Buot Adrien – Guicharnaud Léo – Sabatey Clara

# Rapport Microprocesseurs :

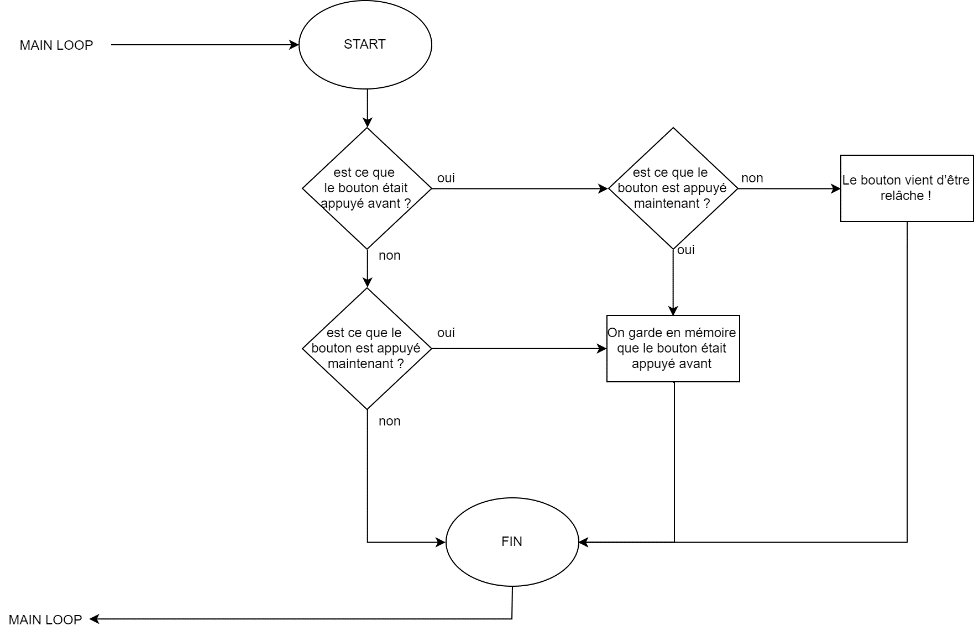
## Rapport TP I :

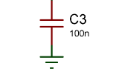
1. On s’attend à trouver en résultat le PORTC qui passe d’une valeur 0 à 1 en continu. Pourtant en utilisant le debugger on se rend compte que la valeur ne change pas et reste à 0. Cela est dû au fait que nous n’avons pas initialisé le port C.
2. On attend en résultat une LED qui s’allume et qui s’éteint en continu. Cependant, rien ne s’allume. L’explication est que nous n’utilisons pas de Delay ainsi la LED s’allume et s’éteint beaucoup trop vite pour que le changement soit perçu par l’œil humain.
3. Pour pouvoir distinguer les 2 états de la LED, il faut ralentir le programme. Pour ce faire, on va devoir créer une fonction Delay.
4. Le programme my\_delay attend d’être sur un état 1 de count 1 et count 2 pour continuer le programme. Tant que count 1 et/ou count 2 ne sont pas à 1, la fonction my\_delay continue de s’appeler. On compte 4 cycles.
5. LED 3 =1/2 fréquence LED 2= 1/4 fréquence de LED 1 = 1/8 fréquence de LED 0
6. On peut lire l’état de la pin RD0 en initialisant le port D comme une entrée digitale. On aura ensuite accès à l’état de PORTD il faut appliquer le masque 0x01 à PORTD, et comme ça, il est possible de savoir si le bouton est relâché ou non :

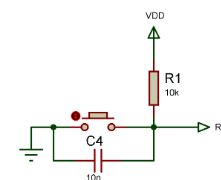
- si le bouton est appuyé, alors PORTD & 0x01 = 1

