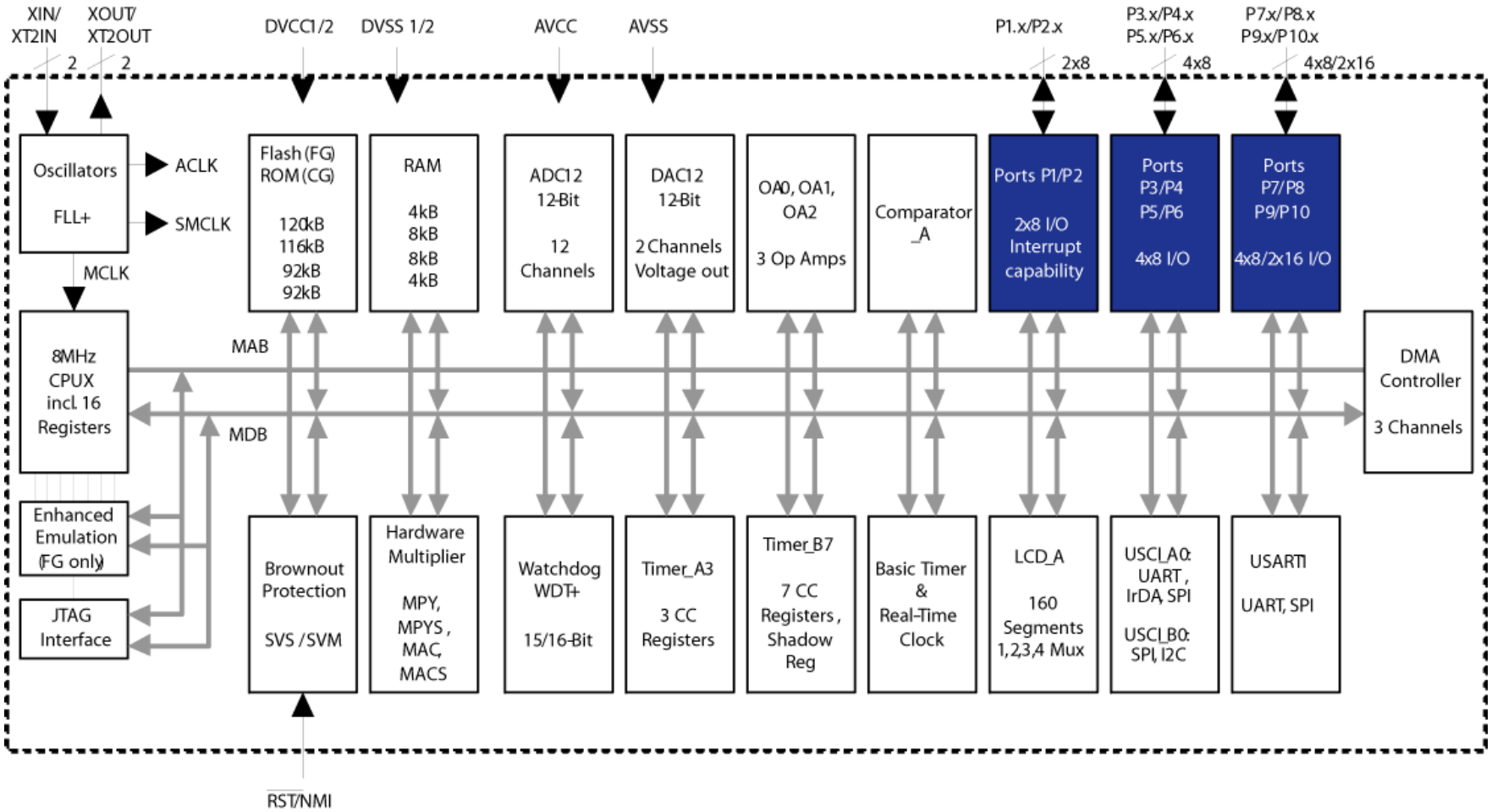
Les entrées-sorties programmables

GPIO = General Purpose Input Output



Caractéristiques principales

Les microcontrôleurs de la famille MSP430 ont jusqu'à 10 ports programmables d'entrée-sortie, P1 à P10.

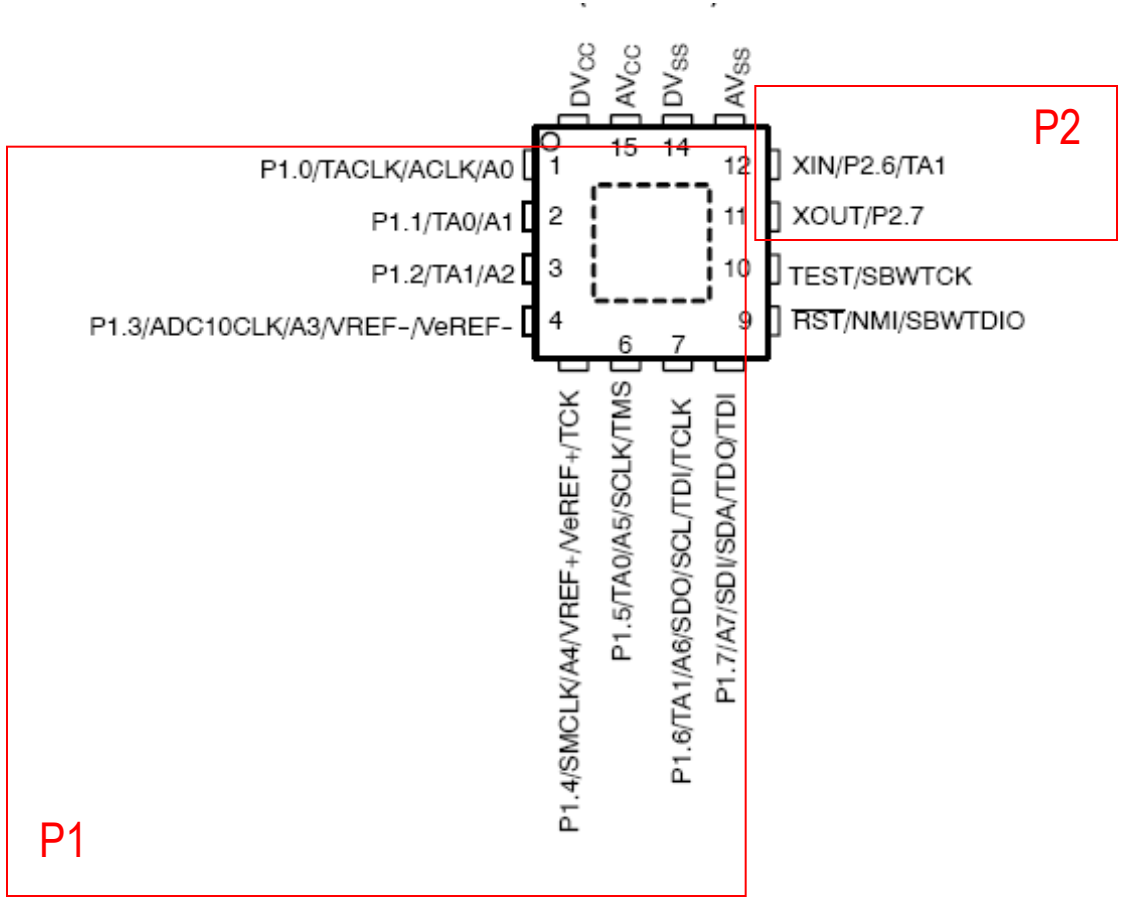
Chaque port a huit broches d'entrée-sortie et chaque broche est configurable individuellement.

Les ports P1 et P2 peuvent être utilisés comme source d'interruption :

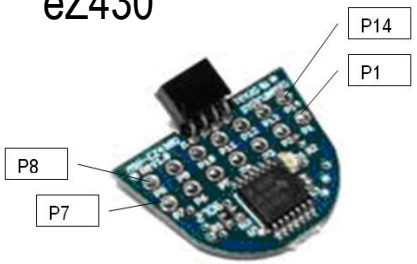
- chaque interruption peut être configurées individuellement (flanc montant ou descendant),
- toutes les lignes source du port P1 forment un vecteur d'interruption unique,
- toutes les lignes source du port P2 forment un vecteur d'interruption unique.

