

# Les entrées-sorties

**Microcontrôleurs pour la commande de systèmes mécaniques**

Jean-Daniel NICOUD et Pierre-Yves ROCHAT

- La solution *Arduino - Wiring*
- Le schéma interne
- Les registres correspondants

- Un numéro arbitraire a été donné pour les pattes d'entrée sortie.

Des procédures spécifiques permettent de les utiliser :

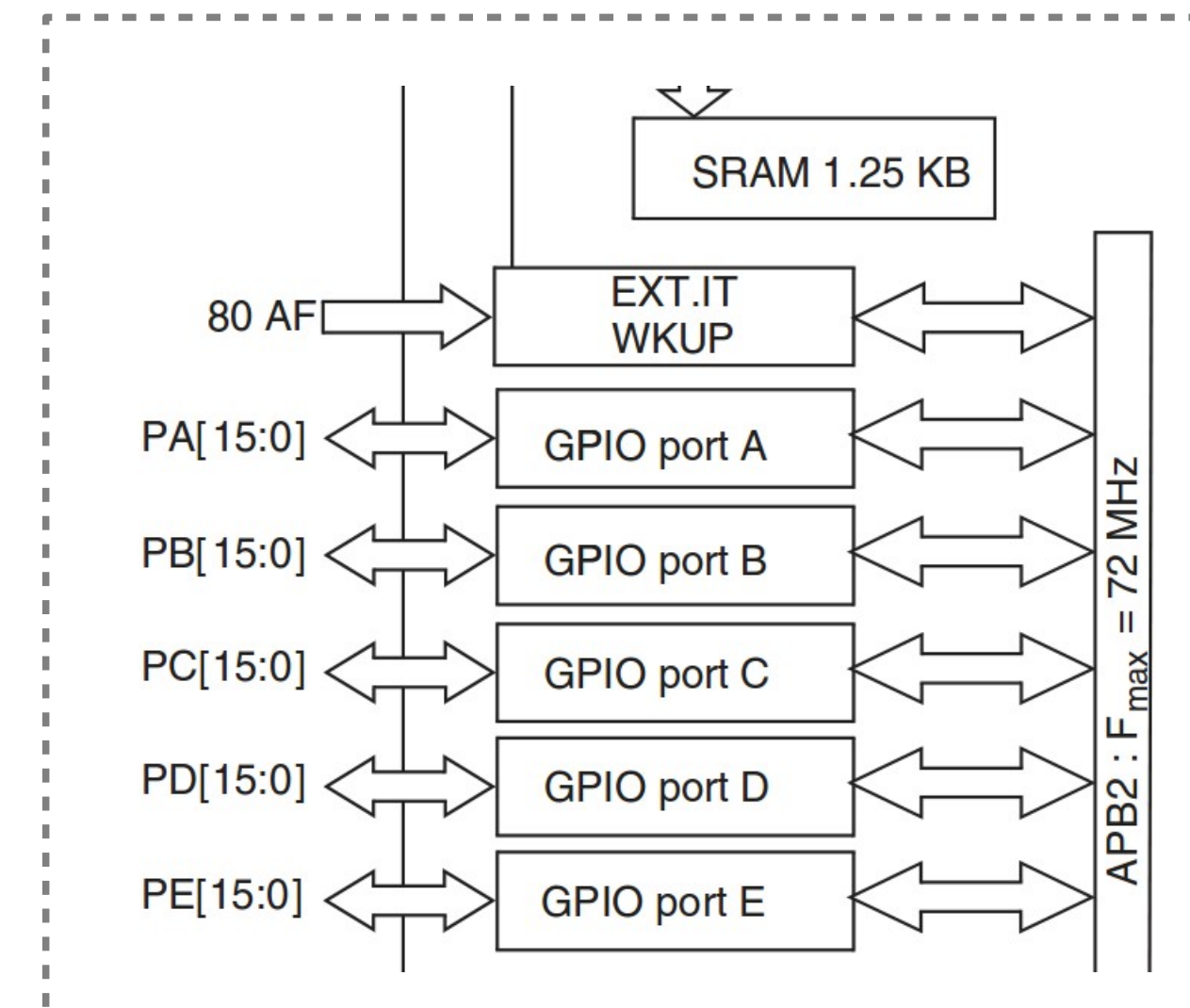