- si le bouton est relâché, alors PORTD & 0x01 = 0

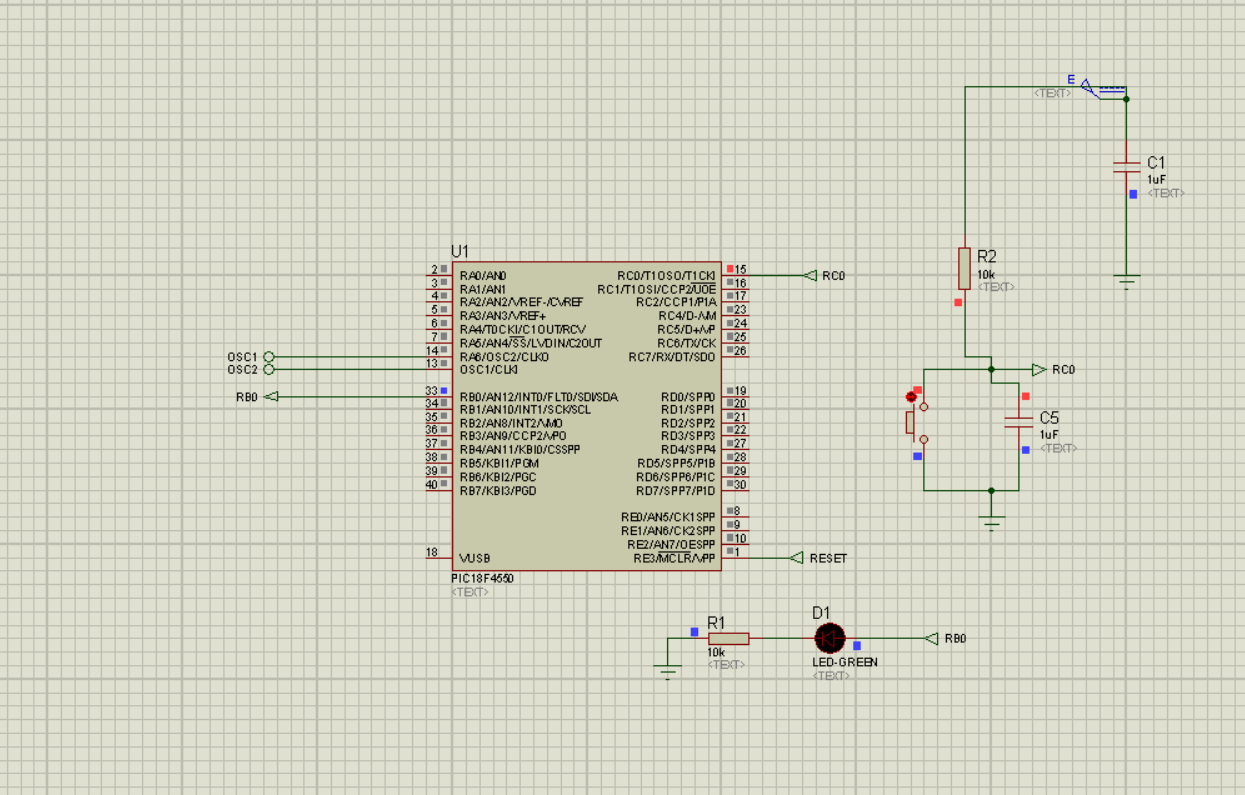
1. Dans le code fourni, le BTFSS agit comme une condition IF ; ainsi lorsque le bouton RD0 est à l’état 1, le programme ira dans la fonction *pressed* et à l’état 0 dans la fonction *not\_pressed*. Inversement, BTFSC agit de la même manière mais à l’inverse : l’état 1 induit la première instruction.
2. Il est en effet possible de faire cette même opération sans bloquer le programme dans une boucle, il faut pour cela utiliser une mémoire : algorithme du code :



1. Le microcontrôleur est alimenté uniquement à l’aide du PC par câble.
2. Les condensateurs C1 et C2 ont pour fréquence 48 MHz.
3. Le condensateur C3 sert de condensateur de découplage. Il est branché juste avant le terminal Ground et permet d’évacuer les harmoniques de hautes fréquences. 
4. La résistance R1 et le condensateur C4 sont reliés au bouton et forment un système antirebond pour le bouton. Ainsi dès que l’on appuiera sur le bouton, le signal sera instantané sans restes de signaux décalés dans le temps. (peut être parler des effet de Gibbs)



