

Bases des systèmes embarqués

A02 – Configuration Ports d'entrées-sorties sur
8051F020

Version 2022 – 09/11/2022 09:26



Bases des systèmes embarqués
A02 – Configuration Ports d'entrées-sorties sur 8051F020

Introduction: Affectation des broches d'entrées-sorties (I/O)



Rôles des ports d'entrées-sorties dans un processeur – Les « I/O » (Input-Output)

Définition: à l'intérieur d'un microcontrôleur, un port est un ensemble de broches destinées à communiquer avec l'environnement externe à travers des opérations d'entrées-sorties (par opposition aux opérations mémoire). Par Port, on fait aussi référence à l'électronique sous-jacente et à sa programmation via des registres.

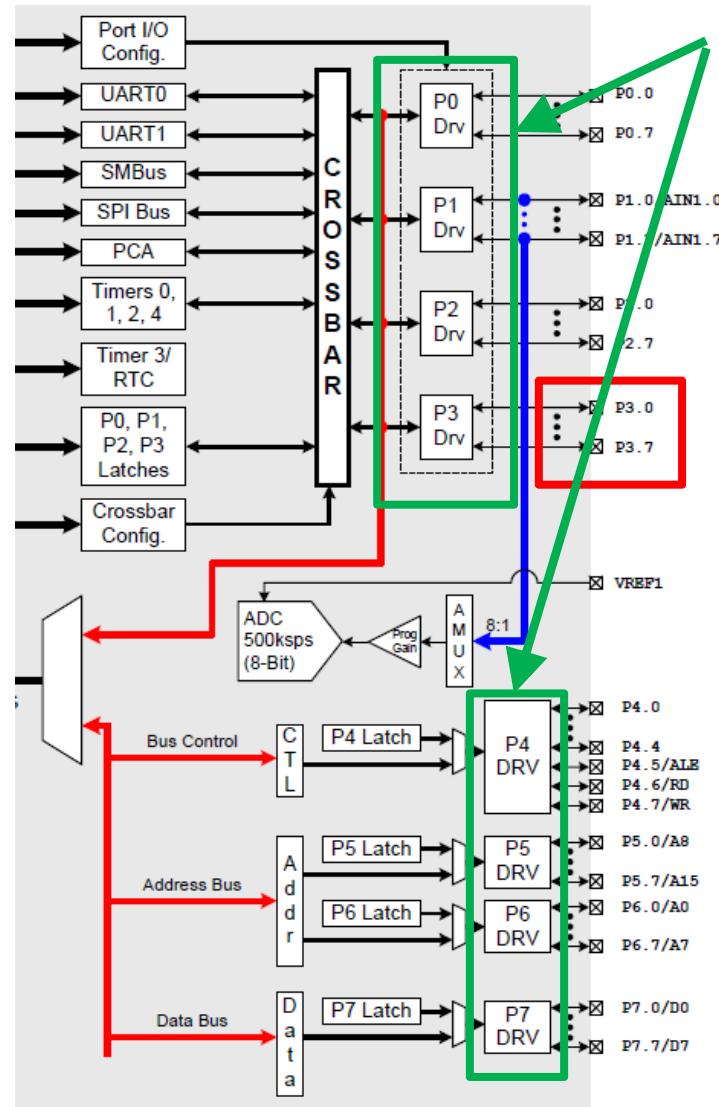
En général, les ports regroupent 8 ou 16 broches (selon les familles de processeur) et permettent

Les broches de ces ports peuvent être:

- **GPIO (General Purpose Input Output)**- Directement manipulées par du code (pilotage de LED ou d'actionneur, lecture de l'état d'un bouton poussoir, d'un capteur TOR)
- **PIO (Peripheral Input Output)**- Gérées par des dispositifs périphériques internes. Dans ce cas précis, on ne manipule donc pas directement l'état de ces broches, mais on le fait pas l'intermédiaire de la configuration et de l'utilisation des périphériques.

Attention, dans un microcontrôleur, une broche I/O peut aussi avoir d'autres fonctionnalités (accès mémoire, entrée analogiques...)

Ports Entrées – Sorties dans le 8051F020



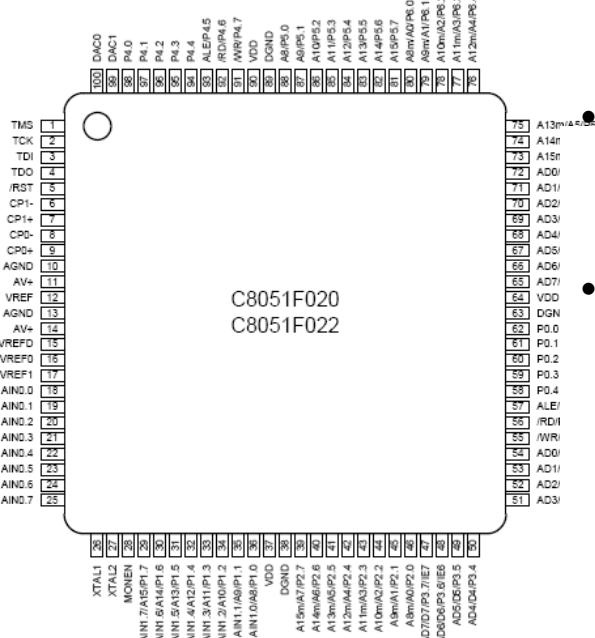
Blocs logiques associés aux divers ports

1 port = 8 broches

Le 8051F020 dispose de 8 ports d'I/O
De P0 à P7 soit 64 broches dédiées aux entrées-sorties (numériques)

Remarque: il existe aussi d'autres broches dédiées aux entrées-sorties analogiques

Le boitier du 8051F020



- Chaque broche peut avoir plusieurs fonctions: **I/O**,ADR, Data, Bus Ctrl, ADC_in

Une fonction port I/O peut être GPIO ou I/O de périphérique interne (PIO)

- Exemples:

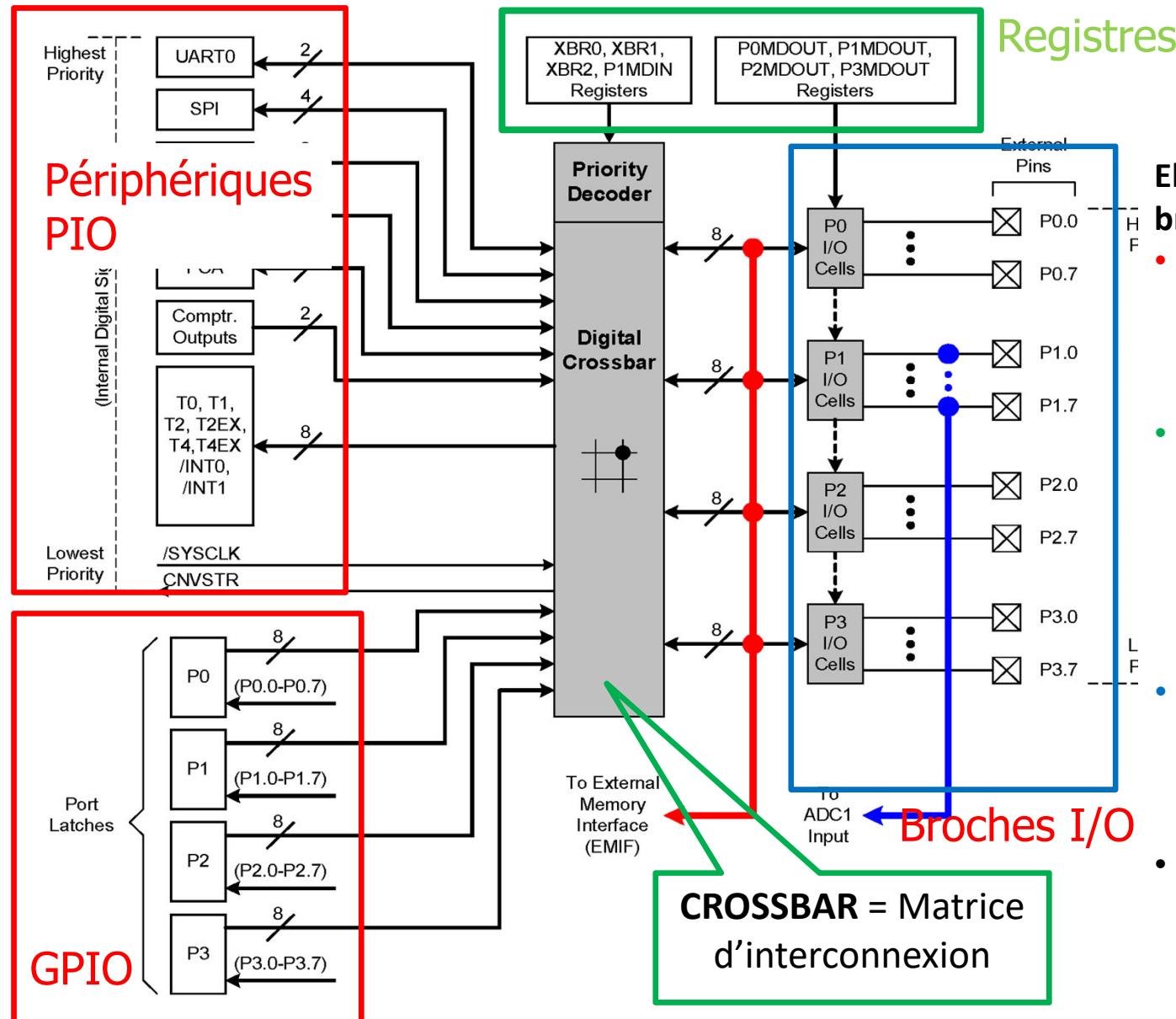
- pin61 – I/O P0.1 (**GPIO ou PIO**)
- Pin54 – Bus processeur AD0, D0 – Port P3.0 (**GPIO ou PIO**)
- Pin47 - Bus processeur AD7, D7 – Port P3.7 (**GPIO ou PIO**) – IE7(entrée INT7)

61	P0.1
60	P0.2
59	P0.3
58	P0.4
57	ALE/P0.5
56	/RD/P0.6
55	/WR/P0.7
54	AD0/D0/P3.0
53	AD1/D1/P3.1
52	AD2/D2/P3.2
51	AD3/D3/P3.3

VDD	37	
DGND	38	
A15m/A7/P2.7	39	
A14m/A6/P2.6	40	
A13m/A5/P2.5	41	
A12m/A4/P2.4	42	
A11m/A3/P2.3	43	
A10m/A2/P2.2	44	
A9m/A1/P2.1	45	
A8m/A0/P2.0	46	
AD7/D7/P3.7/I/E7	47	
AD6/D6/P3.6/I/E6	48	
AD5/D5/P3.5	49	
AD4/D4/P3.4	50	

Structure globale des Port I/O P1 à P3 dans le 8051F020

Figure 17.2. Lower Port I/O Functional Block Diagram



Aiguillage de Ports d'entrée-sorties

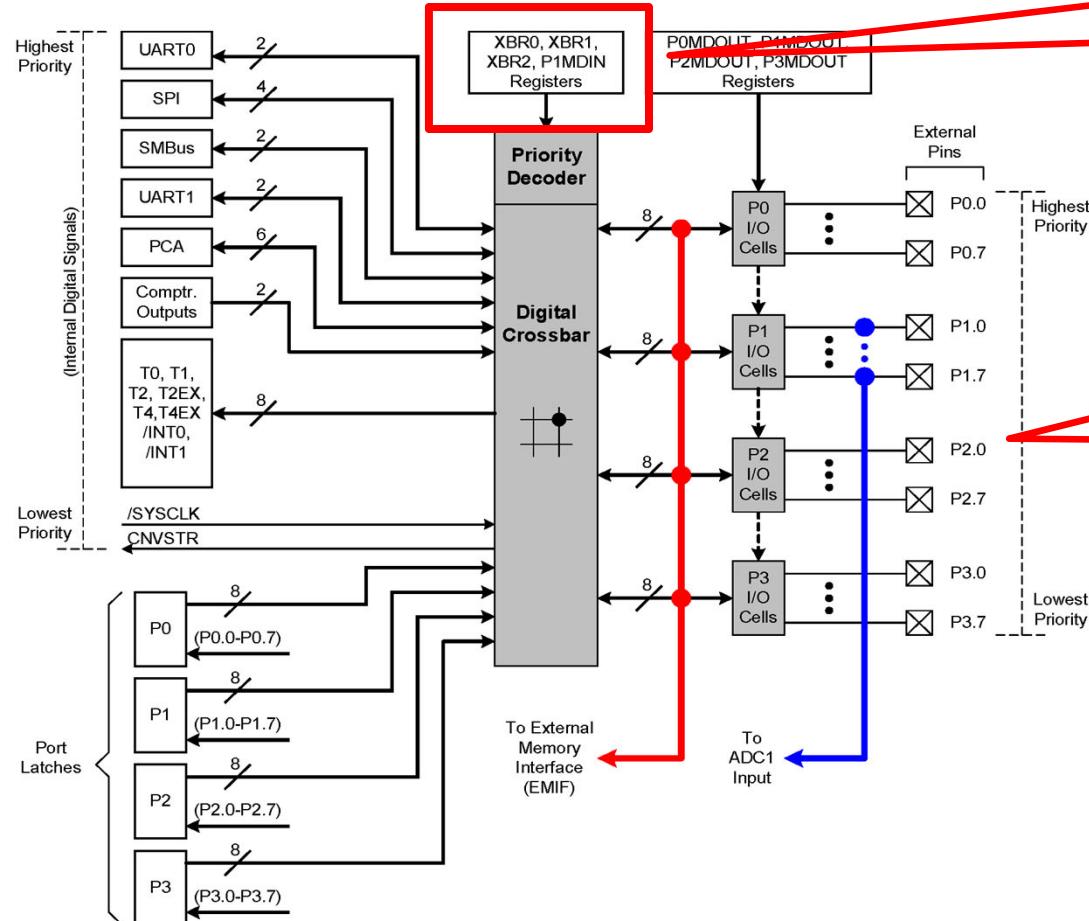
! **IMPORTANT**

Par défaut toutes les broches I/O sont configurées en GPIO

Une action sur le crossbar (par programmation de registres XBRn) permet donc de convertir des GPIO en PIO

Le crossbar, (matrice d'interconnexion) permet de router sur les broches des ports P0 à P3, les signaux d'échange des périphériques que l'on souhaite utiliser

Figure 17.2. Lower Port I/O Functional Block Diagram



Pour configurer cette matrice (Crossbar) on agira sur XBR0, XBR1, XBR2 et P1MDIN

Cette configuration rentre dans la catégorie des configurations globales

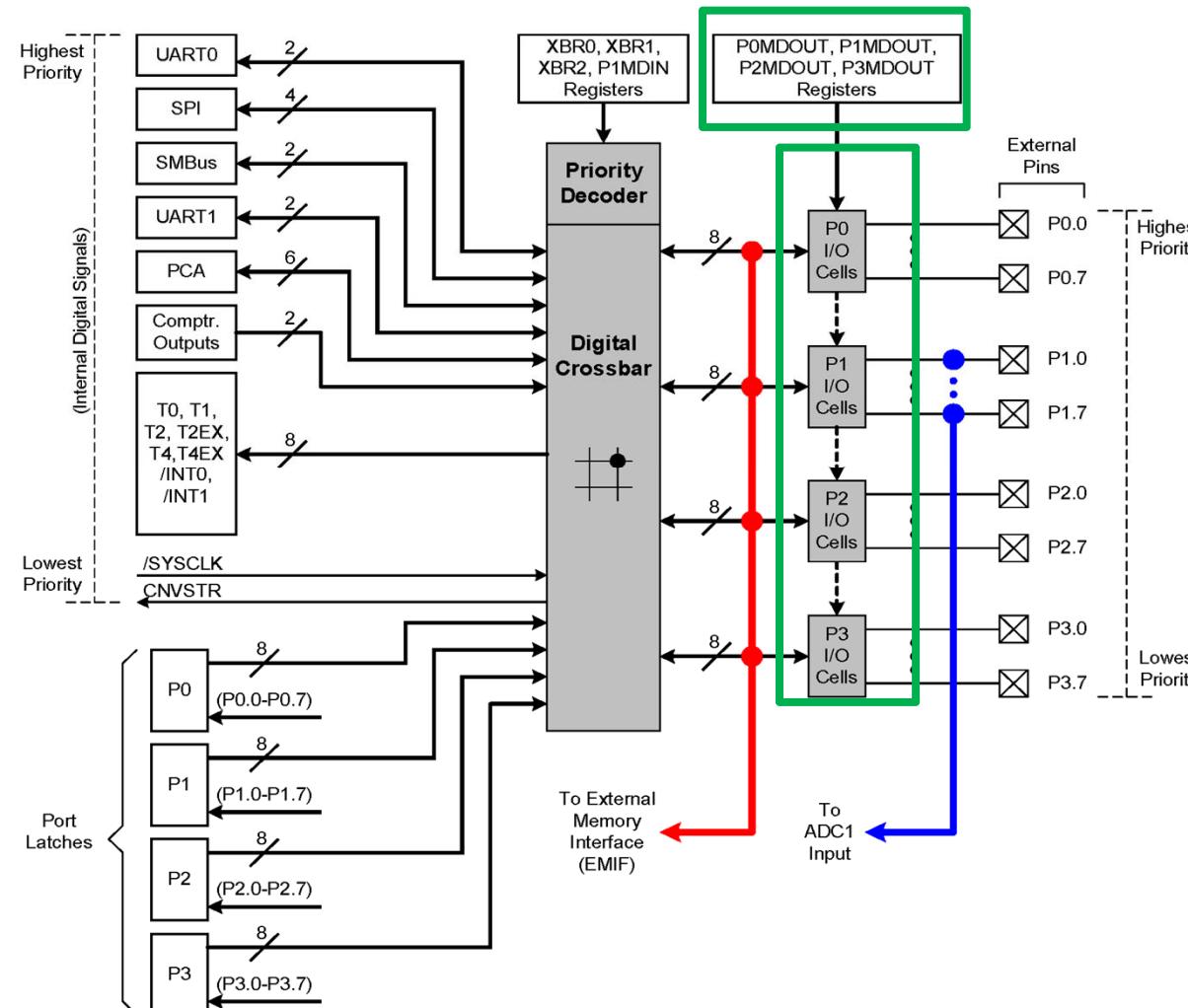
P0 à P3: peuvent être configurés en I/O périphériques internes (PIO) ou GPIO

P4 à P7 ne sont pas branchés sur le crossbar: ils sont uniquement utilisables en GPIO

Configuration des Ports d'entrées-Sorties (Périphériques – GPIO)

Cette configuration matérielle de la broche s'applique aussi bien au fonctionnement de la broche en GPIO et en PIO

