Systèmes Embarqués

Journal - TP.06 : Mini-projet Thermo-Buzzer

Temps effectué hors des heures de classe

Nous avons effectué 4h de plus en dehors des deux périodes mises à disposition pour ce TP.

Synthèse des acquis

Acquis :
- Base du langage C
- Utilisation du thermomètre

Acquis, mais à exercer encore :
- Initialisation du PWM
- Pointeurs de fonction

Questions

Quelle est la fonction des cinq registres internes du thermomètre TMP102 ?

- **T_{HIGH}** : Il permet de définir la température déclenchant l'alarme.
- **T_{LOW}** : Afin d'éviter une "oscillation" du signal d'alarme, celle-ci ne s'arrête pas si la température ne descend pas en dessous d'un seuil, paramétrable. Ce registre définit ce seuil.
- **Configuration :** Registre en lecture et écriture permettant de définir le mode dans lequel fonctionne le thermomètre :
  - Shutdown mode : Éteint le thermomètre
  - Thermostate mode : Le thermomètre est soit en comparateur ou en interruption. Il travaille avec les registres T_{HIGH} et T_{LOW}.
  - Polarity : Définis si la pin ALERT est active à 1 ou à 0.
  - ...
- **Temperature :** Registre en lecture seule comprenant la valeur de la température.
- **Pointeur :** Permets de définir quel registre doit être lu ou écrit.

Comment le TMP102 génère-t-il le signal d'alarme ALERT ?

Selon le mode choisi dans le registre *Configuration*, la sortie est active soit à 1, soit à 0. Dès que la température du thermomètre dépassera le seuil fixé dans le registre T_{HIGH}, le TMP102 générera le signal sur la pin ALERT en fonction du paramètre configuré.

Pour quelles raisons le TPM102 dispose d'un registre T_{HIGH} et T_{LOW} ?

Le registre T_{HIGH} permet de définir à partir de quelle température le TMP102 devra enclencher son ALERT. Le registre T_{LOW} permet d'éviter des oscillations de la température autour du seuil haut, ce qui "enclencherait/déclencherait" en boucle l'ALERT. Grâce à ce registre, l'alerte ne s'arrête pas tant que la température n'est pas passée en dessous de la valeur T_{LOW}.

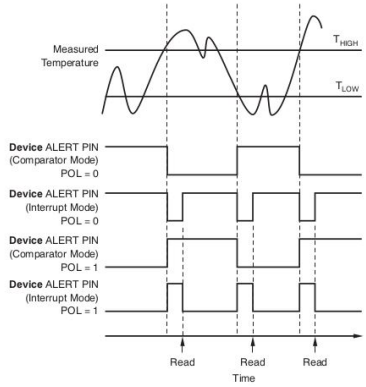

FIGURE 1 – Fonctionnement de l'alarme

Quels sont les domaines d'application des pointeurs de fonctions ?

Les pointeurs de fonction permettent de passer une fonction en paramètre d'une autre fonction. Cela permet également de retourner une fonction comme retour de fonction.
Une autre utilisation des pointeurs de fonction est leurs stockages dans un tableau, ce qui permet ensuite d'appeler ces fonctions grâce à l'utilisation des index du tableau. Un exemple de cette utilisation serait l'appel de différentes fonctions les une après les autres.

Comment déclare-t-on un pointeur de fonction ?

Il existe deux façons de le faire, la deuxième est plus souvent utilisée, car plus courte à écrire.

```
void aFunction(int a, int b) {
  // Some code
}

void (*functionPointeur1) = &(aFunction);
void (*functionPointeur2) = aFunction;
```

Comment utilise-t-on un pointeur de fonction ?

```
void printHello(void){
     printf("Hello\n");
}

int main (void){
     void (*functionPointeur)(void);          // Pointeur declaration
     functionPointeur = printHello;           // Initialisation

     (*functionPointeur)();                   // Call printHello()

     return 0;
}
```

```
double addition(double n1, double n2){
    return n1 + n2;
}
double soustraction(double n1, double n2){
    return n1 - n2;
}
double multiplication(double n1, double n2){
    return n1 * n2;
}
double division(double n1, double n2){
    return n1 / n2;
}

double (*functionTable[4])(double ,double) = {additiona,
                                              soustraction,
                                              multiplication,
                                              division};

double sum = functionTable[0](2, 3);          // sum = 2 + 3 = 5
double division = functionTable[3](4, 2);     // division = 4/2 = 2
```

Comment le compilateur implémente-t-il un pointeur de fonction en assembleur ?

Le compilateur stocke dans un registre l'adresse mémoire à laquelle le code de la fonction est enregistré, puis grâce à une instruction *blx*, celui-ci exécute le code de cette fonction.

Feedback

Comme pour le TP précédent, il est plus intéressant de travailler sur un projet plus conséquent et cela sur plusieurs séances de TP. Il est intéressant de pouvoir combiner tous les éléments que nous avons implémentés pour les précédents TP, tout en ajoutant les composants thermomètre et buzzer.