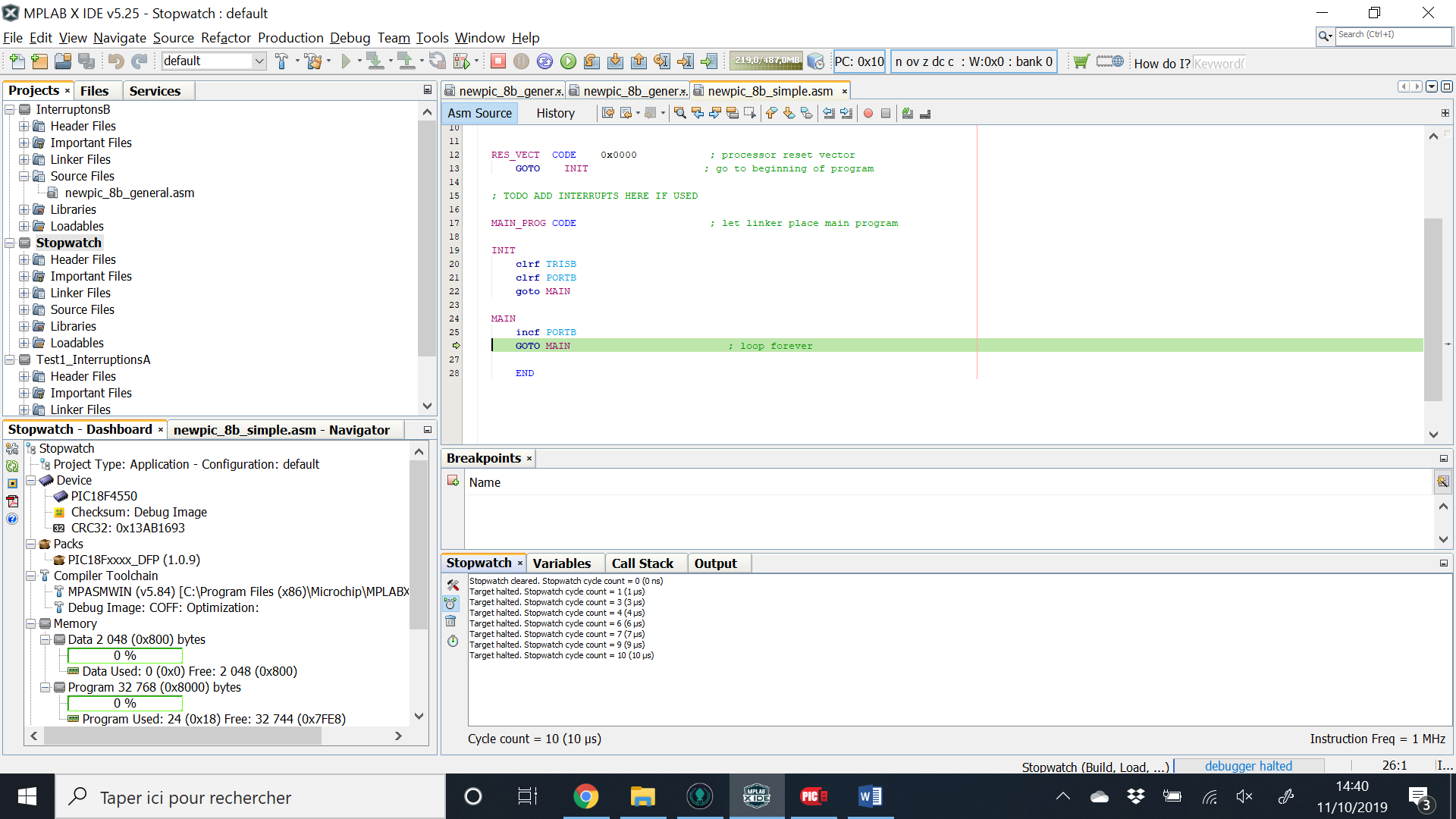
1. Schéma sur ISIS en version debug



