

# Entrée Sorties numériques

---

---

---

---

---

---

---

---

## I. Introduction

- Un système embarqué à microcontrôleur a besoin d'interagir avec le monde "physique", qui est essentiellement analogique :
  - entrées analogiques : capteurs
  - sorties analogiques : actionneurs (moteurs)

---

---

---

---

---

---

---

---

## I. Introduction

- Il doit également interagir avec son environnement technologique, qui est, lui, le plus souvent numérique :
  - entrées numériques
    - gérer des boutons poussoirs
  - sorties numériques
    - Commander des interrupteurs
    - Allumer et éteindre des Leds

---

---

---

---

---

---

---

---

## I. Introduction

- Ceci se fait par l'intermédiaire des GPIOs (Global Peripheral Input Output) que l'on trouve sur tous les microcontrôleurs
- Dispositifs d'entrées sorties permettant des échanges entre une structure microordinateur et son environnement externe.
- Equivalent à des accès à des mots mémoires particuliers appelés **registres d'entrées sorties**

---

---

---

---

---

---

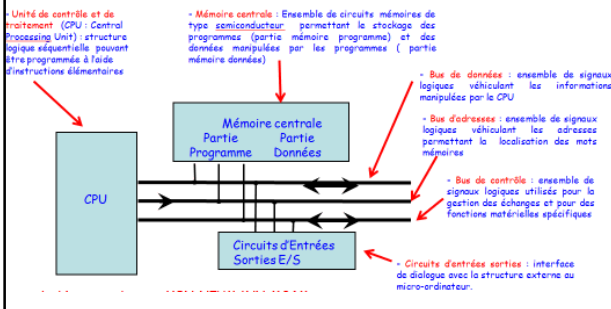
---

---

---

---

## II. Architecture PSoC architecture Von Neuman (R. Fouquet)




---

---

---

---

---

---

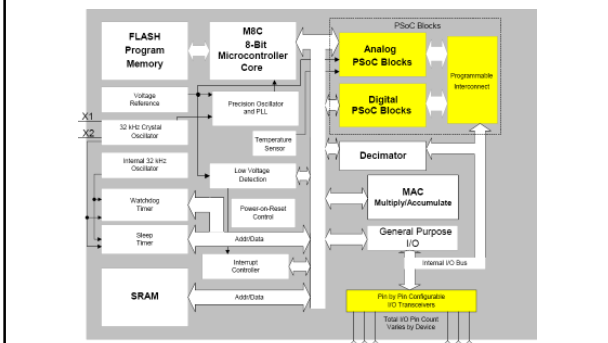
---

---

---

---

## II. Architecture PSoC Parties configurables en jaune




---

---

---

---

---

---

---

---

---

---

## II. Architecture PSoC

### Coeur microcontrôleur

- microcontrôleur **8 bits** d'architecture **Harvard** qui peut exécuter 4 Mips (fréquence d'horloge maximum de **24 MHz**).
- Le CPU intègre 5 registres (non accessibles dans l'espace RAM normal) : le pointeur de pile, le compteur de programme, l'accumulateur, le registre d'index et le registre de drapeaux (flags).

---

---

---

---

---

---

---

---

## II. Architecture PSoC

### Mémoires

- **Mémoire flash**
  - permanente (son contenu est conservé après la coupure de l'alimentation)
  - permet de stocker les programmes ;
  - capacité jusqu'à 32 Ko ;
  - "in-circuit programmable" et aussi "in-application programmable" (IAP), ce qui veut dire que le microprocesseur peut y écrire.

---

---

---

---

---

---

---

---

## II. Architecture PSoC

### Mémoires

- **Mémoire RAM**
  - utilisée pour **stocker les variables d'un projet ainsi que la pile système**.
  - capacités de 256 octets à 2 Ko suivant les familles (pour le CY8C29466 que nous utiliserons, la taille est de 2 Ko).

---

---

---

---

---

---

---

---

## II. Architecture PSoC Contrôleur d'interruption

- Fournit le mécanisme pour qu'une ressource hardware à l'intérieur ou à l'extérieur du PSoC puisse détourner le programme de l'exécution séquentielle des instructions pour aller exécuter le code implanté à une nouvelle adresse, quelle que soit la tâche courante correspondant au code en train d'être exécuté.

