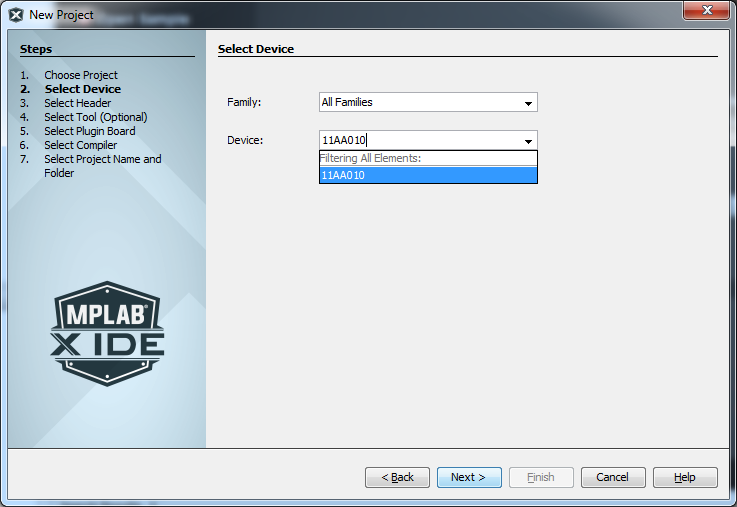
Sous MPLABX 5.10, MPLAB code configurator (3.75), XC16 (1.35)

Dans File, New Project , « Standalone Project »



Choisir le bon processeur (compatible avec MCC) ici un DSPIC33EP512MC502

<https://www.microchip.com/mplab/mplab-code-configurator>

Chercher dans Release Notes le composant

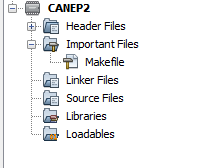
PIC10 / PIC12 / PIC16 / PIC18 MCUs

PIC24 / dsPIC33 / PIC32MM MCUs

PIC32MX MCUs

* Remplir les champs jusqu’à la fin de l’assistant.

Le projet doit contenir ces dossiers (vide)

