Schéma explicatif du fonctionnement du module

Pour pouvoir fonctionner notre module doit pouvoir lire la variation de tension sur notre potentiomètre et pouvoir retranscrire cette valeur en signal PWM

Pour cela il nous faut un CAN et un générateur de PWM pour l’ESC



CONVERTISSEUR ANALOGIQUE NUMÉRIQUE

Pour éviter d’endommager les fonctions de la carte il est conseillé de travailler sur une plage de 3,3V(voir stm32l476\_refmanual page 508, voir Programming with STM32 page 11)

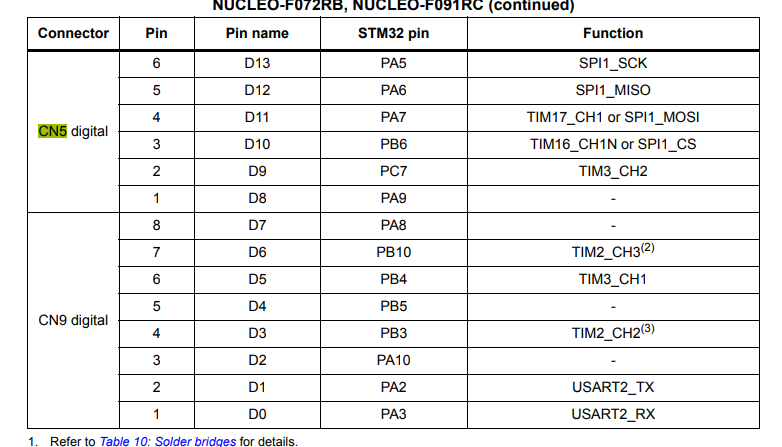
la résolution de nos ports CAN est de 12 bits mais peut être configurée à 10,8 et 6 bits si nécessaire (voir stm32l 476 ref manual page 506)

Ensuite il nous faut déterminer le pas quantique qui est la division de notre voltage par le nombre de bit de codage :

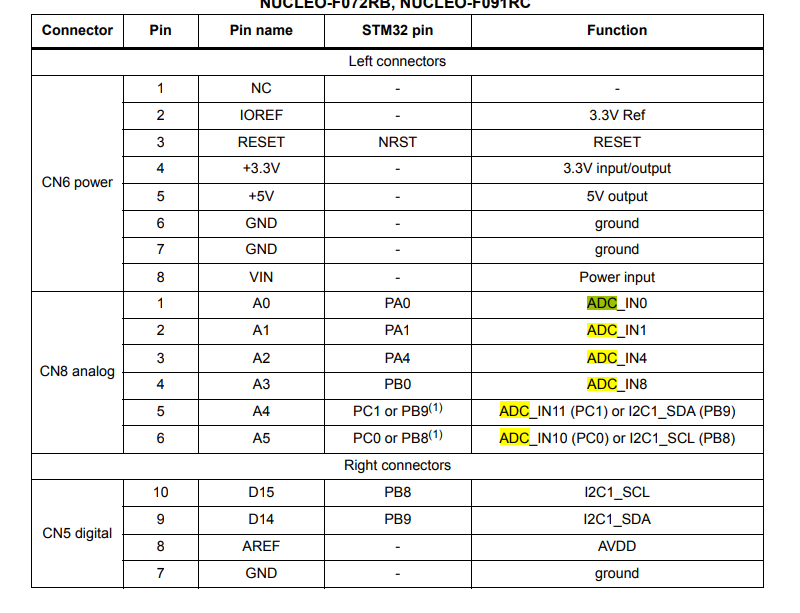
q = avec dV=3,3 et 2expN = 2exp12 = 4096