---

---

---

---

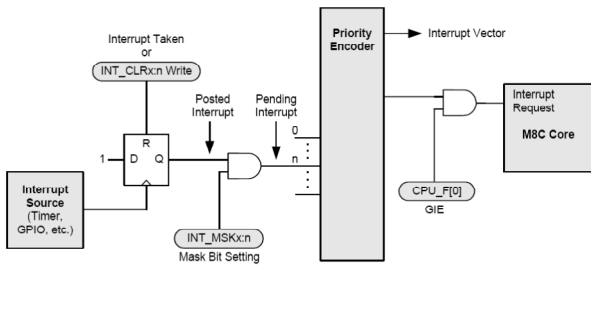
---

---

---

---

## II. Architecture PSoC Contrôleur d'interruption




---

---

---

---

---

---

---

---

## II. Architecture PSoC Fonctionnalités hardware configurables

- Fonctions analogiques / mixtes implantables dans les PSoCs :
  - convertisseurs analogique/numérique (14 bits max.) et numérique/analogique (9 bits max.) ;
  - amplificateurs opérationnels et des amplificateurs d'instrumentation, des filtres et des comparateurs programmables ;
  - ...

---

---

---

---

---

---

---

---

## II. Architecture PSoC

Fonctionnalités hardware configurables

- **Fonctions (User Modules) numériques**
  - Timers, compteurs, PWM 8-16-32 bits ;
  - UARTs supportant les communications RS232 full duplex ;
  - contrôleurs de bus I2C et SPI ;
  - contrôleurs d'afficheurs LCD et 7 segments (modules logiciels) ;
  - ...

---

---

---

---

---

---

---

---

## II. Architecture PSoC

Fonctionnalités hardware configurables ; E / S

- La fonction des broches d'E /S (entrées/sorties) (3 ports 8 bits pour le CY6C29466) est paramétrable :
  - entrée et/ou sortie numérique standard (accessible en lecture et/ou en écriture par le processeur) ;
  - entrée et/ou sortie globale (connectée aux blocs PSoC numériques) ;
  - entrée et/ou sortie analogique.

---

---

---

---

---

---

---

---

## II. Architecture PSoC

Fonctionnalités hardware configurables ; E / S

- Les E/S sont composés de :
  - buffers d'entrées ;
  - drivers de sorties ;
  - registres de données ;
  - registres et logique de configuration.

---

---

---

---

---

---

---

---

## II. Architecture PSoC

Fonctionnalités hardware configurables ; E / S

- Les ports d'entrées/sorties sont **organisés par groupe de 8 bits**.
- Chaque port contient 8 blocs GPIO identiques avec une adresse identique et un numéro de registre de bit pour chaque bloc.

---

---

---

---

---

---

---

---

## II. Architecture PSoC

Fonctionnalités hardware configurables ; E / S

- Chaque broche d'entrée/sortie possède en plus **plusieurs modes de pilotage (drive mode)** et la **possibilité d'être utilisée comme source d'interruptions**.
- Toutes les broches sont identiques et sont configurables en entrées/sorties numériques ; **certaines d'entre elles ne peuvent pas être reliées à des blocs analogiques**.

---

---

---

---

---

---

---

---

## III. Configuration des E/S PSoC

- Les broches d'E/S peuvent être configurées dans la fenêtre Pinout View du Device Editor de PSoC Designer.
- La 1<sup>ère</sup> colonne contient l'identificateur (ID) de la broche du port (par exemple P0[0])

---

---

---

---

---

---

---

---

### III. Configuration des E/S PSoC

Name	Port	Select	Drive	Interrupt
Port_0_0	P0[0]	StdCPU	High Z Analog	DisableInt
Port_0_1	P0[1]	AnalogOutput	log	DisableInt
Port_0_2	P0[2]	Default	log	DisableInt
Port_0_3	P0[3]	GlobalIntEven_0	log	DisableInt
Port_0_4	P0[4]	StdCPU	migrz:analog	DisableInt
Port_0_5	P0[5]	StdCPU	High Z Analog	DisableInt
Port_0_6	P0[6]	StdCPU	High Z Analog	DisableInt
Port_0_7	P0[7]	StdCPU	High Z Analog	DisableInt
Port_1_0	P1[0]	StdCPU	High Z Analog	DisableInt
Port_1_1	P1[1]	StdCPU	High Z Analog	DisableInt
Port_1_2	P1[2]	StdCPU	High Z Analog	DisableInt
Port_1_3	P1[3]	StdCPU	High Z Analog	DisableInt
Port_1_4	P1[4]	StdCPU	High Z Analog	DisableInt
Port_1_5	P1[5]	StdCPU	High Z Analog	DisableInt
Port_1_6	P1[6]	StdCPU	High Z Analog	DisableInt

