

Proposition de correction du TP 3

Informatique Embarquée

Ce document propose des réponses pour certaines questions du TP. Les réponses aux autres questions sont fournies dans des fichiers à part.

Partie 2

Question 1

Pour activer la gestion globale des interruptions, il faut exécuter l'instruction `sei` qui *set* le bit `Interrupt Enable` du registre `SREG`.

Avec `avr-libc`, la macro `sei`, définie dans `avr/interrupt.h`, est remplacée par cette instruction :

```
# define sei() __asm__ __volatile__ ("sei" ::: "memory")
```

Question 2

Le timer 8-bit `Counter0` émet une interruption `TOV` à chaque fois que son compteur passe de `TOP` (= `MAX` = `0xff` = 255 en mode “normal”) à `BOTTOM` (= 0). Le compteur doit donc être incrémenté 256 fois entre chaque interruption. La fréquence f de l'horloge utilisée par le timer est de 16 MHz, c'est à dire que le compteur est incrémenté tous les $t = \frac{1}{f}$ secondes, soit ici $t = 6.25 \times 10^{-8}$ s. Le temps entre deux interruptions est donc :

$$t_{tov} = 256 \times t = 1.6 \times 10^{-5} s$$

Le temps entre deux interruptions `TOV` est donc égal à 16µs.

Question 3

En configurant un *prescaler*, il est possible de réduire la fréquence à laquelle le compteur est incrémenté. Diviser la fréquence de comptage par n , c'est multiplier par n le temps entre deux interruptions `TOV`.

En utilisant le prescaler 1024, on obtient une interruption `TOV` tous les $1024 \times t_{tov}$ s, soit environ 16ms.

Question 4

Le timer 16-bit `Counter1` émet une interruption TOV à chaque fois que son compteur passe de TOP (= MAX = 0xffff = 65535 en mode “normal”) à BOTTOM (= 0). Le compteur doit donc être incrémenté 65536 fois entre chaque interruption. La fréquence f de l’horloge utilisée par le timer est de 16 MHz, c’est à dire que le compteur est incrémenté tous les $t = \frac{1}{f}$ secondes, soit ici $t = 6.25 \times 10^{-8}$ s. Le temps entre deux interruptions est donc :

$$t_{tov} = 65536 \times t = 0.004096s$$

Le temps entre deux interruptions TOV est donc égal à 4.096ms.

En utilisant le prescaler 1024, on obtient une interruption TOV tous les $1024 \times t_{tov}$, soit environ 4.1s.

Question 5

En mode CTC, il est possible de paramétrer la valeur de TOP ($0 \leq \text{TOP} \leq \text{MAX}$). On souhaite qu’une interruption soit émise toutes les 500ms. Il faut trouver un *prescaler* avec lequel il est possible d’atteindre au moins 500ms en comptant jusqu’à MAX.

Les prescalers proposés sont /8, /64, /256 et /1024.

- Avec le prescaler /8, on obtient $8 \times t_{tov} \approx 0.033$ soit environ 33ms. Ce n’est pas suffisant.
- Avec le prescaler /64, on obtient $64 \times t_{tov} \approx 0.262$ soit environ 262ms. Ce n’est pas suffisant.
- Avec le prescaler /256, on obtient $256 \times t_{tov} \approx 1.049$ soit environ 1s. Ce prescaler permet d’obtenir une interruption toutes les secondes. En configurant la valeur de TOP, on devrait pouvoir obtenir une interruption toutes les 500ms.

En notant *presc* la valeur du *prescaler*, *top* la valeur de TOP, et f la fréquence de l’horloge du compteur, on obtient la relation suivante :

$$t_{tov} = \frac{\text{presc} \times (\text{top} + 1)}{f}$$
$$\implies \text{top} = \frac{f \times t_{tov}}{\text{presc}} - 1$$

Ici :

$$\text{top} = \frac{16000000 \times 0.5}{256} - 1 = 31249$$

En configurant `OVR1A = 31249` et le *prescaler* en mode /256, on obtient une interruption TOV toutes les 500ms.

Question 6

Il est possible d’implémenter un compteur dans notre programme : à chaque interruption TOV, on incrémente le compteur. Lorsque le compteur atteint une certaine valeur, on déclenche le traitement attendu. Cela permet d’attendre une durée arbitraire, y compris avec un timer 8 bits.