

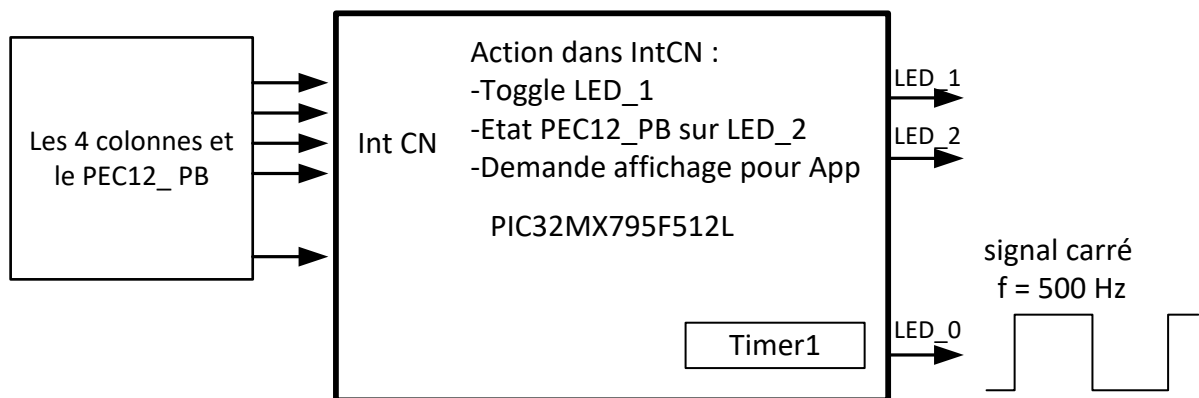
## EXERCICE 5\_2 PIC32MX

### OBJECTIFS

Cet exercice a pour objectif de permettre aux étudiants de mettre en place une interruption de Change Notification pour des entrées et de vérifier concrètement son fonctionnement.

Après l'ébauche sur le papier, l'exercice sera réalisé pratiquement avec un kit PIC32MX795F512L.

Voici le schéma de principe du système demandé :



- Dans la réponse à l'interruption du timer 1, on inverse la LED0 de manière à obtenir un signal carré d'une fréquence de 500 Hz.
- Dans la réponse à l'interruption Change Notification, on effectue les actions suivantes :
  - Inversion (toggle) de la LED1 pour permettre une observation.
  - Lecture état des 5 entrées CN et enregistrement dans appData.
  - Affectation état PEC12\_PB sur LED2.
  - Activation de l'application pour gestion de l'affichage.

## RAPPEL THEORIQUE

### NATURE DES CNx

On dispose de 22 CN, de CN0 à CN21.

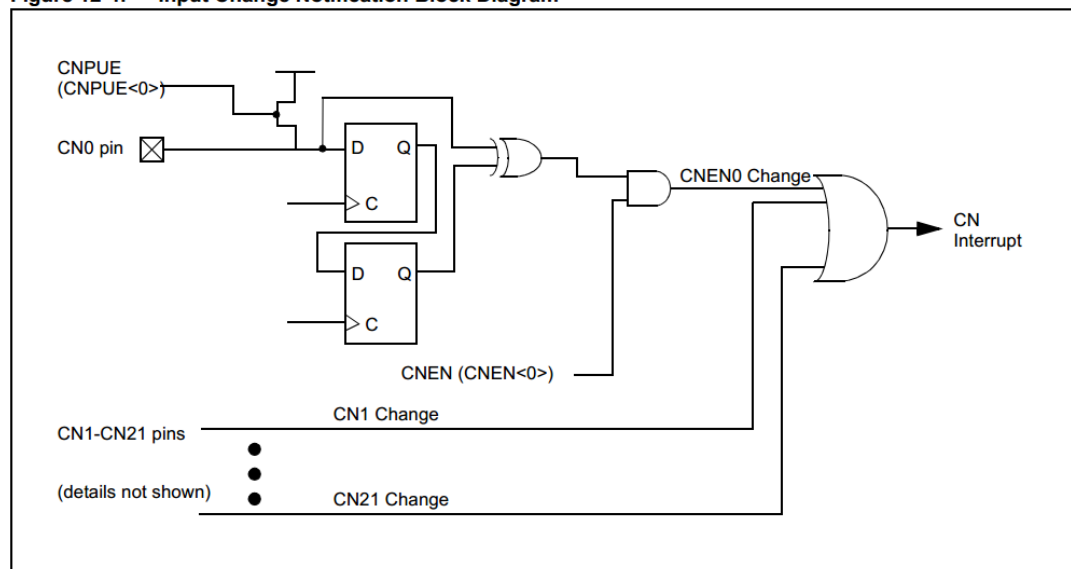
👉 Un CNx est un rôle de plus associé à une broche.

