

Implémentation d'un « voltmètre numérique » sur un composant PSoC Application à l'affichage de la température

Pour étudier, construire et assembler les fonctions électroniques nécessaires à la réalisation d'une chaîne de transmission d'informations vous allez utiliser un composant PSoC. (**Programmable System on Chip**). Ce composant présente les particularités d'être configurable au niveau matériel (hardware) et de permettre l'intégration de fonctions **analogiques et numériques** (système mixte). Il embarque également un cœur **microcontrôleur**, ce qui permet d'ajouter des algorithmes de commande à l'ensemble.

1-Voltmètre numérique

Dans cette première partie l'objectif est de vous faire réaliser une première configuration complète d'un tel composant sur une application simple : un « voltmètre numérique ».

Pour cela vous allez :

- Réaliser la configuration logicielle de l'application en utilisant *PSoC Designer 5.4*
- Réaliser l'implantation de l'application sur un composant PSoC placé sur une carte d'évaluation développée par les enseignants de TSE.
- Vérifier le fonctionnement.

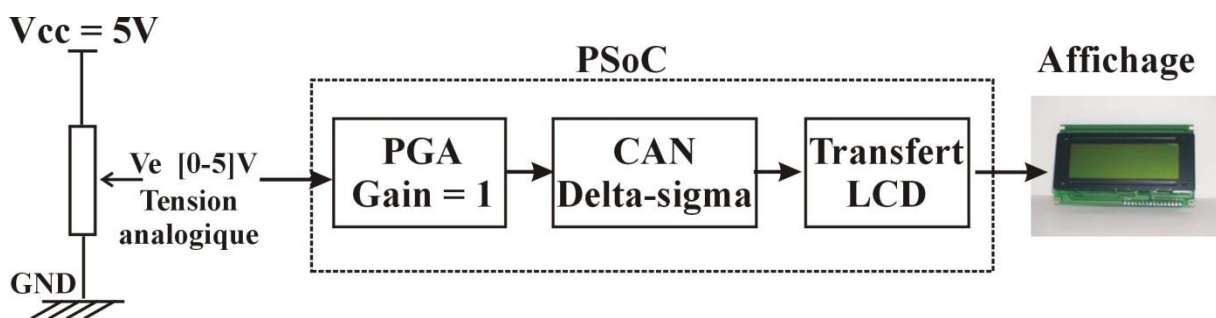


Figure 1 : Schéma fonctionnel de l'application.

Le voltmètre numérique est réalisé en utilisant trois fonctions de base :

- Un amplificateur (PGA) de gain unité pour assurer une séparation d'impédance
- Un convertisseur Analogique/Numérique 8 bits
- Un bloc de transfert pour afficher la valeur convertie en représentation hexadécimale.

Ces trois fonctions sont assurées par le PsoC placé sur la carte d'évaluation. Celle-ci contient également un potentiomètre alimenté sous 5V qui fournira la tension analogique variable en entrée, et un afficheur LCD connecté au port 2 du PsoC.

1.1 Configuration logicielle de l'application

Avec PsoC Designer 5.4, créez un nouveau projet. Placez les trois modules nécessaires à la réalisation de la fonction :

- Un amplificateur de gain programmable (PGA) disponible dans le répertoire : **Amplifiers**. Réglez les propriétés comme ci-dessous :

Name	PGA_1
User Module	PGA
Version	3.2
Gain	1.000
Input	AnalogColumn_InputMUX_0
Reference	AGND
AnalogBus	Disable

- Un convertisseur Numérique Analogique 8 Bits DelSig (DS232) disponible dans le répertoire : **ADC**. Ce convertisseur monopolise deux blocs : un analogique, un numérique. Réglez les paramètres du convertisseur comme suit et assurez-vous que les horloges des deux blocs sont les mêmes. **Attention, ici le nom du bloc a été changé à CAN, plutôt que la dénomination DELSIG8_1 par défaut.**

Name	CAN
User Module	DelSig
Version	1.50
DataFormat	Unsigned
Data Clock	VC1
ClockPhase	Normal
PosInput	ACB00
NegInput	ACB00
NegInputGain	Disconnected
PWM Output	
PulseWidth	1

- Un module LCD (disponible dans le répertoire : **MiscDigital**) qui assure la conversion de données de type texte vers un afficheur LCD à deux lignes. Par constitution, l'afficheur LCD est relié au port 2 du PSoC sur la carte d'évaluation.