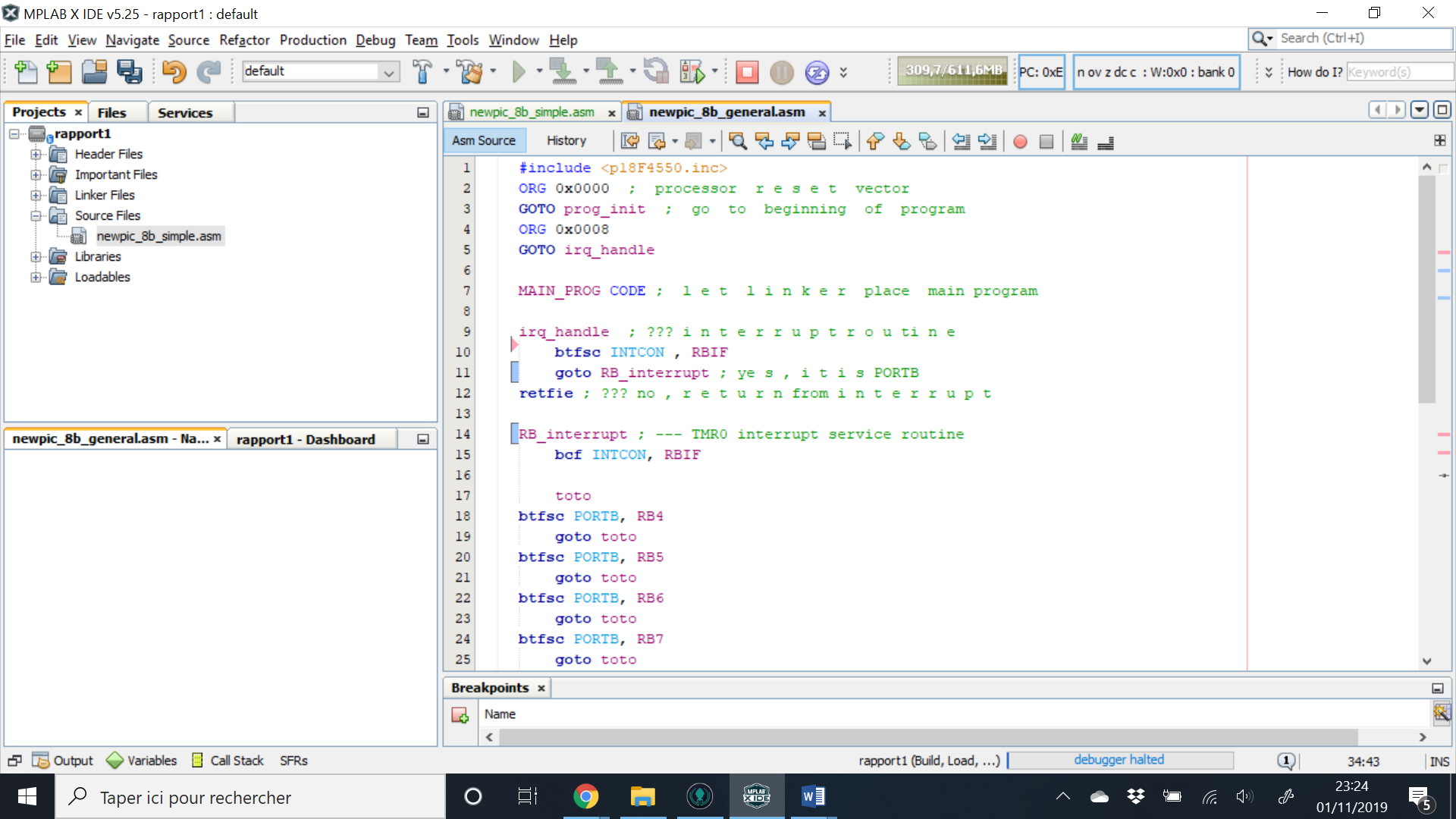
## Rapport TP II :