---

---

---

---

---

---

---

---

---

---

### III. Configuration des E/S PSoC

- La 2<sup>ème</sup> colonne montre le nom de la broche. Port\_2\_5 signifie « broche 5 du port 2 ». On peut le changer en n'importe quel autre plus évocateur.
- La 3<sup>ème</sup> colonne permet de fixer le type de la broche
  - **Analog Input** (uniquement possible sur les ports 0 et 2) ; à utiliser pour les entrées analogiques par exemple sur PGA ou ADC

---

---

---

---

---

---

---

---

---

---

### III. Configuration des E/S PSoC

- **AnalogOutputBuf** (uniquement possible sur le ports 0) ; correspond aux sorties analogiques
- **Default** : ⇒ StdCPU
- **StdCPU** : entrée / sortie « normale » à travers le port (accès par le CPU)

---

---

---

---

---

---

---

---

---

---

### III. Configuration des E/S PSoC

- **Global\_IN, Global\_OUT**
  - Les entrées/sorties globales permettent de **relier aux broches du circuit intégré les signaux depuis/vers les blocs numériques** (compteurs par exemple) du PSoC.
  - Les broches correspondantes **ne seront alors plus accessibles par le processeur**, puisque gérées directement par le hardware
- Autres
  - I2C\_SDA, I2C\_SCL

---

---

---

---

---

---

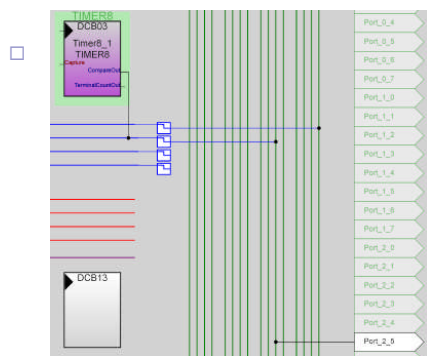
---

---

---

---

### III. Configuration des E/S PSoC




---

---

---

---

---

---

---

---

---

---

### III. Configuration des E/S PSoC

- **Drive Property ou « force » (4<sup>ème</sup> colonne)**

Name	Port	Select	Drive	Interrupt
Port_0_0	P0[0]	StdCPU	High Z Anal	DisableInt
Port_0_1	P0[1]	StdCPU	High Z	bleInt
Port_0_2	P0[2]	StdCPU	High Z Analog	bleInt
Port_0_3	P0[3]	StdCPU	Open Drain High	bleInt
Port_0_4	P0[4]	StdCPU	Open Drain Low	bleInt
Port_0_5	P0[5]	StdCPU	Pull Down	bleInt
Port_0_6	P0[6]	StdCPU	Pull Up	bleInt
Port_0_7	P0[7]	StdCPU	Strong	bleInt
Port_0_6	P0[6]	StdCPU	Strong Slow	bleInt
Port_0_7	P0[7]	StdCPU	High Z Analog	DisableInt
Port_1_0	P1[0]	StdCPU	High Z Analog	DisableInt

---

---

---

---

---

---

---

---

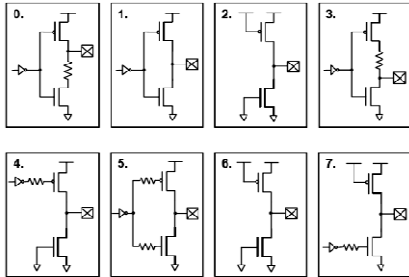
---

---



### III. Configuration des E/S PSoC

□ Drive Property




---

---

---

---

---

---

---

---

### III. Configuration des E/S PSoC

- **High Z** : à utiliser pour les entrées numériques
- **High Z Analog** : à utiliser pour les entrées et les sorties analogiques ou lorsque la broche est inutilisée
- **Open Drain High** : la sortie à l'état haut est pilotée en mode strong ; elle est en haute impédance à l'état bas
- **Open Drain Low**. In this mode, the LOW output is driven with a strong drive and HIGH output is open. This mode is suitable for I2C bus where external pull up resistors are used.

---

---

---

---

---

---

---

---

### III. Configuration des E/S PSoC

- **Strong** : à utiliser si la broche est une sortie ; la broche a une connexion basse impédance à Vss et Vdd ; ne pas utiliser si la broche est une entrée
- **Pull Down** : in this mode, HIGH output is driven strong, and LOW output is through an internal pull down resistor of approximately 5.6K. This mode is used as an input, for example with a switch connected to VCC. This mode is also used as output.

