

## Travaux pratiques Systèmes d'Acquisition

### GitHub et Doxygen

Documentation Doxygen incluse au mail

[https://github.com/LE-P/SCHINI\\_PERTUET\\_TP\\_asserv\\_MCC.git](https://github.com/LE-P/SCHINI_PERTUET_TP_asserv_MCC.git)

### Configuration des pins

Pin STM	Nom Pin STM	Pin Hacheur
PA12	TIM1_CH2N	R_BOT (31)
PA11	TIM1_CH1N	Y_BOT (30)
PA9	TIM1_CH2	R_TOP (13)
PA8	TIM1_CH1	Y_TOP (12)
PC3	ISO_RESET	
PA0	ADC1_IN	35
PA1	TIM2_CH2	A
PA15	TIM2_CH1	B

### Commande MCC Classique

#### Génération de PWM

Afin de générer les signaux PWM pour le moteur, on utilise le Timer 1 en mode Center Aligned. Afin d'avoir une fréquence de 15625Hz à partir d'une

fréquence de 160MHz, on choisit un ARR de 1024-1 et un PSC de 10-1. On inclut aussi un temps mort de 200µs afin de ne pas court-circuiter le moteur. Cette valeur d'ARR nous permet d'avoir une résolution de 10 bits.

## Signal de commande Start

Afin de démarrer le moteur, il faut d'abord activer une pin du hacheur afin de désactiver la sécurité. Cette pin se nomme ISO\_RESET et doit rester active 10ms. Il est ensuite possible de commander le moteur, à condition de ne pas re-déclencher la sécurité à cause d'un trop fort appel de courant entre-autres.

Premièrement, il est possible de lancer la séquence de démarrage en tapant la commande "start" dans la console connectée à la carte STM, mais une interruption EXTI détectée lors de l'appui sur le bouton bleu permet aussi de lancer la séquence de démarrage.

## Premier tests

Après implémentation de la fonction "speed" qui permet de régler le rapport cyclique d'une valeur entre 0 et 100, on remarque que la sécurité du hacheur s'enclenche lors d'un changement trop brusque de vitesse (ex: passer de 70 à 100%)

Pour utiliser cette fonction, il suffit de taper " speed XX " dans la console, cela calcule la valeur du rapport cyclique correspondant au pourcentage de vitesse demandée et l'inscrit dans les registres Pulse correspondant.

## Capteur de courant et de vitesse

### Mise en place du capteur de courant

On utilise comme capteur de courant la résistance shunt du hacheur. D'après la Doc,  $A = 12(2,5 - V)$ . De plus, on active l'ADC sur 12 bits, on a donc 4096 valeurs différentes entre 0 et 3,3V, ce qui nous donne une précision d'environ 0.00025V, donc environ 3mA. De plus, le capteur de courant fonctionne avec le DMA, ce qui permet de mettre à jour la mesure en continu afin d'avoir une valeur disponible lorsqu'on en a besoin.

### Mise en place du capteur de vitesse

On utilise comme capteur de vitesse le codeur incrémental du moteur.

Léo Pertuet

Ludovic Schini

D'après la Doc, on remarque que ce codeur possède deux voies de 1024 incréments. Ce qui correspond alors à 2048 incréments par tour. Afin de ne pas avoir d'erreur d'overflow ou d'underflow, on recharge après chaque mesure le registre de compte à la moitié de son maximum.

La valeur de la vitesse en degrés/secondes vaut:

$$\frac{tim2cnt(t) - tim2cnt(t+\Delta t)}{\Delta t} * \frac{360}{1024}$$

On a, d'après le timer 3 utilisé pour mesurer la vitesse,  $\Delta t = 100$  ms

## Asservissement

Afin d'asservir le moteur en vitesse et en courant, on met en place deux correcteurs PI. On remarque qu'on ne commande que la vitesse du moteur grâce à ces correcteurs. On crée donc une structure de données correcteur nous permettant de gérer facilement ces deux correcteurs. Cette structure de données reprend: l'erreur, le calcul de la partie proportionnelle, le calcul de la partie intégrateur et la sortie. Elle dispose aussi d'une partie écrêtage afin de ne pas surcharger le moteur. Pour le PI de vitesse, on a pris, en gain proportionnel 0.10 et 0.08 en gain intégral. Pour le PI de courant, on a pris en gain proportionnel 0.1 et 2.0 en gain intégral, obtenus d'après les simulations en TP Simulink.