Figure 17.2. Lower Port I/O Functional Block Diagram





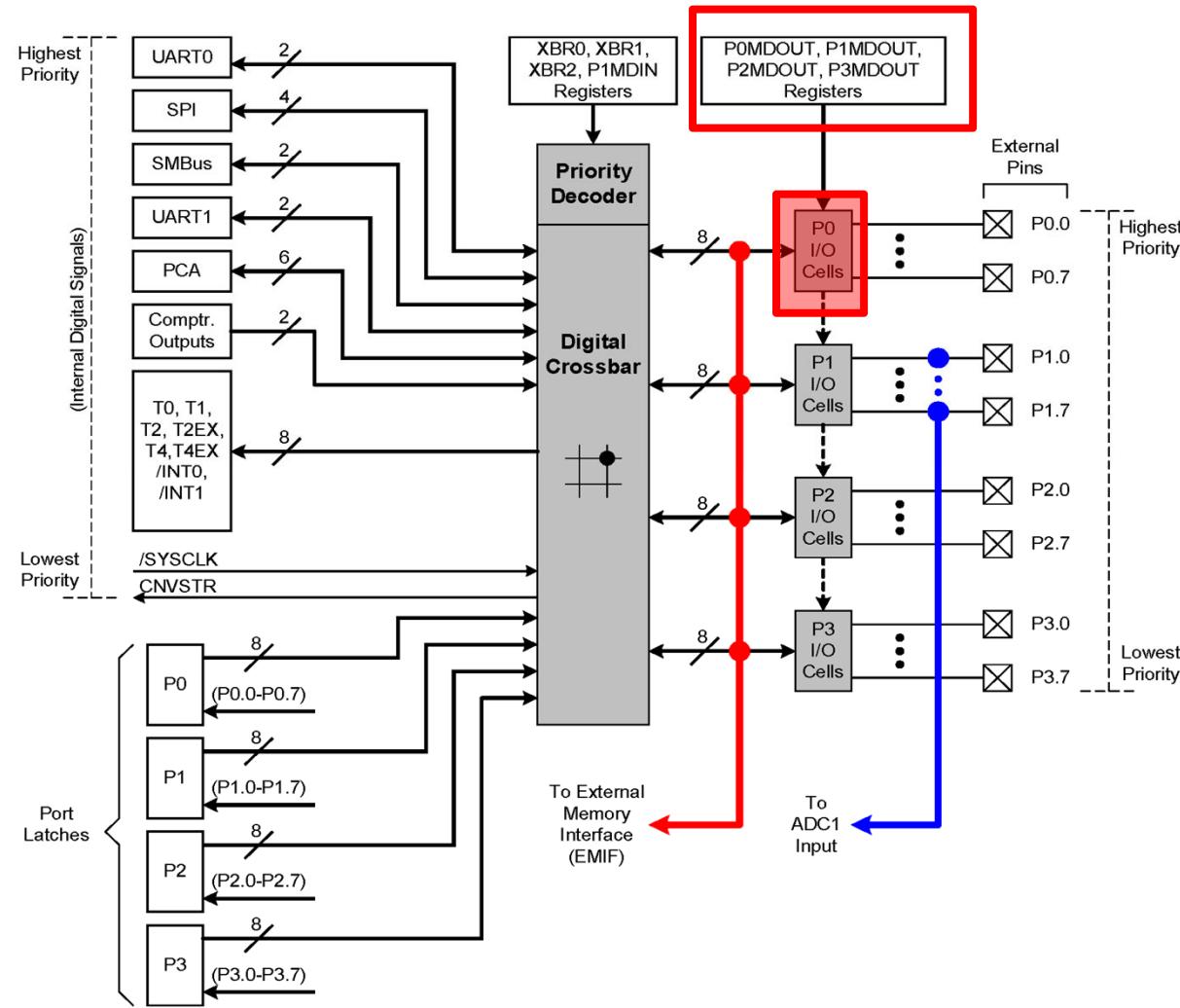
Bases des systèmes embarqués
A02 – Configuration Ports d'entrées-sorties sur 8051F020

Acte 1: Utilisation des Ports d'entrées-sorties en GPIO – Configuration en sortie

Configuration des broches I/O configuration matérielle des broches



Figure 17.2. Lower Port I/O Functional Block Diagram



REMARQUE: Cette configuration s'applique aussi bien aux broches GPIO qu'aux broches configurées en PIO (sauf cas particuliers...)

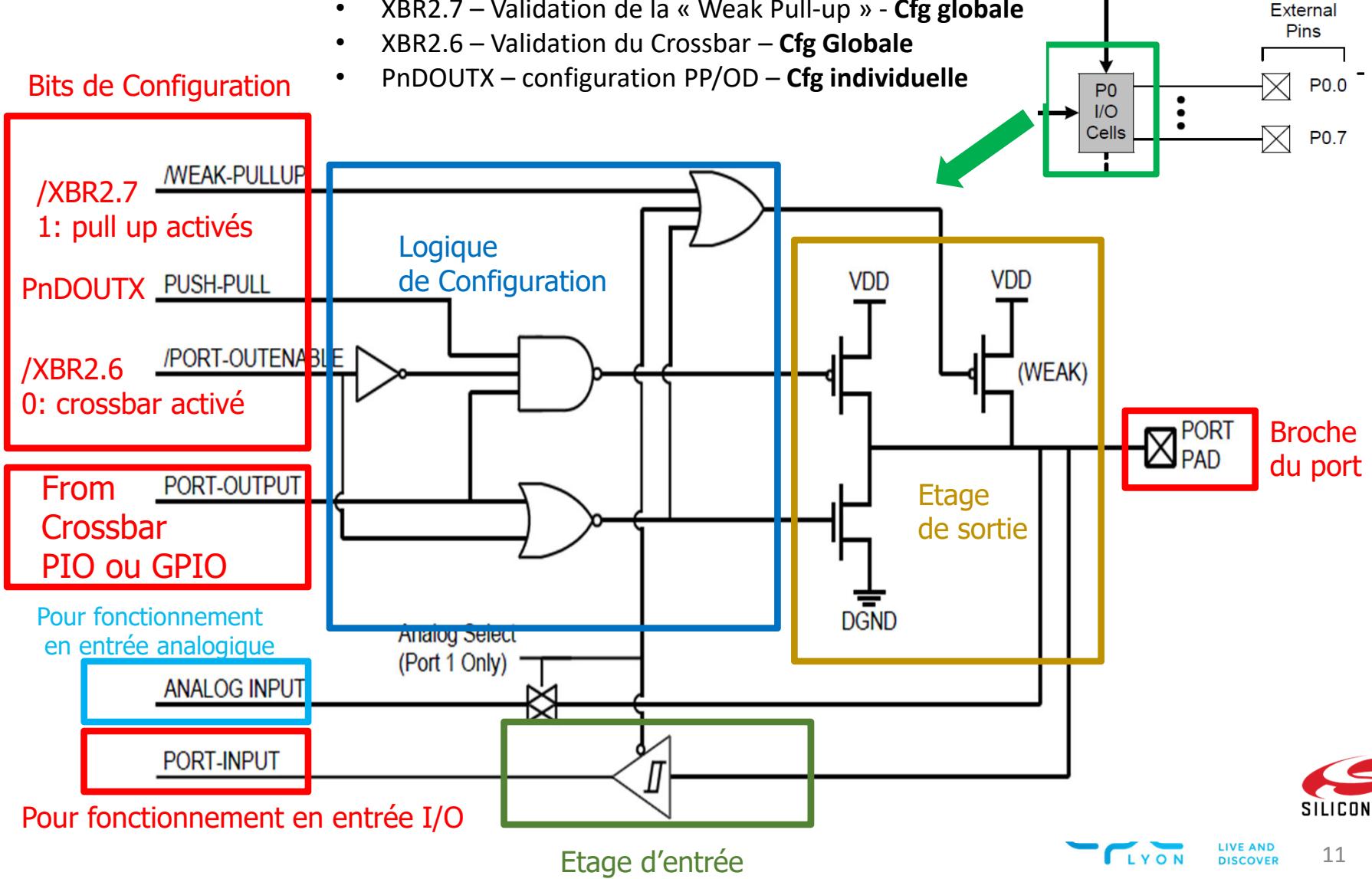
P0 à P3: I/Os périphériques internes (PIO) ou GPIO

P4 à P7: uniquement GPIO

Structure d'une cellule d'entrée-sortie (I/O Cell) dans le 8051F020

La configuration de cette cellule I/O est assurée par 3 bits:

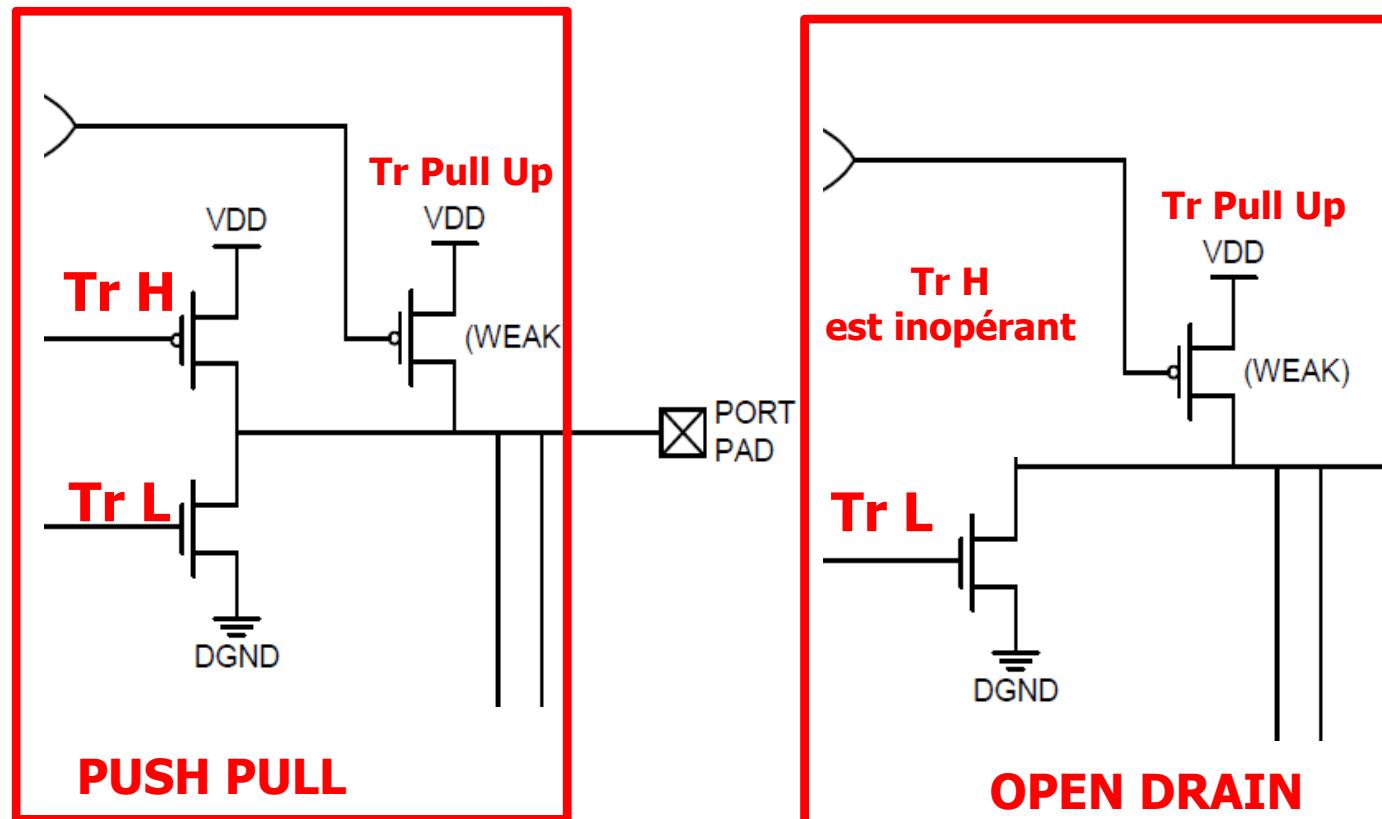
- XBR2.7 – Validation de la « Weak Pull-up » - **Cfg globale**
 - XBR2.6 – Validation du Crossbar – **Cfg Globale**
 - PnDOUTX – configuration PP/OD – **Cfg individuelle**



Etage de sortie d'une cellule d'entrée-sortie (I/O Cell) dans le 8051F020



L'étage de sortie peut être configuré en Push Pull (PP) ou en Drain ouvert (Open Drain)(OD)



! **IMPORTANT**

Les transistors TrH et TrL quel que soit le mode (PP ou OD) fonctionnent en commutation. Ils peuvent donc être dans 2 états possibles:

- Passant – en schéma équivalent on peut les représenter comme une résistance équivalente de 100 Ohm (environ)
- Ouvert – en schéma équivalent, on peut les représenter comme un circuit ouvert.

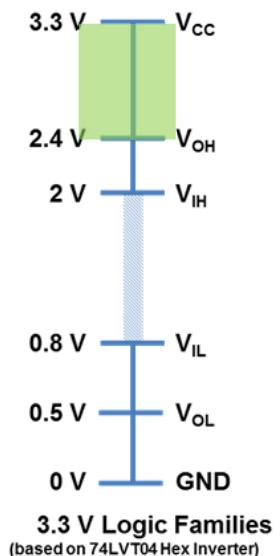
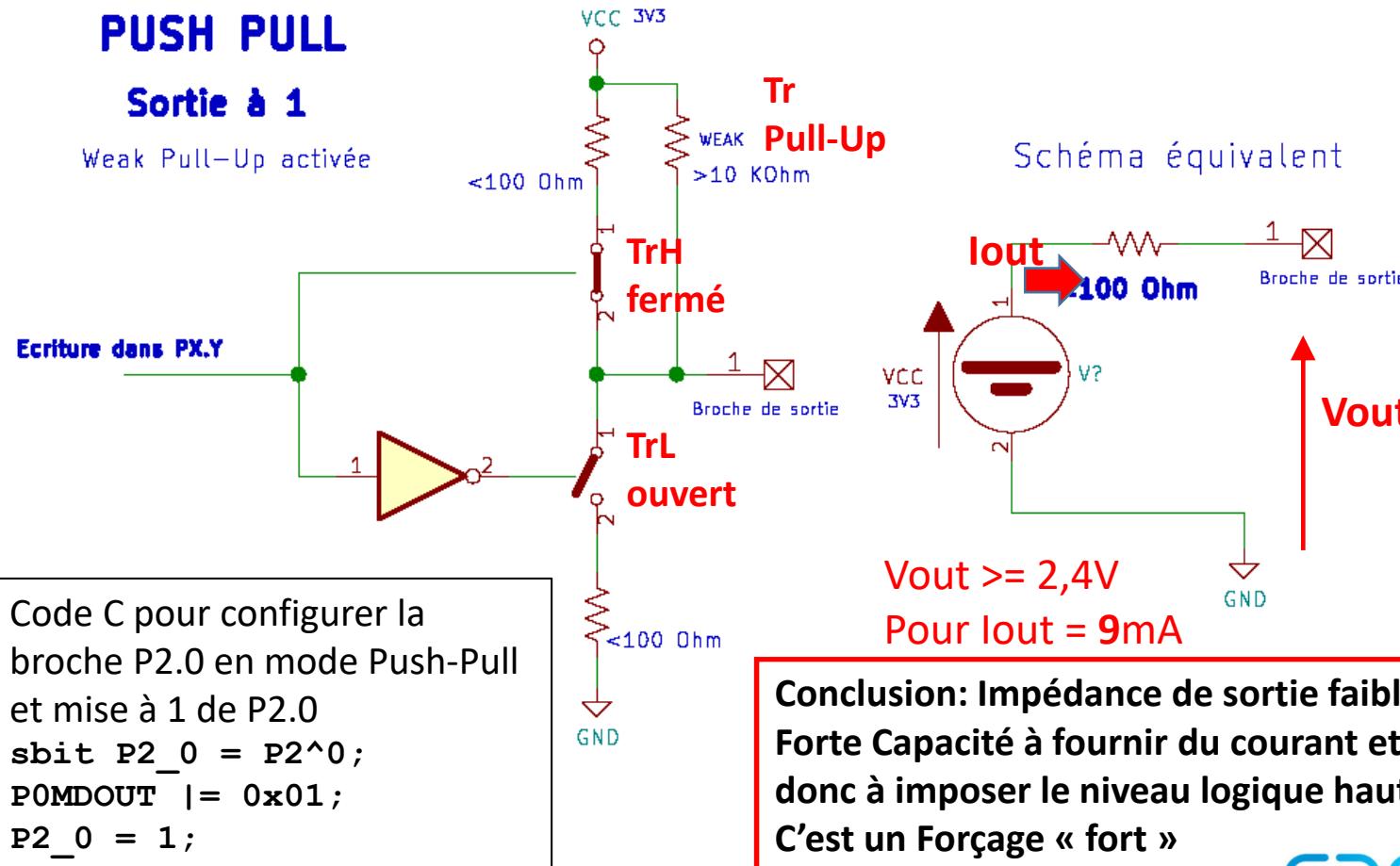
Le transistor Tr Pull Up peut avoir 2 états possibles:
- Ouvert (circuit ouvert)
- Résistance de 100KOhm (environ) (mode de fonctionnement en linéaire)



Schéma équivalent de l'étage de sortie d'une cellule I/O Mode Push Pull – Sortie à 1

Par conception, Tr L et Tr H fonctionnent en opposition (1 ouvert / 1 fermé)

Dans les schémas, par soucis de simplification, Tr L et TR H sont symbolisés par un interrupteur en série avec une résistance de 100 Ohm



IMPORTANT

NB: Le forçage caractérise la capacité d'un signal logique à imposer non niveau

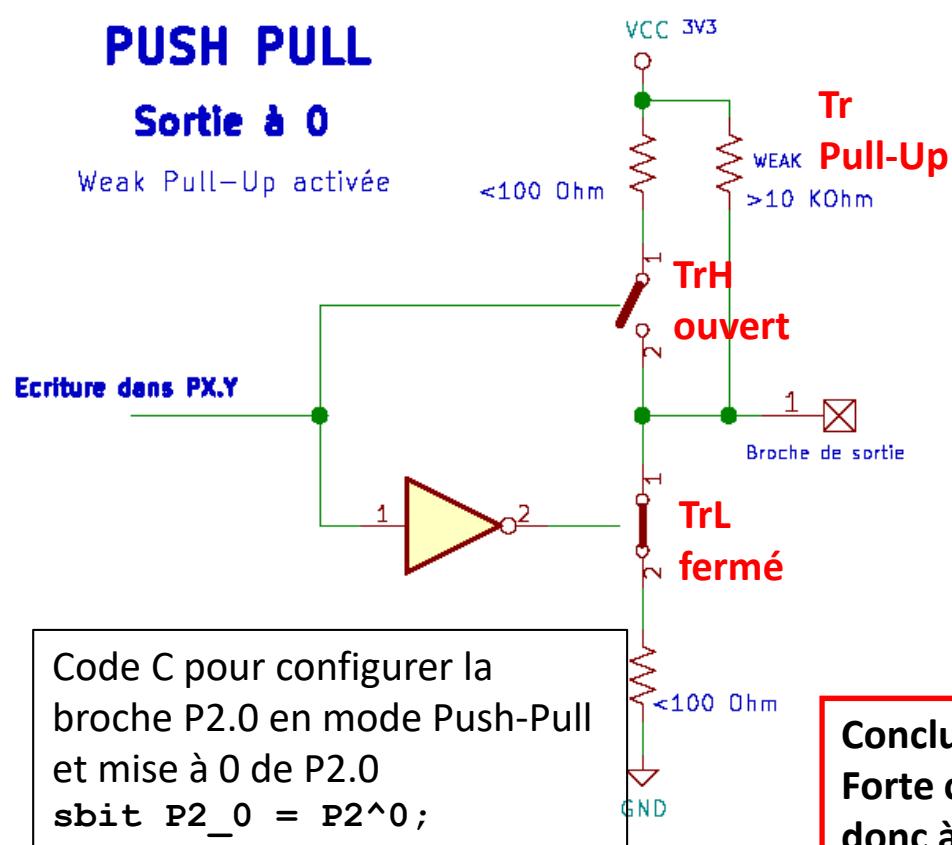
Schéma équivalent de l'étage de sortie d'une cellule I/O Mode Push Pull – Sortie à 0