Name	LCD_1
User Module	LCD
Version	1.5
LCDPort	Port_2
BarGraph	Disable

Reliez le multiplexeur d'entrée de la colonne (Analogcolumn_InputMUX_0) au port **P0(7)** du PsoC. Reliez la sortie du PGA, à l'entrée du convertisseur A/N, et réglez les ressources globales comme suit :

Power Setting [V	5.0V / 24MHz
CPU_Clock	SysClk/8
32K_Select	Internal
PLL_Mode	Disable
Sleep_Timer	512_Hz
VC1= SysClk/N	16
VC2= VC1/N	16
VC3 Source	VC2
VC3 Divider	256
SysClk Source	Internal
SysClk*2 Disable	No
Analog Power	SC On/Ref Low
Ref Mux	(Vdd/2)+/-(Vdd/2)
AGndBypass	Disable
Op-Amp Bias	Low
A_Buff_Power	Low
SwitchModePump	OFF
Trip Voltage [LVC	4.81V (5.00V)
LVDThrottleBack	Disable
Watchdog Enable	Disable

Une fois cette configuration effectuée générez l'application.





Un fichier main.c est généré. Il est accessible dans l'arborescence des fichiers projets. Ouvrez-le et insérez le code ci-dessous :

```
void main(void)
{
    BYTE valeur =0; // Initialisation de la variable valeur
    PGA_1_Start(PGA_1_HIGHPOWER); // Démarrage du PGA -- Mise sous tension
    LCD_1_Start();// Démarrage du LCD
    CAN_Start(CAN_HIGHPOWER);// Démarrage du Convertisseur A/N
    CAN_StartAD(); // Démarrage de la conversion
    M8C_EnableGInt;// Activation des interruptions

    LCD_1_Position(0,1); // Curseur du LCD sur la ligne 0 colonne 1
    LCD_1_PrCString("Tension"); // Ecriture du mot "tension"
    LCD_1_Position(1,1);
    LCD_1_PrCString("V = 0x");

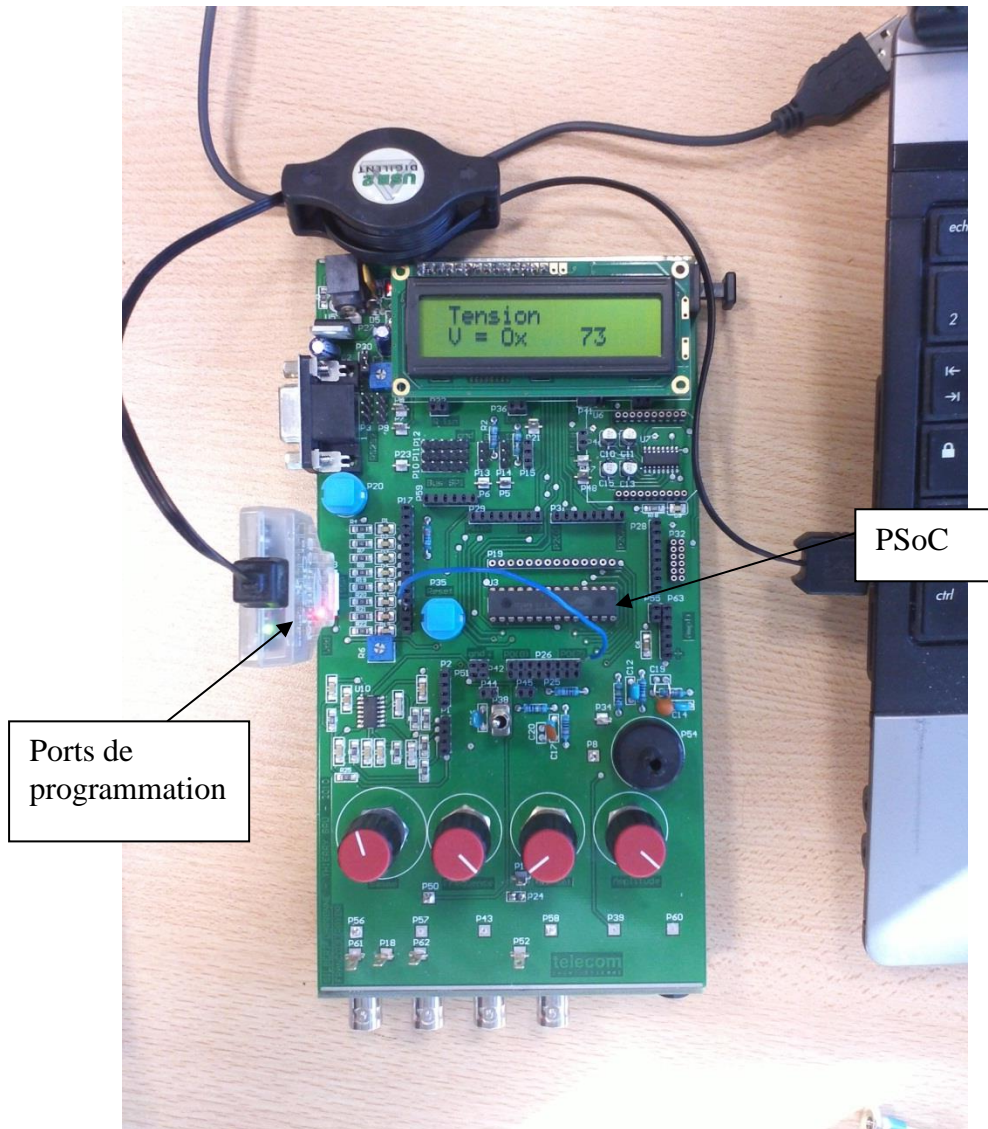
    while(1) // boucle infinie
    {
        if(CAN_fIsDataAvailable()) // attente que la donnée du CAN soit prête
        {
            valeur=CAN_cGetDataClearFlag(); //chargement de la donnée convertie
            LCD_1_Position(1,11);
            LCD_1_PrHexByte(valeur);// affichage de la donnée en hexadécimal
        }
    }
}
```