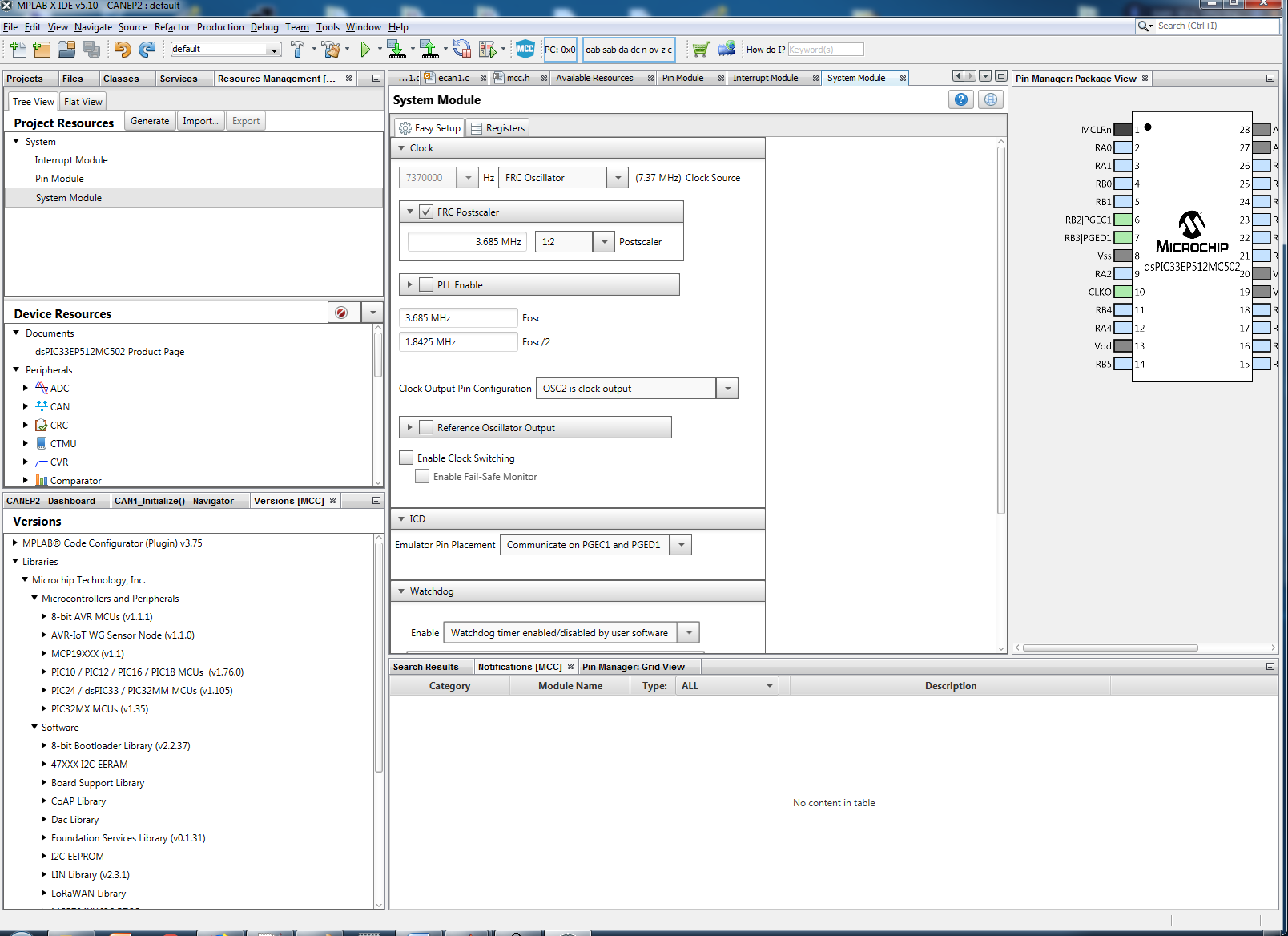


On lance MCC :

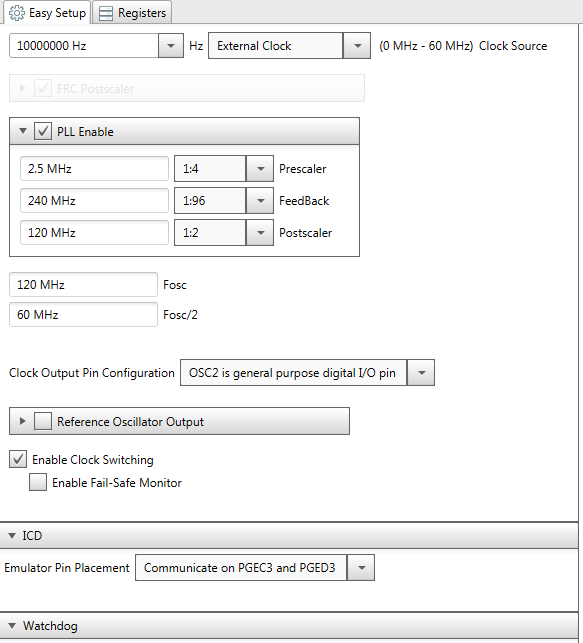
Soit par l’icone : 

Soit par Tools/ Embedded/ MPLAB Code Configurator Open/ Close

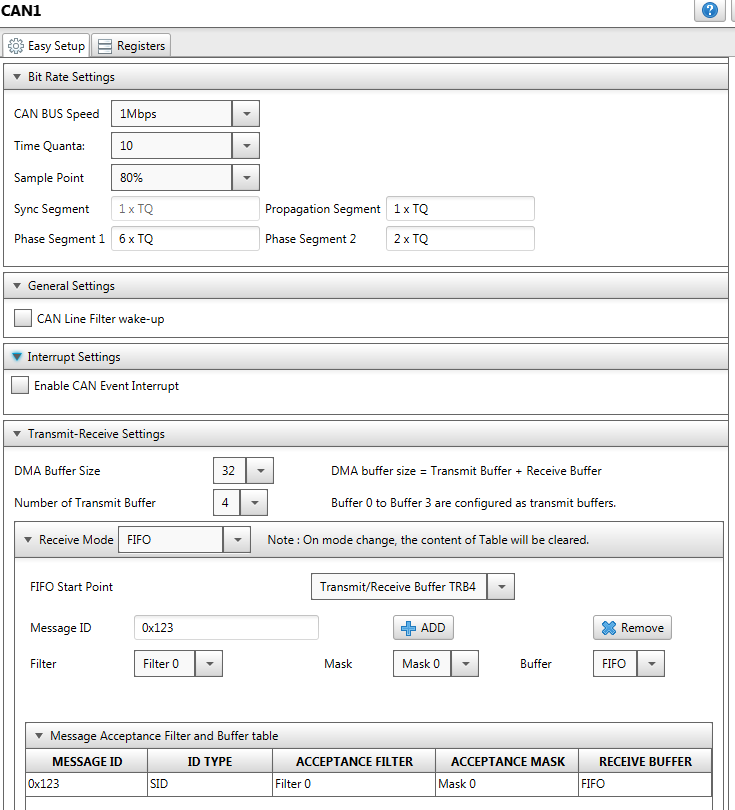
Enregistrer le fichier Myconfig.mc3 dans le bon dossier.