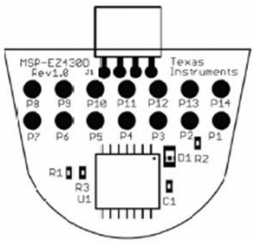
Périphérique : les entrées – sorties programmables (GPIO)

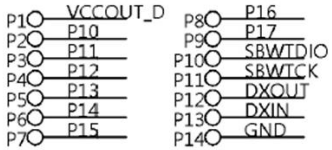
Pinout du MSP430**F2012**



Attention:
Ne pas confondre les Px.y des GPIO avec les Pn (*pin numbers*) du module eZ430

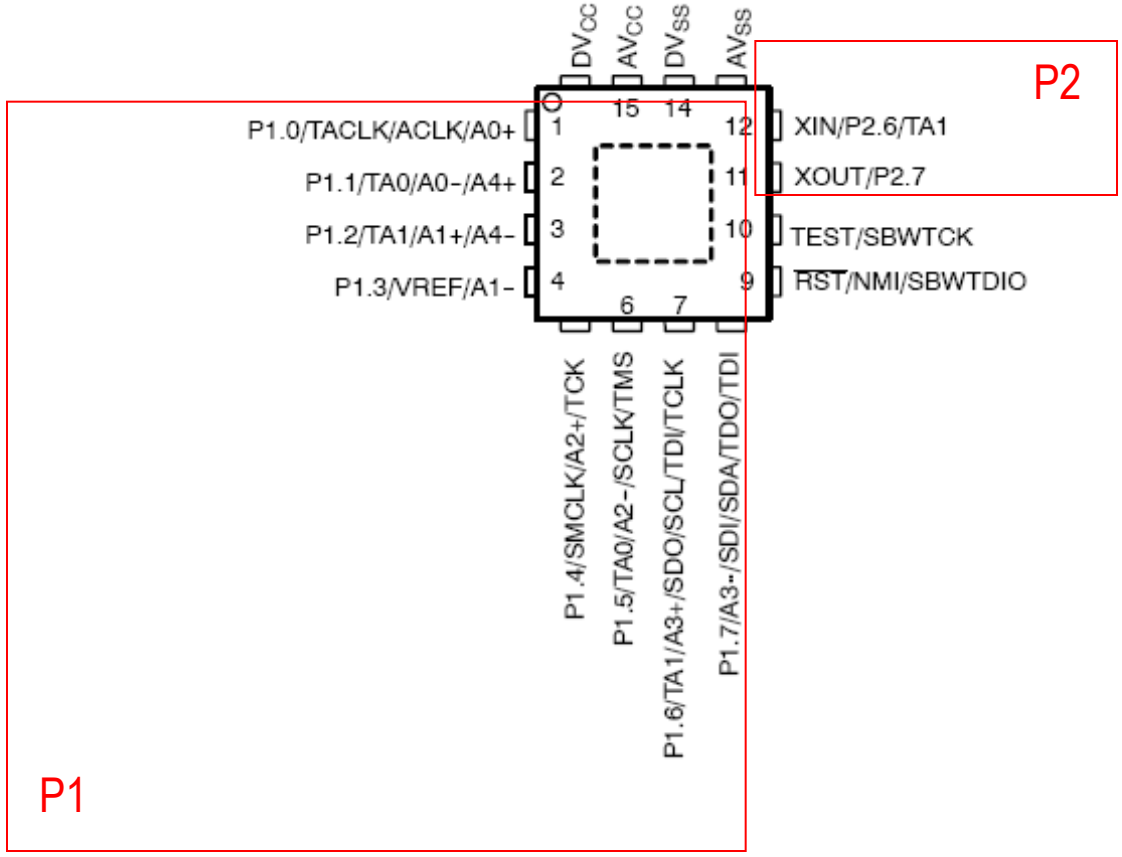




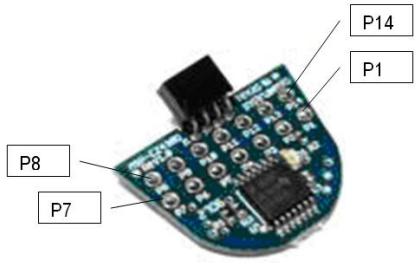


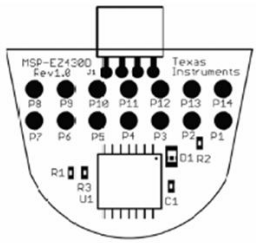
Périphérique : les entrées – sorties programmables (GPIO)

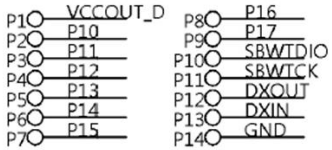
Pinout du MSP430**F2013**



Attention:
Ne pas confondre les Px.y des GPIO avec les Pn du module eZ430





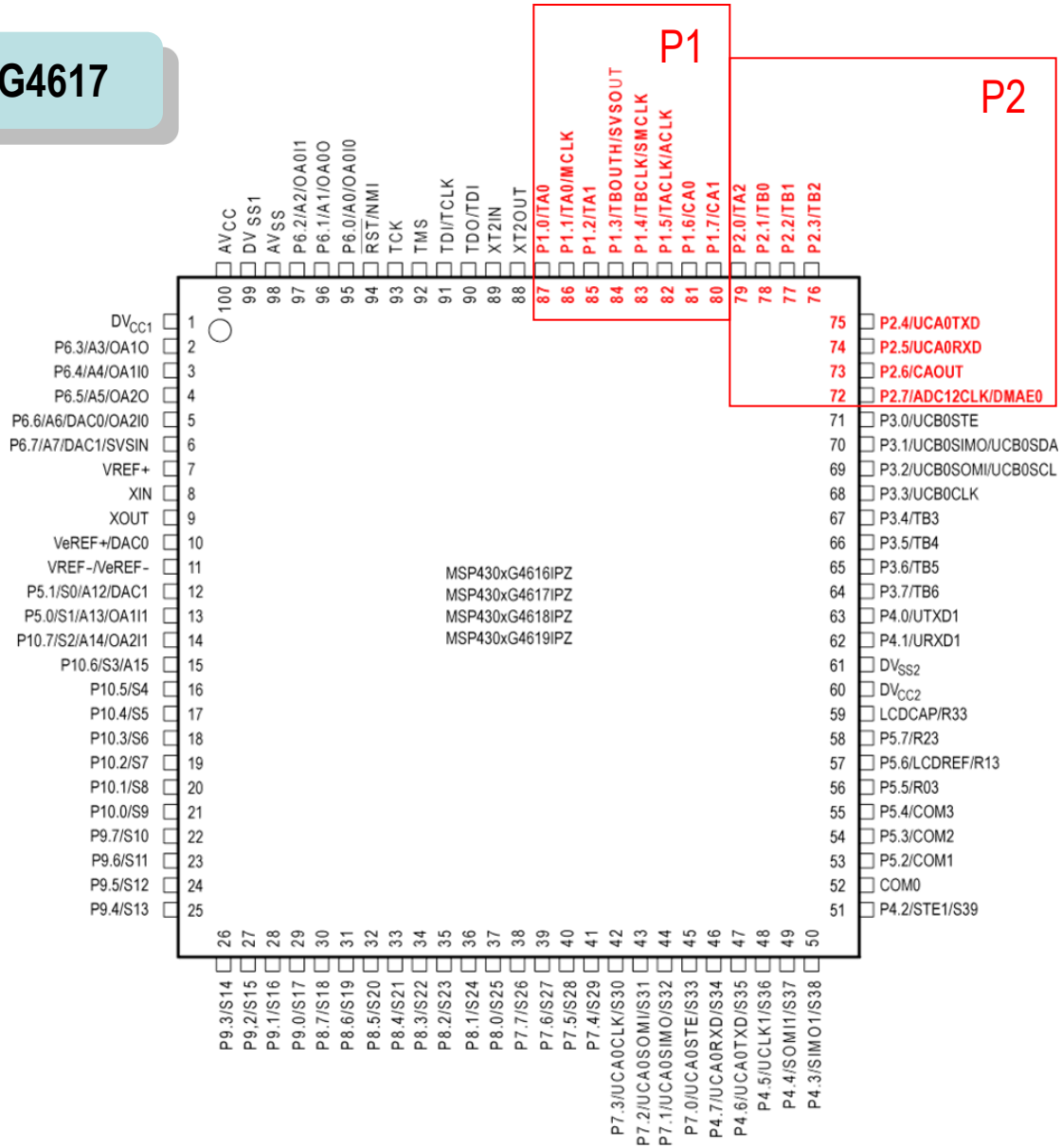


Périphérique : les entrées – sorties programmables (GPIO)