Compilez le code , vérifiez qu'il n'y a pas d'erreur et assemblez le tout . Cette dernière opération génère un fichier de type « .hex » adapté à être chargé sur le composant PSoC.

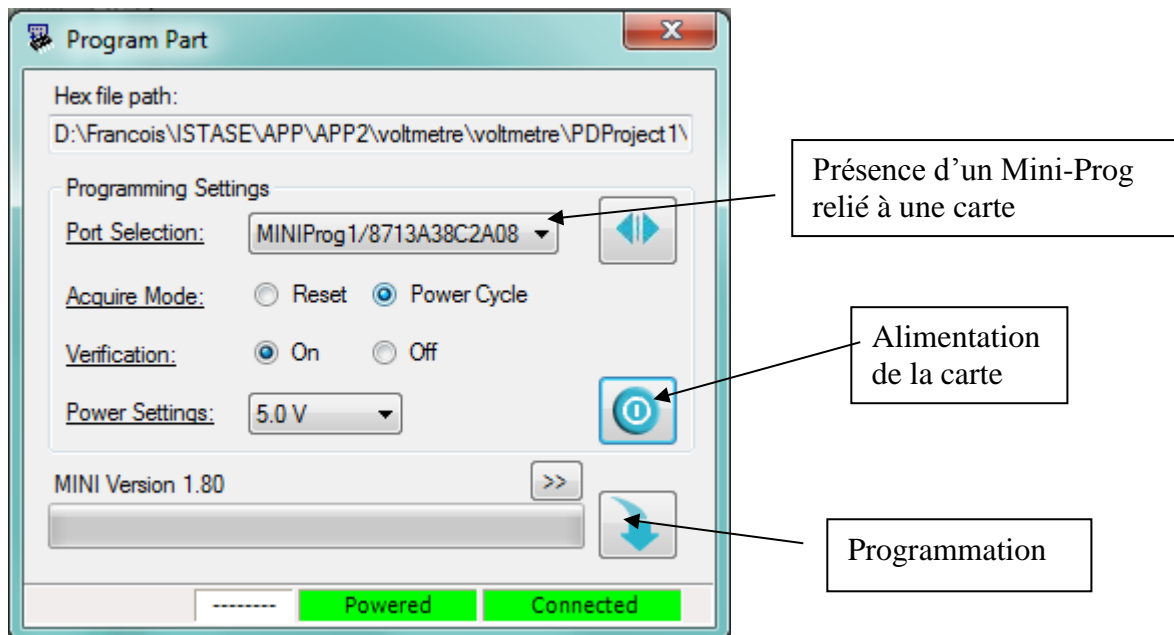
1.2 Implantation de l'application

Le fichier « .hex » conçu à l'étape précédente doit être chargé sur le composant afin que celui-ci réalise l'application « Voltmètre numérique » souhaitée.

Le PsoC est au cœur d'une carte d'évaluation développée à TSE. Autour du composant cette carte réunit différentes fonctionnalités (Tension continue d'entrée, LED d'affichage, LCD, Liaison RS232, Liaison fibre optique, communication sans fil.....) afin de réaliser de nombreux types de démonstrations.



L'étape de programmation se fait à l'aide de PsoC Designer en lançant le Programmeur dans : Programm=> Programm Part.



Reliez la carte au PC en Utilisant le Mini-Prog sur Port USB. Lancez la programmation et vérifiez qu'elle a été réalisée avec succès.

1.3 Vérification du bon fonctionnement

La tension d'entrée du voltmètre est fournie par la carte d'évaluation, au travers d'une tension variable (Voir le plan de la carte disponible sur Mootse).

Utilisez un fil pour relier cette tension au port P0(7) du PsoC. (voir photo précédente).

Alimentez la carte par le port USB du PC, et vérifiez le bon fonctionnement de l'ensemble en modifiant la valeur de la tension d'entrée.

Q : Quelle est la gamme de tension en entrée ?

Q : Comment varie le signal numérique affiché lorsque l'entrée varie sur l'ensemble de la gamme. Sur combien de valeurs différentes le signal d'entrée est-il représenté lors de cette variation ?

2- Thermomètre