---

---

---

---

---

---

---

---

### III. Configuration des E/S PSoC

- **Pull Up** : this mode is the opposite of the Pull Down mode. In this mode, **HIGH output is driven strong and LOW output is through an internal pull down resistor of approximately 5.6K**. This mode is used as an input, for example with a switch connected to GND. When used as input, the corresponding bit in the PRTxDR register must be set to enable the pull up resistor. Once the pull up resistor is enabled, the state of the pin is read using the PRTxDR register. This mode is also used as output.

---

---

---

---

---

---

---

---

### III. Configuration des E/S PSoC

- **Strong Slow** : This mode is similar to the Strong mode, but the slope of the output is slightly controlled so that high harmonics are not present when the output switches.
- La configuration de chaque bit d'un port est mémorisée dans 3 registres appelés PRTxDM0,PRTxDM1 et PRTxDM2.

---

---

---

---

---

---

---

---

### III. Configuration des E/S PSoC

Drive Modes				Diagram Number	Data =	
DM2	DM1	DM0	Drive Mode		0	1
0	0	0	Resistive Pull Down	0	Resistive	Strong
0	0	1	Strong Drive	1	Strong	Strong
0	1	0	High Impedance	2	Hi-Z	Hi-Z
0	1	1	Resistive Pull Up	3	Strong	Resistive
1	0	0	Open Drain, Drives High	4	Hi-Z	Strong (Slow)
1	0	1	Slow Strong Drive	5	Strong (Slow)	Strong (Slow)
1	1	0	High Impedance Analog	6	Hi-Z	Hi-Z
1	1	1	Open Drain, Drives Low	7	Strong (Slow)	Hi-Z