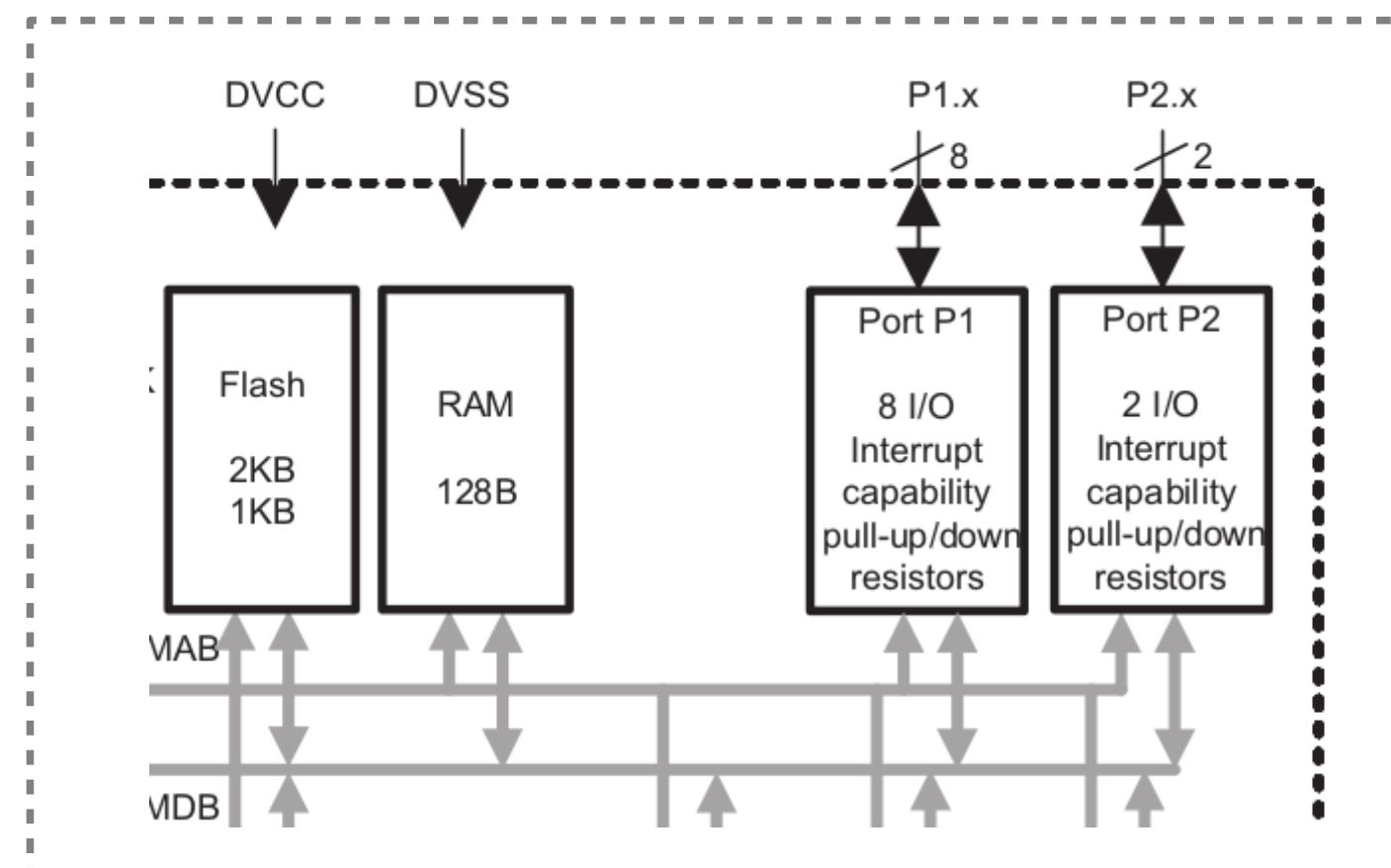
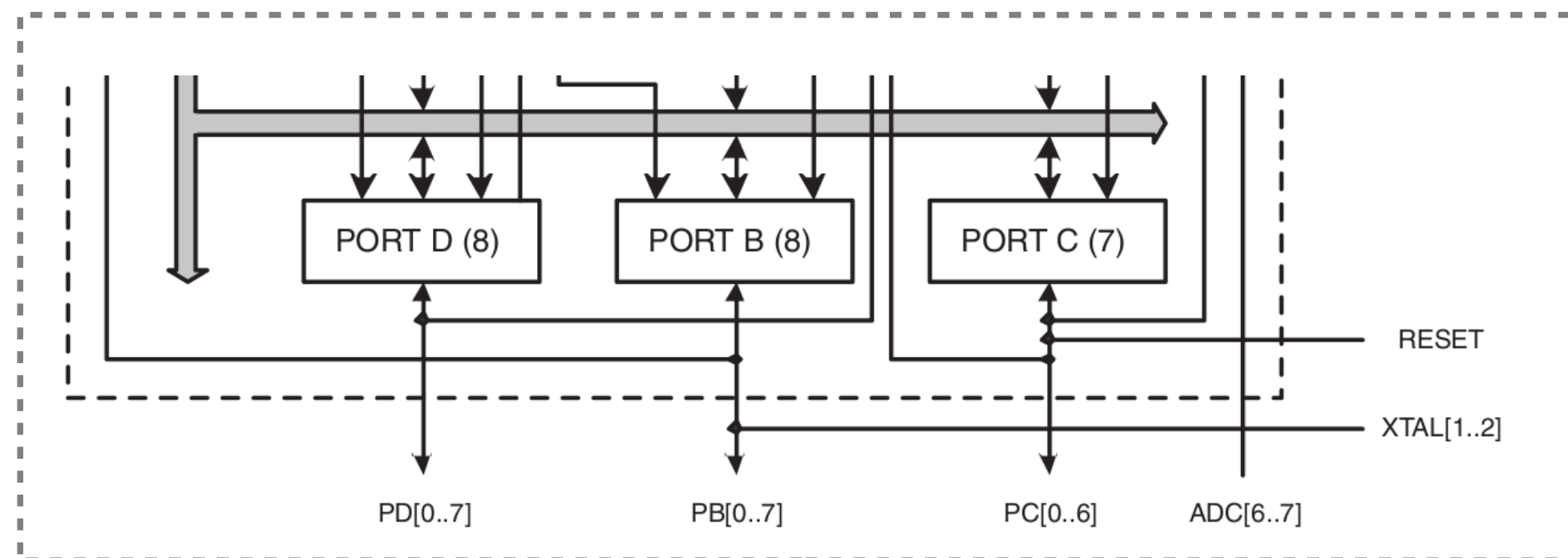
```
pinMode (3, OUTPUT); digitalWrite (3, HIGH);
```

```
pinMode (4, INPUT); if (digitalRead(4)) ...
```