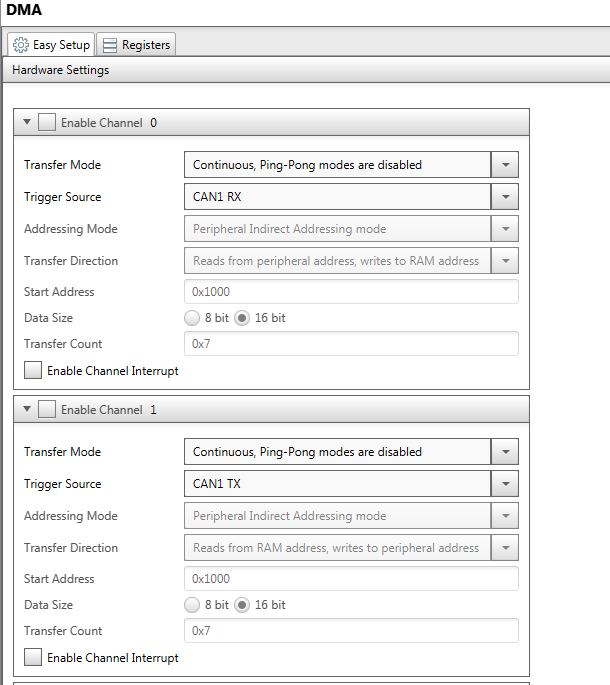
System module :

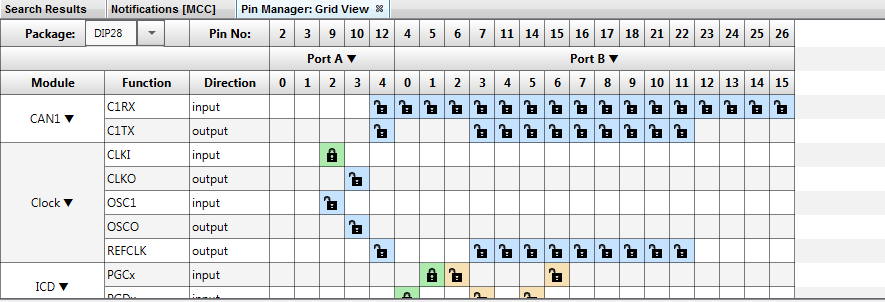


Ajout de CAN1 dans Device Ressources

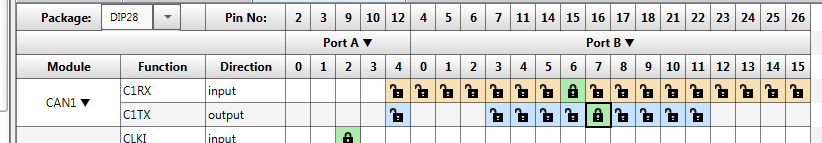


Ajout de DMA

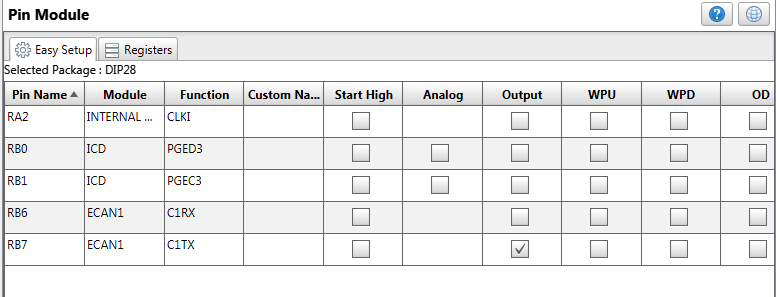




ET choisir pour l’exemple :