---

---

---

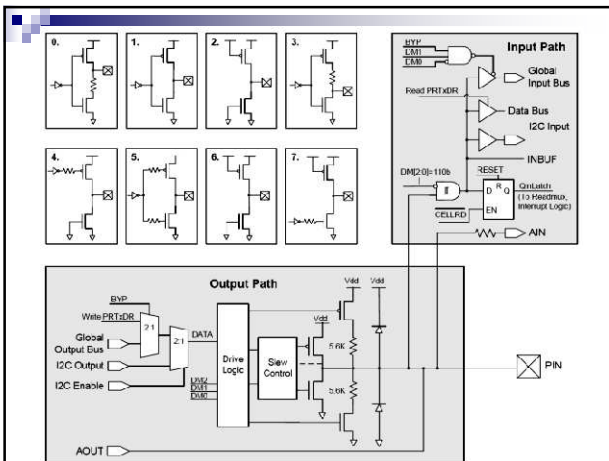
---

---

---

---

---




---

---

---

---

---

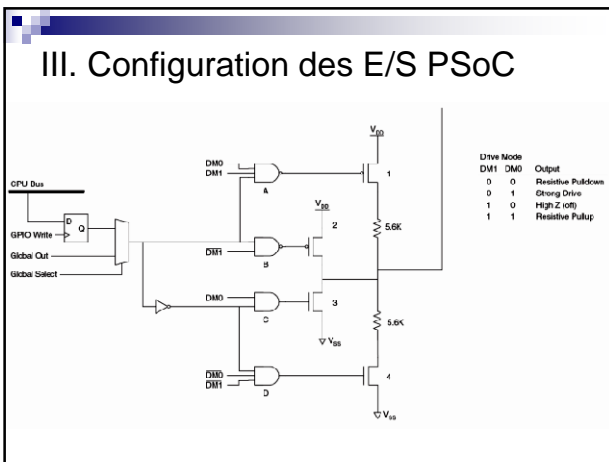
---

---

---

---

---




---

---

---

---

---

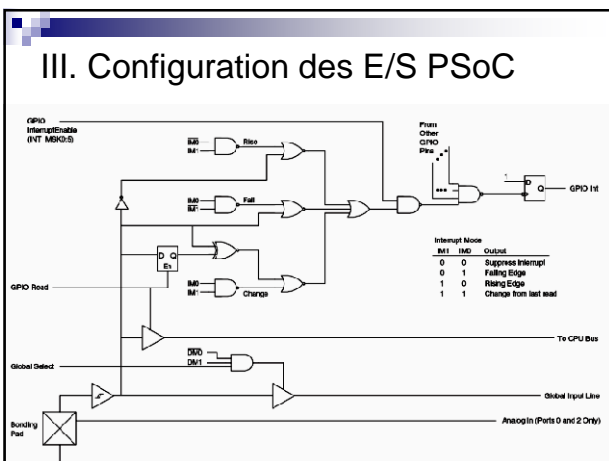
---

---

---

---

---




---

---

---

---

---

---

---

---

---

---

**IV. Utilisation des E/S en mode standard**

- Une des fonctions de base des ports d'E/S est de **permettre au CPU M8C d'envoyer des informations à l'extérieur du PSoC et d'en recevoir en provenance de l'extérieur.**
- Ceci est accompli par le biais de **registres de données de port** (port data register) **PRTxDR.**

---

---

---

---

---

---

---

---

**IV. Utilisation des E/S en mode standard**

- **Lecture sur un port de sortie**
  - pour accéder au contenu d'un port il suffit de lire le registre de 8 bits correspondant au port.
    - `value = PRT1DR ;`
- **Ecriture sur un port de sortie**
  - Les instructions d'écriture du M8C dans le registre PRTxDR stockent la donnée dans ce registre, à raison d'un bit par GPIO.
    - `PRT1DR = value ;`

---

---

---

---

---

---

---

---

**IV. Utilisation des E/S en mode standard**

- **Ecriture sur un port de sortie**
  - En réponse à ce bit de donnée, les drivers de broches positionnent le potentiel de la broche avec une "force" déterminée par les paramètres correspondants
  - **La valeur effective de la tension de sortie dépend donc du "Drive mode" mais aussi de la charge externe.**

---

---

---

---

---

---

---

---

#### IV. Utilisation des E/S en mode standard

- En conclusion :
  - Les broches positionnées en « StdCPU » (paramétrage par défaut) sont des entrées/sorties, c.a.d qu'elles peuvent être utilisées par le CPU à l'intérieur du même programme en lecture et/ou en écriture.

---

---

---

---

---

---

---

---

#### IV. Utilisation des E/S en mode standard ; exemples

- Mise à « 1 » d'un bit d'un port.
  - Instruction C permettant de mettre à 1 la ligne 4 du port 3,
    - `PRT3DR |= 0x10;`
  - Cette instruction se traduit par PRT3DR reçoit PRT3DR OU (0001 0000) en binaire
    - Exemple si PRT3DR = XXXX XXXX, PRT3DR = PRT3DR OU 0001 0000 = XXX1 XXXX
    - On ne modifie ainsi que la ligne 4 du port 3 sans modifier les autres.

---

---

---

---

---

---

---

---

#### IV. Utilisation des E/S en mode standard ; exemples

- Mise à « 0 » d'un bit d'un port.
  - Instruction C permettant de mettre à 0 la ligne 4 du port 3,
    - `PRT3DR &= ~0x10;`
  - Cette instruction se traduit par PRT3DR reçoit PRT3DR et le complément de (0001 0000) en binaire, soit PRT3DR ET (1110 1111) en binaire.

---

---

---

---

---

---

---

---

#### IV. Utilisation des E/S en mode standard ; exemples

- Lecture d'un bit d'un port.
- Pareillement la lecture se fait à l'aide d'un masque en utilisant un test IF.
- Exemple : Suite d'instructions permettant de recopier l'état du bit 0 du port 2 sur le bit 1 du port 0.

---

---

---

---

---

---

---

---

#### IV. Utilisation des E/S en mode standard ; exemples

```

■ if (PRT2DR & 0x01)
■ {
  □ PRT0DR |= 0x02;
■ }
■ else
■ {
  □ PRT0DR &=~ 0x02;
■ }

```

---

---

---

---

---

---

---

---

#### IV. Utilisation des E/S en mode standard ; masques

Bit du port	Masque en binaire	Masque en hexa
0	« 00000001 »	« 01 »
1	« 00000010 »	« 02 »
2	« 00000100 »	« 04 »
3	« 00001000 »	« 08 »
4	« 00010000 »	« 10 »
5	« 00100000 »	« 20 »
6	« 01000000 »	« 40 »
7	« 10000000 »	« 80 »

---

---

---

---

---

---

---

---

## V. Utilisation des E/S en mode Global

- Elles correspondent à des broches reliées à des entrées ou des sorties de blocs PSoC numériques.
- Par défaut, les broches ne sont pas dans cette configuration.

