

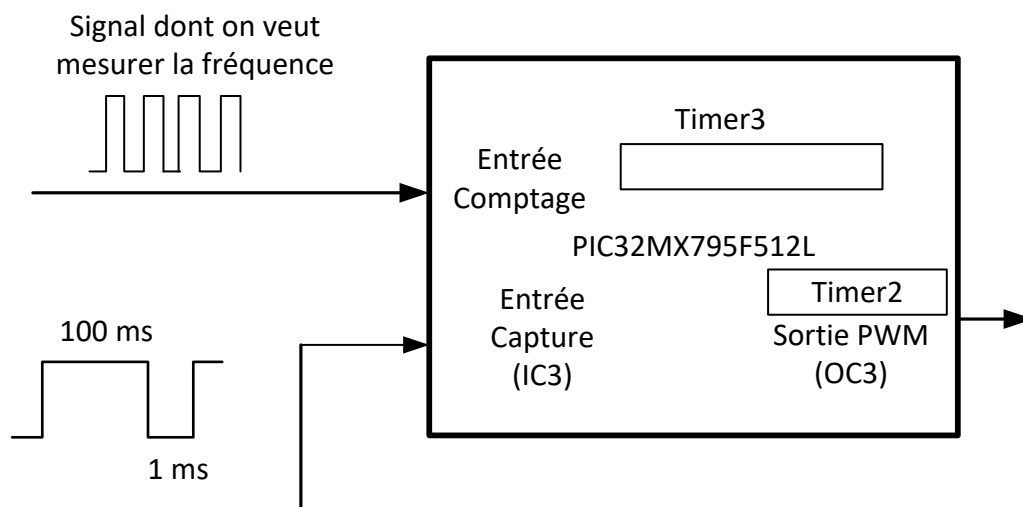
## EXERCICE 6\_2 PIC32MX

### OBJECTIFS

Cet exercice a pour objectif de permettre aux étudiants d'utiliser concrètement les timers, captures et PWM.

Après l'ébauche sur le papier, l'exercice sera réalisé pratiquement avec un kit PIC32MX795F512L.

On souhaite réaliser un fréquencemètre par comptage avec un PIC32MX795F512L configuré pour  $PB\_FREQ = SYS\_FREQ = 80\text{ MHz}$ . Utilisation du timer 3 comme compteur d'impulsions et d'une sortie PWM comme signal de base de temps pour le fréquencemètre. La base de temps du signal PWM est réalisée par le timer2.



Le signal utilisé pour le déclenchement de la capture doit provenir du module OC3 en mode PWM (base de temps timer2). Il doit être au niveau haut pendant 100 ms et au niveau bas seulement pendant 1 ms. La capture est effectuée avec le module IC3.

Il faut effectuer une capture aux deux flancs et réagir à l'interruption de chaque flanc.

En utilisant le timer2 combiné au module OC3 et en capturant le timer 3 avec le module IC3, veuillez fournir les éléments suivants :

a) A quelles broches du pic32MX795F512L (noms et n°) correspondent les 3 éléments suivants :

Entrée de comptage : .....

Entrée de capture : .....

Sortie du signal PWM : .....

b) Déterminez quelles sont les fréquences min et max que l'on peut mesurer avec ce dispositif en partant du principe que le signal à mesurer est périodique et que son rapport cyclique est de 50%. En tenant compte que le timer 3 est configuré sans pré division.

.....  
.....  
.....  
.....  
.....  
.....

c) Quelle est la précision (résolution) de mesure pour les fréquences suivantes :

10 Hz .....  
100 Hz .....  
1000 Hz .....  
10'000 Hz .....  
100'000 Hz .....

d) Déterminez pour le timer2 la valeur de prescaler et de comparaison pour obtenir la période de 101 ms. Pour OC3, déterminez la valeur "PulseWidth" pour obtenir t\_haut = 100 ms et t\_bas 1 ms

.....  
.....  
.....  
.....  
.....

e) Quelle doit être la valeur de comparaison pour que le timer 3 compte jusqu'à son maximum ? Quelles sont les particularités de la configuration pour l'utilisation en comptage externe en vue de la capture.

.....  
.....  
.....  
.....  
.....

## REALISATION PRATIQUE

Créez un projet nommé Ex6\_2 en utilisant le MHC (vous pouvez utiliser une copie de l'exercice 6\_1). Vous devez configurer les drivers suivants :

- Un driver timer statique sans interruption pour le timer 2, période 101 ms.
- Un driver timer statique sans interruption pour le timer 3, **en comptage externe**.
- Un driver OC statique pour fournir le signal demandé avec OC3.
- Un driver IC (avec IC3) avec interruption à chaque flanc, capture du timer 3, priorité 7.

☞ Il sera nécessaire de compléter la fonction DRV\_IC0\_Open() (fournie vide ! – état Harmony 2.06) : activation de l'IC, vidage du fifo des valeurs et activation interruption.

### ACTIONS DANS APP.C

Au niveau de l'application dans le case APP\_STATE\_INIT, il faut :

- lancer les timers,
- lancer l'OC,
- démarrer la capture et autoriser son interruption.

Il faut réaliser l'habituelle fonction APP\_UpdateState et aussi APP\_UpdateFreq qui permet de mettre à jour la variable globale uint32\_t FrequenceSignal.

Dans le case APP\_STATE\_SERVICE\_TASKS, il faut effectuer l'affichage de la fréquence.

### ACTIONS DANS L'ISR DE LA CAPTURE

L'ISR générée est à corriger et à compléter. Lors de l'interruption au flanc descendant de la capture, il faut effectuer le calcul de la fréquence que l'on exprimera en Hz. Réalisation des calculs avec une variable locale, appels des fonctions APP\_UpdateFreq et APP\_UpdateState.

## TEST DU SYSTEME

Il faut utiliser un générateur pour fournir le signal à mesurer et lier la sortie PWM à l'entrée de capture.

☞ Attention au réglage du générateur : Il faut garantir un signal 0 à max 3V3 avant de brancher (branchez la masse en premier !)

Vérifiez l'affichage sur le kit avec celui du générateur (oscilloscope également) en prenant des valeurs de fréquence comme 99 Hz, 999 Hz, 9999 Hz et 99999 Hz.