitera particulièrement son convertisseur analogique numérique intégré. L'entrée de la tension à mesurer se fera sur la broche 0 du port A (PA0).

### Préparation :

- **Raccorder** les cartes : afficheur sur le port C, interface sur le port A.
- **Alimenter** les ainsi que le kit STK500.
- **Charger** le projet « voltmetre » sous Code Visio. **Exécuter** le programme.

### Manipulations et relevés :

- Pour dix valeurs de la tension d'entrée ( $V_e$ ) convenablement réparties entre 0 et 5v **relever** les indications de l'afficheur ( $V_{affich}$ ) et **regrouper** les résultats sous la forme d'un tableau.

N.B: relever  $V_e$ , entre l'entrée E1 et la masse, avec précision.

### Analyses et justifications :

- L'algorithme correspondant au programme est donné en [annexe 1](#). **Repérer** en les surlignant de couleur sur le listing ([annexe 2](#)) du programme en C, les instructions correspondant à chacun des éléments de l'algorithme.

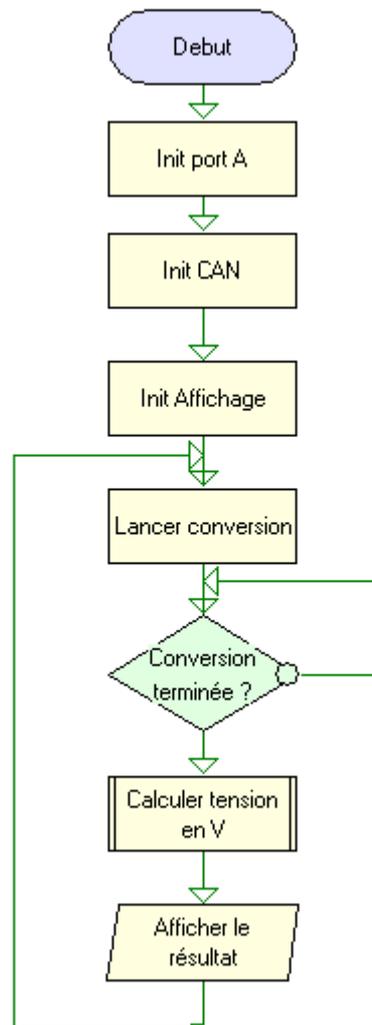
Pour les questions suivantes il sera intéressant de se référer à l'aide en ligne du logiciel.

- **Expliquer** les instructions  $tension\_int = tension\_int + 256*ADCH$
- **Expliquer** les instructions  $tension\_float = (float) tension\_int$ . Pourquoi doit on convertir (trans typer) l'entier en float ?
- **Expliquer** l'intérêt de faire appel à la fonction "ftoa".

On veut afficher à la suite de l'indication "Voltmètre" le résultat de la conversion (en décimal).

- **Proposer** une solution, puis **vérifier** sa validité en modifiant convenablement le programme proposé.

## Document annexe n°1 : ALGORIGRAMME -



## Document annexe n°2 : LISTING DU PROGRAMME EN LANGAGE C

```
#include <mega163.h>           // Définition des registres
#include <stdlib.h>           // Fonctions de transtypage
#include <lcd_v3.h>           // Fonctions d'affichage LCD
#include <adc.h>              // Fonctions de gestion du CAN

// Variables globales

unsigned int tension_int;     // integer pour l'acquisition de la tension
float tension_float;         // float pour le calcul de la tension
char *tension_str="0.000"; // Pointeurs pour l'affichage de la tension

void main(void)

{

DDRA=0x00; // Port A en entrée
PORTA=0x00;
adc_init(DIV64);
channel_select(0);
lcd_init();
lcd_putsf("Voltmetre");
lcd_gotoxy(6,1);
lcd_putsf("V");

while (1)

{

    adc_start();
    while (adc_ready() == 0);

    // Calcul de la tension en volts

    tension_int = ADCL;           // Lecture octet poids faible
    tension_int = ADCL+256*ADCH;  // Concatener avec octet poids fort
    tension_float = (float) tension_int; // Convertir en float
    tension_float = (tension_float / 1023) * 5; // Calcul de la tension

    // Affichage

    ftoa (tension_float, 3, tension_str); // convertir en chaine de caractères
    lcd_gotoxy(0,1); // Curseur 1ème colonne 2ème ligne
    lcd_puts(tension_str); // Affichage de la tension

}

}
```