PUSH PULL

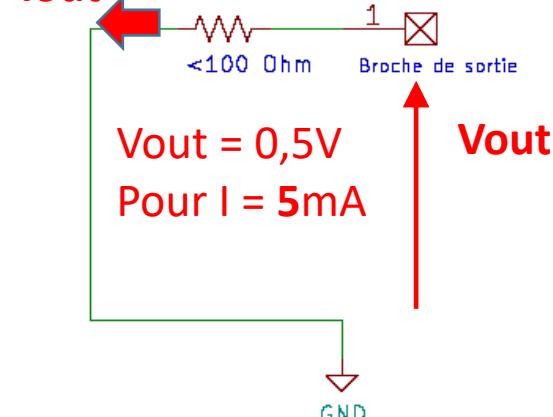
Sortie à 0

Weak Pull-Up activée

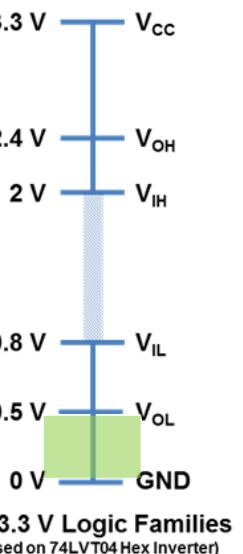


```
Code C pour configurer la
broche P2.0 en mode Push-Pull
et mise à 0 de P2.0
sbit P2_0 = P2^0;
P0MDOUT |= 0x01;
P2_0 = 0;
```

Schéma équivalent
Iout



**Conclusion: Impédance de sortie faible
Forte capacité à absorber du courant et
donc à imposer un niveau logique bas
C'est un Forçage fort**



! IMPORTANT

Schéma équivalent de l'étage de sortie d'une cellule I/O Mode Open Drain – Sortie à 0



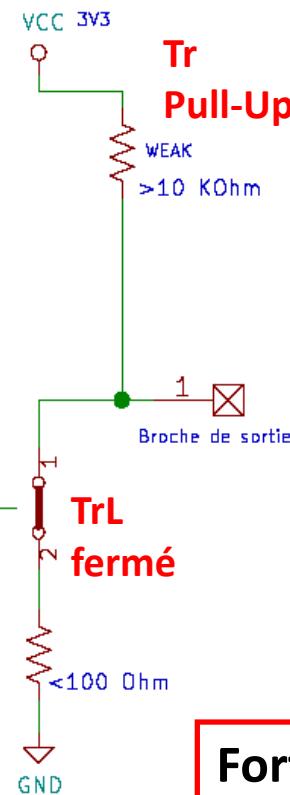
OPEN DRAIN

Sortie à 0

Weak Pull-Up activée

Ecriture dans PX.Y

TrH
inopérant



```
Code C pour configurer la  
broche P2.0 en mode Open  
Drain et mise à 0 de P2.0  
sbit P2_0 = P2^0;  
P0MDOUT &= ~0x01;  
P2_0 = 0;
```

Schéma équivalent

<100 Ohm 1 Broche de sortie

**Idem au
schéma Push
Pull sortie à 0**

GND

**Forte capacité à absorber du
courant et donc à imposer un
niveau logique bas
C'est un Forçage fort**

! **IMPORTANT**

Schéma équivalent de l'étage de sortie d'une cellule I/O Mode Open Drain – Sortie à 1



OPEN DRAIN Sortie à 1

Weak Pull-Up activée

TrH
inopérant

Ecriture dans P.X.Y

Tr
Pull-Up

NB: En fait on est plus proche des 100K que des 10K

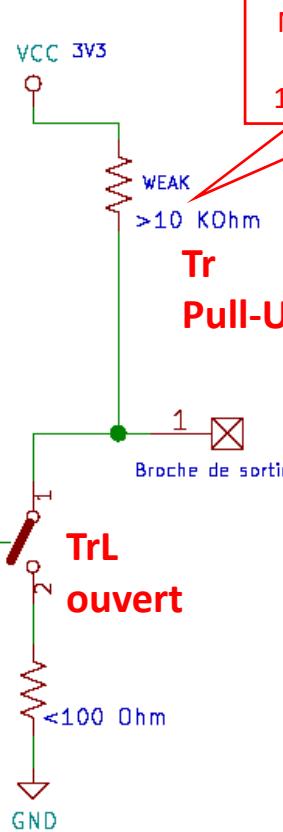
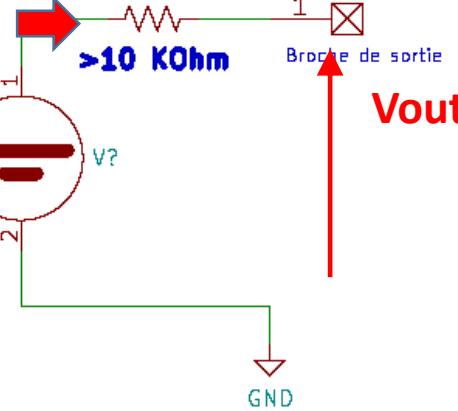


Schéma équivalent

Iout



$$V_{out} = 2,4V$$

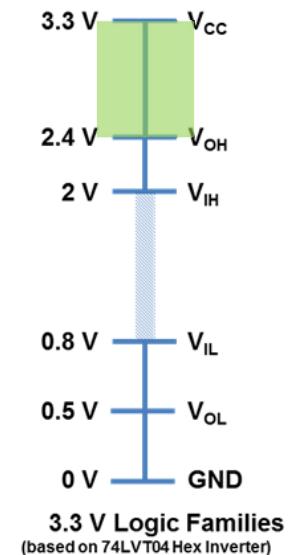
Pour $I_{out} < 0,09mA$

Code C pour configurer la broche P2.0 en mode Open Drain et mise à 1 de P2.0

```
sbit P2_0 = P2^0;
P0MDOUT &= ~0x01;
P2_0 = 1;
```

! **IMPORTANT**

Impédance de sortie élevée
Faible capacité à fournir du courant et donc à imposer un niveau logique bas
C'est un forçage faible



Différence Push-Pull / Drain ouvert

Cas de la génération d'une impulsion 10US

Cas d'une impulsion positive de durée $<10\mu s$ – Uniquement une sonde de d'oscilloscope en X10 branchée sur la broche

Drain ouvert:
Le signal produit est dégradé, mais reste utilisable, toutefois la durée du niveau haut mesurée à VCC/2 est altérée



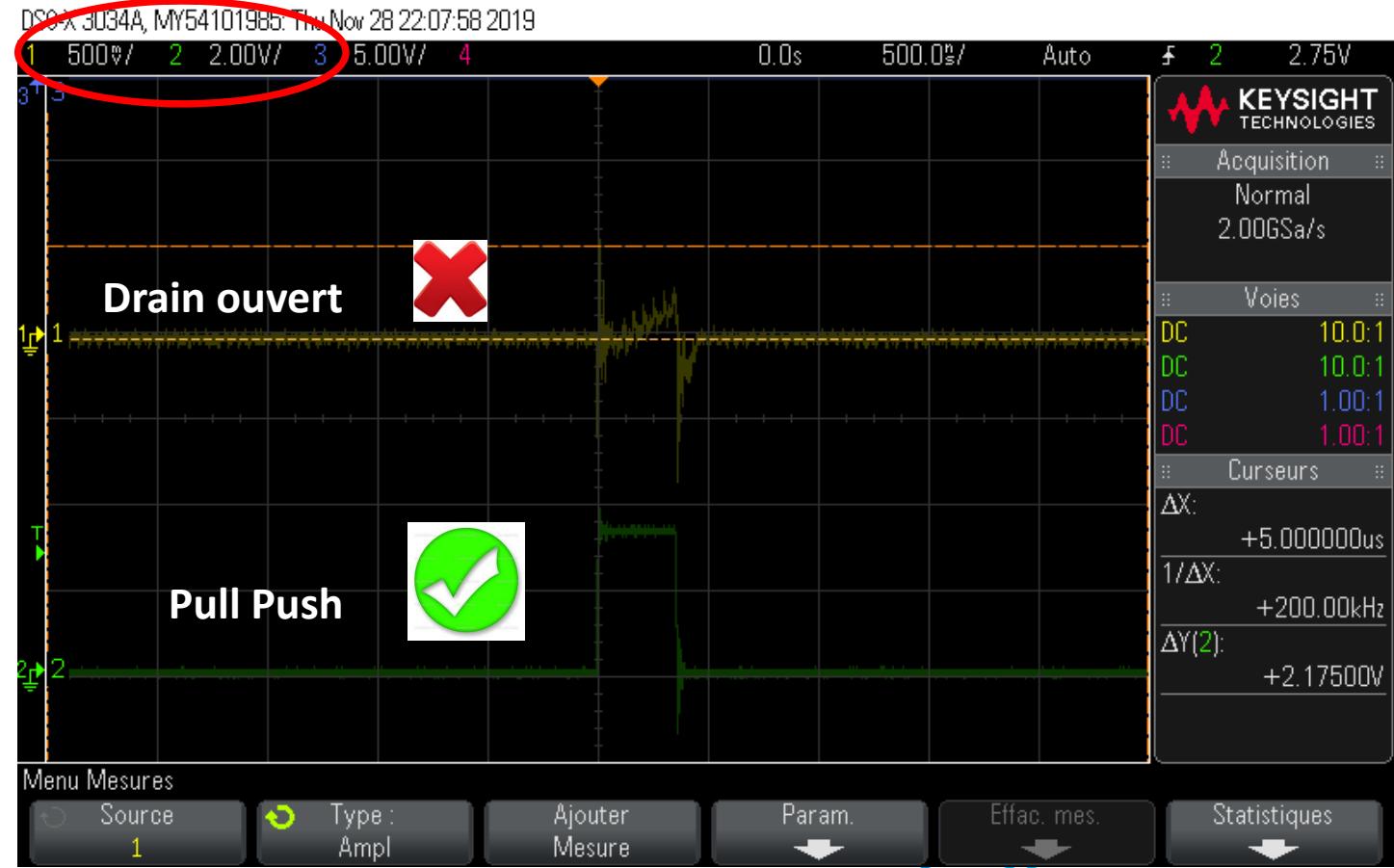
Différence Push-Pull / Drain ouvert

Cas de la génération d'une impulsion 1µs



Cas d'une impulsion positive de durée <1µs – Uniquement une sonde de d'oscilloscope en X10 branchée sur la broche

Drain ouvert:
L'impulsion est à peine visible, son amplitude est trop faible pour être reconnue comme un niveau haut. Il n'y a plus d'impulsion



Production d'une impulsion <1US sur une sortie en drain ouvert

Il faut tenir compte des capacités parasites qui viennent influer sur les temps de montée et de descente . Cette influence sera d'autant plus forte que la résistance de source est élevée

Temps de Montée (2,2 * τ):

$\tau = R_s * C_{parasite}$ Avec **$R_s = 100 \text{ K}\Omega$**

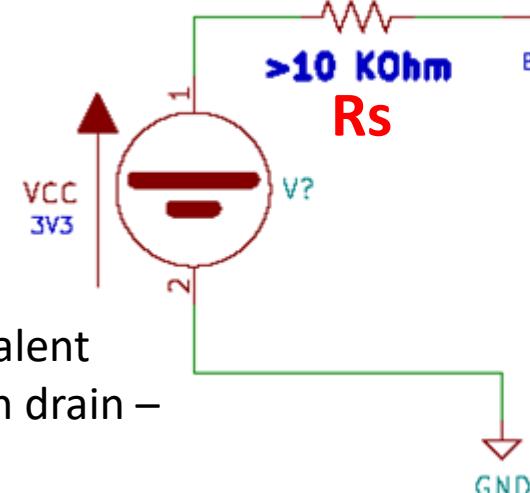
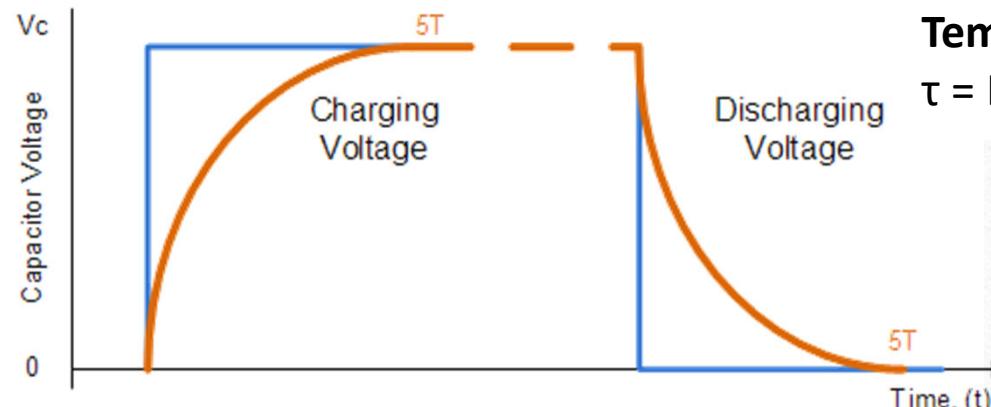


Schéma équivalent
Sortie en Open drain –
sortie à 1

Temps de descente (2,2 * τ):

$\tau = R_s * C_{parasite}$ Avec **$R_s = 100 \Omega$**

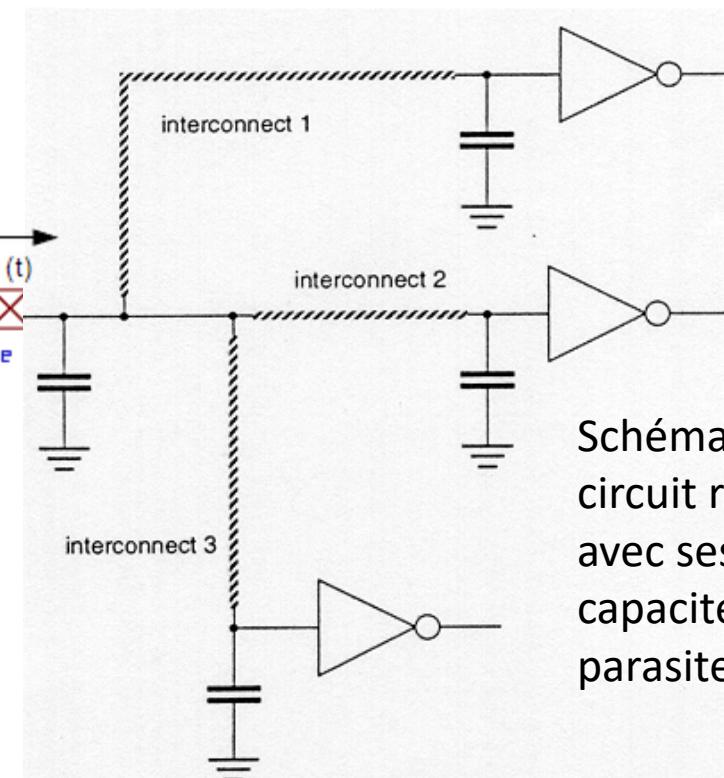


Schéma d'un
circuit réel
avec ses
capacités
parasites....

Evaluation de capacité parasite....

Cas d'une impulsion positive de durée 6,1US – sur chaque broche, n'est connectée qu'une sonde de d'oscilloscope en X10 branchée sur la broche

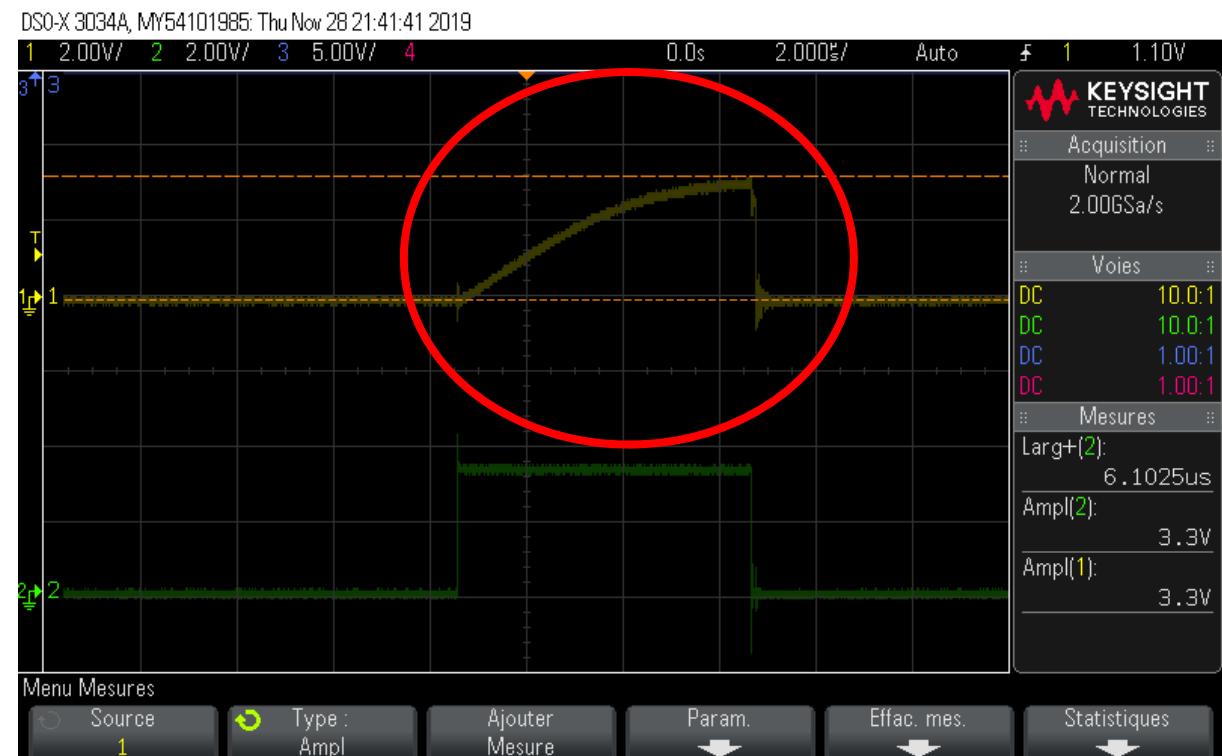
Ce calcul reste approximatif,
mais les ordres de grandeur sont
respectés...

