

Badaoui Yaniss
Génie Electrique 4ème année
Chef de projet: Clément Poirier
9 octobre 2017

Projet de Sous-Traitance: Tutoriel e2 studio driver CAN



Introduction:

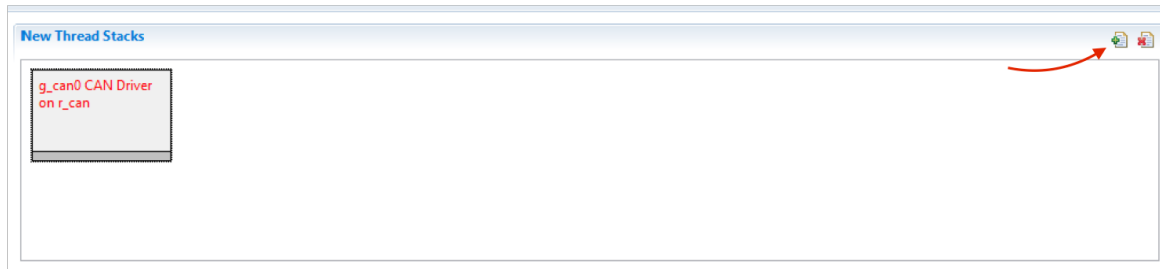
Au cour de ce rapport vous y trouverez le détail pour configurer le driver CAN sur le logiciel e2 studio de Renesas. Il est important de retenir que la configuration va dépendre en grande partie de comment vous souhaitez l'utiliser.

Projet de Sous-Traitance: Tutoriel e2 studio driver CAN	1
Introduction:	2
Configuration du CAN HAL Module	4
1) Parameter Checking Enable	4
2) Name	5
3) Channel	5
4) Baud rate Prescaler	5
5) Time Segment 1 et 2	5
6) Synchronization Jump Width	6
7) Clock source	6
8) Callback	6
9) Overwrite/Overrun Mode	6
10) Standard or Extended ID Mode	7
11) Number of Mailboxes	7
14) Mailboxes ID	7
15) Mailboxes Type	7
17) Interrupt priority	8
Configuration des pins	8
Conclusion	9
Annexe	10

Configuration du CAN HAL Module

Avant de procéder à la configuration du Module CAN, il faut savoir que le logiciel va sur ligné en rouge les erreur. Les blocks possédant une bande grise signifie qu'ils travaillent tout seul (ne dépend pas d'autre module pour être configuré).

Premièrement il faut ajouter le module driver. Ajoutez le en cliquant sur l'icône en haut a droite de la fenêtre: Driver ==> Connectivity ==> CAN Driver on r_can.

