

Laboratoire d'actionneurs intégrés (LAI)

Commande embarquée de moteurs

Laboratoire 4 : Pont triphasé, moteur synchrone

Objectifs : Commande d'un moteur synchrone à l'aide de sondes de hall

Etapes :

- A partir de la feuille annexée, générer la table de commutation pour le fonctionnement dans le sens de rotation positif.
- Reprendre le code du laboratoire 3 : en faire une copie de sauvegarde.

Dans l'interface graphique :

- Désactiver les channels 1 et 2 du timer 1.
- Les pattes PC10 (EN1), PC11 (EN2) et PC12 (EN3) doivent être configurées comme des outputs pour permettre la commutation des phases. Donner un nom (click droit->user label) à ces IO ainsi qu' aux INx et aux Hallx des points suivants
- Les pattes PA8 (IN1), PA9 (IN2), PA10 (IN3) sont des outputs également.
- Les sondes de Hall sont sur les pattes PA15, PB3, PB10. Elles doivent être des inputs (sans pullups).
- Le pwm utilisé pour donner le niveau de courant est le channel 1 du timer 3 (à activer en définissant sa « clock source »), il doit sortir sur PB4 (ctrl->click pour changer).
- Le pwm utilisé pour régler le niveau de tension est le channel 4 du timer 1 (à activer), il doit sortir sur PA11.
- Afin de bloquer le pont en cas de surcourant, il s'agit de configurer PA12 (=CPOUT du L6230) comme TIM1_ETR (et cocher use ETR as clearing source dans la config du timer 1).
- Le timer 3 (channel 1) et le timer 1 (channel 4) sont à configurer en PWM centré de type 1 avec la même période qu'avant.

Vérifier que :

- Le signal de déclenchement de l'interruption (TIM1_CH3) est en « toggle on match » avec un niveau de compare (pulse) à 0
- Le signal de surcourant (Timer 1, clear input) est activé pour le channel 4 en polarité normale (pas inversée) sans prescaler ni filtre.

Dans main.c, seuls les démarrages suivants doivent être effectués :

- Démarrer le PWM sur le channel 1 du timer 3 à l'aide de HAL_TIM_PWM_Start (un seul appel)
- Démarrer le PWM sur le channel 4 du timer 1 à l'aide de HAL_TIM_PWM_Start
- Démarrer OC sur le channel 3 du timer 1 à l'aide de HAL_TIM_OC_Start
- Démarrer l'ADC à l'aide de HAL_ADC_Start_DMA

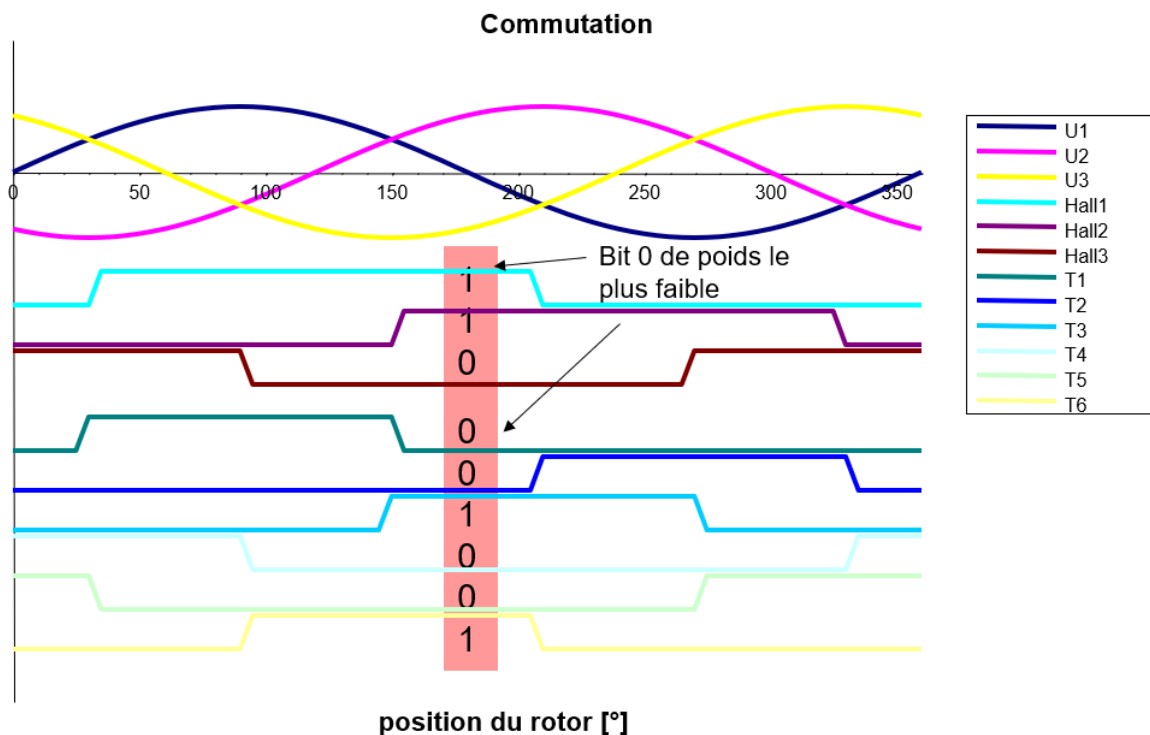
Dans l'interruption :

- Lire l'état des sondes de Hall au moyen de la fonction HAL_GPIO_ReadPin, rassembler les résultats une seule variable (*hall*) de manière que chacun des trois bits de poids le plus faible corresponde à l'état d'une sonde de Hall. Par exemple :
`Hall = (HAL_GPIO_ReadPin(HALL3_GPIO_Port, HALL3_Pin) << 2) + ... (à remplir)`
- Il s'agit ensuite de réaliser la commutation proprement dite. Pour ce faire, il s'agit d'utiliser la variable *hall* comme index dans le tableau de commutation afin de déterminer l'état de commutation du pont.

Dans le cas d'un pont à 6 transistors, l'état de chacun des transistors peut être extrait de la table de commutation et affecté au registre correspondant. Dans le cas d'un pont intégré comme celui qui nous intéresse, il s'agit de distinguer les signaux « Enable » de chaque branche, des signaux « Input » qui déterminent quel transistor de la branche va être passant. Dans ce cas, on peut aussi avoir deux tableaux : un pour les enables et un autre pour les inputs. Par exemple, pour affecter le signal enable de la branche 2, on peut faire :

```
HAL_GPIO_WritePin(EN2_GPIO_Port, EN2_Pin, (enables[hall]>>1)&1);
```

- Le niveau de courant dans le moteur est déterminé par le rapport cyclique sur le channel 1 du timer 3 à l'aide du registre TIM3->CCR1.
- Si l'on désire commander le moteur en tension, il suffit de régler le niveau de tension avec (TIM1->CCR4), si non, il faut le mettre au niveau max (égal à la période).
- Mesurer le courant à l'aide de l'oscilloscope pour divers cas de charge.
- Faire en sorte que le moteur tourne aussi dans le sens négatif (facultatif).



Sondes de Hall	Valeur décimale	Transistors sens positif	En	IN	Transistors sens négatif
000	0	impossible			impossible
001	1				
010	2				
011	3	1 0 0 1 0 0	110 (6)	010 (2)	0 1 1 0 0 0
100	4				
101	5				
110	6				
111	7	impossible			impossible