Pinout du MSP430**FG4617**

Les microcontrôleurs de la famille MSP430 ont jusqu'à 10 ports programmables d'entrée-sortie, P1 à P10.

Chaque port a huit broches d'entrée-sortie et chaque broche est configurable individuellement.



Opérations sur le GPIO

Les entrées – sorties sont configurées par programmation.

Chaque registre dédié au GPIO est un **registre à 8 bits** et est accédé avec des **instructions sur un byte**.

Registres dédiés au port P1

Le tableau ci-dessous donne l'ensemble des registres utiles pour la configuration du port P1.

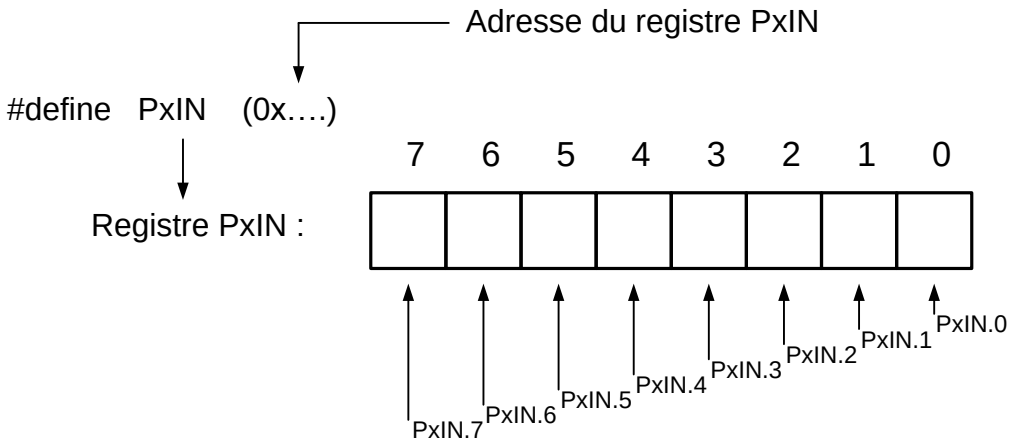
Port	***Registre***	***Nom du registre***	***Adresse***	***Type***	***Etat initial***
P1	Entrée	P1IN	0x020	Lecture seule	-
	Sortie	P1OUT	0x021	lecture / écriture	inchangé
	Direction	P1DIR	0x022	lecture / écriture	entrée
	Flag d'interruption	P1IFG	0x023	lecture / écriture	désactivé
	Sélection du flanc	P1IES	0x024	lecture / écriture	inchangé
	Autorisation des interruptions	P1IE	0x025	lecture / écriture	désactivé
	Sélection port / module	P1SEL	0x027	lecture / écriture	Port I/0

Les autres ports (P2, P3, etc.) sont configurés de la même manière.

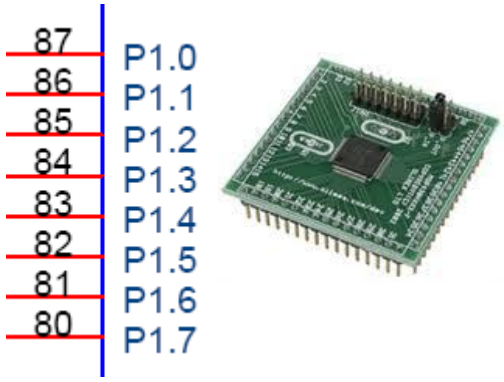
Registre d'entrée : PxIN

Chaque bit de chaque registre PxIN correspond à la valeur logique du signal d'entrée de la broche correspondante, lorsque la broche est configurée en entrée

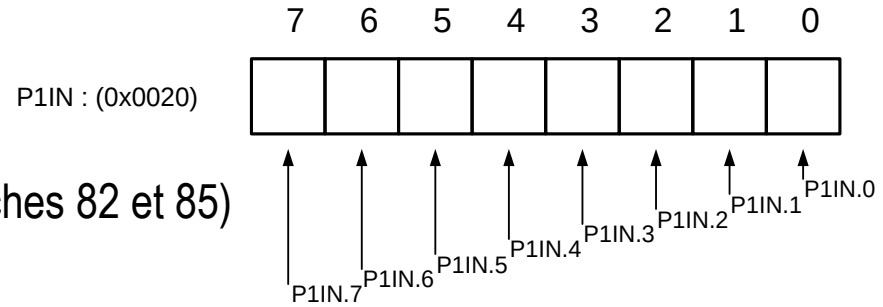
Bit = 0 : l'entrée est à l'état logique bas
Bit = 1 : l'entrée est à l'état logique haut



Exemple : MSP430FG4617
Port 1



Registre d'entrée : P1IN, lecture des entrées



Par exemple, lecture de l'entrée P1.5 et P1.2 (broches 82 et 85)

En C :

```
A=P1IN & (BIT5 | BIT2);
```

Voir ficher d'entête io430.h → io430xG46x.h

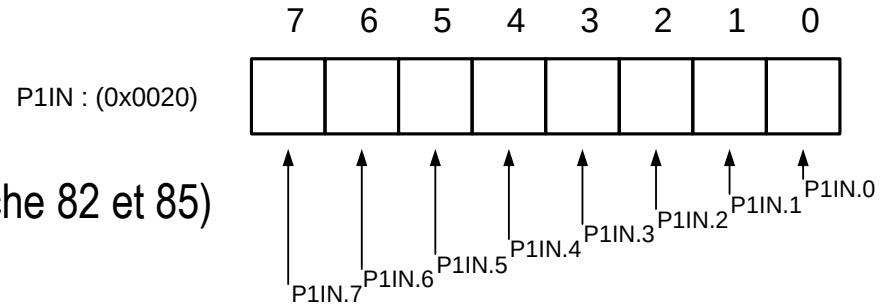
```
__no_init volatile union
 {
  unsigned __READ char P1IN;  /* Port 1 Input */

  struct
  {
    unsigned __READ char P1IN_0  : 1;
    unsigned __READ char P1IN_1  : 1;
    unsigned __READ char P1IN_2  : 1;
    unsigned __READ char P1IN_3  : 1;
    unsigned __READ char P1IN_4  : 1;
    unsigned __READ char P1IN_5  : 1;
    unsigned __READ char P1IN_6  : 1;
    unsigned __READ char P1IN_7  : 1;
  } P1IN_bit;
} @ 0x0020;
```

__no_init est un mot clé étendu (ne fait pas partie du C ANSI), il signifie que toute initialisation est supprimée

```
#define __READ   const

#define BIT5            (0x0020)
```

Périphérique : les entrées – sorties programmables (GPIO)

Registre d'entrée : P1IN, lecture des entrées



Par exemple, lecture de l'entrée P1.5 et P1.2 (broche 82 et 85)

En assembleur :

```
BIT.B  #0x24,&0x0020  ; lecture des entrées P1.5 et P1.2
```

ou :