---



---



---



---



---



---



---

## V. Utilisation des E/S en mode Global

- Pratiquement, le choix de positionner une broche « globale » en entrée se réalise dans PSoC Designer dans la fenêtre Pinout en attribuant au paramètre Select la valeur GlobalOutEven\_0 ou GlobalInEven\_0 (cas de la broche 0 du port 0).

---



---



---



---



---



---



---

## V. Utilisation des E/S en mode Global

- Toujours dans la fenêtre pinout, il faut aussi positionner le Drive mode dans l'état High-Z dans le cas d'une entrée et dans l'un des autres Drive mode proposés (strong, pull Up, pull Down, ..) dans le cas d'une sortie.

---



---



---



---



---



---



---

## VI. Bus I2C et liaison RS232

- La liaison RS232, très utilisée il y a encore quelques années sur les ordinateurs, a tendance à y disparaître aux profits de l'USB. L'I2C datant du début des années 80, n'est pas un bus très rapide.
- Cependant, avec le développement de l'électronique embarquée, ces types de liaison restent très utilisés car la plupart des microcontrôleurs savent très facilement les gérer.

---

---

---

---

---

---

---

---

## VI. Bus I2C et liaison RS232 Bus I2C

- Le bus I<sup>2</sup>C ( Inter Integrated Circuit ) a été développé au début des années 80 par Philips semiconductors pour permettre de relier facilement à un microprocesseur les différents circuits d'un téléviseur.

---

---

---

---

---

---

---

---

## VI. Bus I2C et liaison RS232 Bus I2C : caractéristiques

- Le bus I<sup>2</sup>C permet de faire communiquer entre eux des composants électroniques très divers grâce à **seulement trois fils** :
  - Un signal de donnée (**SDA**),
  - un signal d'horloge (**SCL**),
  - et un signal de référence électrique ( **Masse** ).
- Ceci permet de simplifier les interconnexions, par rapport à un schéma classique (8bits de données, 16 bits d'adresse + les bits de contrôle)

---

---

---

---

---

---

---

---



## VI. Bus I2C et liaison RS232

### Bus I2C : caractéristiques

- Les données sont transmises en série à 100Kbits/s en mode standard et jusqu'à 400Kbits/s en mode rapide.
- De nombreux fabricants ayant adopté le système, la variété des circuits disponibles disposant d'un port I<sup>2</sup>C est très importante
  - Ports d'E/S bidirectionnels, Convertisseurs A/N et N/A, Mémoires (RAM, EPROM, EEPROM, etc...), Circuits Audio ( Egaliseur, Contrôle de volume, ... ) et autre drivers ( LED , LCD , ... )

---

---

---

---

---

---

---

---

## VI. Bus I2C et liaison RS232

### Bus I2C : principe

- Plusieurs circuits pouvant être branchés en même temps sur le même bus
- Afin d'éviter les conflits électriques les Entrées/Sorties SDA et SCL sont de type « Collecteur Ouvert » ou « drain ouvert », ce qui permet de réaliser la fonction ET câblé pour les broches de même nom.
- Les deux lignes SDA et SCL sont chacune bidirectionnelles
- Le fait que l'on réalise simultanément sur une broche unique l'entrée et la sortie d'une information (SDA ou SCL) permet au circuit lui-même de s'auto-espionner.

---

---

---

---

---

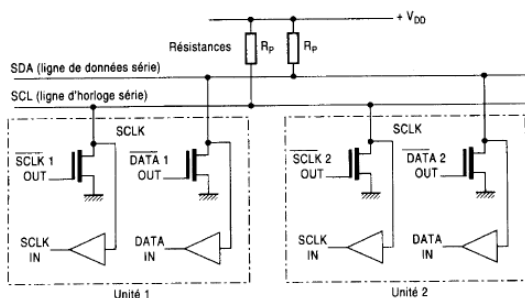
---

---

---

## VI. Bus I2C et liaison RS232

### Bus I2C : principe




---

---

---

---

---

---

---

---

## VI. Bus I2C et liaison RS232

### Bus I2C : protocole I2C

- Afin d'éviter les problèmes dus à une « prise de parole » simultanée de différents modules, il a été nécessaire d'instaurer un protocole entre eux. C'est le **protocole I<sup>2</sup>C**.
- Pour prendre le contrôle du bus, il faut que celui-ci soit au repos. ( SDA et SCL à '1').