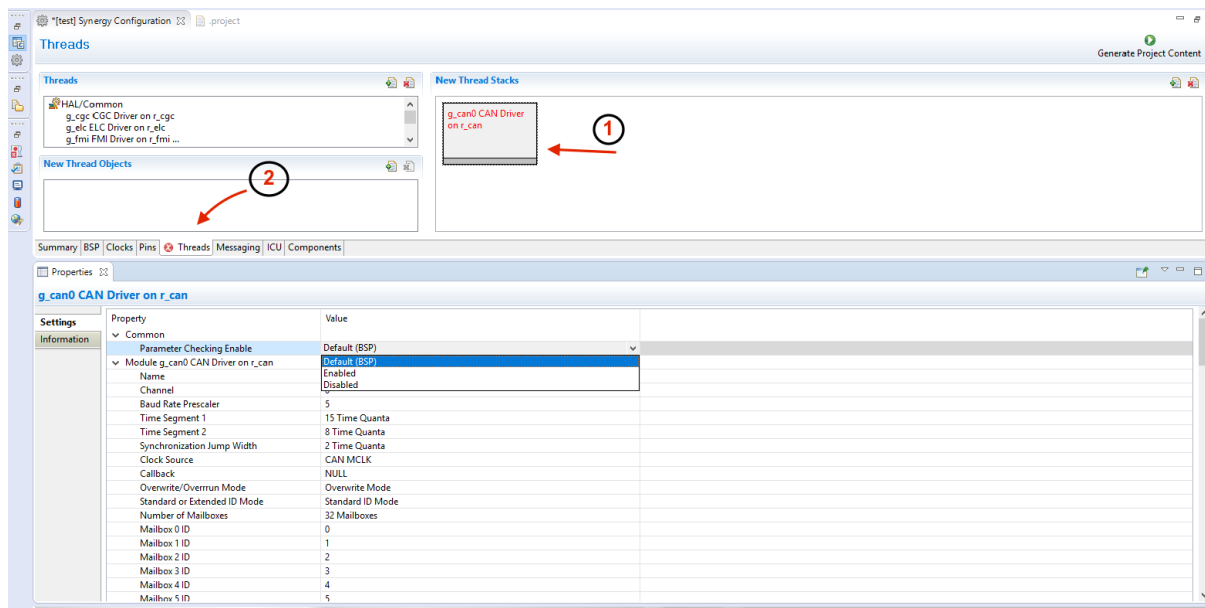


Normalement le nom par défaut est: g_can0. Il pourra être modifié dans les réglages que nous allons voir juste après. Maintenant il va falloir régler un par un les paramètres. À noter que le logiciel configure par défaut tous les modules du CAN. Cela évite les petites erreurs qui auraient pu être faites. En effet le logiciel permet de modifier seulement les module, il va donc bloquer toutes les parties qui pourraient créer des conflits.

1) Parameter Checking Enable

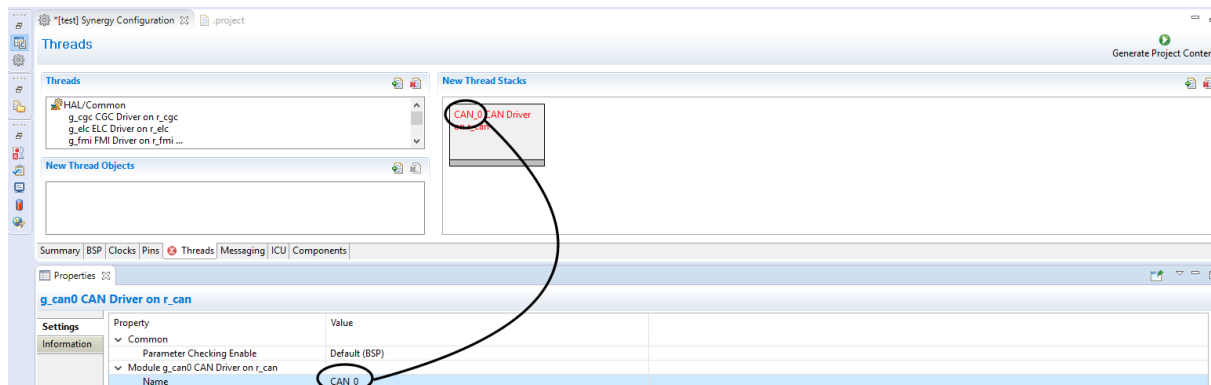
Pour modifier les prochaines étapes il faut aller dans le petit menu Threads après avoir cliquer sur le module can.

« Parameter Checking Enable » permet d'intégrer le code pour le paramètres de vérification dans le « build ». Il est par défaut sur BSP



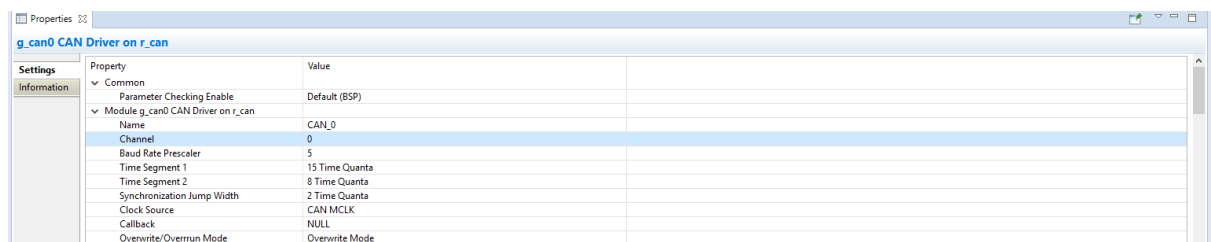
2) Name

À vous de choisir le nom qui vous convient le mieux.



3) Channel

Ici le canal peut être changer seulement sur un S7G2. Il peut être mis a 1 ou 0.