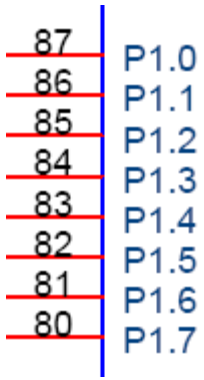
```
BIT.B  #(BIT5|BIT2),&P1IN ; lecture des entrées P1.5 et P1.2
```

Voir ficher d'entête MSP430.h → MSP430xG46x.h

Directive assembleur : `#define DEFC(name, address) sfrb name = address;`

Macro instruction :
```
#define P1IN_          (0x0020) /* Port 1 Input */
READ_ONLY DEFC( P1IN         , P1IN_)
```

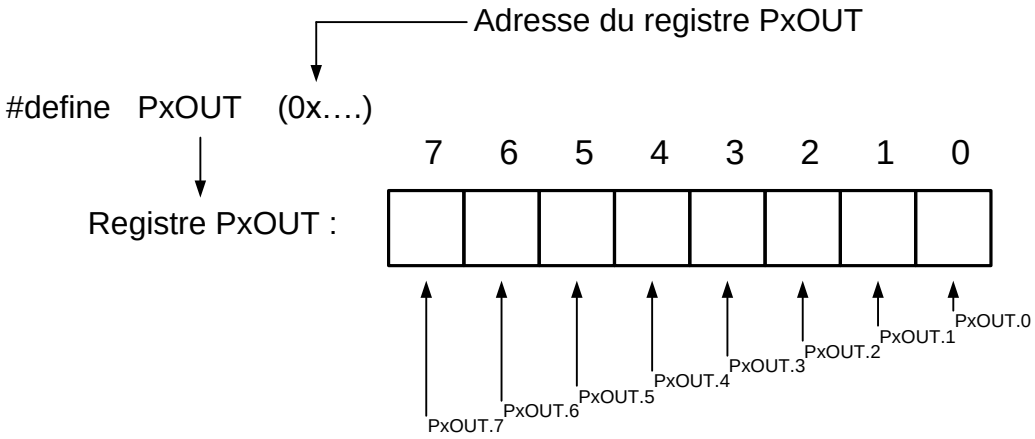
Directive assembleur :
```
…
#define BIT2           (0x0004)
…
#define BIT5           (0x0020)
…
```



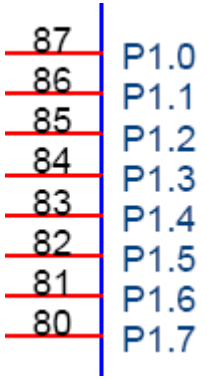
Registre de sortie : PxOUT

Chaque bit de chaque registre PxOUT correspond à la valeur logique du signal de sortie de la broche correspondante, lorsque la broche est configurée en sortie

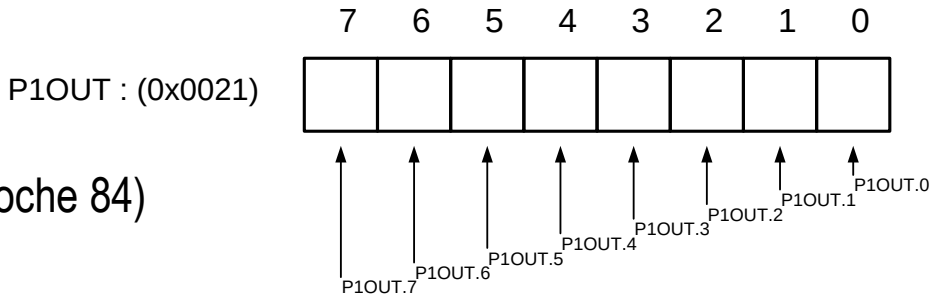
Bit = 0 : la sortie est à l'état logique bas
Bit = 1 : la sortie est à l'état logique haut



Exemple : MSP430FG4617
Port 1



Registre de sortie : P1OUT



Par exemple, écriture sur la sortie P1.3 (broche 84)

En C :

```
P1OUT |= BIT3;
```

Voir ficher d'entête io430.h → io430xxxxxx.h (fichier d'entête spécifique à chaque modèle)

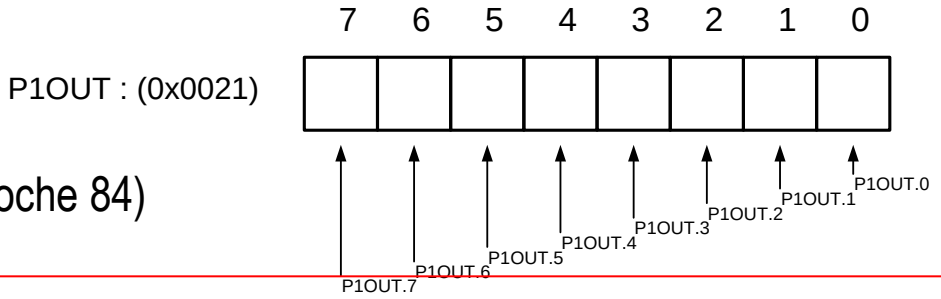
```
__no_init volatile union
{
  unsigned  char P1OUT;  /* Port 1 Output */

  struct
  {
    unsigned  char P1OUT_0  : 1;
    unsigned  char P1OUT_1  : 1;
    unsigned  char P1OUT_2  : 1;
    unsigned  char P1OUT_3  : 1;
    unsigned  char P1OUT_4  : 1;
    unsigned  char P1OUT_5  : 1;
    unsigned  char P1OUT_6  : 1;
    unsigned  char P1OUT_7  : 1;
  } P1OUT_bit;
} @ 0x0021;
```

`__no_init` est un mot clé étendu (ne fait pas partie du C ANSI), il signifie que toute initialisation est supprimée

```
#define BIT3                (0x0008)
```

Périphérique : les entrées – sorties programmables (GPIO)

Registre de sortie : P1OUT



Par exemple, écriture sur la sortie P1.3 (broche 84)

En C :

```
P1OUT_bit.P1OUT_3 = 1 ;
```

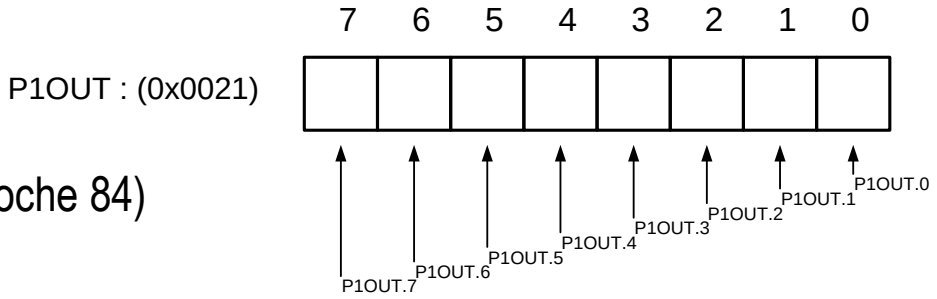
Voir ficher d'entête io430.h → io430xxxxxx.h (fichier d'entête spécifique à chaque modèle)

```
__no_init volatile union
{
  unsigned  char P1OUT;  /* Port 1 Output */

  struct
  {
    unsigned  char P1OUT_0  : 1;
    unsigned  char P1OUT_1  : 1;
    unsigned  char P1OUT_2  : 1;
    unsigned  char P1OUT_3  : 1;
    unsigned  char P1OUT_4  : 1;
    unsigned  char P1OUT_5  : 1;
    unsigned  char P1OUT_6  : 1;
    unsigned  char P1OUT_7  : 1;
  } P1OUT_bit;
} @ 0x0021;
```

__no_init est un mot clé étendu (ne fait pas partie du C ANSI), il signifie que toute initialisation est supprimée

Périphérique : les entrées – sorties programmables (GPIO)

Registre de sortie : P1OUT



Par exemple, écriture sur la sortie P1.3 (broche 84)