---

---

---

---

---

---

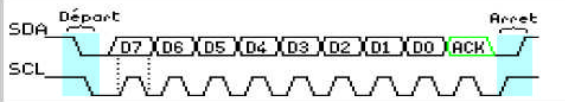
---

---

## VI. Bus I2C et liaison RS232

### Bus I2C : protocole I2C

- Pour transmettre des données sur le bus, il faut donc surveiller deux conditions particulières :
  - La condition de départ. ( SDA passe à '0' alors que SCL reste à '1')
  - La condition d'arrêt. ( SDA passe à '1' alors que SCL reste à '1')




---

---

---

---

---

---

---

---

## VI. Bus I2C et liaison RS232

### Bus I2C : protocole I2C

- Lorsqu'un circuit, après avoir vérifié que le bus est libre, prend le contrôle de celui-ci, il en devient le **maître**. C'est lui qui **génère le signal d'horloge**.
- **Transmission d'un octet**
  - Après avoir imposé la condition de départ, le maître applique sur SDA le bit de poids fort D7.
  - Il valide ensuite la donnée en appliquant pendant un instant un niveau '1' sur la ligne SCL.

---

---

---

---

---

---

---

---

VI. Bus I2C et liaison RS232  
 Bus I2C : protocole I2C

■ **Transmission d'un octet (suite)**

- Lorsque SCL revient à '0', il recommence l'opération jusqu'à ce que l'octet complet soit transmis.
- Il envoie alors un bit ACK à '1' tout en scrutant l'état réel de SDA. L'esclave doit alors imposer un niveau '0' pour signaler au maître que la transmission s'est effectuée correctement.
- Les sorties de chacun étant à collecteurs ouverts, le maître voit le '0' et peut alors passer à la suite.

---

---

---

---

---

---

---

---

VI. Bus I2C et liaison RS232  
 Bus I2C : transmission d'une adresse

- Le nombre de composants qu'il est possible de connecter sur un bus I<sup>2</sup>C étant largement supérieur à deux, il est nécessaire de **définir pour chaque circuit une adresse unique**.
- L'adresse d'un circuit, **codée sur sept bits**, est transmise sous la forme d'un octet au format particulier.

---

---

---

---

---

---

---

---

VI. Bus I2C et liaison RS232  
 Bus I2C : transmission d'une adresse

- On remarque ici que les bits D7 à D1 représentent les adresses A6 à A0, et que le bit D0 est remplacé par le bit de R/W qui permet au maître de signaler s'il veut lire ou écrire une donnée.



Illustration 4 Exemple d'octet d'adresse.

---

---

---

---

---

---

---

---

VI. Bus I2C et liaison RS232

Bus I2C : terminologie

- **Emetteur** : le composant qui envoie des données sur le bus.
- **Récepteur** : le composant qui reçoit les données présentes sur le bus.
- **Maître** : le composant qui initialise un transfert, génère le signal d'horloge et termine le transfert. Un maître peut être soit récepteur soit émetteur.

---

---

---

---

---

---

---

---

VI. Bus I2C et liaison RS232

Bus I2C : terminologie

- **Esclave** : le composant adressé par un maître. Un esclave peut être soit récepteur soit émetteur.
- **Multimaître** : plus d'un maître peut tenter de commander le bus en même temps sans en altérer le message.

---

---

---

---

---

---

---

---

VI. Bus I2C et liaison RS232

liaison RS232

- La liaison RS232 bien que très ancienne (milieu des années 60) est suffisamment universelle et simple d'utilisation pour être quasi systématiquement prévue sur les microcontrôleurs
- Télévisions, amplificateurs HiFi

---

---

---

---

---

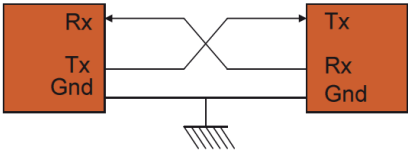
---

---

---

VI. Bus I2C et liaison RS232  
liaison RS232

- Liaison série asynchrone (pas d'horloge).



