



Haute école d'ingénierie et d'architecture Fribourg
Hochschule für Technik und Architektur Freiburg

Systeme embarqués

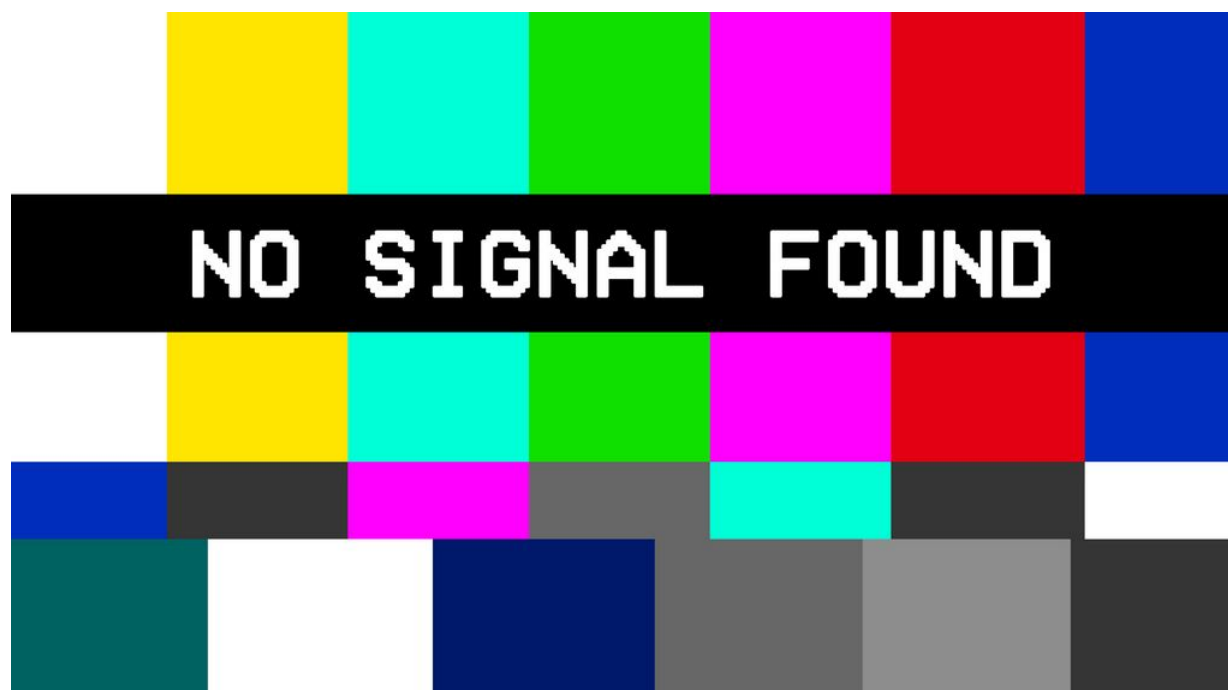
Auteurs :

Marc ROTEN

Sven ROUVINEZ

Professeur :

Daniel GACHET



16 décembre 2018

Table des matières

1	Heure de travail	2
2	Introduction	2
3	Synthèse	2
4	Quelle est la signification du qualificatif volatile et quelle est son utilité quand il est associé à un pointeur ?	2
5	Comment sont placés les champs (membres) d'une structure dans la mémoire ?	3
6	Comment peut-on efficacement définir les registres d'un contrôleur de périphérique situés dans l'espace d'adressage du μ P ainsi que leur contenu en C ?	3
7	Comment peut-on accéder ces registres ?	3
8	Comment générer des nombres aléatoires ?	4
9	A la fréquence maximale (24MHz), le compteur du timer ne permet de compter le temps que sur un intervalle de 3 minutes environ. Décrivez l'algorithme à mettre en place si l'on souhaite compter sur plusieurs années avec la même granularité	4
10	Conclusion	5

1 Heure de travail

12 heures

2 Introduction

Dans ce travail-ci, nous devons utiliser un nouveau composant, le DMTIMER, et d'autres composants tels que les boutons et l'écran LED. Le but avec ce DMTIMER est de faire un minijeu dont le but sera de tester nos reflexes. On devra presser le bouton S2, et dès que le signal se relâche le bouton S2 apparaît, on relâche le dit-bouton et la différence de temps entre les deux sera le temps que l'on affichera sur le LCD.