En assembleur :

```
BIS.B  #0x08,&0x0021 ; 1 logique sur la sortie P1.3
```

ou :

```
BIS.B  #BIT3,&P1IN     ; 1 logique sur la sortie P1.3
```

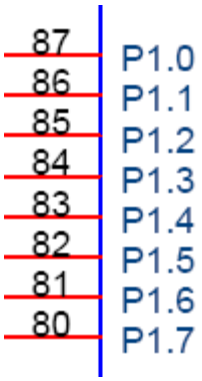
Voir ficher d'entête MSP430.h → MSP430xG46x.h

Directive assembleur : `#define DEFC(name, address) sfrb name = address;`

Macro instruction :
```
#define P1OUT_          (0x0021) /* Port 1 Input */
READ_ONLY DEFC( P1OUT           , P1OUT_)
```

Directive assembleur :
```
…
#define BIT3                (0x0008)
…
```

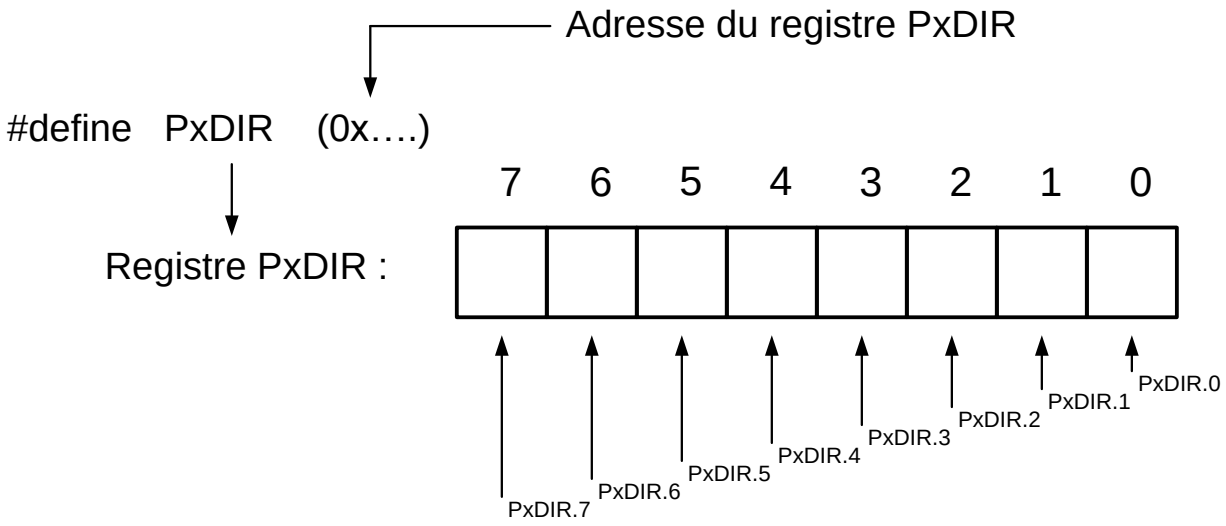
Exemple : MSP430FG4617
Port 1



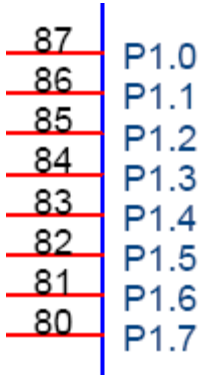
Registre de direction (entrée/sortie) : PxDIR

Chaque bit de chaque registre PxDIR permet de choisir si la broche correspondante est configurée en entrée ou en sortie. Lors de l'utilisation des broches pour d'autres fonction, la configuration de direction doit être réalisée selon les besoins de la fonction choisie.

Bit = 0 : la broche est configurée en entrée
Bit = 1 : la broche est configurée en sortie



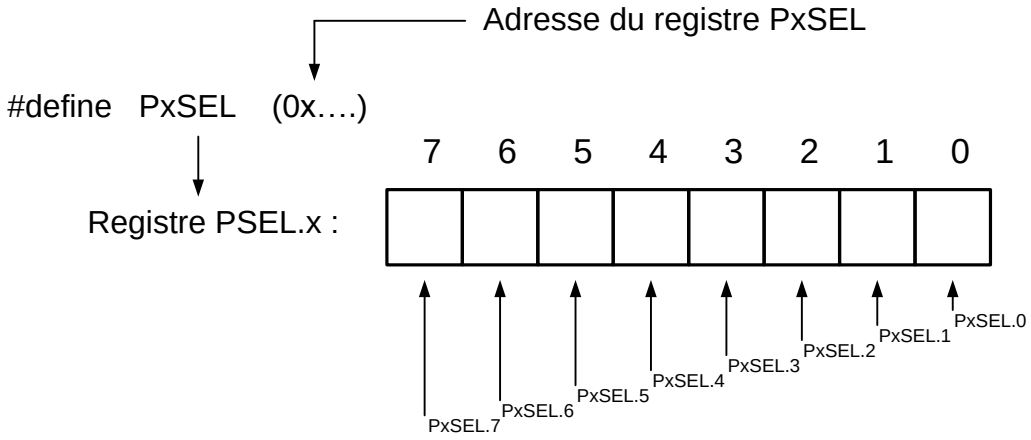
Exemple : MSP430FG4617
Port 1



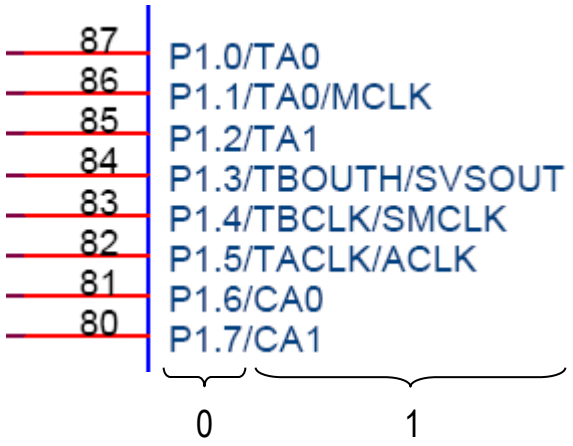
Registre de sélection de fonction : PxSEL

Chaque broche des port P1 à P10 peuvent être multiplexées avec des modules d'autres périphériques. Par conséquent chaque bit des registres PxSEL permet de sélectionner un parmi plusieurs fonctions.

Bit = 0 : la broche est configurée en entrée – sortie
Bit = 1 : la broche est configurée pour une entrée – sortie d'un module spécifique



Exemple : MSP430FG4617
Port 1



Registre de sélection de fonction : PxSEL

Assigner 1 à PxSEL.y ne définit pas automatiquement la direction (entrée ou sortie) de la broche y du port x. Le bit correspondant du registre PxDIR.y doit **aussi** être programmé en fonction des exigences du module pouvant être lié à la broche Px.y.