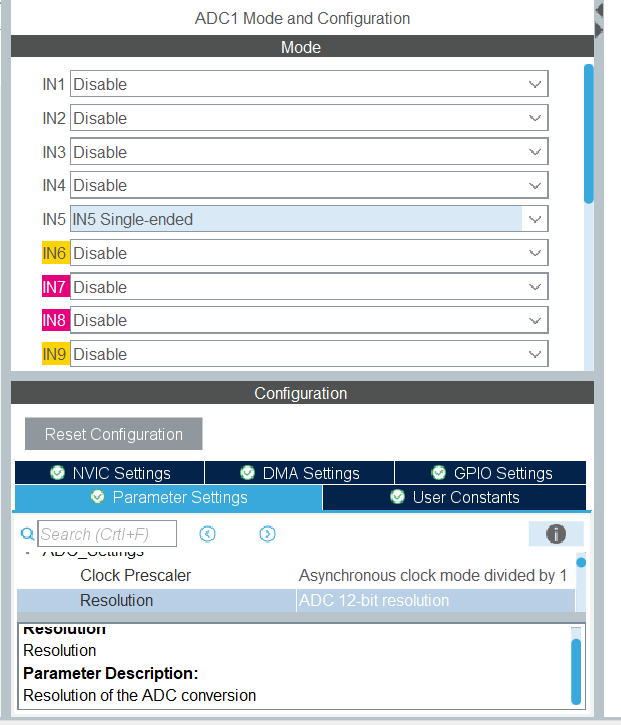
pour la suite nous devons configurer nos timers prenants en compte le signal CAN et PWM

Pour le signal PWM nous choisissons la sortie PA7



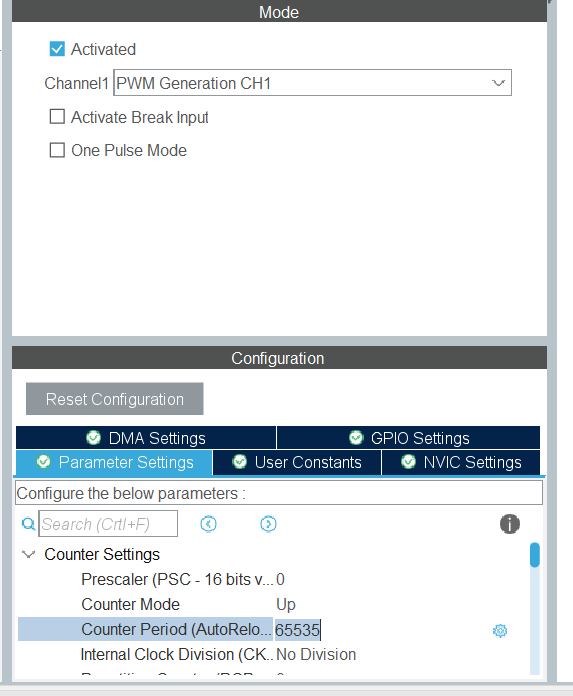
pour le CAN nous prendrons le port PA0



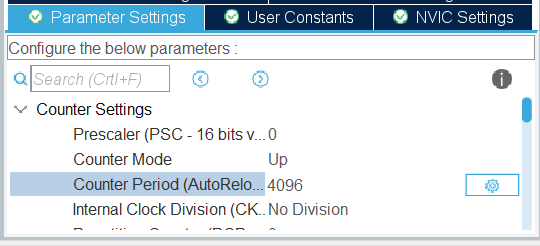


nous configurons le port PB8

Nous devons aussi faire correspondre la période de notre timer au nombre de bits sur notre CAN



une fois la valeur changée :



Nous pouvons ainsi commencer à travailler sur le programme principal.

Une chose qui est intéressante est que nous travaillerons avec la méthode du pooling(surveiller à la porte pour une entrée ou sortie).

Passage de potentiomètre à PWM :

La lecture des données se fait sur une plage de tension qui est ensuite codée en binaire par notre CAN, cette valeur est renvoyée ensuite dans un registre appelé **collect and compare register** qui va lui comparer la valeur produite par notre CAN et va la comparer à notre Timer de PWM(voir code)

Passage de Cablage de L’ESC et De la STM32:

Là reside plusieurs interrogations sur la meilleure maniere de relier l’esc au moteur sachant qu’on a deux câbles de données