3 Synthèse

Sven *Acquis*

- Déclaration de struct pour travailler avec un timer
- Initialisation et écriture sur l'écran OLED

Marc *Acquis*

- Découpage et réflexion d'un problème grâce à une machine d'état déterministe (FSM).
- Lecture de documentation et compréhension du fonctionnement à partir de cette dite documentation.
- Compréhension du fonctionnement de DMTIMER des boutons et du display LCD.
- utilisation de la fonction Debug d'Eclipse sur notre BeagleBone.

4 Quelle est la signification du qualificatif volatile et quelle est son utilité quand il est associé à un pointeur ?

Permet d'indiquer au compilateur de ne pas faire d'optimisation

L'utilité de lier avec un pointer et de ne pas dépendre de l'exécution du code pour que la valeur change, dans le cas de notre compteur, par exemple, même si le processus travaille sur autre chose, notre timer va continuer à s'incrémenter

5 Comment sont placés les champs (membres) d'une structure dans la mémoire ?

De façon continue selon la taille du type choisit

6 Comment peut-on efficacement définir les registres d'un contrôleur de périphérique situés dans l'espace d'adressage du μP ainsi que leur contenu en C ?

En utilisant des struct

```
struct timer_reg {  
    uint32_t tidr;    //00  
    uint32_t res1[3]; //04--0f  
    uint32_t tiocp_cfg; //10  
    uint32_t res2[3]; //14--1f  
    uint32_t irq_eoi;  
    uint32_t irqstatus_raw;  
    uint32_t irqstatus;  
    uint32_t irqenable_set;  
    uint32_t irqenable_clr;  
    uint32_t irqwakeen;  
    uint32_t tclr;  
    uint32_t terr;  
    uint32_t tldr;  
    uint32_t ttgr;  
    uint32_t twps;  
    uint32_t tmar;  
    uint32_t tear1;  
    uint32_t tsier;  
    uint32_t tear2;  
};
```

7 Comment peut-on accéder ces registres ?

En affectant une variable static volatile

```
static volatile struct timer_reg* dmtimer[] = {  
    (volatile struct timer_reg*) 0x48040000,  
};
```

```
(volatile struct timer_reg*) 0x48042000 ,  
(volatile struct timer_reg*) 0x48044000 ,  
(volatile struct timer_reg*) 0x48046000 ,  
(volatile struct timer_reg*) 0x48048000 ,  
(volatile struct timer_reg*) 0x4804A000  
};  
  
uint32_t dmtimer1_get_counter(enum dmtimer_timers timer){  
    volatile struct timer_reg* ctrl = dmtimer[timer];  
    return ctrl->tctr;  
}
```

8 Comment générer des nombres aléatoires ?

Cette fonction permet de générer des nombres aléatoire selon un intervalle donné

```
rand();  
uint32_t random_time() {  
    return random() % ((MAX_VALUE_TIMER + MIN_VALUE_TIMER) + 1)  
        + MAX_VALUE_TIMER;  
}
```

9 A la fréquence maximale (24MHz), le compteur du timer ne permet de compter le temps que sur un intervalle de 3 minutes environ. Décrivez l'algorithme à mettre en place si l'on souhaite compter sur plusieurs années avec la même granularité

Pour calculer le temps écoulé, on calcule le delta entre le moment où le timer a été démarré et le moment où l'on veut qu'il s'arrête.

Dans notre cas, la fréquence étant de 24Mhz, un timer peut compter jusqu'à environ 3 minutes donc pour aller plus loin il faut faire un pointage maximum jusqu'à 3 minutes afin d'éviter de louper une interruption, et donc d'incrémenter une variable qui garde le nombre de fois que l'on a contrôlé notre timer. Pour calculer le delta, il faut utiliser :

$$\Delta = sp - st$$

, où sp est le stop time et st le start time

10 Conclusion

On a pu au travers de ce TP, concevoir les blocs DMTIMER et rajouter les fonctions `displayString` et `displayChar` dans notre fichier `display.c`. Ce TP s'est bien passé et était bien documenté.