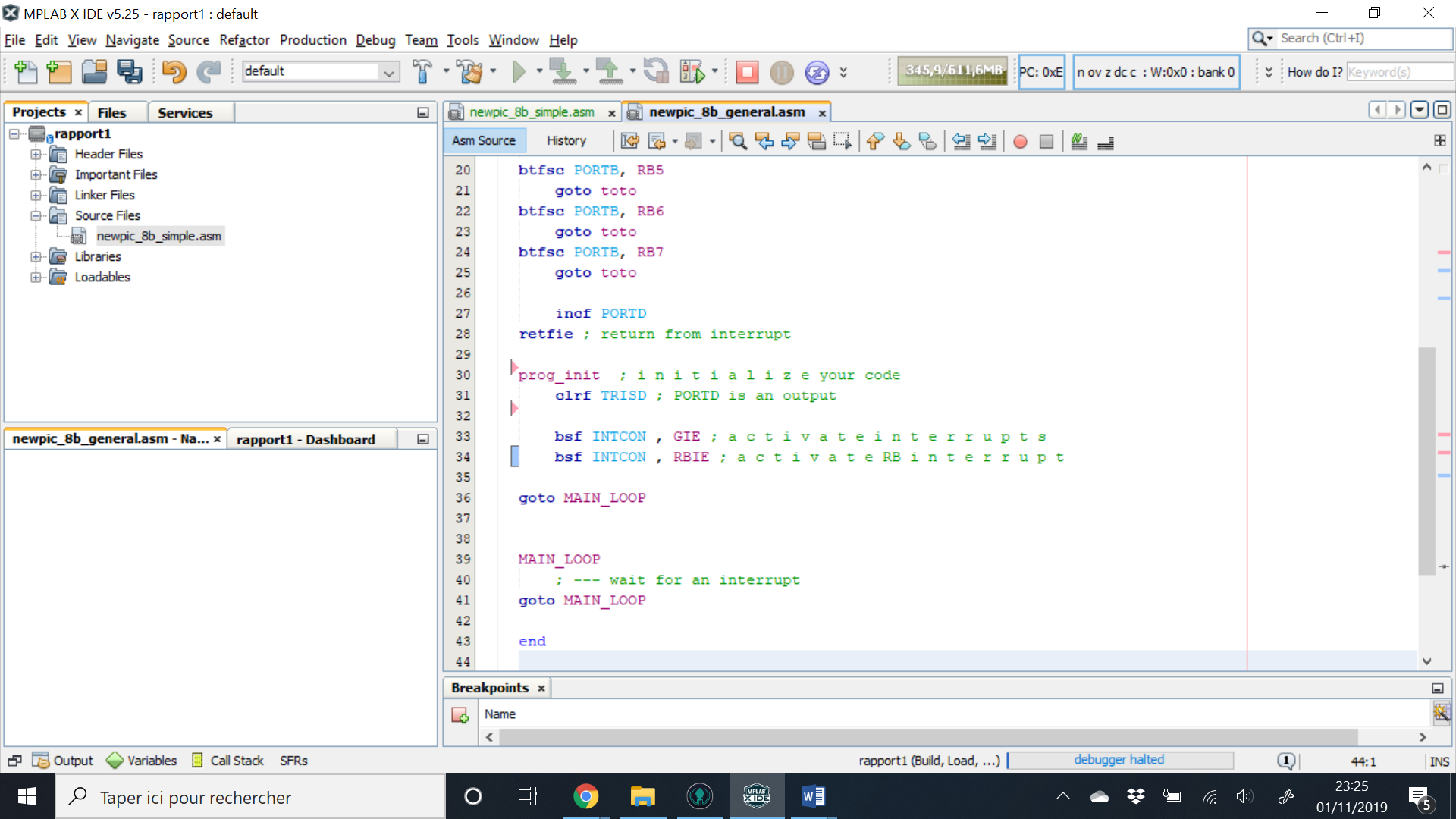
15) Pour échapper à la boucle infinie, on utilise des interruptions. Une interruption va lancer une partie du code avec une priorité plus élevé que la boucle infinie de al boucle principale, ce qui fait que on s’en échappe.

16) Le temps des interruptions n’est pas en secondes mais en microseconde. D’après le test des stopwatchs sur le code, le code s’interrompt toutes les microsecondes environ.



Task 9





Task 10

Ici on utilise 2 sources d’interuptions : un timer et un bouton

On compte les interuptions du timer sur les leds du port C et celles du bouton sur les leds du port D

