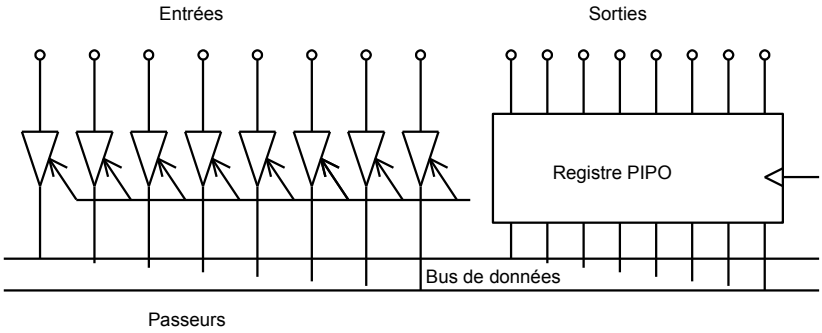
Les entrées-sorties des microcontrôleurs

Pierre-Yves Rochat

ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

Dans les années 1970, au moment où les microprocesseurs sont apparus, des circuits intégrés logiques discrets étaient utilisée pour réaliser les entrées et les sorties du système. Pour les entrées, des **passeurs** permettaient de lire les données lors de la sélection de l'adresse correspondante. Pour les sorties, des **registres *PIPO*** (*Parallel In Parallel Out)* étaient utilisés pour mémoriser les valeurs.



Pour simplifier les schémas, les fabricants de microprocesseurs ont proposé des circuits intégrés contenant tout ce qui est nécessaire pour les entrées-sorties. Comme le nombre d'entrées et le nombre de sorties varient d'une application à une autre, ces circuits offraient la possibilité de programmer certaines pattes comme des entrées et d'autres comme des sortie. *Intel* a proposé son circuit 8255, où des groupes de 4 ou 8 pattes pouvaient être définis ensemble comme entrées ou comme sorties. Mais *Motorola* a fait école avec son *PIA* (*MC6820*), en offrant une flexibilité complète, chaque patte pouvant être définie individuellement comme une entrée ou une sortie.

Ports des microcontrôleurs

Dans les microcontrôleurs se trouvent des pattes groupées par 8 bit (parfois par 16 ou 32 bits). On les appelle des Ports. Les ports ont des noms, par exemple :

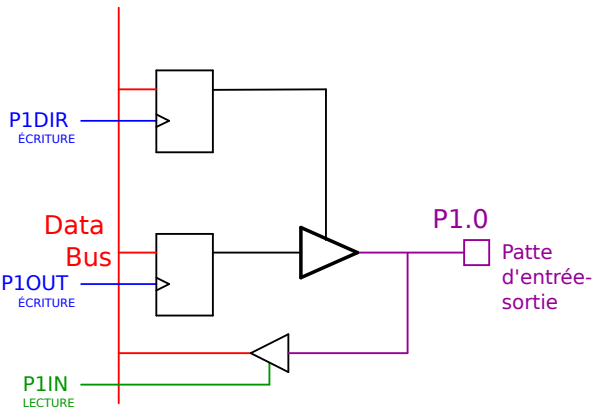
- PORTA, PORTB, PORTC, pour les AVR ou les PIC
- P1, P2, pour les MSP430.

Chaque patte a aussi un nom, dérivé du nom du port :

- PA0, PA1, PA2, pour les AVR
- P1.0, P1.1, P1.2 (pour les MSP430).

Des registres spécialisés permettent de manipuler les ports, par exemple :

- PORTA, PINA, DDRA pour les AVR
- PORTA, PORTA, TRISA pour les PIC
- P1OUT, P1IN, P1DIR pour le MSP430.



Voici le schéma simplifié correspondant à chaque patte d'entrée-sortie, illustré ici pour la patte P1.0 d'un MSP430G.

On y trouve deux passeurs. L'un permet à tout instant de lire l'état de la patte. L'autre permet d'imposer une valeur logique à la patte, lorsqu'il s'agit d'une sortie.

On y trouve aussi deux bascules. Chacune fait partie d'un registre 8 bit. L'une de ces

bascules est appelée ***PxDIR*** *(Port Direction).* Elle permet d'activer ou non la sortie. L'autre est appelée ***PxOUT*** *(Port Output).* Elle donne l'état devant être passé à la sortie.

Ce schéma est très pratique. Il permet en effet de choisir le rôle de chaque patte d'entrée-sortie et de ceci à tout instant. Certains dispositifs de communication nécessitent en effet qu'un patte soit une entrée à certains moments et une sortie à d'autres. C'est le cas par exemple de l'interface de claviers PS-2 des PC.

Le tableau suivant donne la fonctionnalité correspondant à l'état des deux bascules, pour un des bits du port (ici le bit 6).

Dir	Out	Rôle de la patte
0	0	Entrée
0	1	Entrée
1	0	Sortie à 0
1	1	Sortie à 1

Bit 6

Voici donc un exemple d'initialisation des ports, pour avoir P1.0 et P0.6 en sortie et P1.3 en entrée :

```
P1DIR = 0b01000001;
```

Le pattes non utilisées sont laissées en entrée.

En écrivant : `P1OUT = 1;` on va passer la sortie P10. À l'état haut. Mais on va aussi mettre la sortie P1.6 à zéro !

Manipulation de champs de bits

On sent le besoin de pouvoir agir de manière séparée sur chaque bit à l'intérieur d'un registre. Le document « Manipulation de champs de bits » présente ce sujet en détail.

Pour le résumer :

- Set-bit (mettre un bit à 1). Exemple : `P1OUT |= (1<<6);`
- Clear-bit (mettre un bit à 0). Exemple : `P1OUT &=~(1<<6);`
- Test-bit (tester la valeur d'un bit). Exemple : `if (P1IN & (1<<3)) ...`