- Envoi de caractères, poids faible en premier

---

---

---

---

---

---

---

---

VI. Bus I2C et liaison RS232  
liaison RS232

- Envoi de caractères, poids faible en premier
- Vitesses normalisées 9600baud, 19200,....
- Synchronisation par les bits de « stop », « start »
- Possibilité de transmettre un bit de parité

---

---

---

---

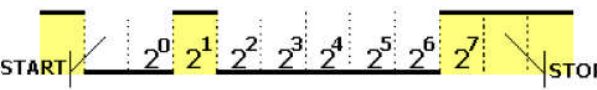
---

---

---

---

VI. Bus I2C et liaison RS232  
liaison RS232



---

---

---

---

---

---

---

---

**VI. Bus I2C et liaison RS232**  
liaison RS232 ; implantation PSoC Tx8

- The TX8 User Module is an 8-bit RS-232 data-format compliant serial transmitter with **programmable clocking and selectable interrupt or polling style operation.**
  - 8-bit serial transmitter with selectable clocking to 48 MHz, yielding maximum 6 Mbit data rate
  - Data framing consists of start, optional parity, and stop bits
  - RS-232 serial-data compliant format with even, odd, or no parity
  - **Optional interrupt on transmit buffer empty condition**

---

---

---

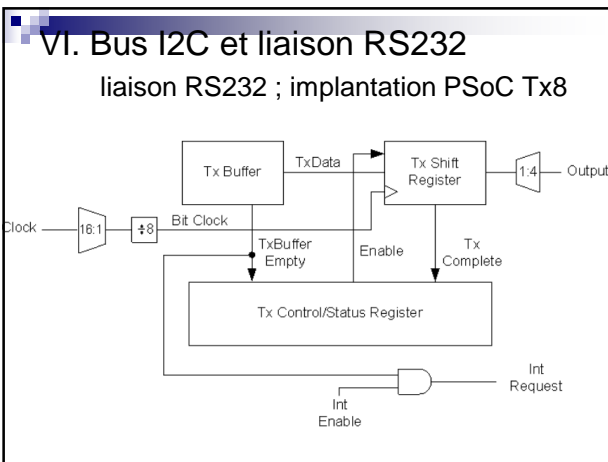
---

---

---

---

---




---

---

---

---

---

---

---

---

**VI. Bus I2C et liaison RS232**  
liaison RS232 ; implantation PSoC Tx8

- A data byte to transmit is written by an API routine into the Buffer register, clearing the Buffer Empty status bit in the Control register.
- The rising edge of the next bit clock transfers the data to the Shift register and sets the Buffer Empty bit of the Control register. If the interrupt enable mask is enabled, an interrupt will be triggered. This interrupt enables the queuing of the next byte to transmit, so that upon completion of transmission of the current data byte, the new byte will be transmitted on the next available transmit clock.

---

---

---

---

---

---

---

---

**VI. Bus I2C et liaison RS232**  
liaison RS232 ; implantation PSoC Tx8

- The start bit is transmitted at the same time that the data byte is transferred from the Buffer register to the Shift register. Successive bit clocks shift a serial bit stream to the output.
- The stream is composed of each bit of the data byte, least significant bit first, an optional parity bit, and a final stop bit.
- Upon completion of transmission of the stop bit, the Control register's Tx Complete status bit is set. This bit will remain valid until read.

---

---

---

---

---

---

---

---

---

---

**VI. Bus I2C et liaison RS232**  
liaison RS232 ; timing PSoC Tx8

---

---

---

---

---

---

---

---

---

---

**VI. Bus I2C et liaison RS232**  
liaison RS232 ; implantation UART PSoC

---

---

---

---

---

---

---

---

---

---

**VI. Bus I2C et liaison RS232**  
liaison RS232 ; implantation PSoC

- Asynchronous receiver and transmitter
- Data-format compliant with RS-232 serial-data format
- Burst rates up to 6 Mbits/second
- Data framing consists of start, optional parity, and stop bits
- **Optional interrupt on receive register full and/or transmit buffer empty**
- Parity, overrun, and framing error detection
- High level transmit and receive functions

---

---

---

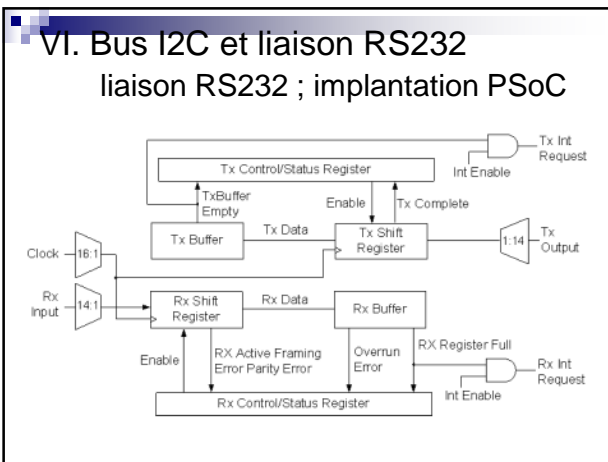
---

---

---

---

---




---

---

---

---

---

---

---

---