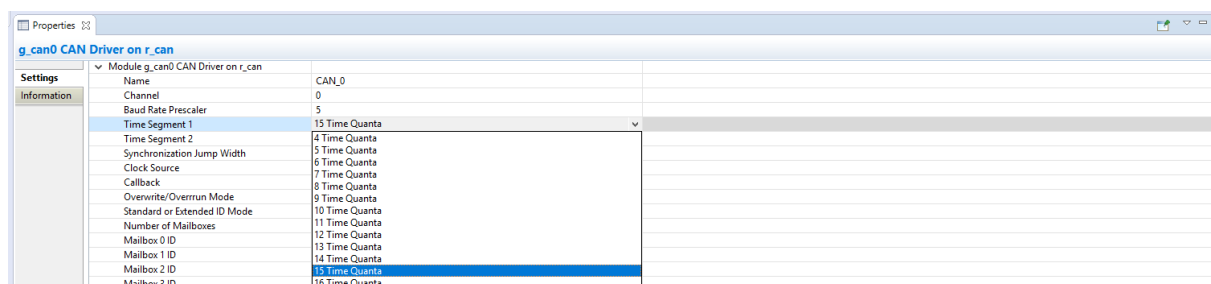
4) Baud rate Prescaler

Le prescaler du CAN peut varier de 0-1023. Ce prescaler permet de modifier la fréquence d'horloge de communication du can. Si la valeur du prescaler vaut x, alors la fréquence du CAN sera divisé par x+1.

5) Time Segment 1 et 2

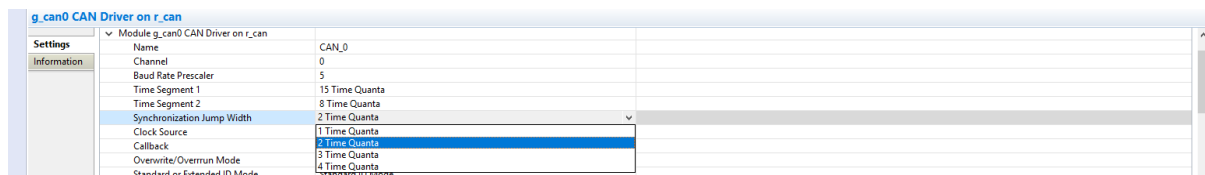
Le « time segment 1 » : permet de spécifier la taille total du temps de propagation du segment ainsi que la phase du buffer en fonction du time quantum (noté Tq). Varie de 2 à 8.

Le « time segment 2 » : permet de spécifier la taille de la phase du buffer en fonction du Tq. Sa taille doit être inférieur à celle du time segment 1. Varie de 4 à 16.



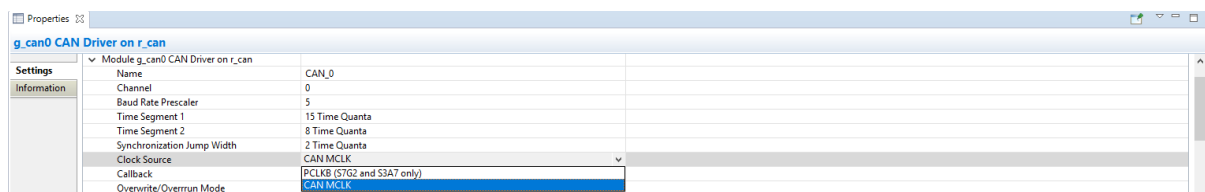
6) Synchronization Jump Width

Permet de spécifier la longueur du saut pour la synchronisation. Cette valeur doit être égale ou inférieure au time segment 2.



7) Clock source

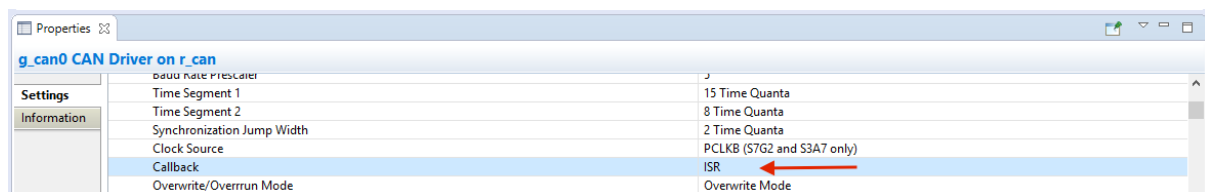
PCKLB : « Peripheral module clock » La source de l'horloge sera celle du CAN
CAN MCLK : L'horloge sera produite en externe via la pin « EXTAL »