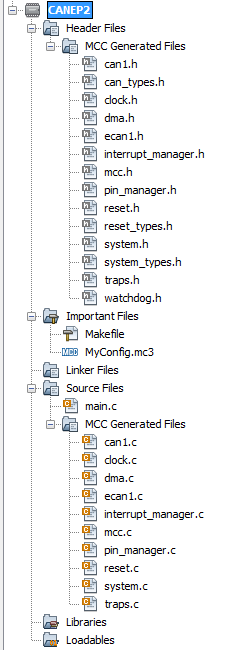


Verifier dans Pin module :



Faire « Generate »

Des fichiers ont été ajoutés dans le Projet



Récupérer dans le dossier mcc\_generates\_files les fichiers :

* can\_types.h
* can1.c
* can1.h
* ecan1.c
* ecan1.h
* dma.c
* dma.h

Puis dans pin\_manager.c copier la partie PPS

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* Set the PPS

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

\_\_builtin\_write\_OSCCONL(OSCCON & 0xbf); // unlock PPS

RPOR2bits.RP39R = 0x000E; //RB7->ECAN1:C1TX

RPINR26bits.C1RXR = 0x0026; //RB6->ECAN1:C1RX

\_\_builtin\_write\_OSCCONL(OSCCON | 0x40); // lock PPS

Cette partie de code est à ajouter dans la fonction void ECAN1\_Simulink\_Init(void)

Pour les filtres, vérifier dans can1.c :

C1FEN1 = 0x07; //FLTEN8 disabled; FLTEN7 disabled; FLTEN9 disabled; FLTEN0 enabled; FLTEN2 enabled; FLTEN10 disabled; FLTEN1 enabled; FLTEN11 disabled; FLTEN4 disabled; FLTEN3 disabled; FLTEN6 disabled; FLTEN5 disabled; FLTEN12 disabled; FLTEN13 disabled; FLTEN14 disabled; FLTEN15 disabled;

/\* select acceptance masks for filters \*/

C1FMSKSEL1bits.F0MSK = 0x0; //Select Mask 0 for Filter 0

C1FMSKSEL1bits.F1MSK = 0x1; //Select Mask 1 for Filter 1

C1FMSKSEL1bits.F2MSK = 0x2; //Select Mask 2 for Filter 2

/\* Configure the masks \*/

C1RXM0SIDbits.SID = 0x7ff;

C1RXM1SIDbits.SID = 0x7ff;

C1RXM2SIDbits.SID = 0x7ff;

/\* Configure the filters \*/

C1RXF0SIDbits.SID = 0x123;

C1RXF1SIDbits.SID = 0x200;

C1RXF2SIDbits.SID = 0x300;

C1RXF0SIDbits.EID = 0x0;

C1RXF1SIDbits.EID = 0x0;

C1RXF2SIDbits.EID = 0x0;

C1RXF0EID = 0x00;

C1RXF1EID = 0x00;

C1RXF2EID = 0x00;

C1RXF0SIDbits.EXIDE = 0x0;

C1RXF1SIDbits.EXIDE = 0x0;

C1RXF2SIDbits.EXIDE = 0x0;

/\* FIFO Mode \*/

C1BUFPNT1bits.F0BP = 0xf; //Filter 0 uses FIFO

C1BUFPNT1bits.F1BP = 0xf; //Filter 1 uses FIFO

C1BUFPNT1bits.F2BP = 0xf; //Filter 2 uses FIFO

Le fichier fonctions.c contient les 3 fonctions utiles à Simulink.

void ECAN1\_Simulink\_Init(void);

void ECAN1\_Simulink\_Transmit(unsigned int Valid, unsigned int Id, unsigned char Dlc, unsigned char\* Data);

unsigned int ECAN1\_Simulink\_Receive(unsigned int\* Id,unsigned char\* Data,unsigned char\* Dlc);