

## ::: Architecture :::

semaine n°7  
Interruptions et Timers  
TD

- Nous allons réaliser un compteur de vitesse pour vélo à l'aide d'un micro-contrôleur MCS251. Le compteur est constitué d'un capteur fixé sur le bras avant du cadre et d'un déflecteur placé sur les rayons. Lorsque le déflecteur passe devant le capteur, celui-ci envoie un signal sur la broche INT1 du micro-contrôleur.
- Le temps mis par la roue pour faire un tour est mesuré par un temporisateur. Le périmètre de la roue est de 1,5 mètres.

- Dans un premier temps, le passage du déflecteur devant le capteur sera réalisé par scrutation :
  - Lorsque le capteur détecte le passage du déflecteur, doit-il envoyer 0Volts ou 5Volts sur la broche INT1# ?
  - Dans quel mode doit-on configurer la détection du signal INT1# ? Front descendant ou niveau 0 ?
  - Quel bit / registre SFR permet de détecter l'activation du signal INT1 ?
  - Nous ne souhaitons pas que ce signal déclenche d'interruption, déduisez en la procédure d'initialisation.

- Dans un premier temps, le passage du déflecteur devant le capteur sera réalisé par scrutation :
  - Lorsque le capteur détecte le passage du déflecteur, doit-il envoyer 0Volts ou 5Volts sur la broche INT1# ?
    - 0 Volt : le # à la fin du signal indique qu'il est actif à l'état bas (0 volt)
  - Dans quel mode doit-on configurer la détection du signal INT1# ? Front descendant ou niveau 0 ?
    - Le passage du capteur devant le déflecteur peut être plus long que le traitement de ce passage, pour que l'on distingue bien un nouveau passage du passage précédent, la configuration sur front est préférable.

- Quel bit / registre SFR permet de détecter l'activation du signal INT1 ?

Voir page 23, il s'agit du bit IE1 du registre TCON

- Nous ne souhaitons pas que ce signal déclenche d'interruption, déduisez en la procédure d'initialisation.

Il faut interdire les interruption sur INT1 en mettant à 0 le bit EX1. L'initialisation complète est la suivante :

```
IT1 = 1;    // détection sur front
IE1 = 0;    // met à zéro le flag de détection
EX1 = 0;    // désactive l'interruption INT1
```

- Nous allons devoir compter le temps mis par la roue pour faire un tour. Sachant que la roue mesure 1,5 mètres de périmètre et que nous souhaitons mesurer des vitesses allant de 6 à 100km/h soit ( 1,5 à 28m/s ) déduisez-en l'intervalle de temps mis par la roue pour faire 1 tour.
- Ce temps est-il compatible avec les timers que nous avons à notre disposition ?
- Proposer 2 solutions

- Déduisez-en l'intervalle de temps mis par la roue pour faire 1 tour.
  - Au minimum la roue mettra 50 ms pour réaliser un tours (à 100km/h) ; au maximum 1s (5,4 km/h).
- Ce temps est-il compatible avec les timers que nous avons à notre disposition ?
  - En dessous de 50 km/h la roue mettra plus de 100ms, hors nous avons vu que nos temporisateurs ne peuvent compter d'un coup, plus de 98ms ( $2^{16} * 1,5 * 10^{-6}$ )

► Proposer 2 solutions

- Baisser la fréquence du micro-contrôleur à 400Khz par exemple alors 1 pas de temporisateur ne sera plus 1,5us mais :  $12 / 400000 = 30\text{us}$ .

=> à 100 Km /h = 50ms/tour =  $1666 * 30 \text{ us}$   
 $< 2^{16}$

=> à 5,4 Km/h = 1s/tour =  $33333 * 30 \text{ us}$   
 $< 2^{16}$

- Mettre en place un système capable de compter plusieurs cycle de temporisation au lieu d'un seul.

- Considérons tout d'abord la première des solutions : la fréquence du micro-contrôleur est de 400KHz et la période du temporisateur sera donc de 30 us.
- Ecrire un programme de calcul sans interruptions. Il réalisera les opérations suivantes :
  - Initialisation du signal INT1
  - Initialisation du temporisateur Timer 0 et démarrage
  - Attente d'un signal du capteur
  - Arrêt relance du temporisateur et calcul de la vitesse.

```
// Init des signaux du capteur
IT1 = 1;           // détection sur front
IE1 = 0;           // met à zéro le flag de détection
EX1 = 0;           // désactive l'interruption INT1
// Init du temporisateur
TR0 = 0 ; TMOD = (TMOD & 0xF0) | 0x1; // Temporisateur 0, mode 1
TH0 = 0 ; TL0 = 0;
TF0 = 0 ; TR0 = 1;           // Démarrage
while (1) {
    // Attente active du passage du capteur
    while ( IE1 == 0 );           // Attente active
    temps=(TH0<<8)+TL0;           // calcule le temps mis avant le clear
    TH0 = TL0 = 0;               // on relance tout de suite le timer.
    if ( TF0 == 0 ) {           // Si TF0 est actif alors le timer a
                                // débordé le calcul ne sera pas possible

        // Calcul de la vitesse en m/s
        vit = 1,5 / ( ( temps * 30 E-6) ;
    }
    TF0 = 0;
    IE1 = 0;           // Reset signal IE
}
```

- Solution n°2 : Notre micro-contrôleur fonctionne à 8Mhz, les pas sont de 1,5 us. Nous devons donc mettre en place un compteur de temps permettant de couvrir l'intervalle nécessaire.
- Pour cela : écrire une fonction de temporisation sous interruption comptant les ms. Cette fonction sera déclenchée par le mécanisme des interruptions toutes les 1 ms et viendra mettre à jour une variable globale qui pourra être lue par la fonction de calcul.
  - Initialiser un timer d'une milliseconde avec interruption
  - Ecrire la fonction d'interruption qui met à jour la variable globale et relance le timer pour 1 ms.

```
int compteurMs;
```

```
void initTimer1ms() {  
    compteurMs = 0; // Compteur global  
    // 1ms = 667 * 1,5 us  
    TR0 = 0 ; TMOD = (TMOD & 0xF0) | 0x1; // Temporisateur 0, mode 1  
    TH0 = ( ( 65536 - 667 ) >> 8 );  
    TL0 = ( ( 65536 - 667 ) & 0xFF );  
    TF0 = 0 ; TR0 = 1; // Démarrage  
    ET0 = 1; // Active les interruptions  
}
```

```
_interrupt(1) void Timer1ms() {  
    compteurMs++;  
    TH0 = ( ( 65536 - 667 ) >> 8 );  
    TL0 = ( ( 65536 - 667 ) & 0xFF );  
}
```

- Ecrire la fonction main effectuant le calcul à partir de ce nouveau compteur de millisecondes.

```
void main() {  
    IT1 = 1;           // détection sur front  
    IE1 = 0;          // met à zéro le flag de détection  
    EX1 = 0;          // désactive l'interruption INT1  
    initTimer1ms();  
    while (1) {  
        while ( IE1 == 0 );           // Attente active  
        int temps = compteurMs;  
        initTimer1ms();  
        IE1 = 0;  
  
        // Calcul de la vitesse en m/s  
        vit = 1,5 / ( compteurMs * 1 E-3) ;  
    }  
}
```