Tmontée estimé à 6US (voir
oscillogramme)

$$T_m = 2,2 \tau := 2,2RC$$

$$C = T_m / 2,2R = 27\text{pF}$$

La valeur est faible, mais suffisante pour qu'en drain ouvert il soit impossible de produire proprement des impulsions inférieures en durée à quelques microsecondes.



Bilan - Configuration des cellules d'entrée-sorties en sortie

Les cellules d'entrée-sorties sur lesquelles sont reliées les broches des ports peuvent être configurées de 2 manières différentes:

- Push-Pull – Ce mode permet d'avoir un forçage fort quel que soit le niveau de sortie (H ou L), c'est-à-dire que l'on peut fournir ou absorber jusqu'à 10mA tout en restant dans des niveaux de tension compatibles avec les niveaux standard des logiques 3V3
- Open drain – Ce mode a un comportement très différent selon le niveau logique produit. Au niveau bas, il se comporte comme un PushPull (Forçage faible au niveau bas). Par contre au niveau haut, il n'est pas en mesure de fournir du courant (ce courant est fourni via la résistance de PullUp de 10Kohm minimum) (Forçage faible au niveau bas)

Quel est donc l'intérêt du mode Open drain?



Port I/O – Drain ouvert – Le « OU » câblé

Quel est l'intérêt de configurer en drain ouvert une broche pour fonctionner en sortie: le OU câblé.

Où comment connecter plusieurs sorties sur le même fil.

On s'appuie sur la dissymétrie de chaque sortie:

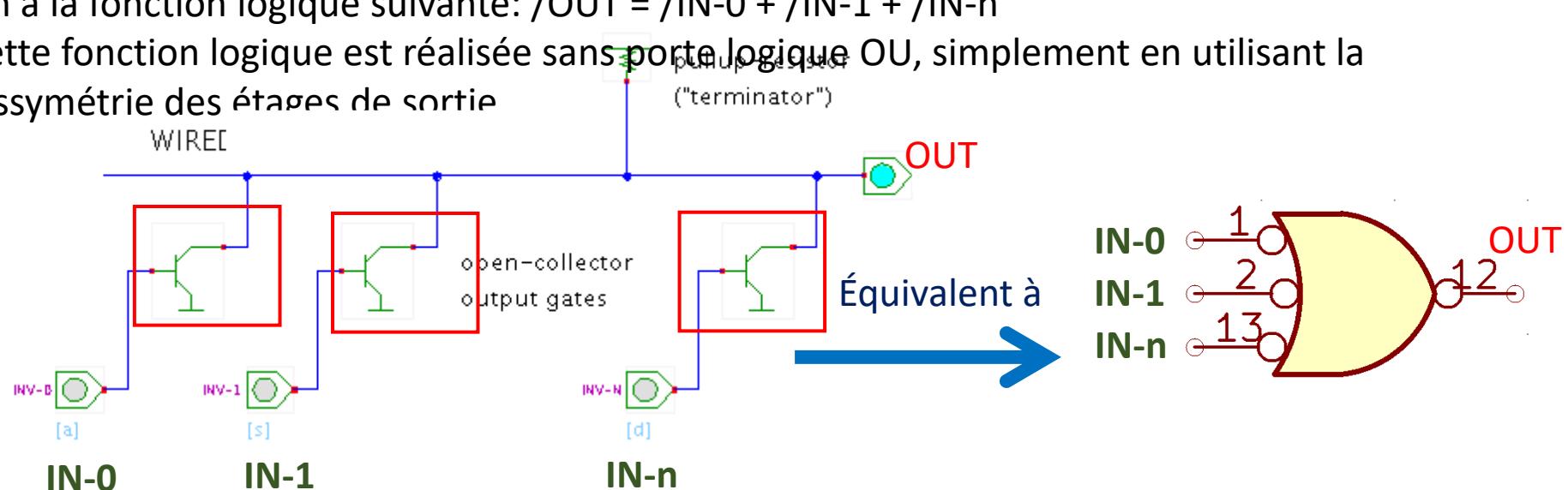
- Un niveau avec un forçage faible (en règle générale le niveau haut)
- Un niveau avec un forçage fort (le niveau bas)

Dans l'exemple ci-dessous:

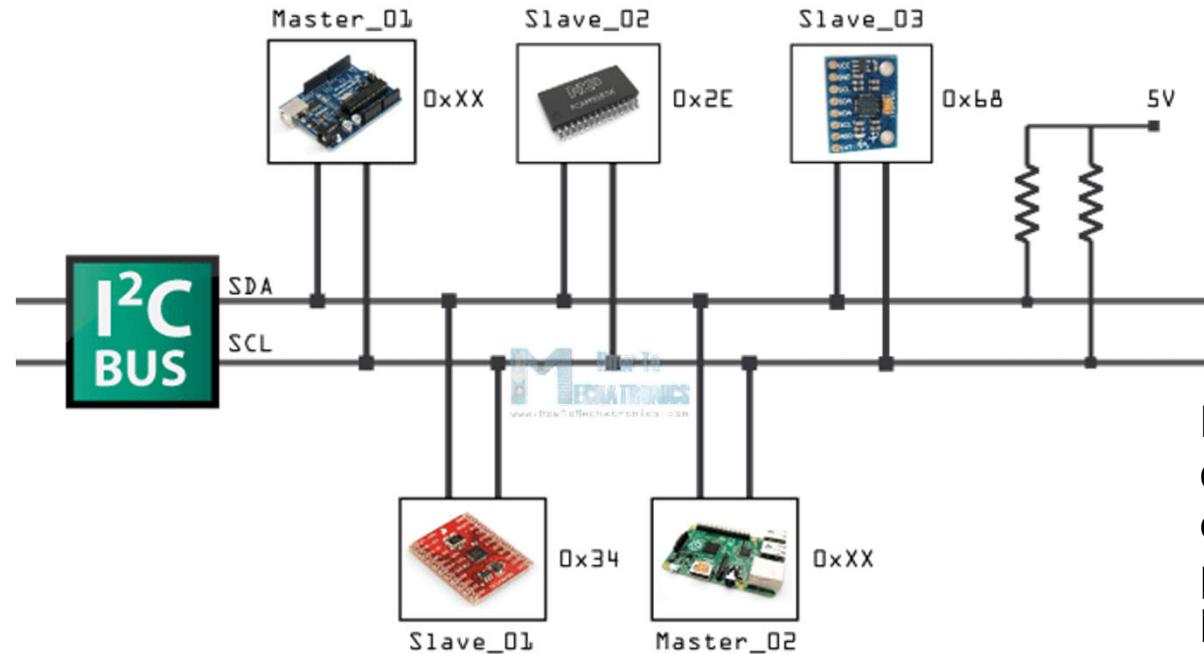
Le signal OUT passe au niveau bas si une des entrées IN-0, IN-1 ou IN-n passe à 0

On a la fonction logique suivante: $/OUT = /IN-0 + /IN-1 + /IN-n$

Cette fonction logique est réalisée sans porte logique OU, simplement en utilisant la dissymétrie des étages de sortie

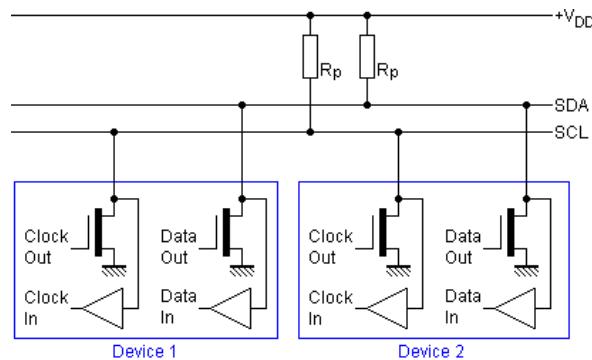


Port I/O – Drain ouvert – liaison I2C



Bus I2C / SMBus

Exemple d'utilisation
d'une sortie configurée
en drain ouvert: OU câblé
pour mise en œuvre d'un
bus série



Registres de Gestion des ports de P0 à P3

Exemple du Port P3

Figure 17.17. P3: Port3 Data Register

R/W	Reset Value 11111111 SFR Address: 0xB0							
P3.7	P3.6	P3.5	P3.4	P3.3	P3.2	P3.1	P3.0	
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	(bit addressable)

Bits7-0: P3.[7:0]: Port3 Output Latch Bits.
(Write - Output appears on I/O pins per XBR0, XBR1, XBR2, and XBR3 Registers)
0: Logic Low Output.
1: Logic High Output (open if corresponding P3MDOUT.n bit = 0).
(Read - Regardless of XBR0, XBR1, XBR2, and XBR3 Register settings).
0: P3.n pin is logic low.
1: P3.n pin is logic high.

Note: P3.[7:0] can be driven by the External Data Memory Interface (as AD[7:0] in Multiplexed mode, or as D[7:0] in Non-multiplexed mode). See **Section “16. EXTERNAL DATA MEMORY INTERFACE AND ON-CHIP XRAM” on page 145** for more information about the External Memory Interface.

Figure 17.18. P3MDOUT: Port3 Output Mode Register

R/W	Reset Value 00000000 SFR Address: 0xA7							
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	

Bits7-0: P3MDOUT.[7:0]: Port3 Output Mode Bits.
0: Port Pin output mode is configured as Open-Drain.
1: Port Pin output mode is configured as Push-Pull.

Note: SDA, SCL, and RX0 (when UART0 is in Mode 0) and RX1 (when UART1 is in Mode 0) are always configured as Open-Drain when they appear on Port pins.

0 pour Open drain
1 pour Push Pull

**Pour chaque Port de P0 à P3,
2 registres:**

Pn (P3) – pour l'utilisation en GPIO – Ecriture et lecture dans le port

PnMDOUT (P3MDOUT) – pour la configuration Push-Pull ou drain ouvert



Port I/O dans le 8051F020 – Configuration PP/OD dans P3 (idem pour P0, P1 et P2) -- Registre P3MDOUT

Ports P0 à P3

P0MDOUT à P3MDOUT

Figure 17.18. P3MDOUT: Port3 Output Mode Register

	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
Bit7 Bit6 Bit5 Bit4 Bit3 Bit2 Bit1 Bit0							00000000

SFR Address: 0xA7

Bits7:0. P3MDOUT.[7:0]: Port3 Output Mode Bits.
0: Port Pin output mode is configured as Open-Drain.
1: Port Pin output mode is configured as Push-Pull.

Note: SDA, SCL, and RX0 (when UART0 is in Mode 0) and RX1 (when UART1 is in Mode 0) are always configured as Open-Drain when they appear on Port pins.

Ces registres ne sont pas accessibles bit à bit.

La configuration d'un bit en particulier est accomplie par des opérations de masquage

Par défaut (après Reset), tous les ports sont configurés en drain Ouvert

Exemple de configuration:

```
P3MDOUT |= ((1<<6) | (1<<0)) ; // P3.6 et P3.0 en Push-Pull - les autres ports sont inchangés  
P3MDOUT &= ~((1<<4) | (1<<0)) ; // P3.4 et P3.0 en Drain ouvert - les autres ports sont inchangés
```

Port I/O dans le 8051F020 – Accès au port en mode GPIO Sortie Registre P3 – « Bit addressable » (idem pour P0, P2 et P3)

Figure 17.17. P3: Port3 Data Register

R/W	Reset Value							
P3.7	P3.6	P3.5	P3.4	P3.3	P3.2	P3.1	P3.0	11111111
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	SFR Address: (bit addressable) 0xB0

Bits7-0: P3.[7:0]: Port3 Output Latch Bits.
(Write - Output appears on I/O pins per XBR0, XBR1, XBR2, and XBR3 Registers)
0: Logic Low Output.
1: Logic High Output (open if corresponding P3MDOUT.n bit = 0).
(Read - Regardless of XBR0, XBR1, XBR2, and XBR3 Register settings).
0: P3.n pin is logic low.
1: P3.n pin is logic high.

Note: P3.[7:0] can be driven by the External Data Memory Interface (as AD[7:0] in Multiplexed mode, or as D[7:0] in Non-multiplexed mode). See **Section “16. EXTERNAL DATA MEMORY INTERFACE AND ON-CHIP XRAM” on page 145** for more information about the External Memory Interface.

```
sbit LED = P3^6;  
sbit BP = P3^0;
```

```
BP = 1; //Sortie P3.0 mise à 1  
LED = 0; //sortie P3.6 mise à zéro  
P3 = 0x55H; //P3.0 à 1, P3.1 à 0, P3.2 à 1....
```

Ports P0
à P3

Registres P0 à P3

Registres de Gestion des ports P0 à P3 – Exemples d'utilisation en mode sortie GPIO



Exemple d'utilisation du port 3

```
sbit OUT1 = P3^7; // Déclaration - OUT1 corresponds au bit 7 du port 3  
P3MDOUT = 0x0F; // Configuration octet: P3.7 à P3.4 en OD; P3.3 à P3.0 en PP  
  
P3MDOUT |= (1<<6); // Configuration bit: P3.6 en Push-Pull  
P3 = 00; // Ecriture octet: Toutes les sorties P3 sont mises à zéro  
OUT1 = 1; // Ecriture bit: mise à 1 du bit 7 du port 3
```

Cas des autres ports:

P0: registres P0 et P0MDOUT

P1: registres P1 et P1MDOUT (+ P1MDIN...)

P2: registres P2 et P2MDOUT

P3: registres P3 et P3MDOUT



**Dans les ports P0 à P3,
Chaque broche est individuellement
configurable en PP ou en OD**



LIVE AND
DISCOVER

Registres de Gestion des ports de P4 à P7



Exemple du Port P4

Figure 17.21. P4: Port4 Data Register

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
P4.7	P4.6	P4.5	P4.4	P4.3	P4.2	P4.1	P4.0	11111111
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	SFR Address: 0x84
Bits7-0: P4.[7:0]: Port4 Output Latch Bits. Write - Output appears on I/O pins. 0: Logic Low Output. 1: Logic High Output (Open-Drain if corresponding P74OUT bit = 0). See Figure 17.20. Read - Returns states of I/O pins. 0: P4.n pin is logic low. 1: P4.n pin is logic high.								
Note: P4.7 (/WR), P4.6 (/RD), and P4.5 (ALE) can be driven by the External Data Memory Interface. See Section “16. EXTERNAL DATA MEMORY INTERFACE AND ON-CHIP XRAM” on page 145 for more information.								

Figure 17.20. P74OUT: Ports 7 - 4 Output Mode Register

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
P7H	P7L	P6H	P6L	P5H	P5L	P4H	P4L	00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	SFR Address: 0xB5
Bit7: P7H: Port7 Output Mode High Nibble Bit. 0: P7.[7:4] configured as Open-Drain. 1: P7.[7:4] configured as Push-Pull.								
Bit6: P7L: Port7 Output Mode Low Nibble Bit. 0: P7.[3:0] configured as Open-Drain. 1: P7.[3:0] configured as Push-Pull.								
Bit5: P6H: Port6 Output Mode High Nibble Bit. 0: P6.[7:4] configured as Open-Drain. 1: P6.[7:4] configured as Push-Pull.								
Bit4: P6L: Port6 Output Mode Low Nibble Bit. 0: P6.[3:0] configured as Open-Drain. 1: P6.[3:0] configured as Push-Pull.								
Bit3: P5H: Port5 Output Mode High Nibble Bit. 0: P5.[7:4] configured as Open-Drain. 1: P5.[7:4] configured as Push-Pull.								
Bit2: P5L: Port5 Output Mode Low Nibble Bit. 0: P5.[3:0] configured as Open-Drain.								

Ports P4 à P7:

Les bits de ces ports ne sont pas configurables individuellement.
La configuration du mode se fait par demi-octet (nibble) dans le registre P74OUT

Exemple:

Si le bit 7 de P74OUT est mis à 1,
alors les broches P7.4 à P7.7 sont configurées en PushPull

**Pour chaque Port de P4 à P7,
2 registres:**

**Pn (P4) – pour
l'utilisation en
mode GPIO–**

Ecriture et lecture
dans le port

P74OUT – 1
registre commun à
P4 à P7 pour
configurer en PP
ou OD

Port I/O dans le 8051F020 – Config PP/OD de P4 (idem pour P5, P6 et P7) dans le Registre P74OUT

Exemple de configuration du Port P4

Figure 17.20. P74OUT: Ports 7 - 4 Output Mode Register

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
P7H	P7L	P6H	P6L	P5H	P5L	P4H	P4L	00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	SFR Address: 0xB5
Bit7:	P7H: Port7 Output Mode High Nibble Bit. 0: P7.[7:4] configured as Open-Drain. 1: P7.[7:4] configured as Push-Pull.							
Bit6:	P7L: Port7 Output Mode Low Nibble Bit. 0: P7.[3:0] configured as Open-Drain. 1: P7.[3:0] configured as Push-Pull.							
Bit5:	P6H: Port6 Output Mode High Nibble Bit. 0: P6.[7:4] configured as Open-Drain. 1: P6.[7:4] configured as Push-Pull.							
Bit4:	P6L: Port6 Output Mode Low Nibble Bit. 0: P6.[3:0] configured as Open-Drain. 1: P6.[3:0] configured as Push-Pull.							
Bit3:	P5H: Port5 Output Mode High Nibble Bit. 0: P5.[7:4] configured as Open-Drain. 1: P5.[7:4] configured as Push-Pull.							
Bit2:	P5L: Port5 Output Mode Low Nibble Bit. 0: P5.[3:0] configured as Open-Drain. 1: P5.[3:0] configured as Push-Pull.							
Bit1:	P4H: Port4 Output Mode High Nibble Bit. 0: P4.[7:4] configured as Open-Drain. 1: P4.[7:4] configured as Push-Pull.							
Bit0:	P4L: Port4 Output Mode Low Nibble Bit. 0: P4.[3:0] configured as Open-Drain. 1: P4.[3:0] configured as Push-Pull.							

Il n'est pas possible de configurer en PP ou OD chaque broche séparément. On ne peut configurer que par demi-octet.

Ainsi si l'on souhaite configurer P4.0 en PP, on va automatiquement configurer P4.1, P4.2, P4.3 en PP

P74OUT |= (1<<0) // Cfg de P4.0 à P4.3 en PP

Port I/O dans le 8051F020 – Accès au port en mode GPIO sortie du port - Exemple du registre P4 (idem dans P5, P6 et P7)

Ports P4 à P7

Registres P4 à P7

Ces registres ne sont pas accessibles bit à bit, il est donc impossible de faire des déclarations sbit sur les bits de ces ports. Il faut donc procéder par masquage

Figure 17.21. P4: Port4 Data Register

R/W	Reset Value							
P4.7	P4.6	P4.5	P4.4	P4.3	P4.2	P4.1	P4.0	11111111
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	SFR Address: 0x84

Bits7-0: P4.[7:0]: Port4 Output Latch Bits.
Write - Output appears on I/O pins.
0: Logic Low Output.
1: Logic High Output (Open-Drain if corresponding P74OUT bit = 0). See Figure 17.20.