Par exemple en C:

```
// sortie de l'horloge ACLK sur la broche P1.5
P1SEL & = BIT5;                 // sélection de l'horloge ACLK en sortie sur P1.5
P1DIR & = BIT5;                 // assigne P1.5 en sortie
```

ou

```
// sortie de l'horloge ACLK sur la broche P1.5
P1SEL_bit.P1SEL_5 = 1;          // sélection de l'horloge ACLK en sortie sur P1.5
P1DIR_bit. P1DIR_5 = 1;         // assigne P1.5 en sortie
```

En assembleur :

```
; sortie de l'horloge ACLK sur la broche P1.5
BIS.B  #BIT5,&P1SEL             ; sélection de l'horloge ACLK en sortie sur P1.5
BIS.B  #BIT5,&P1DIR             ; assigne P1.5 en sortie
```

MSP430FG4617 : Registres dédiés au port P1, broches P1.0 à P1.5

Nom de la broche (P1.x)	x	Fonction	Bits de contrôle / signals	
			P1DIR.x	P1SEL.x
P1.0/TA0	0	P1.0 (I/0)	0 →I ; 1 →O	0
		Timer_A3.CCI0A	0	1
		Timer_A3.TA0	1	1
P1.1/TA0/MCLK	1	P1.1 (I/0)	0 →I ; 1 →O	0
		Timer_A3.CCI0B	0	1
		MCLK	1	1
P1.2/TA1	2	P1.2 (I/0)	0 →I ; 1 →O	0
		Timer_A3.CCI1A	0	1
		Timer_A3.TA1	1	1
P1.3/TBOUTH/SVSOUT	3	P1.3 (I/0)	0 →I ; 1 →O	0
		Timer_B7.TBOUTH	0	1
		SVSOUT	1	1
P1.4/TBCLK/SMCLK	4	P1.4 (I/0)	0 →I ; 1 →O	0
		Timer_B7.TBCLK	0	1
		SMCLK	1	1
P1.5/TACLK/ACLK	5	P1.5 (I/0)	0 →I ; 1 →O	0
		Timer_A3.TACLK	0	1
		ACLK	1	1

MSP430FG4617 : Registres dédiés au port P1, broches P1.6 et P1.7

Nom (P1.x)	***x***	***Fonction***	***Bits de contrôle / signals***		
			CAPD.x	***P1DIR.x***	***P1SEL.x***
P1.6 / CA0	6	P1.6 (I/0)	0	0 →I ; 1 →O	0
		CA0	1	-	-
P1.7 / CA1	7	P1.7 (I/0)	0	0 →I ; 1 →O	0
		CA1	1	-	-

Exercices: lecture/écriture des ports P1 et P2 en C

1. Activer en sortie le premier et le quatrième bit du port P2 .
2. Lire tous les bits du port P1 en entrée et les répercuter en sortie sur le port P2 .
3. Programmer le port P1.0 (premier bit du port 1) en bouton pressoir pour activer et désactiver le port P2.3 .

Périphérique : les entrées – sorties programmables (GPIO)

Registres dédiés aux Ports P3 à P6

Port	Register	Short Form	Address	Register Type	Initial State
P3	Input	P3IN	018h	Read only	–
	Output	P3OUT	019h	Read/write	Unchanged
	Direction	P3DIR	01Ah	Read/write	Reset with PUC
	Port Select	P3SEL	01Bh	Read/write	Reset with PUC
P4	Input	P4IN	01Ch	Read only	–
	Output	P4OUT	01Dh	Read/write	Unchanged
	Direction	P4DIR	01Eh	Read/write	Reset with PUC
	Port Select	P4SEL	01Fh	Read/write	Reset with PUC
P5	Input	P5IN	030h	Read only	–
	Output	P5OUT	031h	Read/write	Unchanged
	Direction	P5DIR	032h	Read/write	Reset with PUC
	Port Select	P5SEL	033h	Read/write	Reset with PUC
P6	Input	P6IN	034h	Read only	–
	Output	P6OUT	035h	Read/write	Unchanged
	Direction	P6DIR	036h	Read/write	Reset with PUC
	Port Select	P6SEL	037h	Read/write	Reset with PUC

Pseudo-ports de 16 bits PA et PB

Sur les modèles MSP430 sur lesquels ils sont présents, les ports **P7−P8** et **P9−P10** sont arrangées de manière à pouvoir être adressés sous la forme d'un port unique de **16 bits** par couple.

La couple P7/P8 désigné sous le nom de la PA et du P9/P10 est mentionnée sous le nom de PB. Par exemple, pour écrire dans les registres (1 byte) de sélection P7SEL et P8SEL simultanément, il est possible de travailler avec un registre de (16 bits) de nom PASEL qui est une concaténation des registres P7SEL et P8SEL.

Par exemple en assembleur

```
BIS.B  #0x01,&P7SEL
BIS.B  #0x05,&P8SEL   } BIS.W  #0x0501,&PASEL
```

Périphérique : les entrées – sorties programmables (GPIO)

Registres dédiés aux Ports P7 à P10

Deux port de 8 bits P7 et P8 ou un port de 16 bits PA

Deux port de 8 bits P9 et P10 ou un port de 16 bits PB

Port	Register	Short Form	Address	Register Type	Initial State
P7 PA	Input	P7IN	038h	Read only	–
	Output	P7OUT	03Ah	Read/write	Unchanged
	Direction	P7DIR	03Ch	Read/write	Reset with PUC
	Port Select	P7SEL	03Eh	Read/write	Reset with PUC
P8	Input	P8IN	039h	Read only	–
	Output	P8OUT	03Bh	Read/write	Unchanged
	Direction	P8DIR	03Dh	Read/write	Reset with PUC
	Port Select	P8SEL	03Fh	Read/write	Reset with PUC
P9 PB	Input	P9IN	008h	Read only	–
	Output	P9OUT	00Ah	Read/write	Unchanged
	Direction	P9DIR	00Ch	Read/write	Reset with PUC
	Port Select	P9SEL	00Eh	Read/write	Reset with PUC
P10	Input	P10IN	009h	Read only	–
	Output	P10OUT	00Bh	Read/write	Unchanged
	Direction	P10DIR	00Dh	Read/write	Reset with PUC
	Port Select	P10SEL	00Fh	Read/write	Reset with PUC

Interruption sur les ports P1 et P2

Chaque broche des ports P1 et P2 peuvent être utilisés pour générer des interruptions.

La configuration des interruptions est générée à l'aide des registres PxIGF, PxIE et PXIES.

L'ensemble des broches du port P1 permettent d'activer un premier vecteur d'interruption.

L'ensemble des broches du port P2 permettent d'activer un second vecteur d'interruption.

Source d'interruption	***Bits d'état des interruption***	***Adresse***	***Priorité***
I/0 Port P1 (8 lignes)	P1IFG.0 – P1IFG.7	0x0FFE8	20
I/0 Port P2 (8 lignes)	P2IFG.0 – P2IFG.7	0x0FFE2	17

Interruption sur les ports P1 et P2 : registres de configuration des interruptions P1IFG et P2IFG

Chaque bit du registre PxIFG.y correspond à un flag en relation avec la broche y du port x.

Ce flag est mis à 1 lors d'un flanc montant ou descendant du signal sur la broche y.

Chaque flag mis à 1 provoque une demande d'interruption pour autant que le flag correspondant du registre d'activation des interruptions PxIEy soit mis à 1 et que le flag d'activation générale des interruptions GIE soit également à 1.

Chaque flag du registre PxIFG.y doit être explicitement remis à 0 dans la routine d'interruption.
Il est également possible d'initier une interruption software en forçant à 1 un flag particulier.

Bit = 0 : pas d'interruption pendante
Bit = 1 : interruption active

Les interruptions ne sont activées que par des transitions des signaux placés sur les broches.

La détection de flancs (montant ou descendant) permet d'assurer qu'une demande d'interruption est prise en compte même si une interruption plus prioritaire est en cours.

Sur une même broche, deux flancs actifs successifs ne seront pris en compte que si la première interruption a été validée entre-temps (par la remise à 0 du flag PxIFG.y dans la routine d'interruption).

Interruption sur les ports P1 et P2: registres flancs actifs pour les interruptions P1IES et P2IES

Chaque bit PxIES.y permet de choisir le flanc actif pour déclencher une interruption:

Bit = 0 : déclenchement sur le flanc montant (↑)
Bit = 1 : déclenchement sur le flanc descendant (↓)

L'écriture de la configuration des flancs actifs dans les registres PxIES peut provoqué l'activation d'une interruption.

