
Fiche explicative : Généralités sur la Pulse Width Modulation

Periph'Team - INSA de Toulouse

1 1. Généralités sur les signaux de type *PWM*

Même si ils portent un nom assez peu explicite, les signaux de type *PWM* sont des signaux simples.

1.1 Définition

Un signal *PWM* (Pulse Width Modulation) est un signal numérique **binaire**, plus précisément une succession d'impulsions qui s'enchaînent à une période donnée T_{PWM} . Chaque impulsion possède une durée qui lui est propre notée T_{pulse} ou T_h (pour durée du signal à l'état haut). Chacune des impulsions porte donc en elle une information contenue dans la durée de l'état haut qui peut aller de 0 à T_{PWM} .

La figure 1 illustre un tel signal dans l'hypothèse où l'état haut correspond à une tension de 3.3V.

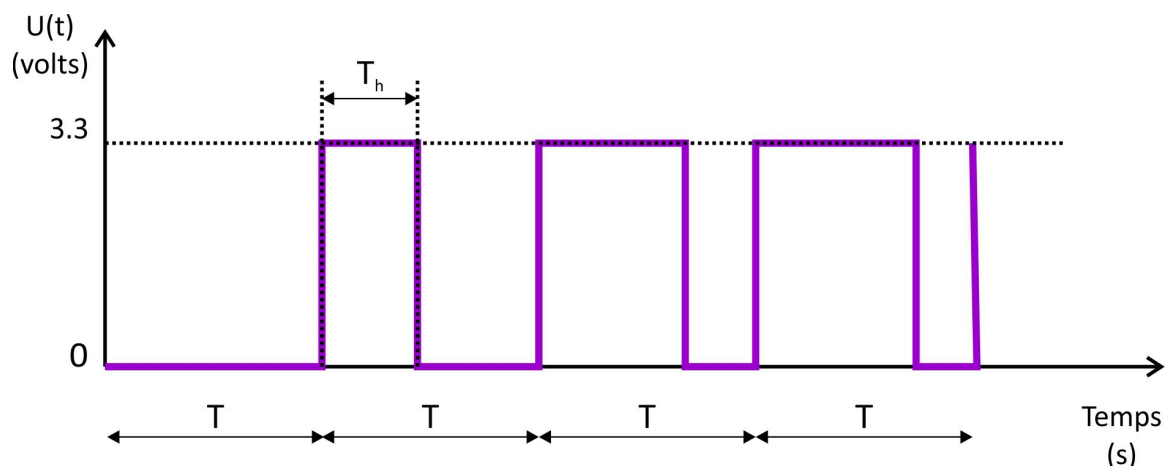


Figure 1: Le signal PWM avec T_h non fixe au cours du temps

Un tel signal s'utilise notamment dans deux cas bien distincts :

- **La transmission d'information** à l'instar de que l'on peut faire avec une modulation *AM*, *FM*. La *PWM* peut alors être vue comme une technique de transmission d'information. Chaque nouvelle impulsion est un symbole correspondant à la durée de l'impulsion T_h . Un exemple typique de cette utilisation est le signal

envoyée à un servomoteur. La durée T_h correspond à la position angulaire que l'on souhaite obtenir de la part du servomoteur. Le signal PWM code la consigne que l'on impose au système asservi.

- **La conversion numérique analogique.** Dans ce cas, le signal PWM est caractérisé par une durée d'impulsion (T_h) qui est mise à jour à période fixe T_s (avec obligatoirement $T_s \geq T_{PWM}$). La grandeur qui nous intéresse n'est plus directement T_h mais le ratio $\alpha = \frac{T_h}{T_{PWM}}$, α est appelé le rapport cyclique instantané du signal qui s'étend de 0% (signal toujours à l'état bas) à 100% (signal toujours l'état haut).

Le signal est alors systématiquement suivi par un dispositif de filtrage explicite ou implicite afin d'obtenir une grandeur qui n'est plus numérique (succession d'impulsions) mais analogique comme le montre la figure 2 pour un ratio $\alpha = 0.4$. La figure 3 montre quant à elle comment le filtre de nature passe-bas supprime dans le spectre des fréquences, les grandeurs qui ne correspondent pas à la composante continue du signal. Deux exemples d'application de ce cas d'utilisation de la PWM :

- Un convertisseur DAC (Digital Analog Converter) typiquement le schéma de la figure 2 où la grandeur analogique de sortie est une tension. Le filtrage apparaît ici de manière explicite.
- Le commande d'un moteur à courant continu. Dans ce cas, le signal est directement appliquée au moteur. C'est le moteur qui joue le rôle de filtre (il n'est naturellement pas en capacité de faire varier sa vitesse avec une grande fréquence). La sortie de type analogique du système pris dans sa globalité est la vitesse du moteur.