- Les avantages sont la simplicité et une certaine compatibilité entre des environnements différents (Arduino, LaunchPad MSP430, ...)
- Mais nous avons envie de **comprendre** ce qui se cache derrière !

# Ce qui se trouve dans le Microcontrôleur

- Dans les microcontrôleurs se trouvent des pattes groupées par 8 bit (parfois par 16 ou 32 bits). On les appelle des **Ports**.



# Ce qui se trouve dans le Microcontrôleur

- 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 l'AVR ou le PIC
  - `P1`, `P2` pour le 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 le MSP430

# Ce qui se trouve dans le Microcontrôleur

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

**PORTA, PINA, DDRA** pour les AVR

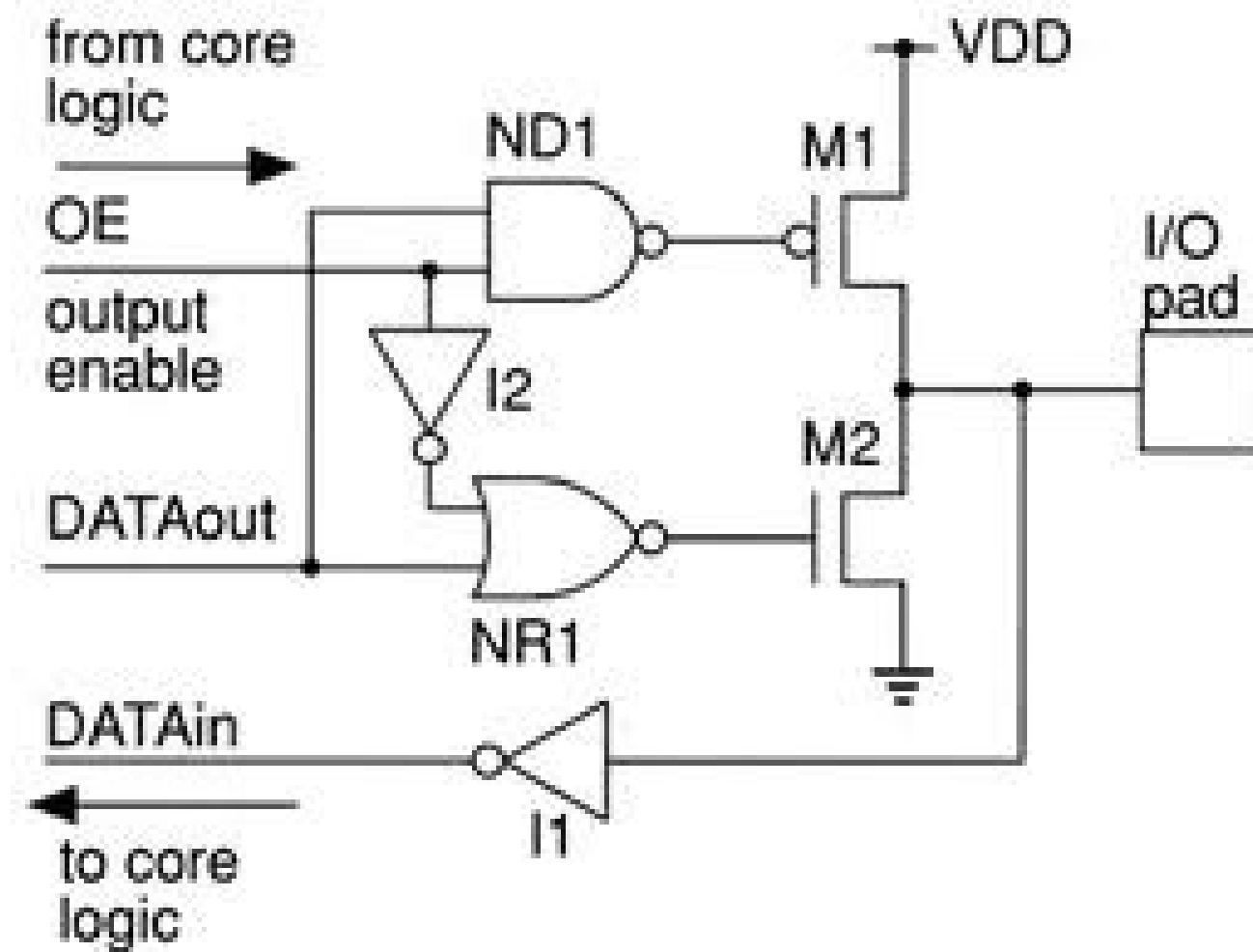
**PORTA, PORTA, TRISA** pour les PIC

**P1OUT, P1IN, P1DIR** pour le MSP430

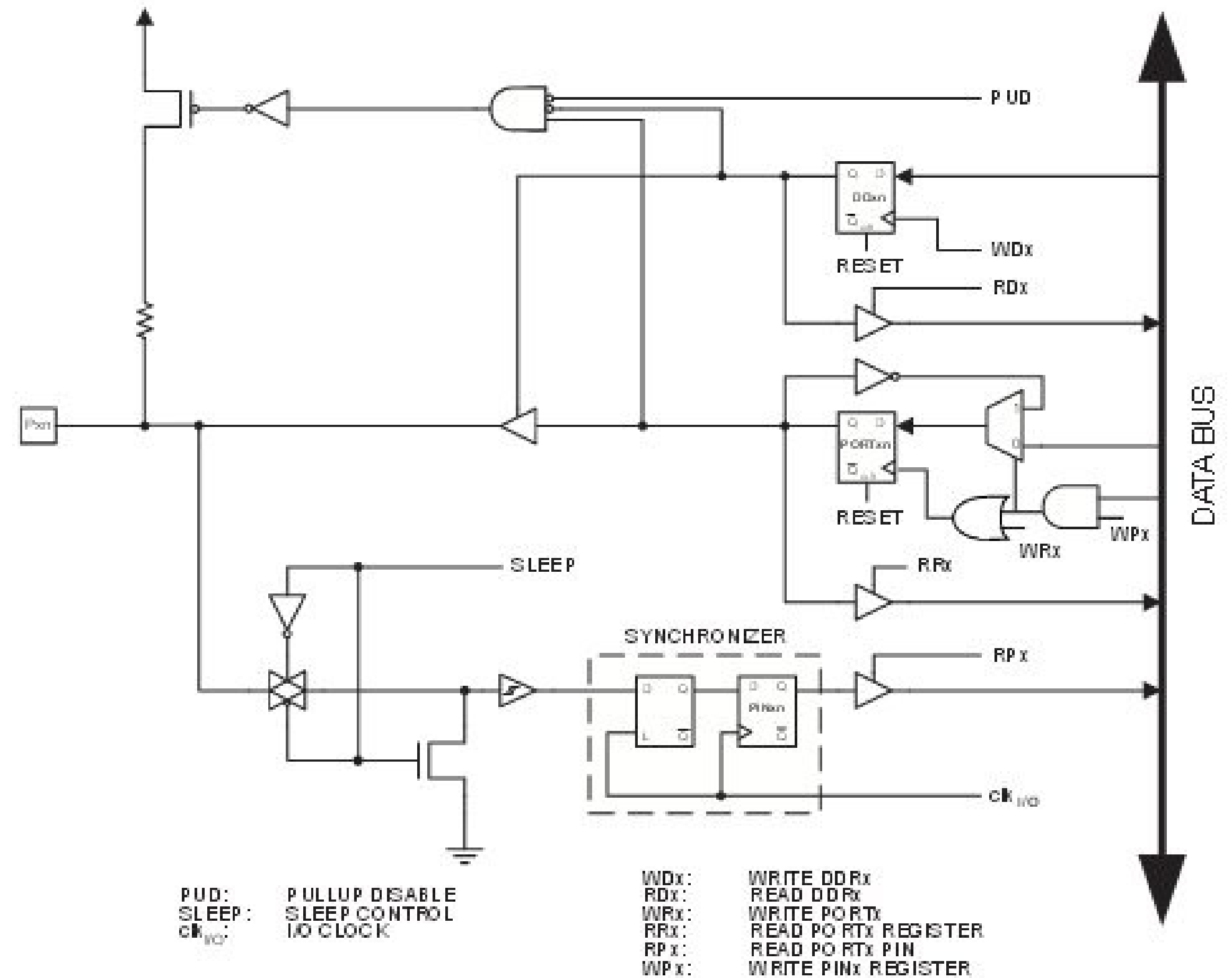
- Essayons d'y voir clair !

- Pour une **sortie**, on doit pouvoir décider de la valeur qui doit être mémorisée ensuite jusqu'à une nouvelle valeur : il faut un **registre**.
- Pour une **entrée**, on doit pouvoir saisir la valeur à un instant donné, il faut un **passeur** (*tri-state driver*)
- Pour qu'une même patte puisse être tantôt une entrée, tantôt une sortie, il faut un second **passeur**.