Voici la liste des CN pour le PIC32MX795F512L en boîtier 100 broches :

<b>CNx</b>	<b>Nom complet de la pin</b>	<b>No pin</b>
CN0	SOSC0/T1CK/CN0/RC14	74
CN1	SOSCI/CN1/RC13	73
CN2	PGED1/AN0/CN2/RB0	25
CN3	PGEC1/AN1/CN3/RB1	24
CN4	AN2/C2IN-/CN4/RB2	23
CN5	AN3/C2IN+/CN5/RB3	22
CN6	AN4/C1IN-/CN6/RB4	21
CN7	AN5/C1IN+/Vbuson/CN7/RB5	20
CN8	ECOL/SCK2/U6TX/_U3RTS/PMA5/CN8/RG6	10
CN9	ECRS/SDA4/SDI2/U3RX/PMA4/CN9/RG7	11
CN10	ERXDV/AERXDV/ECRSDV/AECRSDV SCL4/SDO2/U3TX/PMA3/CN10/RG8	12
CN11	ERXCLK/AERXCLK/EREFCLK/AEREFCLK _SS2/U6RX/_U3CTS/PMA2/CN11/RG9	14
CN12	AN15/ERXD3/AETXD2/OCFB/PMALL /PMA0/CN12/RB15	44
CN13	OC5/PMWR/CN13/RD4	81
CN14	PMRD/CN14/RD5	82
CN15	ETXEN/PMD14/CN15/RD6	83
CN16	ETXCLK/PMD15/CN16/RD7	84
CN17	SDA5/SDI4/U2RX/PMA9/CN17/RF4	49
CN18	SCL5/SDO4/U2TX/PMA8/CN18/RF5	50
CN19	ETXD3/PMD13/CN19/RD13	80
CN20	AETXD0/_SS3/U4RX/_U1CTS/CN20/RD14	47
CN21	AETXD1/SCK3/U4TX/_U1RTS/CN21/RD15	48

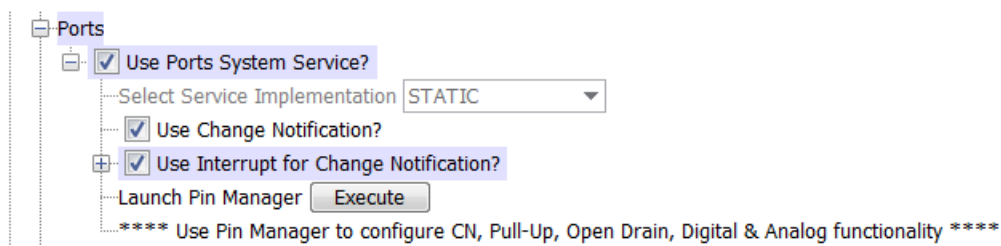
## SCHEMA BLOC DU MECANISME D'INPUT CHANGE NOTIFICATION

Figure 12-4: Input Change Notification Block Diagram



## ACTION POSSIBLE AU NIVEAU DU MHC

Au niveau du MHC, dans Harmony Framework Configuration > System Services > Ports, il est possible d'établir la configuration suivante :



Avec  , on aboutit au Pin Manager.

Le système offre la possibilité d'activer graphiquement un CN pour les broches qui en ont la possibilité (indiqué par les carrés bleus).