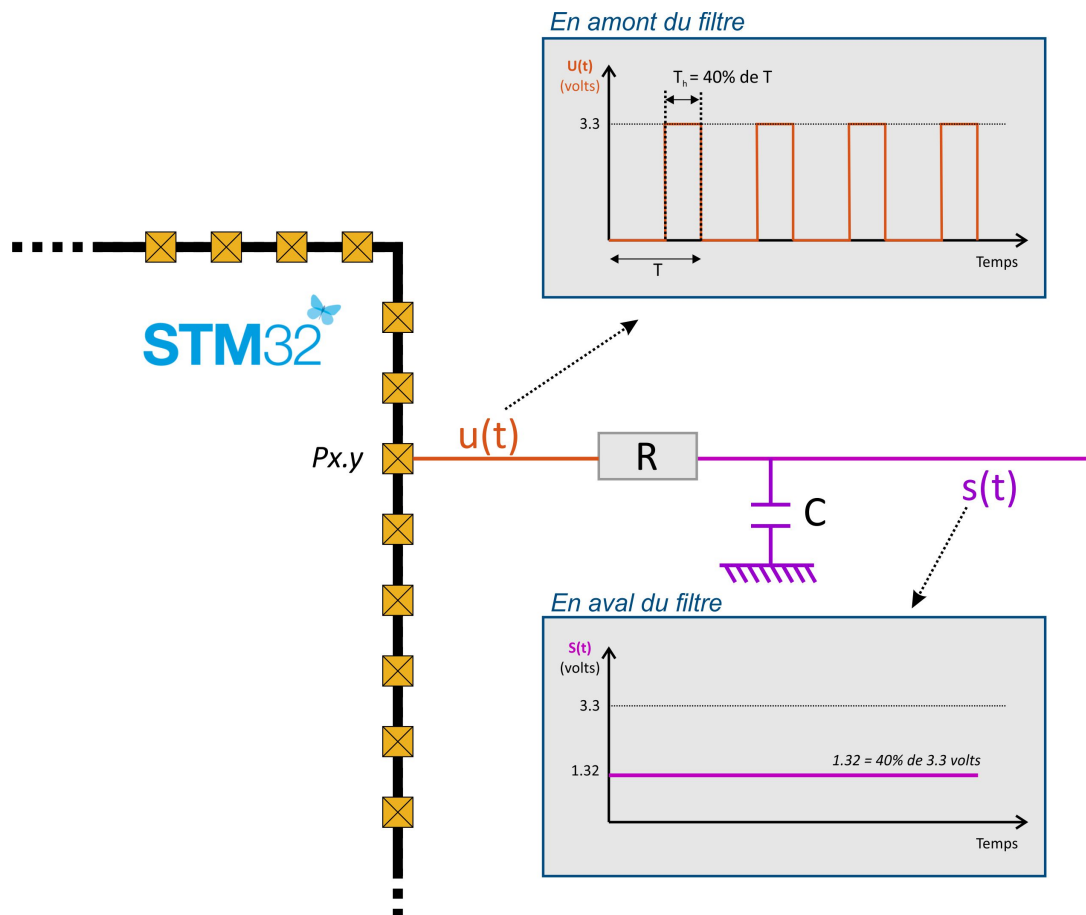


Figure 2: Signal PWM filtré

2 Mise en œuvre de la PWM dans un μ contrôleur STM32F10x

Dans un μ contrôleur, la fonction permettant d'obtenir un tel signal est systématiquement associée à un module *Timer*.

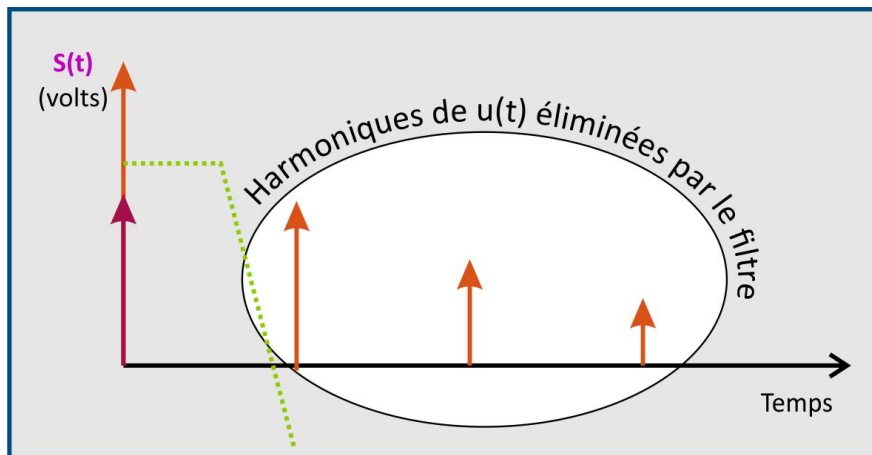


Figure 3: Signal PWM filtré - Aspect fréquentiel

Sur Moodle vous disposez d'un lien de téléchargement d'un fichier LTSpice permettant de bien comprendre le procédé de réalisation de la PWM.