8) Callback

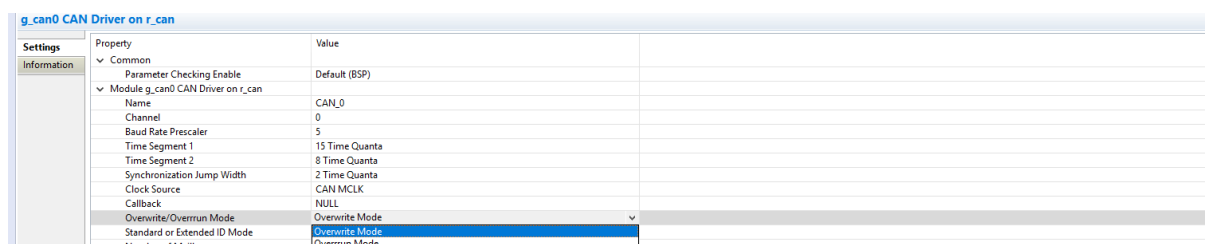
IMPORTANT

C'est la seule chose que je n'ai pas trop compris. Après plusieurs recherches il s'avérait que « Callback » serait appelé par une interruption d'où le ISR à la place du NULL par défaut. Il faut faire attention ici car c'est le seul paramètre qui doit être modifié pour enlever le surgissement rouge sur le driver CAN.



9) Overwrite/Overrun Mode

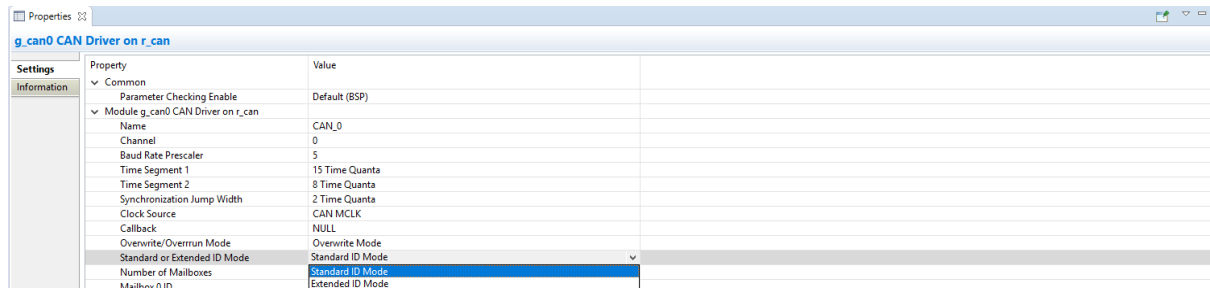
Permet de choisir de réécrire sur les « mailboxes » ou si elles seront débordé (« overrun »).



10) Standard or Extended ID Mode

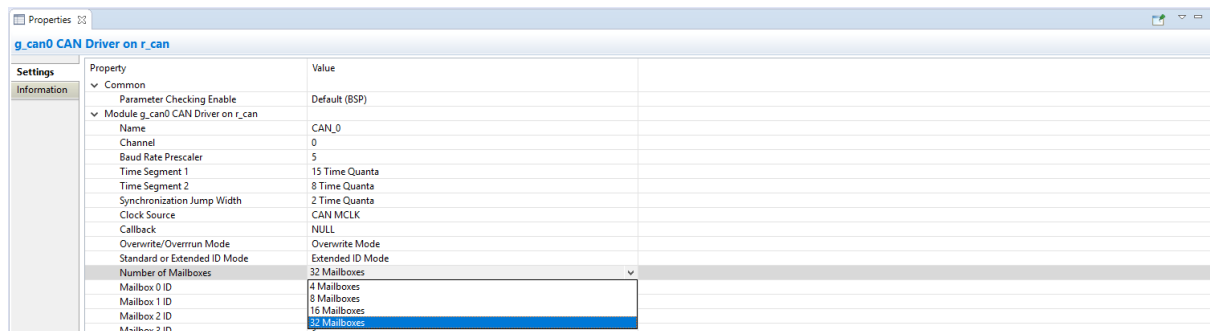
- Le mode « extended » permet de recevoir des messages « extended » et compare avec la « mailbox » correspondante.