P4 non adressable bit à bit!!

En écriture

//Mise à 1 de P4.2

P4 |= (1<<4); // Ecriture très explicite
ou P4 |= P4 | 0x04; // Ecriture moins explicite

//Mise à 0 de P4.4

P4 &= ~(1<<4);
ou P4 &= 0xEF;

ALE) can be driven by the External Data Memory Interface.
MEMORY INTERFACE AND ON-CHIP XRAM” on

Registres de Gestion des ports P4 à P7 – Exemples d'utilisation en mode sortie GPIO



Exemple d'utilisation du port 4

```
P74OUT |= (1<<0) // P4.0 à P4.3 en PP  
P4 |= (1<<6) ; // P4.6 mis à 1  
P4 &= ~ (1<<5) ; // P4.5 mis à 0  
If ( ((P4 & (1<<4)) == 0) ... // Test P4.4 à zéro?
```

Cas des autres ports:

P4: registres P0 et P74OUT

P5: registres P1 et P74OUT

P6: registres P2 et P74OUT

P7: registres P3 et P74OUT



LIVE AND
DISCOVER



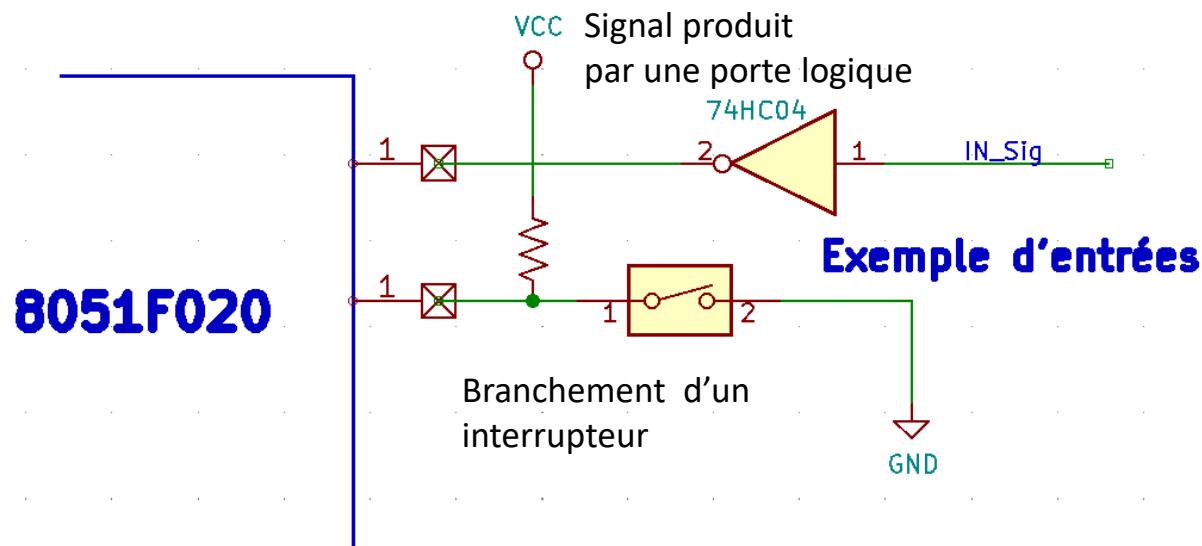
Bases des systèmes embarqués
A02 – Configuration Ports d'entrées-sorties sur 8051F020

Acte 2: Utilisation des Ports d'entrées-sorties en GPIO – Configuration en entrée

Utilisation d'une broche I/O en entrée

Rappel: On sait configurer l'étage de sortie des blocs d'entrée sortie selon 2 modes possibles (PP et OD), on connaît leur différences et on sait sur quels registres agir pour configurer ces étages de sortie et produire des signaux logiques sur ces broches.

Désormais la question est la suivante: comment faire pour utiliser une broche d'I/O en entrée?

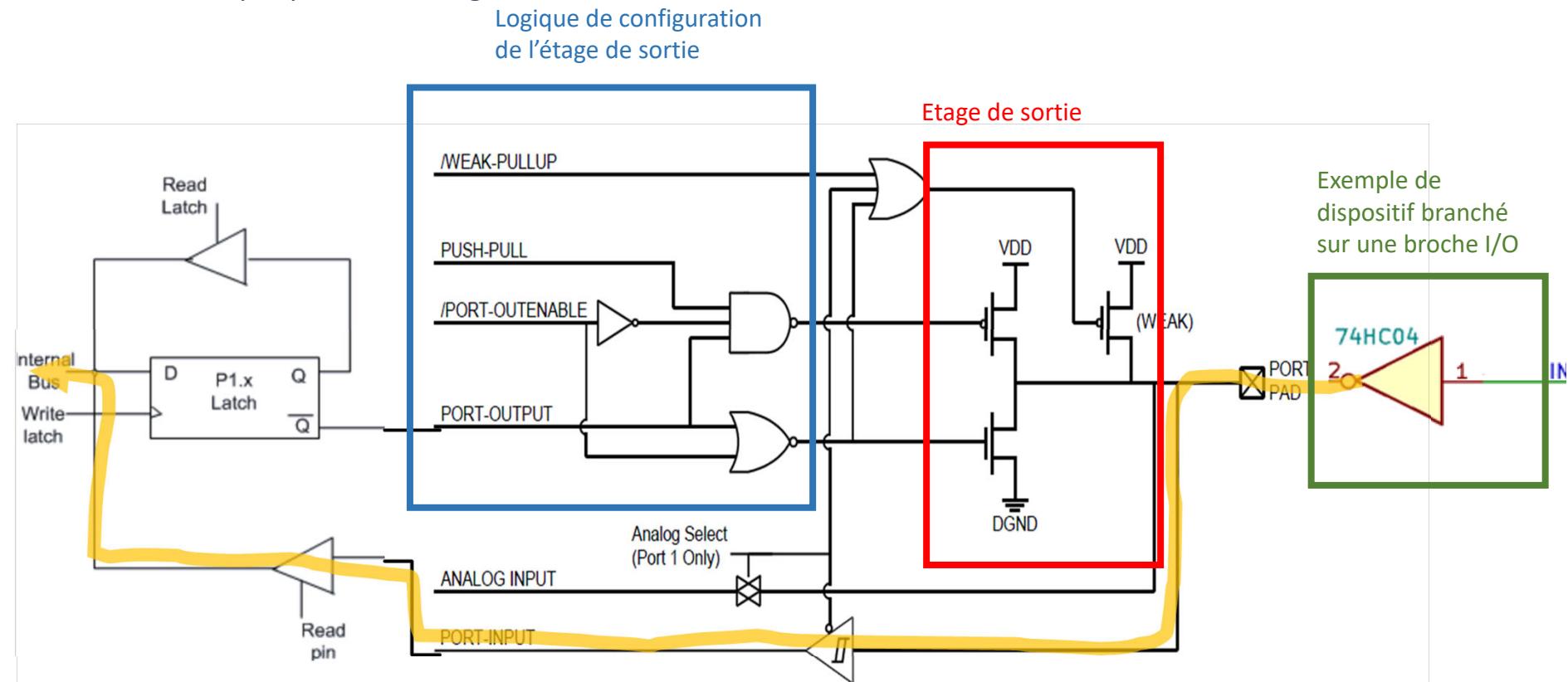


Analyse du fonctionnement d'une broche I/O utilisée en entrée



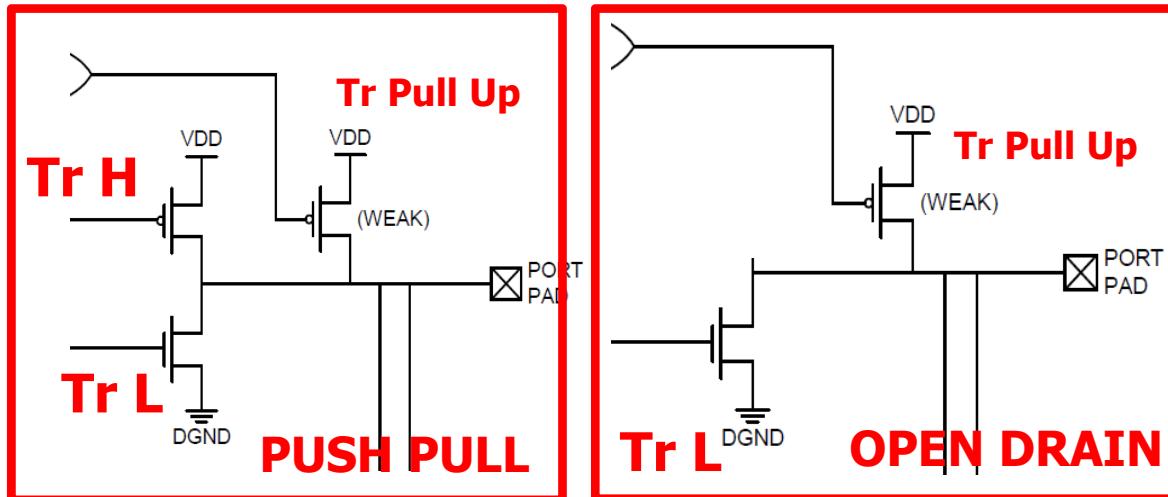
Lors d'un opération de lecture sur un port (par exemple en C, « `var = P0 ;` ») l'information suit le chemin indiqué ci-dessous par le trait orange.

Mais pour que cette lecture soit correcte, c'est-à-dire pour que l'on lise bien une valeur qui correspond au signal logique produit par l'inverseur 74HC04 (dans l'exemple du schéma ci-dessous), il faut que l'étage de sortie de la broche ne vienne pas perturber le signal entrant.



Configuration de l'étage de sortie d'une broche I/O pour fonctionner en entrée.

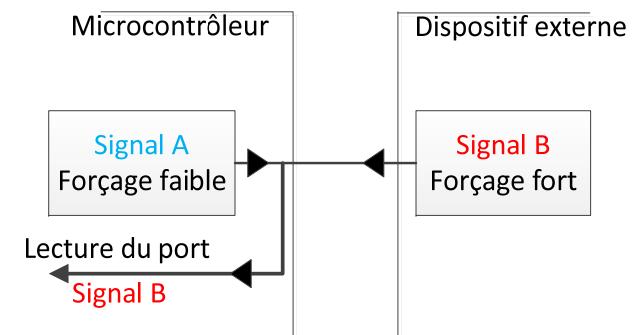
Rappel: L'étage de sortie ne peut être configuré qu'en Push Pull ou en Drain ouvert (Open Drain)



Pour que cet étage de sortie n'influe pas sur l'état du signal externe connecté sur la broche, il faut faire en sorte que les transistors TrH et TrL soient ouverts, pour que cet étage de sortie soit en mode forçage faible.

Ainsi, à la condition bien sûr, que le signal externe impose un forçage fort (c'est le cas pour une porte logique standard, pour un interrupteur, etc...), l'information qui va être lue, correspond bien à l'information produite par le signal externe.

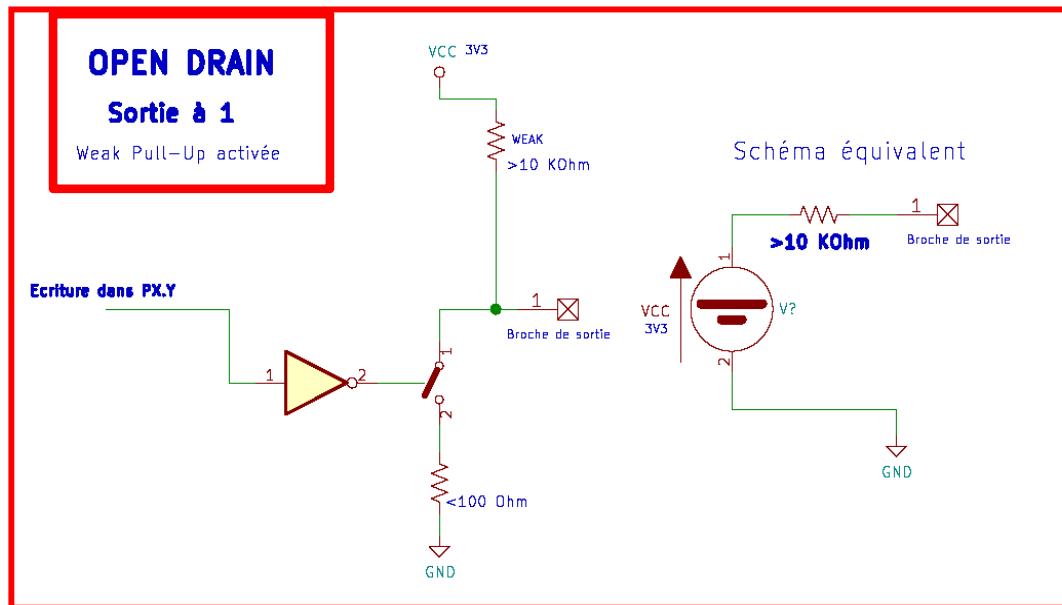
Pour fonctionner en entrée, la seule configuration du l'étage de sortie (IO Cell) qui permet ceci est le mode Drain Ouvert avec la sortie mise à 1



Le signal B (forçage fort) « l'emporte »
sur le signal A (forçage faible)

Le signal B IMPOSE son niveau

Port IO en entrée – Schéma équivalent



Pour fonctionner en entrée (bis!):

1. Mode Drain Ouvert
2. Sortie placée à 1

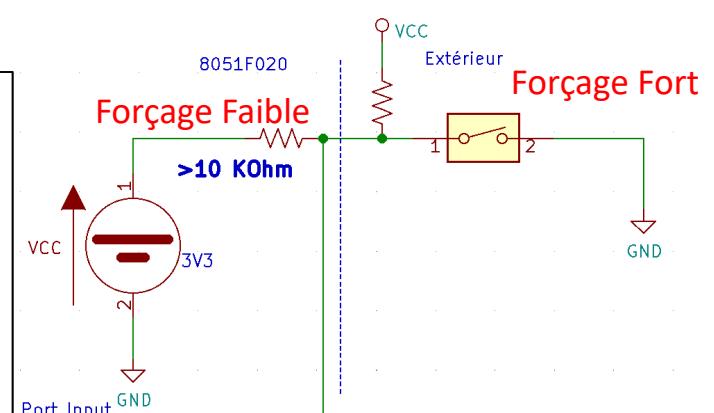
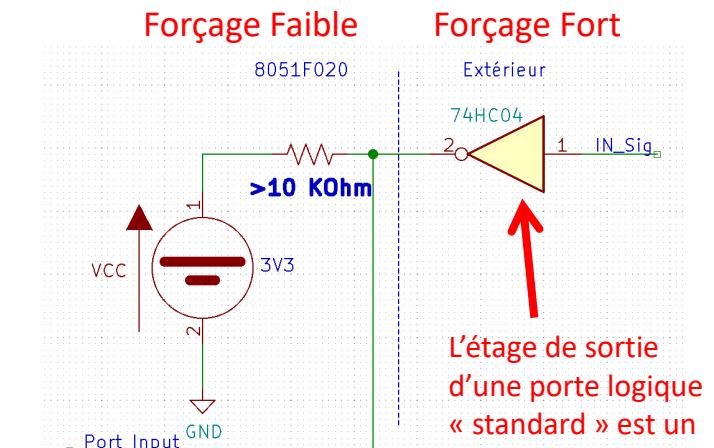


Code C pour configurer la broche P2.0 afin que celle-ci puisse fonctionner en entrée

```
sbit P2_0 = P2^0;
P0MOUT &= ~0x01; // 1 Cfg OD
P2_0 = 1; // 2 sortie à 1
```

Exemple Utilisation:

```
If (P2_0 == 0) { ... }
```

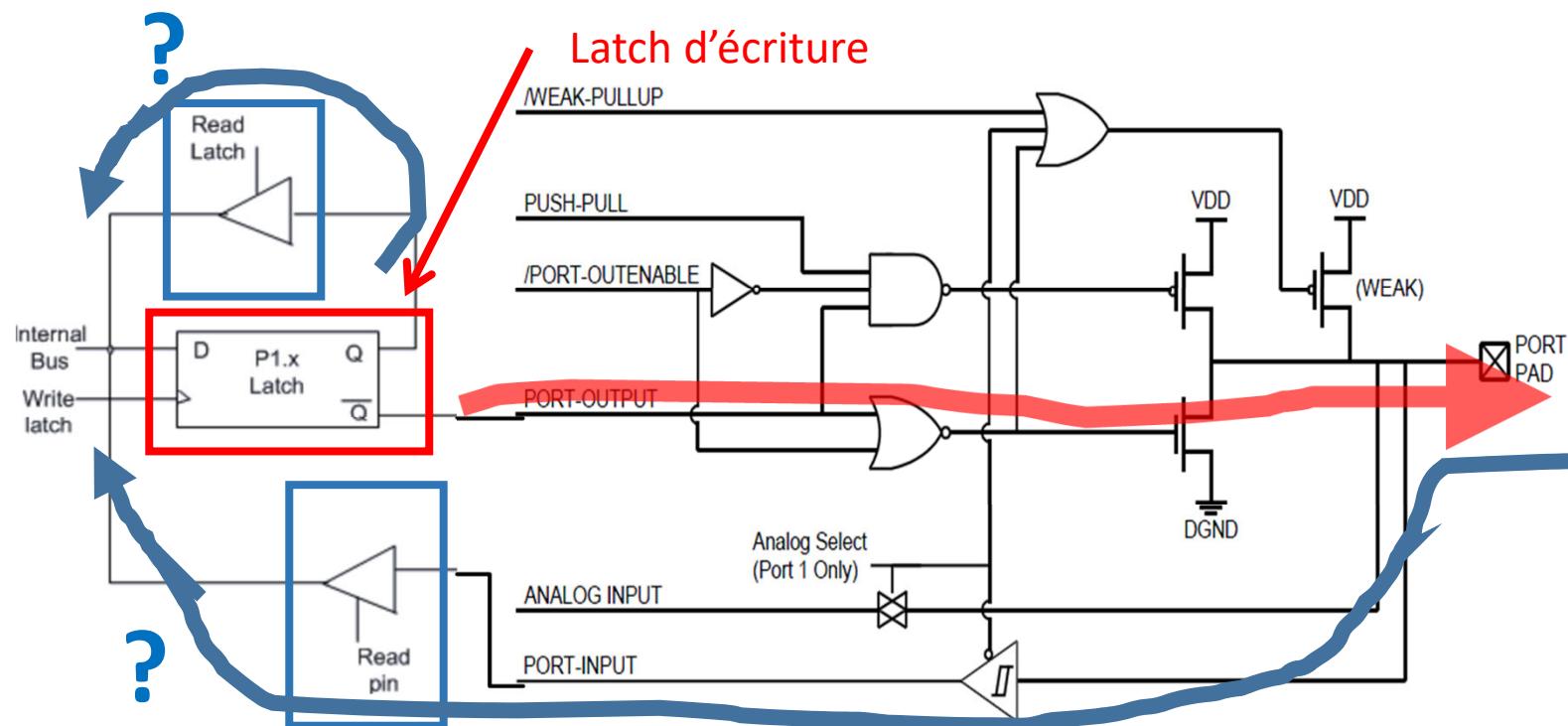


Accès lecture – écriture dans un port

Ecriture dans le port: En C, la ligne de code « P1 = 0x22; » se traduit au niveau du processeur par une écriture dans le circuit latch du port. Le signal est ensuite propagé vers les broches des ports via les étages de sortie.