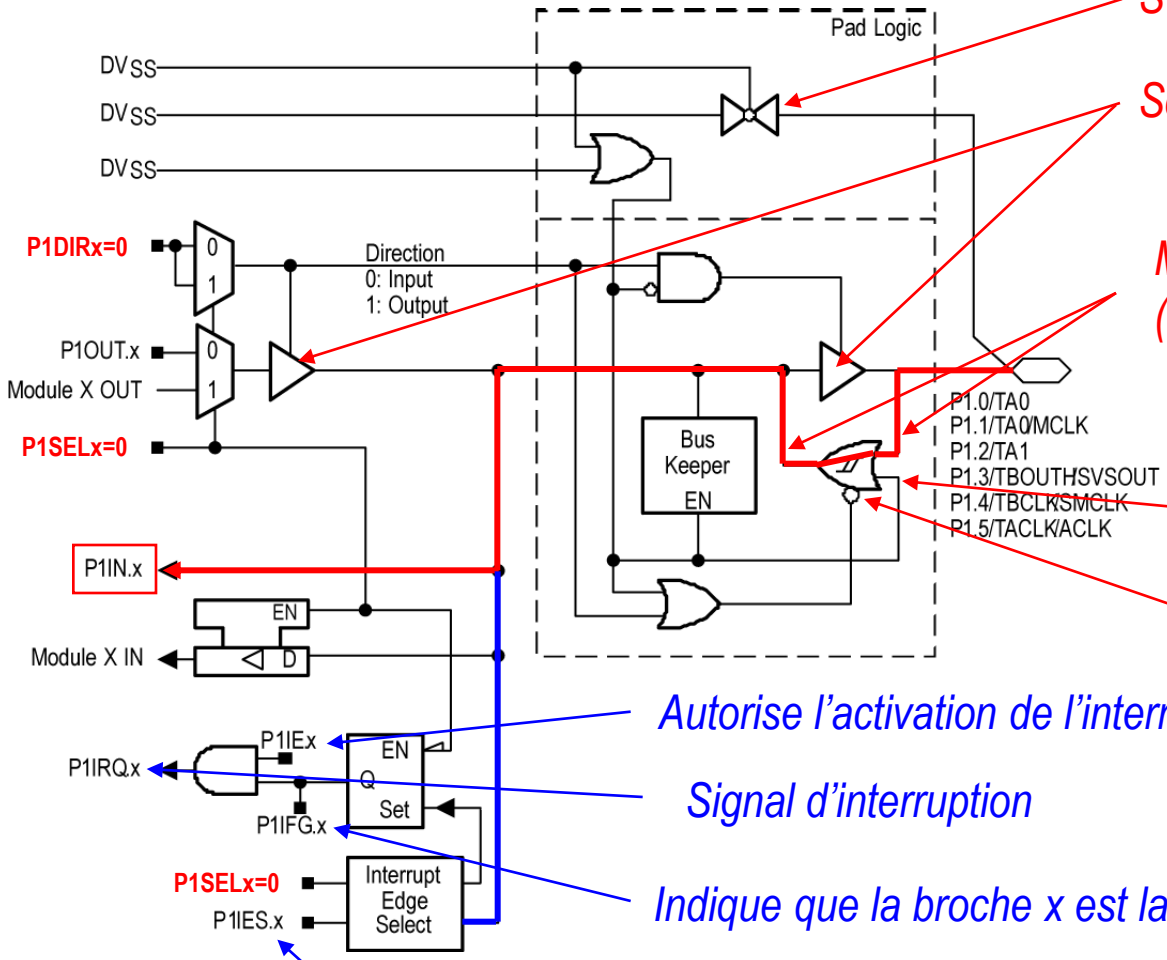
Interruption sur les ports P1 et P2: registres d'activation des interruptions P1IE et P2IE

Chaque bit PxIE.y permet de choisir d'activer ou non le déclenchement d'interruption.
Les registres PxIE sont associés bit à bit aux registres PxIFG:

Bit = 0 : interruption désactivée
Bit = 1 : interruption activée

Périphérique : les entrées – sorties programmables (GPIO)

Caractéristiques principales du port P1 configuré en sortie



Switch analogique ouvert

Sorties haute impédance

Même niveau logique (défini de l'extérieur sur la broche P1x)

Niveau logique bas

Sortie active

Autorise l'activation de l'interruption du port P1

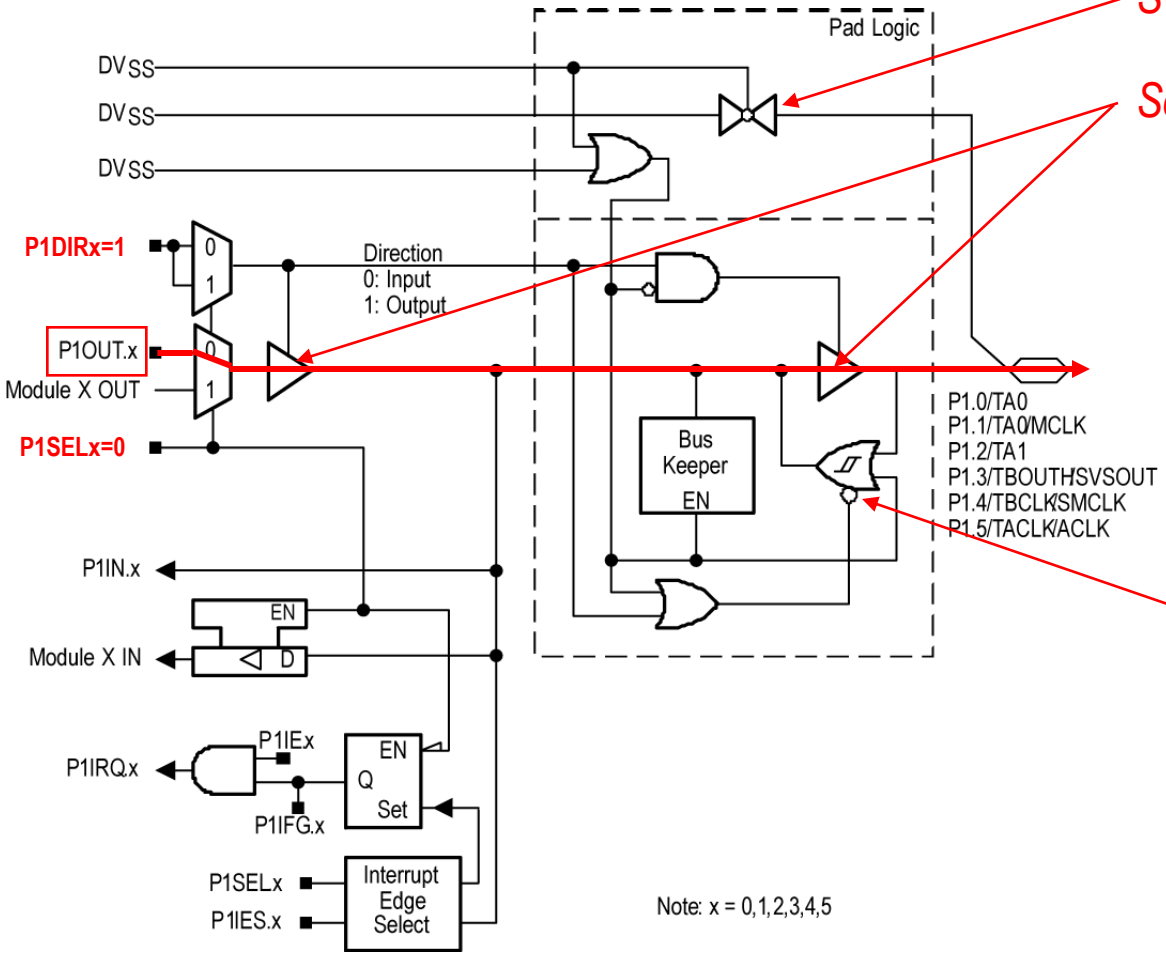
Signal d'interruption

Indique que la broche x est la source d'une interruption

Interruption sur flanc montant (P1IESx=0) ou sur flanc descendant (P1IESx=0)

Périphérique : les entrées – sorties programmables (GPIO)