La figure 4 montre l'architecture simplifiée du principe de la PWM sur STM32F10x. Sur cette figure sont représentés les 4 registres essentiels :

- les trois registres propres au *Timer* générique : le compteur *CNT*, le registre de reload *ARR*, et le prescaler contenant le diviseur de fréquence *PSC* . Ce trio permet d'obtenir le comptage "en dent de scie" et le débordement du *Timer* à la période voulue T_{PWM}
- Le quatrième registre *CCRx* est celui de la comparaison pour engendrer le signal PWM. Il contient la ratio de la valeur *ARR*, soit donc une valeur comprise entre 0 et *ARR* pour avoir rapport cyclique souhaité. Attention la valeur est obligatoirement entière ! Si, par exemple, on souhaite un ratio de 37,28% il y aura très certainement une troncature à faire avant d'écrire dans ce registre.

Remarque 1 : le signal PWM produit en sortie du Timer (1 à 4) ne peut sortir du μ contrôleur que si la pin associée (il faut trouver laquelle dans la spécification du STM32F103RB) est en mode alternate output.

Remarque 2 : la mise en œuvre demande de lire le manuel de référence (bien entendu!). On précise juste que le Timer 1 est légèrement différent et plus complexe que le 3 autres Timers. Aussi pour cette unité et pour que le signal PWM sorte bien du Timer1, le bit MOE du registre TIM1_CR2 doit être mis à 1.

Le figure 5 présente une visualisation des 4 signaux référencés dans la figure précédente et montre comment le signal final de sortie se construit à l'aide d'une simple horloge et de deux comparateurs.

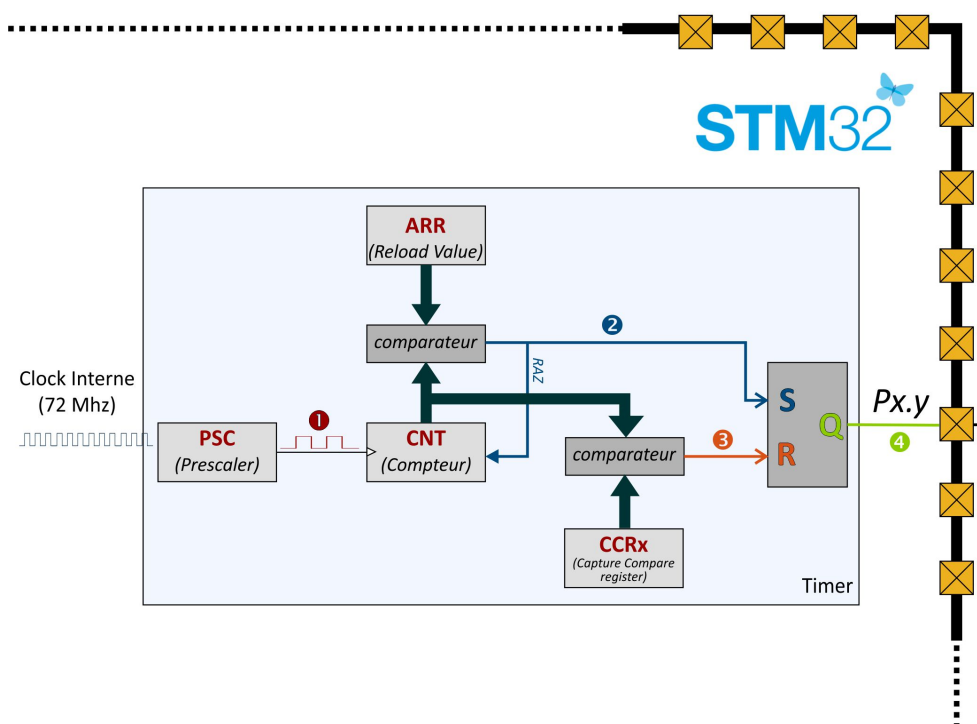


Figure 4: Architecture simplifiée de la fonctionnalité PWM d'une unité Timer

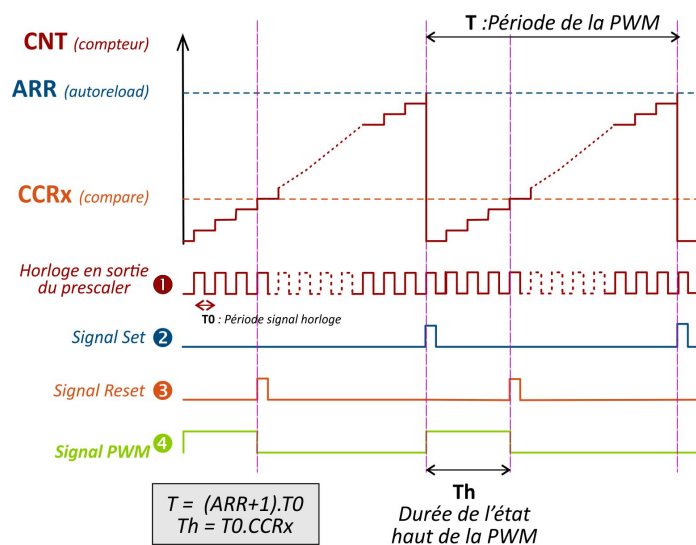


Figure 5: Architecture simplifiée de la fonctionnalité PWM d'une unité Timer