Lecture dans le port: on constate sur le schéma ci-dessous, qu'en lecture, on dispose de 2 chemins: le chemin qui vient de la broche via la porte trois états pilotée par le signal « read Pin » et un chemin qui permet de relire le contenu du latch du port via la porte trois états pilotée par le signal Read Latch.

Y a-t-il un risque de confusion lors du codage ?



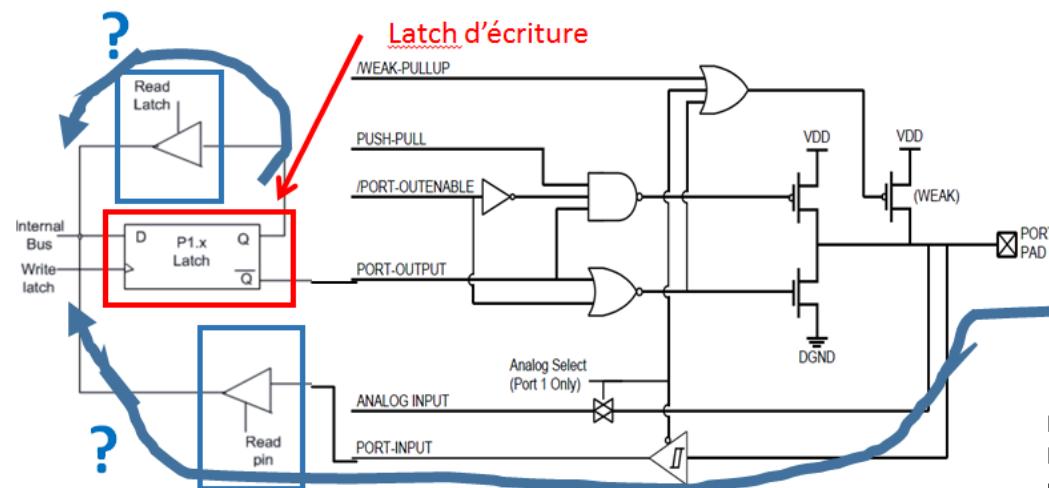
Risque de confusion Latch/Port Pin en lecture

Questions posées: est-il possible de choisir entre une lecture du latch ou une lecture du niveau logique sur la broche? Et, si ce n'est pas le cas, comment savoir si dans mon code, je fais une lecture « Latch » ou une lecture « Pin »?

Tout est une question d'assembleur! C'est la nature des instructions assemblleur qui va déterminer le type d'accès lecture.

Toutes les instructions de type « **Read-Modify-Write** » quand l'opérande de destination est un port ou un bit de port vont faire une lecture sur le latch, toutes les autres vont faire une lecture sur la pin.

Ces instructions sont:



- ANL (logical AND, e.g., ANL P1,A)
 - ORL (logical OR, e.g., ORL P2,A)
 - XRL (logical EX-OR, e.g., XRL P3,A)
 - JBC (jump if bit = 1 and clear bit, e.g., JBC P1.1,LABEL)
 - CPL (complement bit, e.g., CPL P3.0)
 - INC (increment, e.g., INC P2)
 - DEC (decrement, e.g., DEC P2)
 - DJNZ (decrement and jump if not zero, e.g., DJNZ P3,LABEL)
 - MOV PX.Y,C (move carry bit to bit Y of Port X)
 - CLR PX.Y (clear bit Y of Port X)
 - SET PX.Y (set bit Y of Port X)
- Source: <http://www.8051projects.info/modify.php>

Pour les 3 dernières instructions, ce n'est pas évident, mais
L'instruction va lire les 8 bits du port, modifie le bit concerné, puis va réécrire les 8 bits

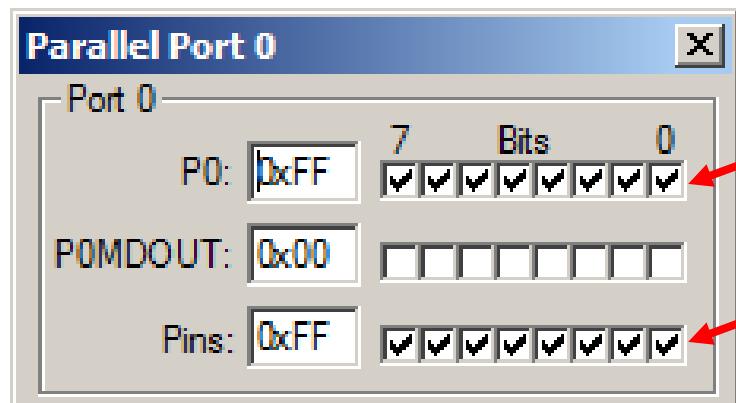
Risque de confusion Latch/Port Pin en lecture dans le code C

En C, sauf cas exceptionnel, le compilateur en analysant le fichier source est tout à fait en mesure de produire le code assembleur adéquat , pour éviter toute confusion.

Exemple:

```
    40:           if (P0_0 == 0) var_bit = 0;
C:0x000C      208002   JB      P0_0(0x80.0),C:0011 Lecture PIN de P0.0
C:0x000F      C200     CLR     var_bit(0x20.0)
    41:           P0_1 = P0_1;
C:0x0011      A281     MOV     C,P0_1(0x80.1) Lecture PIN de P0.1
C:0x0013      9281     MOV     P0_1(0x80.1),C
    42:           P1 = P1 | 0x80;
C:0x0015      439080   ORL     P1(0x90),#P0(0x80)  Lecture Latch de P1
                                            Instruction RMW
```

Interprétation de la fenêtre d'affichage
d'un port durant le débogage:



Lecture Latch

Equivalent de la lecture PIN

Port I/O – Port en entrée ou port en sortie



Résumé: Choix des modes Push Pull et Open Drain

- Port utilisé en **sortie**:
 - Le mode Open Drain (Drain ouvert) permet de réaliser un « ou » câblé. Ce mode est utilisé dans des cas particuliers
 - Le mode Push Pull permet d'avoir une *sortance* identique pour les niveaux haut et bas (faible impédance). **C'est le mode à utiliser par défaut.**
- Port utilisé en **entrée**: 2 opérations de configuration **obligatoires**
 - Port à configurer en Drain ouvert (Transistor « Tr H » ouvert)
 - Ecrire un « 1 » dans le port (Transistor « Tr L » ouvert)





Exemple de configuration et d'utilisation de Port en sortie

Ports P0 à P3

P1.0 configuré en sortie (sauf cas particulier, configuration en Push Pull)

Déclaration

```
Sbit OUT1 = P1^0;
```

Configuration

```
P1MDOUT |= (1<<0); // Cfg PP sur P1.0
```

Utilisation

```
OUT1 = 0; // OUT1 (P1.0) mis à zéro
```

Ports P4 à P7

P4.2 configuré en sortie (sauf cas particulier configuration en Push Pull)

Déclaration

Pas de déclaration sbit possible sur P4.2

Configuration

```
P74OUT |= (1<<0); // Cfg PP sur P4.0 à P4.3
```

Utilisation

```
P4 &= ~ (1<<2); // (P4.2) mis à zéro
```





Exemple de configuration et d'utilisation de Port en entrée

Ports P0 à P3

P1.0 configuré en entrée (configuration obligatoire en drain ouvert (OD))

Déclaration

```
sbit IN1 = P1^0;
```

Configuration

```
P1MDOUT &= ~(1<<0); // Cfg OD sur P1.0
```

```
IN1 = 1; // Ecriture de '1' dans le port P1.0 pour pouvoir fonctionner en entrée
```

Utilisation

```
if (IN1 == 0) ... // Lecture de IN1 (P1.0) et teste si égal à 0
```

Ports P4 à P7

P4.2 configuré en entrée (configuration obligatoire en drain ouvert (OD))

Déclaration

Pas de déclaration sbit possible sur P4.2

Configuration

```
P74OUT &= ~(1<<0); // Cfg OD sur P4.0 à P4.3
```

```
P4 |= (1<<2); // Ecriture de '1' dans le port P4.2 pour pouvoir fonctionner en entrée
```

Utilisation

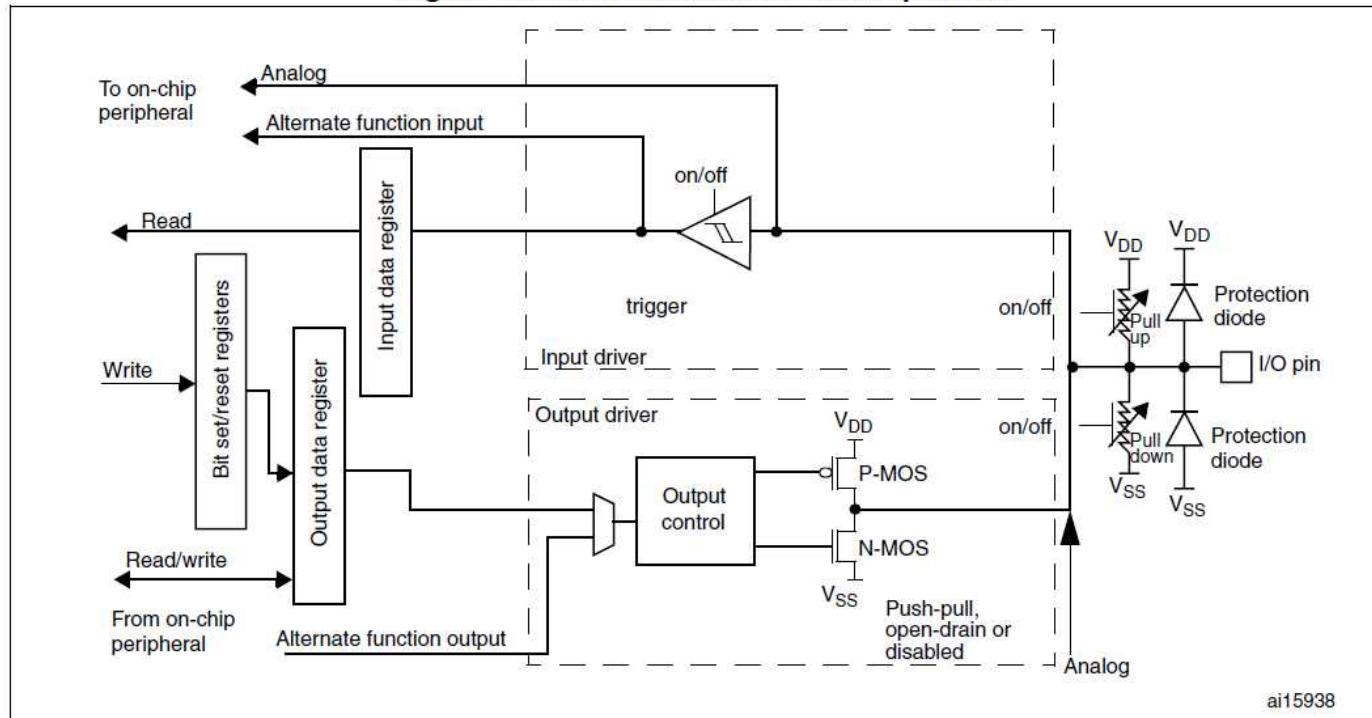
```
if ( (P4 & (1<<4)) == 0) ... // Lecture de P4.2) et teste si égal à 0
```



Autres horizons: Les GPIO sur STM32F3 - Cortex M4 Core



Figure 40. Basic structure of an I/O port bit



Une structure d'IO Cell proche de celle du 8051F020...

On retrouve la possibilité de configuration Push-Pull/Drain Ouvert

Mais en plus, il est possible de configurer la valeur de la résistance de pull up

Et on dispose d'une résistance de pull Down configurable.

Pause GPIO...



Soit la configuration:

XBR2 = 0x40 ;

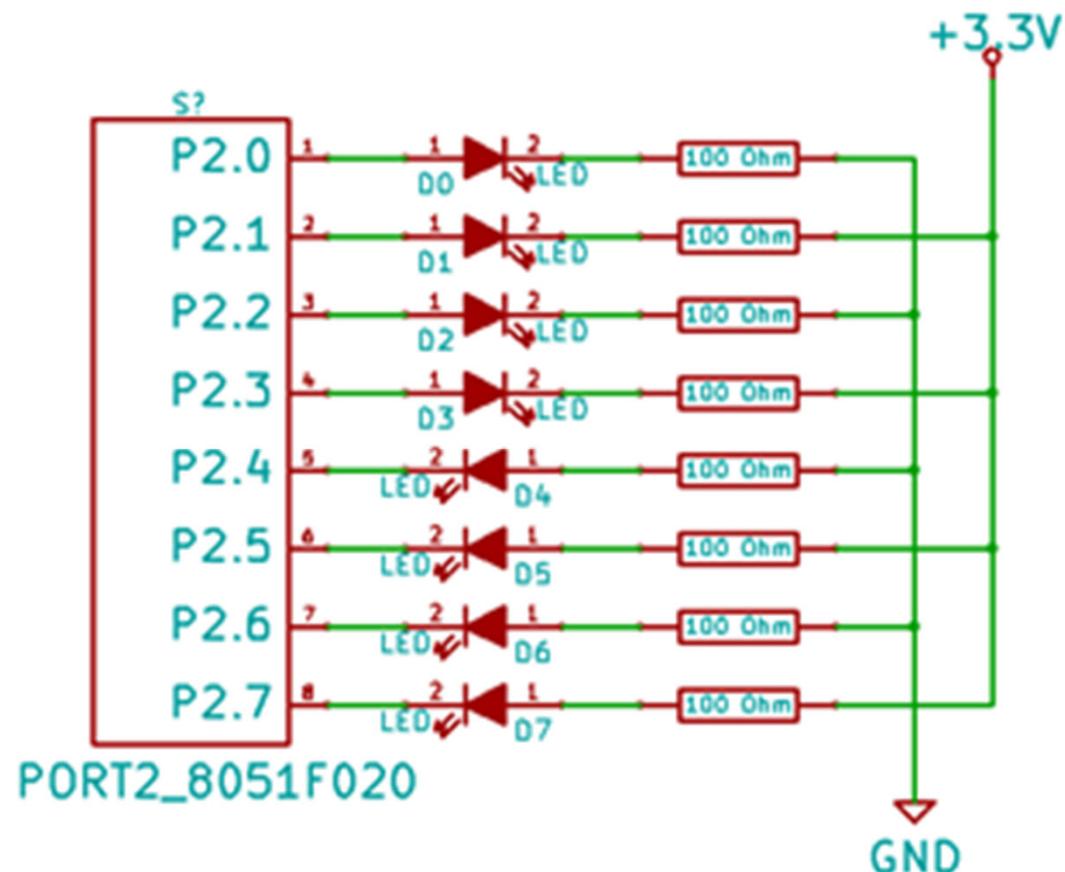
P2MDOUT = 0x33 ;

P2 = 0x2D ;

Déterminer l'état de chaque LED (allumée – éteinte)

Rappel: VDled = 1,8V

I Led minimum = 2mA



Pause...



Soit la configuration:

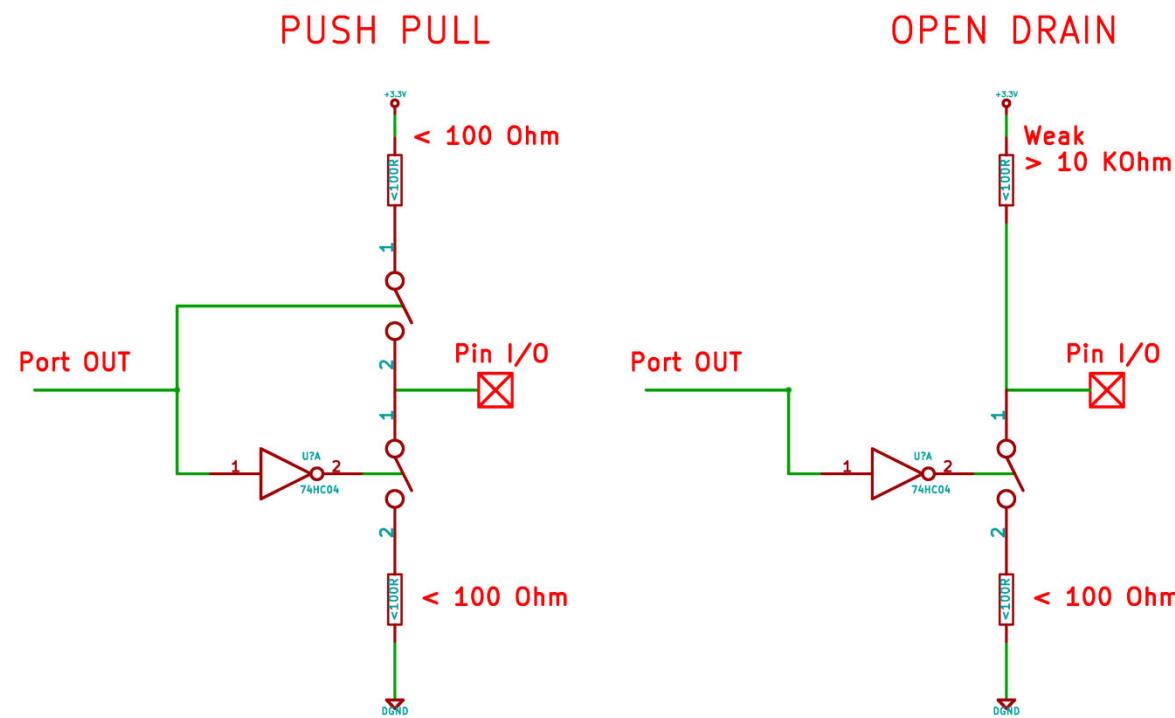
XBR2 = 0x40;

P2MDOUT = 0x33;

P2 = 0x2D;