Caractéristiques principales port P1 configuré en sortie



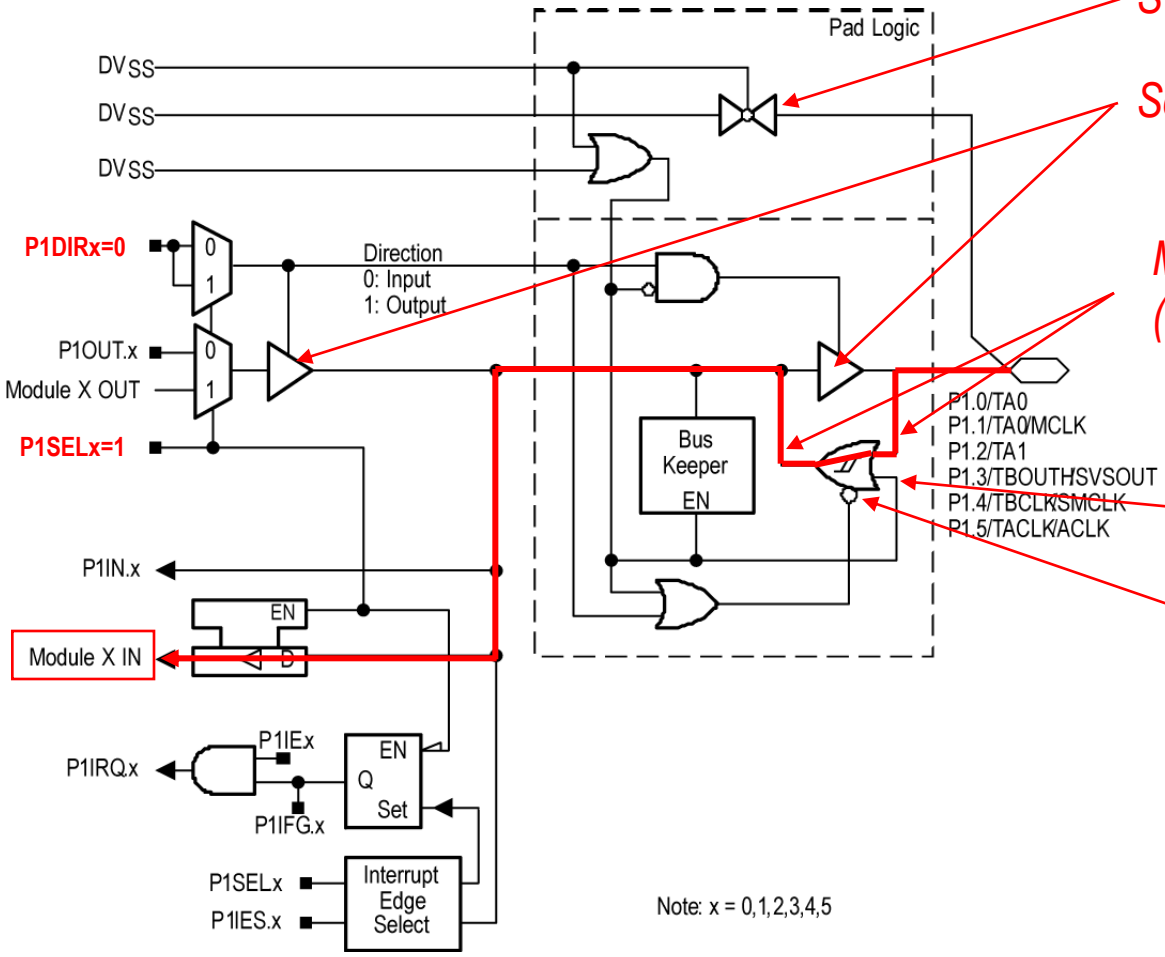
Switch analogique ouvert

Sorties active

Sortie haute impédance

Périphérique : les entrées – sorties programmables (GPIO)

Caractéristiques principales port P1 configuré en sortie



Switch analogique ouvert

Sorties haute impédance

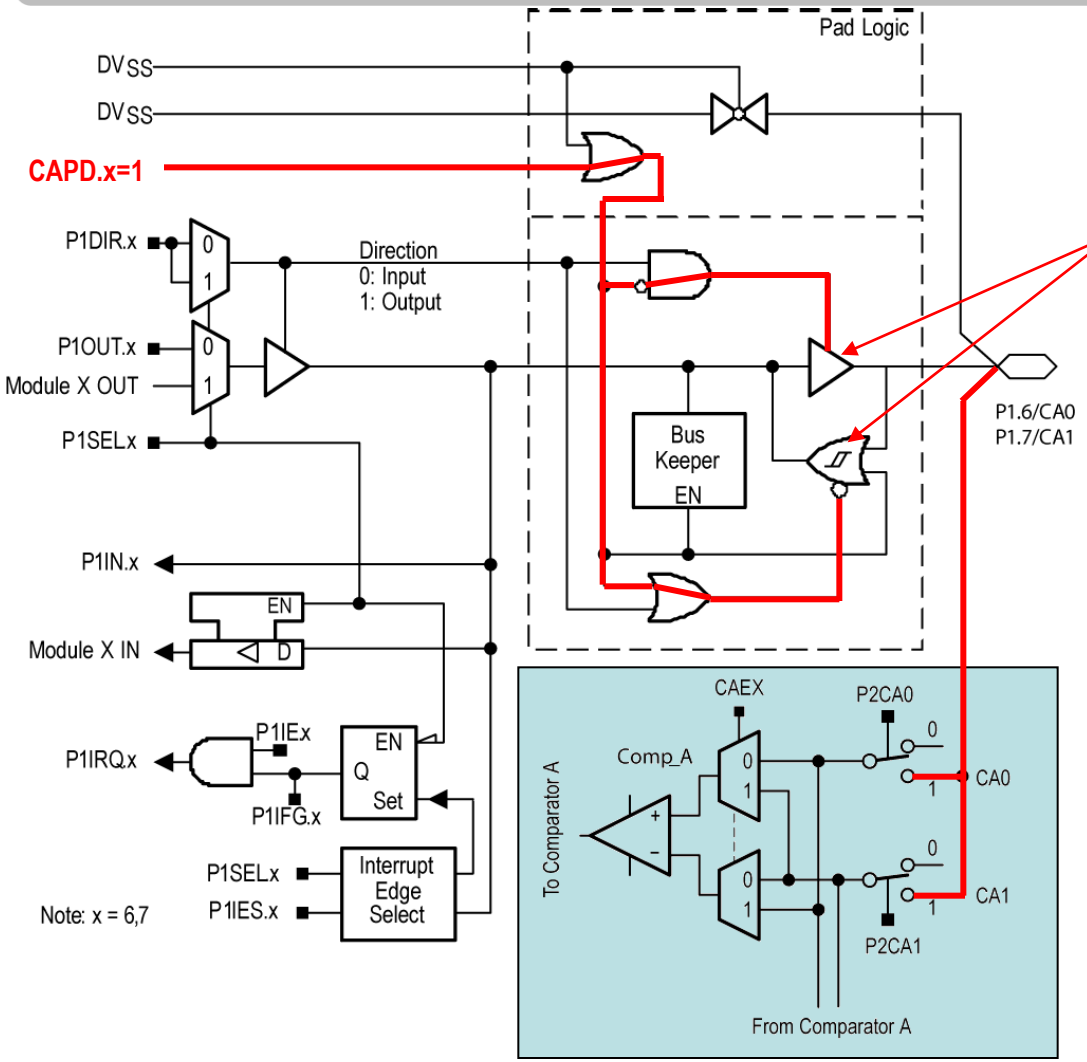
Même niveau logique
(défini de l'extérieur sur la broche P1x)

Niveau logique bas

Sortie active

Périphérique : les entrées – sorties programmables (GPIO)

Caractéristiques principales port P1 configuré entrée comparateur



Sortie haute impédance

Partie active

Exercices: configuration du port P1 en C

1. Sur le port P1, configurer en entrée les quatre premiers bits et en sortie les quatre autres.
2. Activer une interruption au flanc descendant sur le 2ème bit du port P1.
3. Répercuter à la fois sur les deux derniers bits (en sortie) du port P1 l'état du premier bit (P1.0).