L'objectif de cette deuxième partie est d'utiliser une l'application proche de la précédente pour réaliser **l'affichage de la température ambiante de la pièce sur le LCD**. Pour cela on utilisera un capteur de température TMP 36, donc les caractéristiques principales sont données à la fin du document.

Il s'agit ici de s'intéresser plus particulièrement à la **résolution** de la chaîne de mesure. Celle-ci est définie comme la **variation minimale de température détectable par la chaîne d'acquisition**.

La mise en œuvre se fait à partir d'un projet fourni que vous allez modifier pour faire évoluer les caractéristiques de la chaîne.

2.1. Téléchargez depuis Mooste le fichier Temperature.zip. De-zippez-le sur le bureau dans un répertoire portant le même nom.

Ouvrez le projet Temperature.app., et programmez votre carte avec ce projet et constatez que l'affichage contient trois parties : la donnée convertie en décimale, la tension d'entrée du composant (fournie par le capteur), et la température en degrés.

2.2. Le capteur de température est un composant à trois pattes : 2 doivent être reliées à l'alimentation (PIN 1 et 3) et la troisième (PIN 2) fournie une tension image de la température avec une loi de variation linéaire donnée en annexe.

Alimentez ce capteur avec les broches (GND et Vcc) de la carte, et reliez la sortie de ce capteur au port P0_7 du PSoC.

Vérifiez que l'affichage de la température est cohérent. Chauffez le capteur avec votre main et constatez l'élévation. Analysez le code, et expliquez le rôle des lignes suivantes :

```
x = valeur ;  
x1 = (4.8*x)/256 ; // calcul de la tension fournie par le capteur  
T=100*(x1-0.5) ; // calcul de la température
```

Vérifiez que la variation minimale de température obtenue (**résolution** de la chaîne) est de **l'ordre de 2°C**. Sachant que la configuration est ici faite d'un amplificateur de gain 1 et un convertisseur 8 bits, **justifiez cette valeur**.