Rappel: VD led = 1,8V

I Led minimum =
2mA



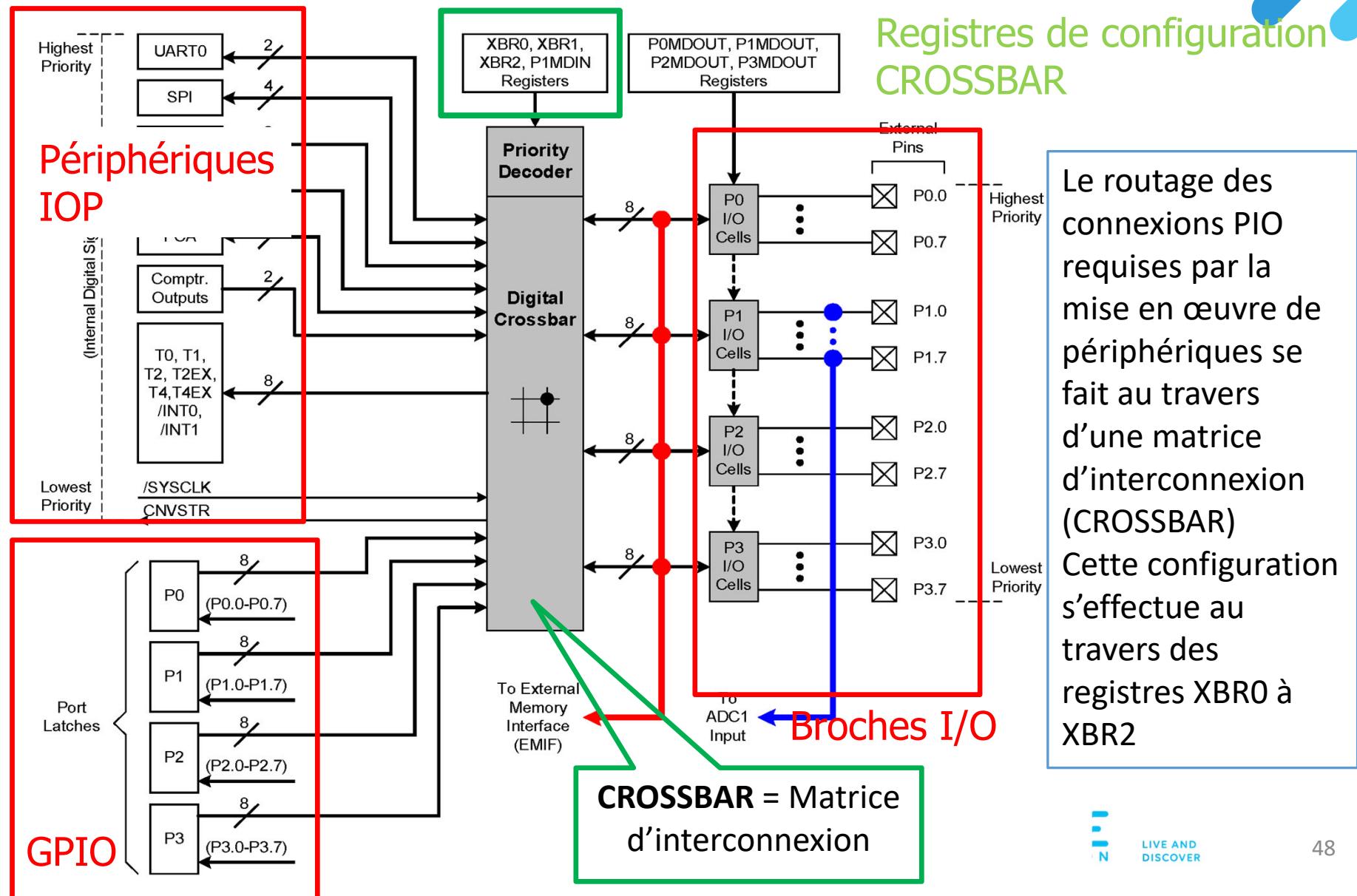


Bases des systèmes embarqués
A02 – Configuration Ports d'entrées-sorties sur 8051F020

Acte 3: Utilisation des Ports d'entrées-sorties en PIO (Peripheral Input _ Output) Configuration du CROSSBAR

RAPPEL: Structure globale des Port I/O P1 à P3 dans le 8051F020

Figure 17.2. Lower Port I/O Functional Block Diagram



Etude du Registre XBR0

Figure 17.7. XBR0: Port I/O Crossbar Register 0

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
CP0E	ECI0E		PCA0ME		UART0EN	SPI0EN	SMB0EN	00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	SFR Address: 0xE1
<p>Bit7: CP0E: Comparator 0 Output Enable Bit. 0: CP0 unavailable at Port pin. 1: CP0 routed to Port pin.</p> <p>Bit6: ECI0E: PCA0 External Counter Input Enable Bit. 0: PCA0 External Counter Input unavailable at Port pin. 1: PCA0 External Counter Input (ECI0) routed to Port pin.</p> <p>Bits5-3: PCA0ME: PCA0 Module I/O Enable Bits. 000: All PCA0 I/O unavailable at Port pins. 001: CEX0 routed to Port pin. 010: CEX0, CEX1 routed to 2 Port pins. 011: CEX0, CEX1, and CEX2 routed to 3 Port pins. 100: CEX0, CEX1, CEX2, and CEX3 routed to 4 Port pins. 101: CEX0, CEX1, CEX2, CEX3, and CEX4 routed to 5 Port pins. 110: RESERVED 111: RESERVED</p> <p>Bit2: UART0EN: UART0 I/O Enable Bit. 0: UART0 I/O unavailable at Port pins. 1: UART0 TX routed to P0.0, and RX routed to P0.1.</p> <p>Bit1: SPI0EN: SPI0 Bus I/O Enable Bit. 0: SPI0 I/O unavailable at Port pins. 1: SPI0 SCK, MISO, MOSI, and NSS routed to 4 Port pins.</p> <p>Bit0: SMB0EN: SMBus0 Bus I/O Enable Bit. 0: SMBus0 I/O unavailable at Port pins. 1: SMBus0 SDA and SCL routed to 2 Port pins.</p>								

Pour chaque périphérique, on ne spécifie pas un numéro de broche.
On indique seulement si on réserve des broches pour ce périphérique.
Le nombre de broches réservées dépend de la fonctionnalité du périphérique

Cas particulier pour UART0:
Affectation sur 2 broches bien identifiées P0.0 et P0.1

Registre XBR1

Figure 17.8. XBR1: Port I/O Crossbar Register 1

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
SYSCKE	T2EXE	T2E	INT1E	T1E	INT0E	T0E	CP1E	00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	SFR Address: 0xE2

Bit7: SYSCKE: /SYSCLK Output Enable Bit.
0: /SYSCLK unavailable at Port pin.
1: /SYSCLK routed to Port pin.

Bit6: T2EXE: T2EX Input Enable Bit.
0: T2EX unavailable at Port pin.
1: T2EX routed to Port pin.

Bit5: T2E: T2 Input Enable Bit.
0: T2 unavailable at Port pin.
1: T2 routed to Port pin.

Bit4: INT1E: /INT1 Input Enable Bit.
0: /INT1 unavailable at Port pin.
1: /INT1 routed to Port pin.

Bit3: T1E: T1 Input Enable Bit.
0: T1 unavailable at Port pin.
1: T1 routed to Port pin.

Bit2: INT0E: /INT0 Input Enable Bit.
0: /INT0 unavailable at Port pin.
1: /INT0 routed to Port pin.

Bit1: T0E: T0 Input Enable Bit.
0: T0 unavailable at Port pin.
1: T0 routed to Port pin.

Bit0: CP1E: CP1 Output Enable Bit.
0: CP1 unavailable at Port pin.
1: CP1 routed to Port pin.

Idem que pour XBR0, pour chaque périphérique, on ne spécifie pas un numéro de broche.
On indique seulement si on réserve des broches pour ce périphérique

Registre XBR2

Figure 17.9. XBR2: Port I/O Crossbar Register 2

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value					
WEAKPUD	XBARE	-	T4EXE	T4E	UART1E	EMIFLE	CNVSTE	00000000					
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	SFR Address: 0xE3					
Bit7:	WEAKPUD: Weak Pull-Up Disable Bit. 0: Weak pull-ups globally enabled. 1: Weak pull-ups globally disabled.	Validation globale des résistances de Pull-UP sur les ports		Pull-UP validées après le Reset									
Bit6:	XBARE: Crossbar Enable Bit. 0: Crossbar disabled. All pins on Ports 0, 1, 2, and 3, are forced to Input mode. 1: Crossbar enabled.	Validation globale du crossbar. Indispensable pour utiliser les ports en GPIO et IOP		Le Crossbar est dévalide après le Reset!!									
Bit5:	UNUSED. Read = 0, Write = don't care.												
Bit4:	T4EXE: T4EX Input Enable Bit. 0: T4EX unavailable at Port pin. 1: T4EX routed to Port pin.												
Bit3:	T4E: T4 Input Enable Bit. 0: T4 unavailable at Port pin. 1: T4 routed to Port pin.												
Bit2:	UART1E: UART1 I/O Enable Bit. 0: UART1 I/O unavailable at Port pins. 1: UART1 TX and RX routed to 2 Port pins.												
Bit1:	EMIFLE: External Memory Interface Low-Port Enable Bit. 0: P0.7, P0.6, and P0.5 functions are determined by the Crossbar or the Port latches. 1: If EMI0CF.4 = '0' (External Memory Interface is in Multiplexed mode) P0.7 (/WR), P0.6 (/RD), and P0.5 (ALE) are 'skipped' by the Crossbar and their output states are determined by the Port latches and the External Memory Interface.	Idem que pour XBR0, pour chaque périphérique, on ne spécifie pas un numéro de broche. On indique seulement si on réserve des broches pour ce périphérique											
Bit0:	If EMI0CF.4 = '1' (External Memory Interface is in Non-multiplexed mode) P0.7 (/WR) and P0.6 (/RD) are 'skipped' by the Crossbar and their output states are determined by the Port latches and the External Memory Interface.												
Bit0:	CNVSTE: External Convert Start Input Enable Bit. 0: CNVSTR unavailable at Port pin. 1: CNVSTR routed to Port pin.												



Cas particulier du Port 1 - Registre P1MDIN

Chaque broche du port P1, en plus de pouvoir fonctionner en GPIO et en PIO, peut aussi fonctionner en entrée analogique pour le convertisseur analogique-numérique ADC1. Un registre supplémentaire P1MDIN permet de gérer cette configuration

Figure 17.13. P1MDIN: Port1 Input Mode Register

R/W	Reset Value 0xBD	SRK Address. 0xBD							
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0		
								11111111	

Bits7-0: P1MDIN.[7:0]: Port 1 Input Mode Bits.

0: Port Pin is configured in Analog Input mode. The digital input path is disabled (a read from the Port bit will always return '0'). The weak pull-up on the pin is disabled.

1: Port Pin is configured in Digital Input mode. A read from the Port bit will return the logic level at the Pin. The state of the weak pull-up is determined by the WEAKPUD bit (XBR2.7, see Figure 17.9).

Lire plutôt « digital Input-Output » mode

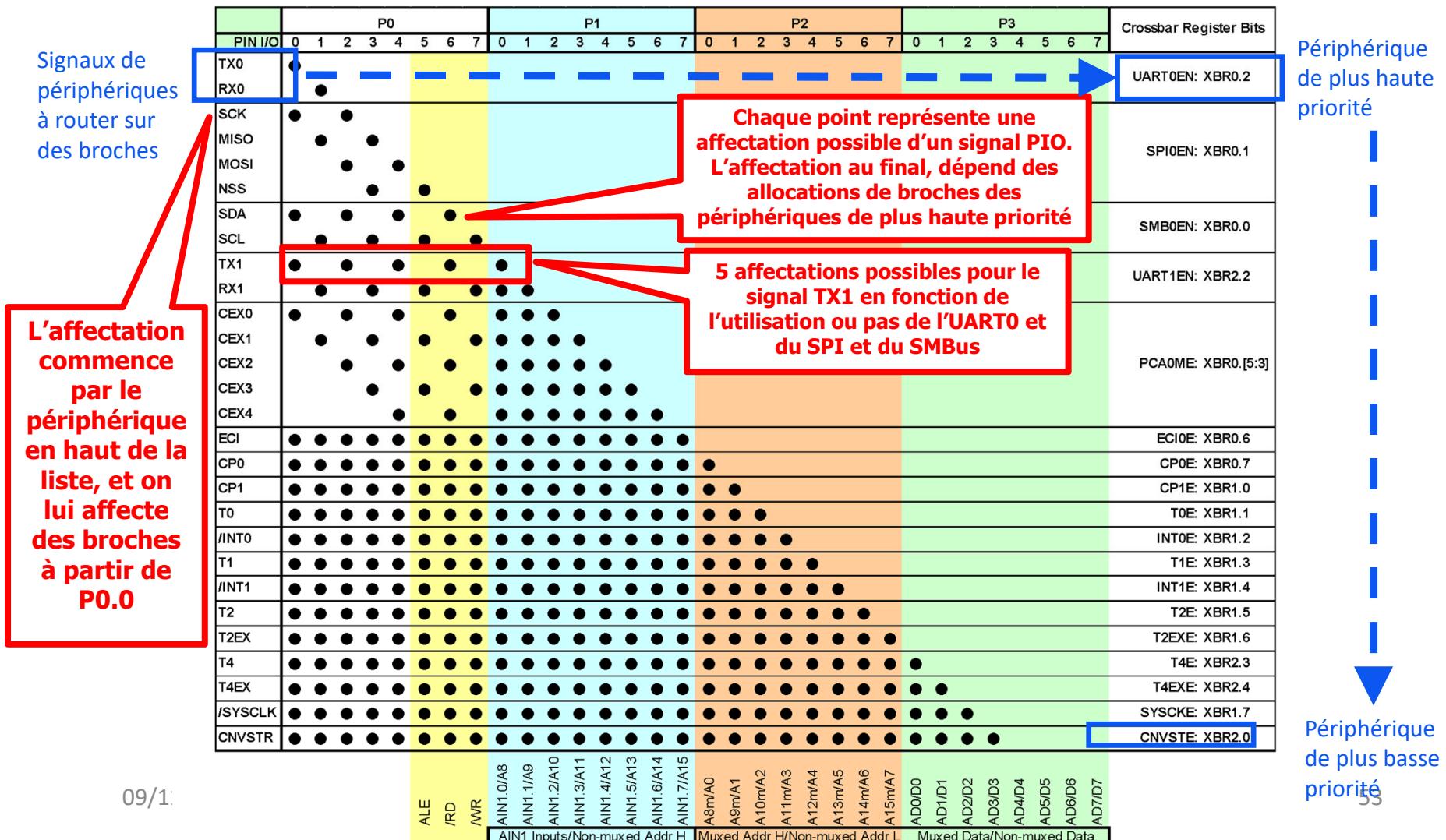
Configuration particulière des broches de P1:

- Par défaut (après Reset), configurées en Entrées-Sorties digitales (GPIO ou I/O périphérique)
- Elles peuvent être configurées en entrée analogique pour l'ADC1 (convertisseur analogique-numérique) par action sur P1MDIN

Règle d'affectation des broches PIO

L'affectation des broches aux périphériques obéit à des règles précises:

- On affecte d'abord des broches au périphérique utilisé de plus haute priorité (voir colonne à droite pour les priorités)
- On commence par affecter la broche P0.0, puis P0.1, etc.... (Attention, tenir compte des réservations éventuelles sur le port P1 de broches comme entrées analogiques pour l'ADC1)

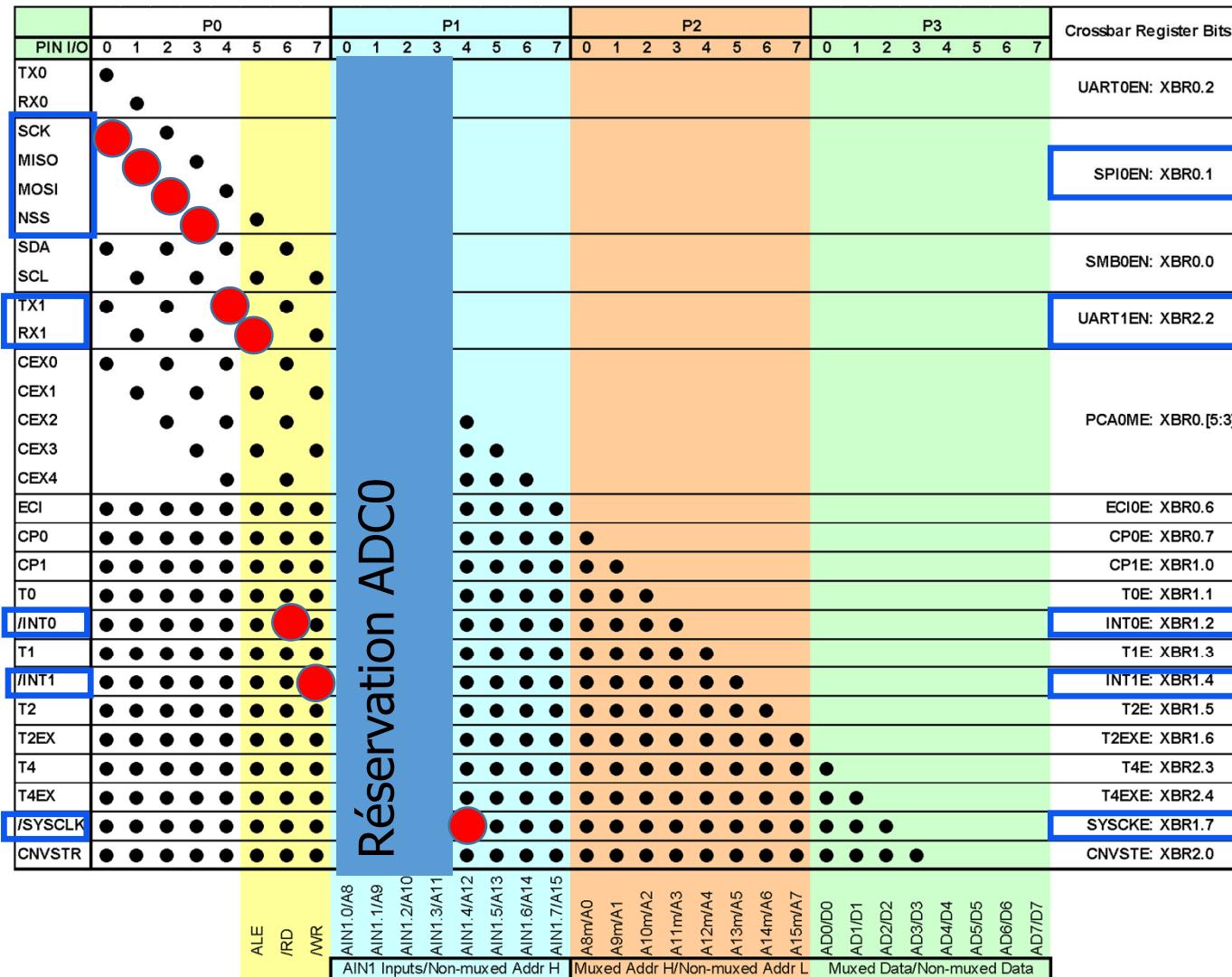


Exemple d'affectation des broches sur le Crossbar



On souhaite utiliser les périphériques: SPI, UART1, INT0, INT1 et SYSCLK. Par ailleurs on veut réservé les broches P1.0 à P1.3 comme entrées analogiques pour l'ADC1.