Output MPLAB® Harmony Configurator

Output Pin Table ×

Package: TQFP

Module	Function	S_MINU.	VDD	LCD_DB..	LCD_DB..	LCD_DB..	BM1_H..	BM2_H..	SD_DET.	RC4	LIGNE1	LIGNE2	RG8	MCLR	RG9	VSS	VDD	LED_0	PEC12_..	PEC12_..	COLONN.	COLONN.	COLONN.	COLONN.	POT1	POT0	RB6	RB7	RA9	RA10	AVDD	AVSS	STBY_H.	RB9
Change Notification	XBEE_RESET																																	
	CN0																																	
	CN1																																	
	CN2																																	
	CN3																																	
	CN4																																	
	CN5																																	
	CN6																																	
	CN7																																	
	CN8																																	
	CN9																																	
	CN10																																	
	CN11																																	
	CN12																																	
	CN13																																	
	CN14																																	
	CN15																																	
	CN16																																	
	CN17																																	
	CN18																																	
	CN19																																	
	CN20																																	
	CN21																																	
OSC1																																		

Cette sélection remplace la définition du BSP par le CNx correspondant.

☺ Lors de la génération du code, l'interruption CN sera activée pour les pins sélectionnées, et l'ISR générée.  
L'ISR est partagée entre tous les CN.

## LES CN VUS DU PIN TABLE

Voici l'ensemble des CN et leur correspondance avec les broches et les définitions du BSP.

Output MPLAB® Harmony Configurator*																											
Output Pin Table x																											
Package: TQFP																											
Pin conflict resolved. See output window.																											
Module	Function	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26
Change Notification	XBEE_RESET																										
	CN0																										
	CN1																										
	CN2																										
	CN3																										
	CN4																										
	CN5																										
	CN6																										
	CN7																										
	CN8																										
	CN9																										
	CN10																										
	CN11																										
	CN12																										
	CN13																										
	CN14																										
	CN15																										
	CN16																										
	CN17																										
	CN18																										
	CN19																										
	CN20																										
	CN21																										

Output MPLAB® Harmony Configurator*																											
Output Pin Table x																											
Package: TQFP																											
Pin conflict resolved. See output window.																											
Module	Function	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71	72
Change Notification	XBEE_RESET																										
	CN0																										
	CN1																										
	CN2																										
	CN3																										
	CN4																										
	CN5																										
	CN6																										
	CN7																										
	CN8																										
	CN9																										
	CN10																										
	CN11																										
	CN12																										
	CN13																										
	CN14																										
	CN15																										
	CN16																										
	CN17																										
	CN18																										
	CN19																										
	CN20																										
	CN21																										

a) En vous basant sur les copies d'écran des pin table, indiquez à quels éléments du kit PIC32MX795F512L correspondent les CNx.

CNx	Nom complet de la pin	No pin	Rôles kit
CN0	SOSC0/T1CK/CN0/RC14	74	
CN1	SOSCI/CN1/RC13	73	
CN2	PGED1/AN0/CN2/RB0	25	
CN3	PGEC1/AN1/CN3/RB1	24	
CN4	AN2/C2IN-/CN4/RB2	23	
CN5	AN3/C2IN+/CN5/RB3	22	
CN6	AN4/C1IN-/CN6/RB4	21	
CN7	AN5/C1IN+/Vbuson/CN7/RB5	20	
CN8	ECOL/SCK2/U6TX/_U3RTS/PMA5/CN8/RG6	10	
CN9	ECRS/SDA4/SDI2/U3RX/PMA4/CN9/RG7	11	
CN10	ERXDV/AERXDV/ECRSDV/AECRSDV SCL4/SDO2/U3TX/PMA3/CN10/RG8	12	
CN11	ERXCLK/AERXCLK/EREFCLK/AEREFCLK _SS2/U6RX/_U3CTS/PMA2/CN11/RG9	14	
CN12	AN15/ERXD3/AETXD2/OCFB/PMALL /PMA0/CN12/RB15	44	
CN13	OC5/PMWR/CN13/RD4	81	
CN14	PMRD/CN14/RD5	82	
CN15	ETXEN/PMD14/CN15/RD6	83	
CN16	ETXCLK/PMD15/CN16/RD7	84	
CN17	SDA5/SDI4/U2RX/PMA9/CN17/RF4	49	
CN18	SCL5/SDO4/U2TX/PMA8/CN18/RF5	50	
CN19	ETXD3/PMD13/CN19/RD13	80	
CN20	AETXD0/_SS3/U4RX/_U1CTS/CN20/RD14	47	
CN21	AETXD1/SCK3/U4TX/_U1RTS/CN21/RD15	48	

b) Pour obtenir un signal carré de 500 Hz en inversant LED0 à chaque interruption du timer 1, quelle doit être sa période ? Déterminez également le prescaler.

