

1 ACQUISITION DE TEMPERATURE AVEC LE CAPTEUR LM335

Ce projet permet d'afficher la température ambiante d'un système en degré Celsius, à l'aide d'un capteur analogique de référence « LM335 » et d'indiquer par un voyant rouge le dépassement des seuils haut et bas d'une plage de température. Le capteur LM335 est constitué d'un semi-conducteur dont la caractéristique est de fournir une tension directement proportionnelle à la température exprimée en Kelvin « Tk ». La résolution du capteur est de **10 mV/Kelvin**.

Pour réaliser la mesure de température et l'afficher en degrés Celsius, le microcontrôleur doit effectuer les opérations suivantes :

- Conversion de la tension issue du capteur en grandeur numérique « N » codée sur 12 bits
- Mise à l'échelle de la grandeur numérique afin d'obtenir une température en degrés Celsius « Tc »
- Affichage du résultat dans la console toutes les secondes
- Mise en fonctionnement de la led rouge en cas de dépassement des seuils

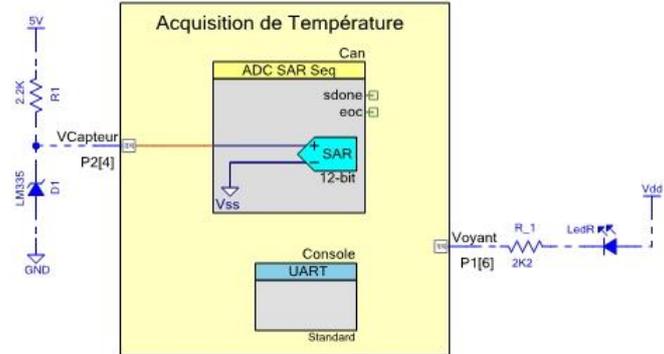
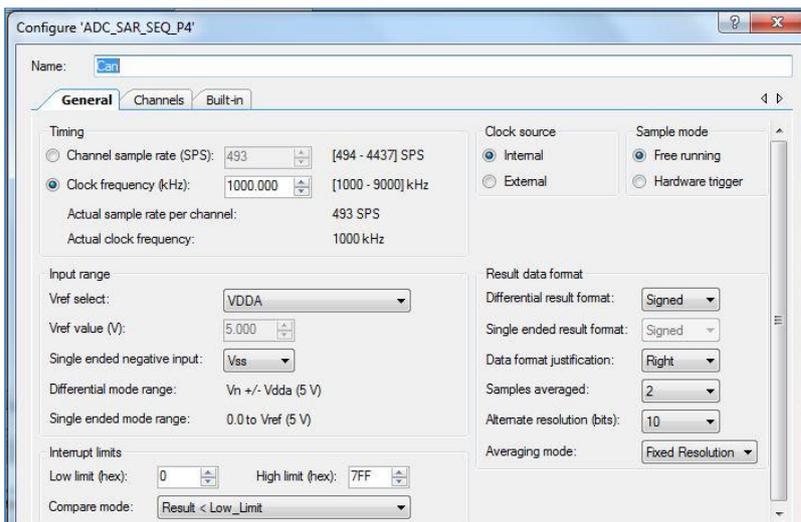


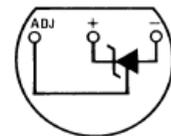
Schéma de l'application

	Name	Port	Pin	Lock
<input checked="" type="checkbox"/>	\Console: rx\	P0 [4]	28	<input checked="" type="checkbox"/>
<input checked="" type="checkbox"/>	\Console: tx\	P0 [5]	29	<input checked="" type="checkbox"/>
<input checked="" type="checkbox"/>	VCapteur	P2 [4]	6	<input checked="" type="checkbox"/>
<input checked="" type="checkbox"/>	Voyant	P1 [6]	43	<input checked="" type="checkbox"/>

Affectation des broches d'entrées/sorties



Configuration du module d'acquisition



Brochage LM335
Vue de dessous

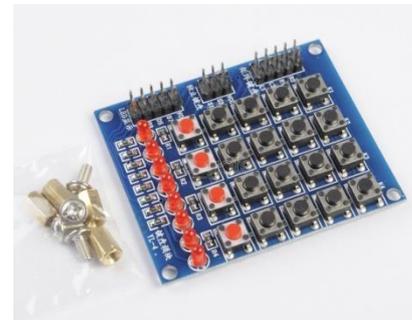
<i>Fonction</i>	<i>Rôle</i>
<code>void Can_Start(void) ;</code>	Initialisation du module d'acquisition avec les valeurs saisies dans la boîte de configuration
<code>void Can_StartConvert(void) ;</code>	Déclenchement des acquisitions en mode « free running »
<code>int16 Can_GetResult16 (uint32 channel) ;</code>	Lecture de la valeur de conversion correspondante à une voie
<code>uint32 Can_IsEndConversion (uint32 retMode) ;</code>	Retourne l'état de la conversion
<code>int16 Can_CountsTo_mVolts (uint32 chan, int16 adcCounts) ;</code>	Retourne la valeur de la tension en millivolts d'une voie
<code>void Console_Start(void) ;</code>	Initialisation de l'UART avec les valeurs saisies dans la boîte de configuration

Travail demandé :

- 1) Ajouter le projet « Activite07 » à l'espace de travail.
- 2) Configurer l'UART en 9600 bits/s, 8 bits de données, pas de parité, 1 bit de stop.
- 3) Configurer le convertisseur A/N sur 12 bits, entrée simple, mode « free running »
- 4) Définir les affectations des broches d'entrées / sorties.
- 5) Déterminer la tension délivrée par le capteur LM335 pour une température de 20 °C
- 6) Exprimer la tension délivrée par le capteur en fonction de la valeur de conversion « N » issu du convertisseur A/N 12 bits avec une tension de référence de 5V
- 7) Exprimer l'expression de la température en °C Tc en fonction de « N »
- 8) Ecrire le programme en langage C correspondant au cahier des charges
- 9) Câbler le système hors tension (voir les documentations techniques de la carte de développement et du capteur)
- 10) Compiler le projet puis transférer le programme dans le PSoC.
- 11) Vérifier le bon fonctionnement