Sur quelles broches vont être affectés les signaux des périphériques utilisés?



Résultat:

P0.0: SCK

P0.1: MISO

P0.2: MOSI

P0.3: NSS

P0.4: TX1

P0.5: RX1

P0.6: /INT0

P0.7: /INT1

P1.0: Réservé ADC

P1.1: Réservé ADC

P1.2: Réservé ADC

P1.3: Réservé ADC

P1.4: /SYSCLK

P1.5 et tous les autres:
Dispo GPIO

Code à produire:

P1MDIN = 0xF0 ;

XBR0 = 0x02 ;

XBR1 = 0x94 ;

XBR2 = 0x44 ;



Pause Crossbar...

Soit la ligne de configuration: **XBR0 = 0x93**;

On considère par ailleurs que le Crossbar est validé (XBR2.6 =1)

Déterminer l'affectation des broches pour le port P0. Peut-on la déterminer pour toutes les broches de P0?

Pause crossbar

Figure 17.7. XBR0: Port I/O Crossbar Register 0

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
CP0E Bit7	ECI0E Bit6		PCA0ME Bit5		UART0EN Bit3	SPI0EN Bit2	SMB0EN Bit1	00000000
								SFR Address: 0xE1

Bit7: CP0E: Comparator 0 Output Enable Bit.
0: CP0 unavailable at Port pin.
1: CP0 routed to Port pin.

Bit6: ECI0E: PCA0 External Counter Input Enable Bit.
0: PCA0 External Counter Input unavailable at Port pin.
1: PCA0 External Counter Input (ECI0) routed to Port pin.

Bits5-3: PCA0ME: PCA0 Module I/O Enable Bits.
000: All PCA0 I/O unavailable at Port pins.
001: CEX0 routed to Port pin.
010: CEX0, CEX1 routed to 2 Port pins.
011: CEX0, CEX1, and CEX2 routed to 3 Port pins.
100: CEX0, CEX1, CEX2, and CEX3 routed to 4 Port pins.
101: CEX0, CEX1, CEX2, CEX3, and CEX4 routed to 5 Port pins.
110: RESERVED
111: RESERVED

Bit2: UART0EN: UART0 I/O Enable Bit.
0: UART0 I/O unavailable at Port pins.
1: UART0 TX routed to P0.0, and RX routed to P0.1.

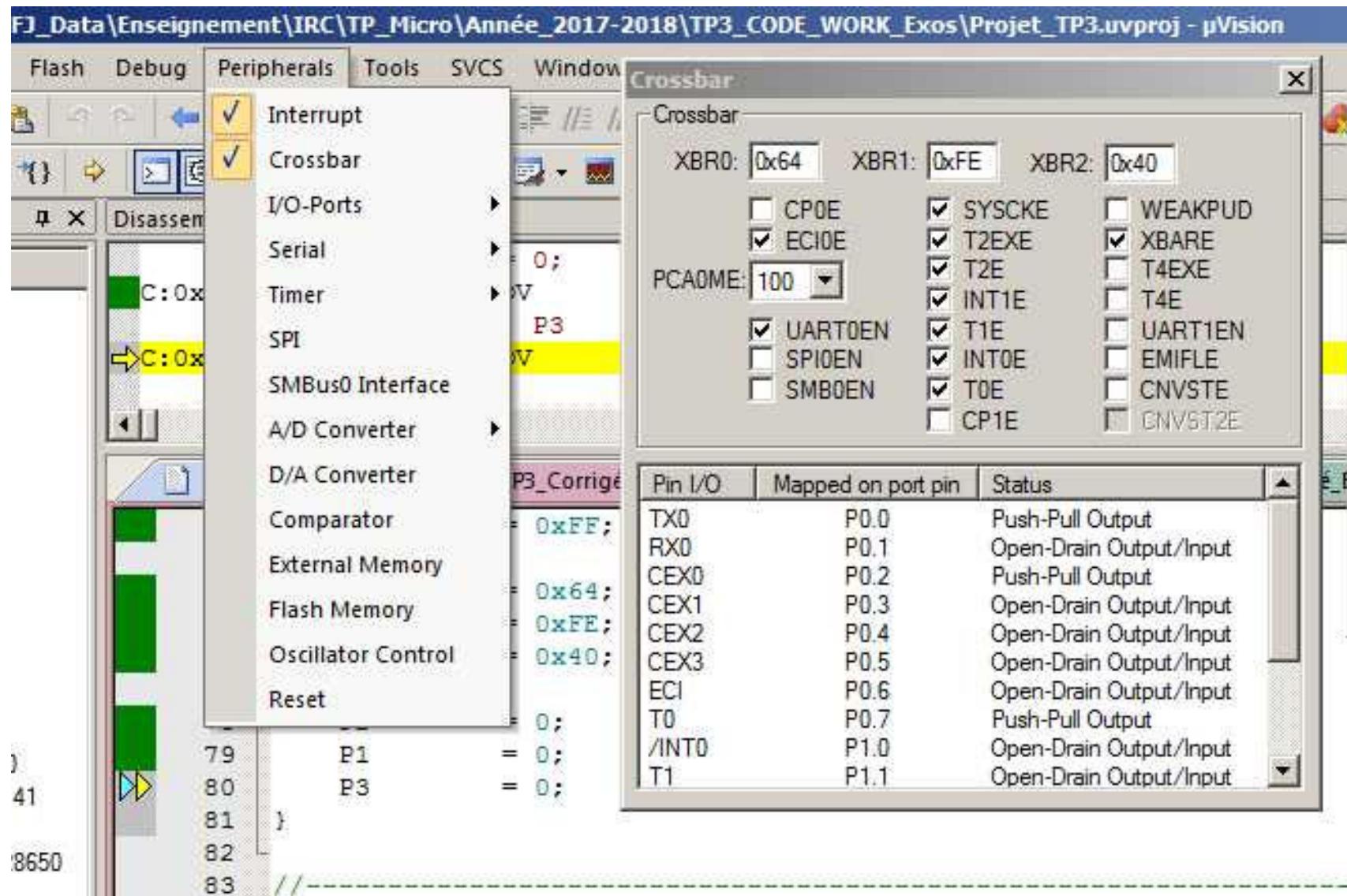
Bit1: SPI0EN: SPI0 Bus I/O Enable Bit.
0: SPI0 I/O unavailable at Port pins.
1: SPI0 SCK, MISO, MOSI, and NSS routed to 4 Port pins.

Bit0: SMB0EN: SMBus0 Bus I/O Enable Bit.
0: SMBus0 I/O unavailable at Port pins.
1: SMBus0 SDA and SCL routed to 2 Port pins.

Pause Crossbar

PIN I/O	P0								P1								P2								P3								Crossbar Register Bits
	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	
TX0	●																													UART0EN: XBR0.2			
RX0		●																															
SCK	●		●																												SPI0EN: XBR0.1		
MISO		●		●																													
MOSI		●		●																													
NSS			●																														
SDA	●		●		●																									SMB0EN: XBR0.0			
SCL		●		●					●																								
TX1	●		●		●				●																					UART1EN: XBR2.2			
RX1		●		●					●																								
CEX0	●		●		●				●																								
CEX1		●		●					●																								
CEX2		●		●					●																					PCA0ME: XBR0.[5:3]			
CEX3		●		●					●																								
CEX4		●		●					●																								
ECI	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	ECI0E: XBR0.6				
CP0	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	CP0E: XBR0.7				
CP1	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	CP1E: XBR1.0				
T0	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	TOE: XBR1.1				
/INT0	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	INT0E: XBR1.2				
T1	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	T1E: XBR1.3				
/INT1	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	INT1E: XBR1.4				
T2	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	T2E: XBR1.5				
T2EX	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	T2EXE: XBR1.6				
T4	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	T4E: XBR2.3				
T4EX	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	T4EXE: XBR2.4				
/SYSCLK	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	SYSCKE: XBR1.7				
CNVSTR	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	CNVSTE: XBR2.0				
	ALE	/RD	/WR	AIN1.0/A8	AIN1.1/A9	AIN1.2/A10	AIN1.3/A11	AIN1.4/A12	AIN1.5/A13	AIN1.6/A14	AIN1.7/A15	A8m/A0	A9m/A1	A10m/A2	A11m/A3	A12m/A4	A13m/A5	A14m/A6	A15m/A7	AD0/D0	AD1/D1	AD2/D2	AD3/D3	AD4/D4	AD5/D5	AD6/D6	AD7/D7						
	AIN1 Inputs/Non-muxed Addr H								Muxed Addr H/Non-muxed Addr L								Muxed Data/Non-muxed Data																

Vérifier la configuration du crossbar durant le débogage...





Configuration des ports dans la partie Configuration globale – Fonction Init_Device()

Objectif de la fonction Port_IO_Init:

- Configurer la matrice d'interconnexion (Crossbar) au travers des registres d'aiguillage (XBRn) pour affecter des broches aux périphériques que l'on va utiliser.
- Configurer les broches utilisées par les périphériques et par les GPIO
 - En entrée?
 - En sortie Push Pull?
 - En sortie Drain ouvert?

Remarque: Pour éviter tout problème, on se gardera bien de modifier la configuration des registres XBR en dehors de cette fonction de configuration Port_IO_Init, appelée une seule fois, après le démarrage du processeur

Exemple de Configuration complète de Port I/O

```
void Port_IO_Init()
{
    P1MDIN      = 0xFA; // Réservation de broches de P1 comme entrées analogiques
    P0MDOUT     = 0x15; // Configurations Push/Pull Drain ouvert
    P1MDOUT     = 0x00;
    P2MDOUT     = 0x00;
    P3MDOUT     = 0x00;
    P74OUT      = 0x00;
    P0          = 0x7F; //Initialisation de l'état des broches en sorties, et configuration en entrées de certaines broches
    ...
    P7          = 0xFF;
    XBR0        = 0x86; // Configuration du Crossbar – Réservation de broches pour les périphériques
    XBR1        = 0x80;
    XBR2        = 0x04;
    XBR2        |= 0x40; // Précaution: Validation du Crossbar – broches activées à la fin de la configuration
}
```



Squelette de Code Init_Device()

Rappel « Initialisation globale »: les initialisations qui vont impacter la totalité de l'application développée et qui vont concerner tous les périphériques sont considérés comme des initialisations globales

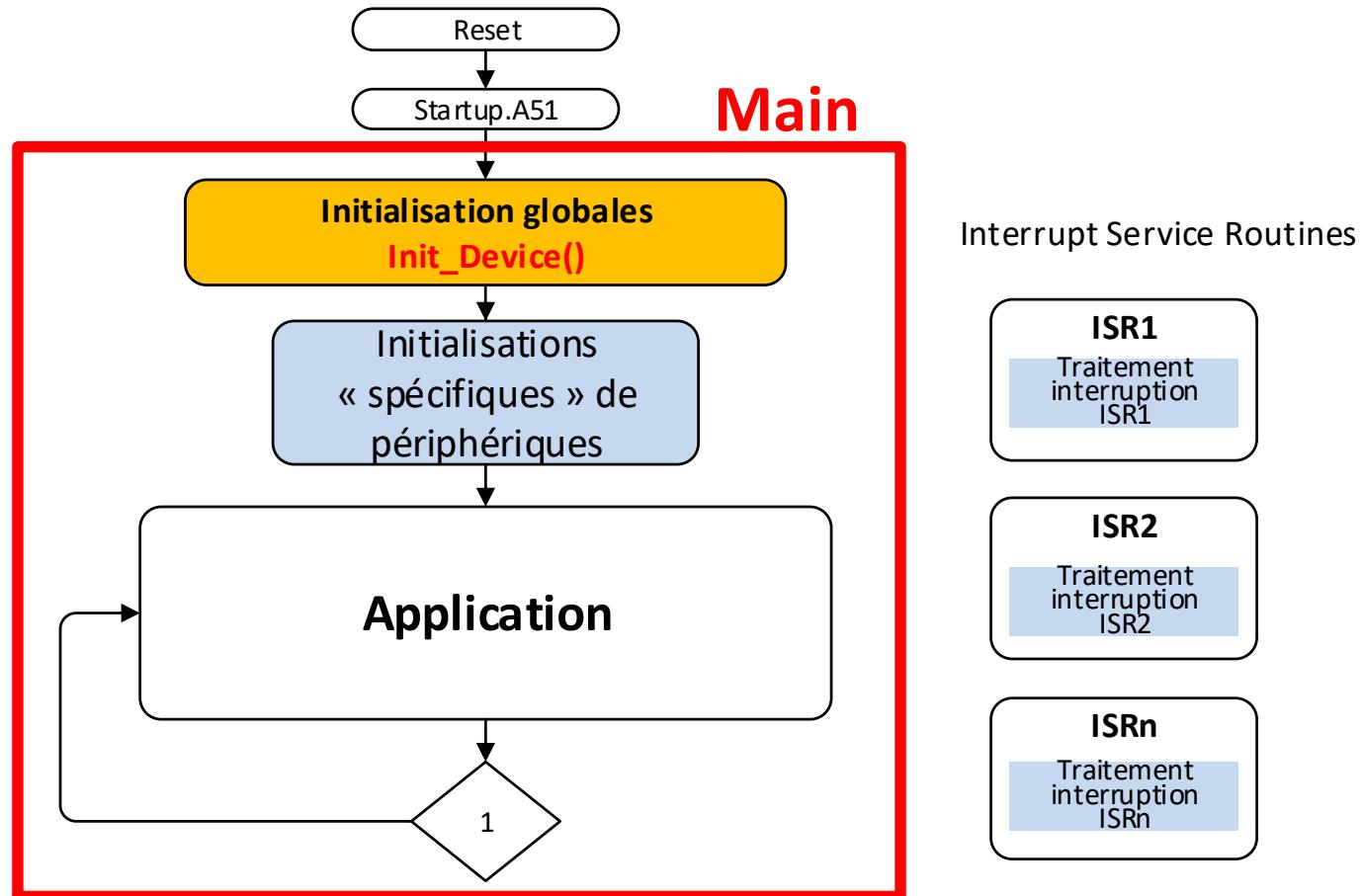
Convention utilisée pour les codes d'initialisation globale du 8051F020:

La fonction `Init_Device` contient toutes les configurations globales

Squelette d'Init_device:

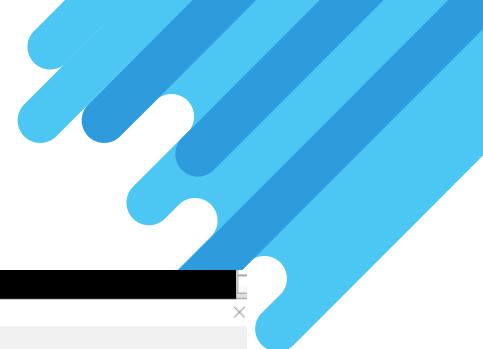
```
void Init_Device(void)
{
    Reset_Sources_Init(); //Sources Reset et watchdog
    EMI_Init(); // Configuration de la mémoire - Timings
    Port_IO_Init(); //affectation des broches aux périphériques internes et configuration de ces
                    broches
    Oscillator_Init(); // Configuration de l'oscillateur interne
}
```

Positionnement de Init_Device dans l'algorithme global d'une application



Exemple de Configuration de Port I/O

Le « Wizard » de configuration pour vous servir...



Pin I/O	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7
UART0 TX0																																
UART0 RX0																																
UART0 SCK																																
SPI0 MISO																																
SPI0 MOSI																																
SPI0 NSS																																
SMBus SDA																																
SMBus SCL																																
UART1 TX1																																
UART1 RX1																																
PCA CEX0																																
PCA CEX1																																
PCA CEX2																																
PCA CEX3																																
PCA CEX4																																
CP0 CPO																																
CP0 CP0																																
CP0 CP1																																
Timer0 T0																																
Timer0 INT0																																
Timer1 T1																																
Timer1 INT1																																
Timer2 T2																																
Timer2 T2EX																																
Timer4 T4																																
Timer4 T4EX																																
SYSCLK SYSLCK																																
CNVSTR CNVSTR																																

```

#include "compiler_defs.h"
#include "C8051F020_defs.h"

// Peripheral specific initialization functions,
// Called from the Init_Device() function
void Port_IO_Init()
{
    // P0.0 - TX0 (UART0), Open-Drain, Digital
    // P0.1 - RX0 (UART0), Open-Drain, Digital
    // P0.2 - CEX0 (PCA), Open-Drain, Digital
    // P0.3 - CP0 (CmpR0), Open-Drain, Digital
    // P0.4 - T0 (Timer0), Open-Drain, Digital
    // P0.5 - T1 (Timer1), Open-Drain, Digital
    // P0.6 - T2EX (Timer2), Open-Drain, Digital
    // P0.7 - T4EX (Timer4), Open-Drain, Digital

    // P1.0 - Unassigned, Open-Drain, Digital
    // P1.1 - Unassigned, Open-Drain, Digital
    // P1.2 - Unassigned, Open-Drain, Digital
    // P1.3 - Unassigned, Open-Drain, Digital
    // P1.4 - Unassigned, Open-Drain, Digital
    // P1.5 - Unassigned, Open-Drain, Digital
    // P1.6 - Unassigned, Open-Drain, Digital
    // P1.7 - Unassigned, Open-Drain, Digital

    // P2.0 - Unassigned, Open-Drain, Digital
    // P2.1 - Unassigned, Open-Drain, Digital
    // P2.2 - Unassigned, Open-Drain, Digital
    // P2.3 - Unassigned, Open-Drain, Digital
    // P2.4 - Unassigned, Open-Drain, Digital
    // P2.5 - Unassigned, Open-Drain, Digital
    // P2.6 - Unassigned, Open-Drain, Digital
    // P2.7 - Unassigned, Open-Drain, Digital

    // P3.0 - Unassigned, Open-Drain, Digital
    // P3.1 - Unassigned, Open-Drain, Digital
    // P3.2 - Unassigned, Open-Drain, Digital
    // P3.3 - Unassigned, Open-Drain, Digital
    // P3.4 - Unassigned, Open-Drain, Digital
    // P3.5 - Unassigned, Open-Drain, Digital
    // P3.6 - Unassigned, Open-Drain, Digital
    // P3.7 - Unassigned, Open-Drain, Digital

    XBR0 = 0x8C;
    XBR1 = 0x4A;
    XBR2 = 0x50;
}

```



L'outil « Configuration Wizard » est un générateur de code de configuration qui pourra vous aider à configurer les divers périphériques

Mais, malheureusement (ou heureusement?) il ne vous dispensera pas de devoir comprendre COMMENT fonctionnent ces périphériques

09/11/2016



LIVE AND
DISCOVER

66



LIVE AND
DISCOVER

Contact

François JOLY
Tél. : 04 72 43 13 36
francois.joly@cpe.fr
www.cpe.fr

- CPE Lyon – Sciences du Numérique
- Domaine Scientifique de la Doua
- 43, bd du 11 novembre 1918 – Bâtiment Hubert Curien
- B.P. 2077 – 69616 Villeurbanne cedex – France