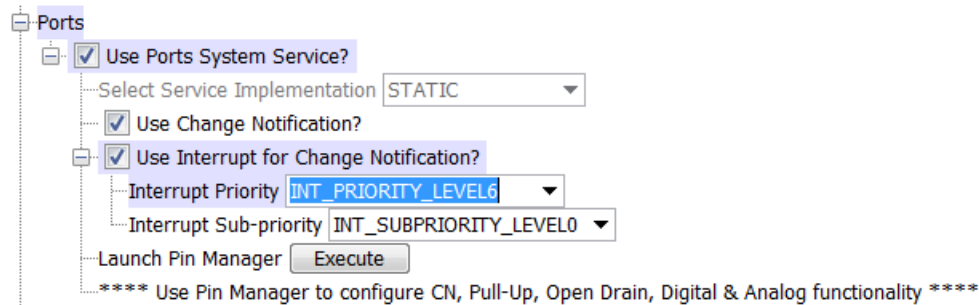
Période d'un signal carré 500 Hz : .....

Période du timer 1 : .....

Prescaler du timer 1 : .....

## REALISATION PRATIQUE

- Création d'un projet avec Harmony pour le kit PIC32MX
- Utilisation du timer 1 avec interruption. prendre niveau 4.
- Utilisation des CN pour les pins voulus, avec interruption. Prendre niveau 6.



## MODIFICATION DU FICHIER SYSTEM\_INTERRUPT.C

- Compléter l'ISR du timer 1 avec l'inversion de la LED0. Utilisez la fonction BSP\_LEDToggle. Incrémentez le champ Elapsedms de appData.
- Compléter l'ISR de CHANGE\_NOTICE avec les actions suivantes :
  - L'inversion de la LED1 avec la fonction BSP\_LEDToggle.
  - Lire les 4 colonnes et affecter les 4 booléens correspondants dans appData en utilisant la fonction PLIB\_PORTS\_PinGet.
  - Lire l'état du PEC12\_PB et l'affecter à la LED2 par action directe.
  - Ajouter l'appel à la fonction APP\_UpdateState avec (APP\_STATE\_SERVICE\_TASKS) en paramètre.
  - Effectuer le SourceFlagClear après la lecture des CN ! (nécessaire au mécanisme).

## MODIFICATION DES FICHIER APP.H & APP.C

- Ajoutez l'état WAIT et la fonction APP\_UpdateState. Mettre en place le switch avec les 3 états.
- Ajouter les éléments suivants dans le typedef APP\_DATA :
 

```
bool EtatCol1;
bool EtatCol2;
bool EtatCol3;
bool EtatCol4;
bool EtatPec12Pb;
uint32_t Elapsedms;
```

Actions **case APP\_STATE\_INIT** :

- Init. lcd et backlight ON
- Mettre à 0 Elapsedms
- Pour que la lecture des colonnes fonctionne, il faut un niveau '0' sur les lignes. Etablir les 4 lignes en sortie, à 0 (utilisation des fonctions PLIB\_PORTS).
- Etablir appData.state à WAIT

Action **case APP\_STATE\_WAIT** :

- Si Elapsedms > 1000, éteindre le rétro-éclairage.

Actions **case APP\_STATE\_SERVICE\_TASKS** :

- Remettre ON le rétro-éclairage et Elapsedms à 0
- Afficher état des 5 entrées CN
- Etablir appData.state à WAIT

## TESTS ET OBSERVATIONS

- Timer :  
Vérifiez à l'oscilloscope si le signal sur LED0 est bien à 500 Hz.
- CN :
  - Observez si une pression sur une des touches provoque une interruption (observation LED1)...
  - ...ainsi que l'affichage correspondant.
  - Vérifiez si la LED2 reflète l'état de PEC12\_PB.