2 CHENILLARD GERE PAR INTERRUPTION

Ce projet consiste à réaliser un chenillard à 8 leds, comportant 16 séquences possibles. Le changement de séquence et la temporisation entre chaque pas sont saisis au clavier matricé 16 touches lors d'un appui sur le bouton poussoir de test SW2 (interruption du système). Le cadencement du chenillard est géré par un timer qui déclenche une interruption au bout d'une temporisation (multiple d'une base de temps).



N° séquence	Nombre de pas	Pas
0	8	\$01 \$02 \$04 \$08 \$10 \$20 \$40 \$80
1	8	\$80 \$40 \$20 \$10 \$08 \$04 \$02 \$01
2	13	\$03 \$06 \$0C \$18 \$30 \$60 \$C0 \$60 \$30 \$18 \$0C \$06 \$03
3	11	\$07 \$0E \$1C \$38 \$70 \$E0 \$70 \$38 \$1C \$0E \$07
4	
.....		A l'initiative du programmeur
.....	
12	
13	16	\$01 \$03 \$07 \$0F \$1F \$3F \$7F \$FF \$FE \$FC \$F8 \$F0 \$E0 \$C0 \$80 \$00
14	7	\$81 \$42 \$24 \$18 \$24 \$42 \$81
15	5	\$C3 \$66 \$3C \$66 \$C3

Tableau des séquences

Remarques :

- ⇒ La première valeur du tableau de séquence indique le nombre de pas de la séquence.
- ⇒ La base de temps est fixée à 20 ms.
- ⇒ La console permet de visualiser les valeurs saisies au clavier matricé

```
char8 Clavier_Key[4][4] = {
    {'1','2','3','A'}, // ligne 1
    {'4','5','6','B'}, // ligne 2
    {'7','8','9','C'}, // ligne 3
    {'*','0','#','D'} // ligne 4
};
```

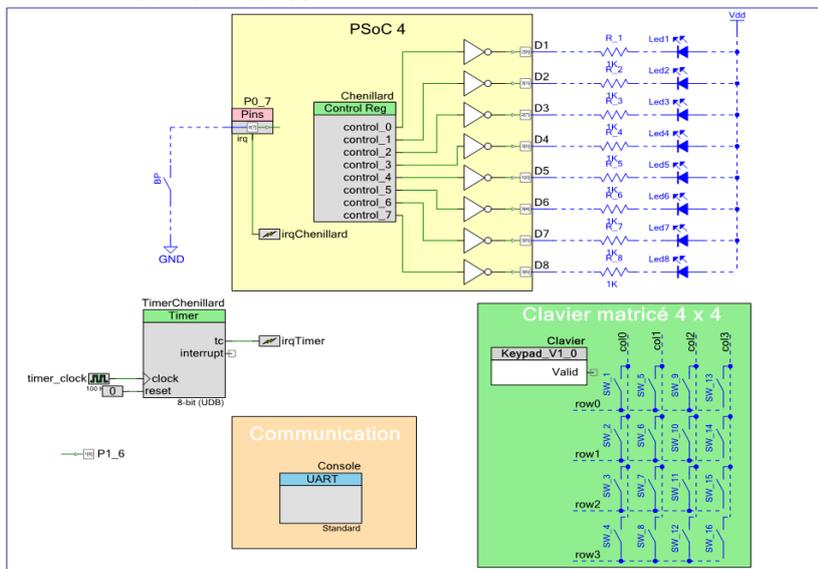
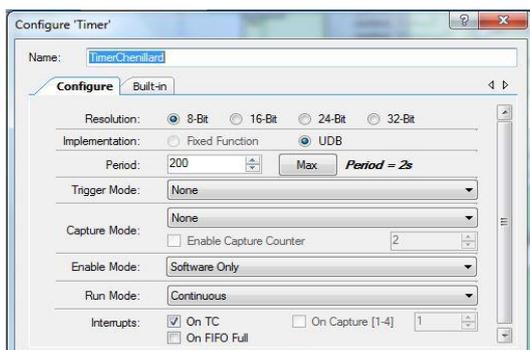


Schéma de l'application

Name	Port	Pin
\Clavier:Column[0]\	P1[3]	40
\Clavier:Column[1]\	P2[2]	4
\Clavier:Column[2]\	P1[4]	41
\Clavier:Column[3]\	P2[3]	5
\Clavier:Row[0]\	P1[1]	38
\Clavier:Row[1]\	P2[0]	2
\Clavier:Row[2]\	P1[2]	39
\Clavier:Row[3]\	P2[1]	3
\Console:rx\	P0[4]	28
\Console:tx\	P0[5]	29
D1	P2[6]	8
D2	P2[7]	9
D3	P3[6]	17
D4	P1[0]	37
D5	P3[4]	15
D6	P3[5]	16
D7	P3[0]	11
D8	P0[0]	24
P0_7	P0[7]	31
P1_6	P1[6]	43

Affectation des broches



Configuration du timer

Travail demandé :

- 1) Ajouter le projet « Activite08 » à votre espace de travail.
- 2) Configurer les entrées et les sorties.
- 3) Définir les affectations des broches d'entrées / sorties.
- 4) Ecrire les sous-programmes d'interruption du timer et du bouton poussoir en langage C
- 5) Compiler le projet puis transférer le programme dans le PSoC.
- 6) Vérifier le bon fonctionnement