2.3. Modifiez la valeur du gain à 2, modifiez le calcul de la tension en conséquence, et déterminez la nouvelle valeur de la résolution. Expliquez la valeur obtenue

Toute modification de l'application nécessite de reprendre toutes les étapes de compilation et programmation de la carte.

2.4. Modifiez le convertisseur pour un convertisseur 10 bits. Pour cela, faites un clic-droit sur le bloc CAN et sélectionner « Selection Options », puis choisissez 10 bits. Modifiez le code pour tenir compte de cette modification, car le format de la donnée convertie a changé : voir la déclaration de la variable valeur, et le chargement de la donnée convertie dans la boule infinie.

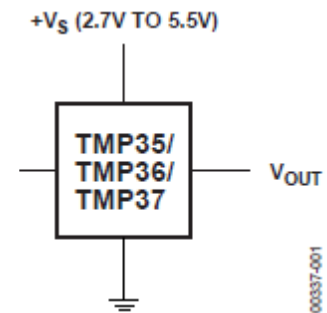
De même, modifiez le calcul de la tension en conséquence.

Programmez la carte avec la nouvelle application, puis déterminez la nouvelle valeur de la résolution, et expliquez la valeur obtenue.

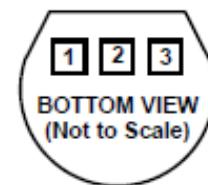
En déduire une expression générale de la résolution pour une chaîne de mesure de ce type.

Extrait des données constructeur du TMP 36

- Low voltage operation (2.7 V to 5.5 V)
- Calibrated directly in °C
- 10 mV/°C scale factor (20 mV/°C on TMP37)
- ±2°C accuracy over temperature (typ)
- ±0.5°C linearity (typ)
- Stable with large capacitive loads
- Specified -40°C to +125°C, operation to +150°C
- Less than 50 µA quiescent current
- Shutdown current 0.5 µA max
- Low self-heating
- Qualified for automotive applications



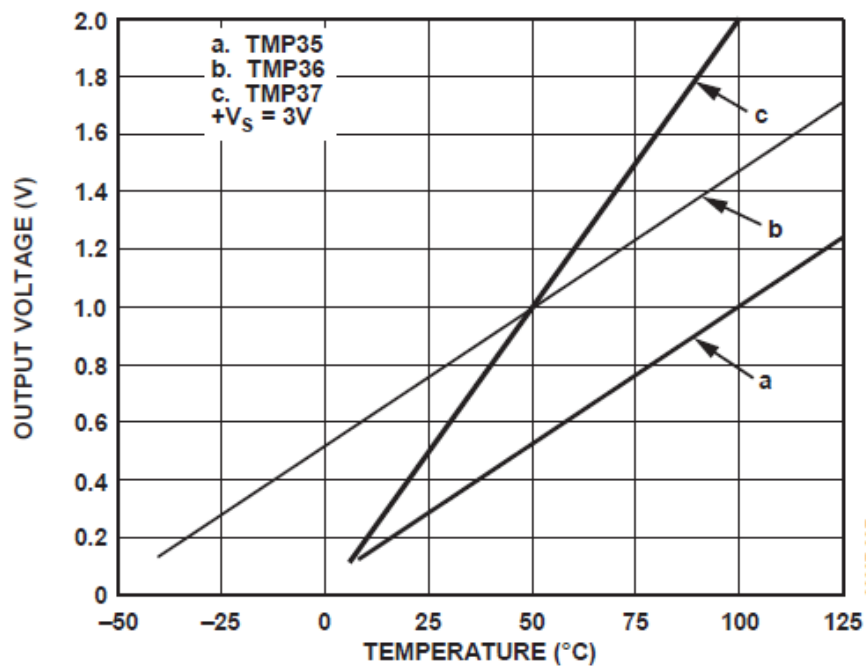
000337-001



PIN 1, +V_S; PIN 2, V_{OUT}; PIN 3, GND

000337-004

Figure 4. T-3 (TO-92)



000337-007

Figure 6. Output Voltage vs. Temperature