- Le mode « standard » permet de recevoir des messages « standard » mais aussi « extended ». Tout comme le mode extended il compare la valeur avec la « mailbox » correspondante.



11) Number of Mailboxes

Le nom parle de lui même, varie de 4 à 32.



14) Mailboxes ID

Permet de définir l'ID des « mailboxes ». Cela sert entre autre « standard and extended mode ». L'ID va permettre de choisir avec quelle « mailbox » nous voulons communiquer.

15) Mailboxes Type

Les « mailboxes » peuvent être définies en tant que receveur ou transmetteur.16) Mailboxes Frame Type

Permet ou « mailboxes » de type receveur, de capturer des données ou de les enlever. Cela ne fonctionne pas avec les « mailboxes » transmetteur.

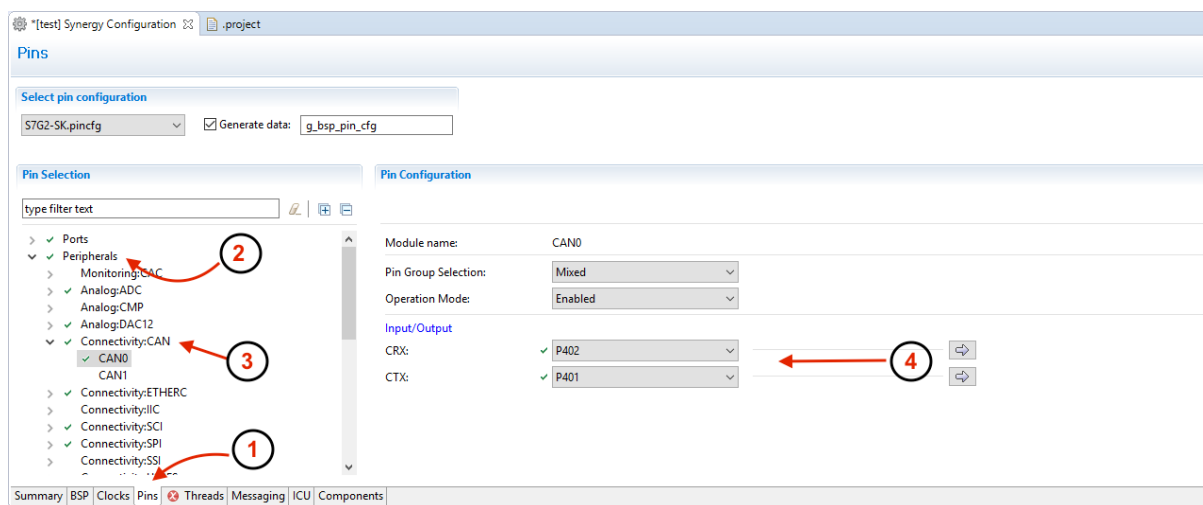
17) Interrupt priority

Ici, il y a 3 types d'interruption. Par défaut elles sont désactivées, mais nous avons le choix avec plusieurs niveaux de priorités.

Error Interrupt Priority	Disabled
Receive Mailbox Interrupt Priority	Disabled
Transmit Mailbox Interrupt Priority	Disabled

Configuration des pins

Pour cela il suffit de suivre les indication sur les images si-dessous. Par défauts les pins seront bien assignés.



Conclusion

Le logiciel e2 studio facilite grandement les choses en ayant implanter ce système de module. Il programme par défauts des pins, des valeurs ou même des états (ex :transmission / réception) . Il faut cependant faire attention au paramètre de callback mais aussi à la manière que vous devez l'utiliser.

Annexe

Datasheet de e2 studio:

<https://www.renesas.com/en-us/doc/products/renesas-synergy/doc/r01um0001eu0120-synergy-s7g2.pdf?key=83e1c64d2b10245ff523bf113b87e38d>

Datasheet du CAN HAL Module:

<https://www.renesas.com/en-eu/doc/products/renesas-synergy/apn/r11an0065eu0101-synergy-can-hal-mod-guide.pdf>