# Le schéma interne

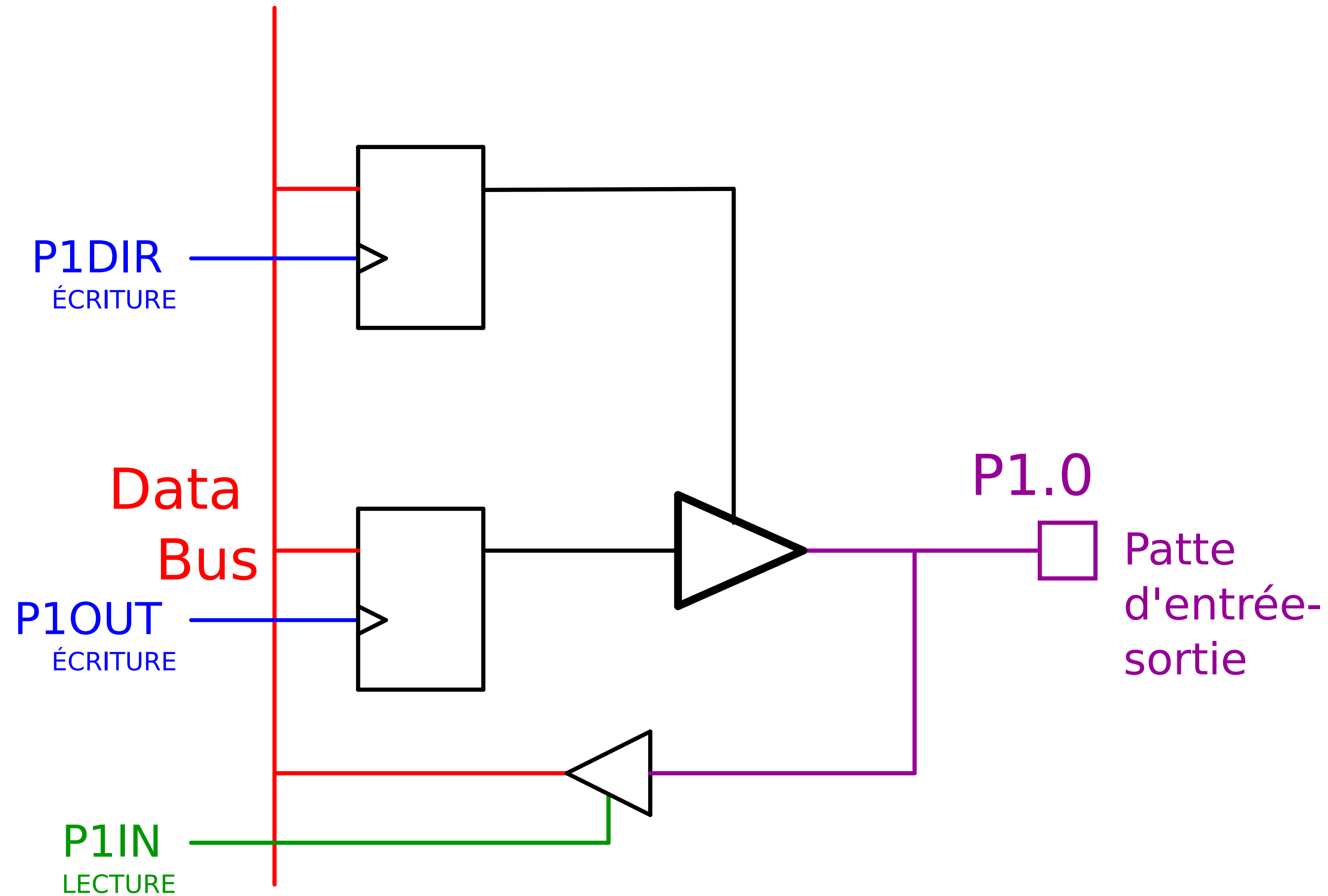


e 13-2. General Digital I/O<sup>(1)</sup>

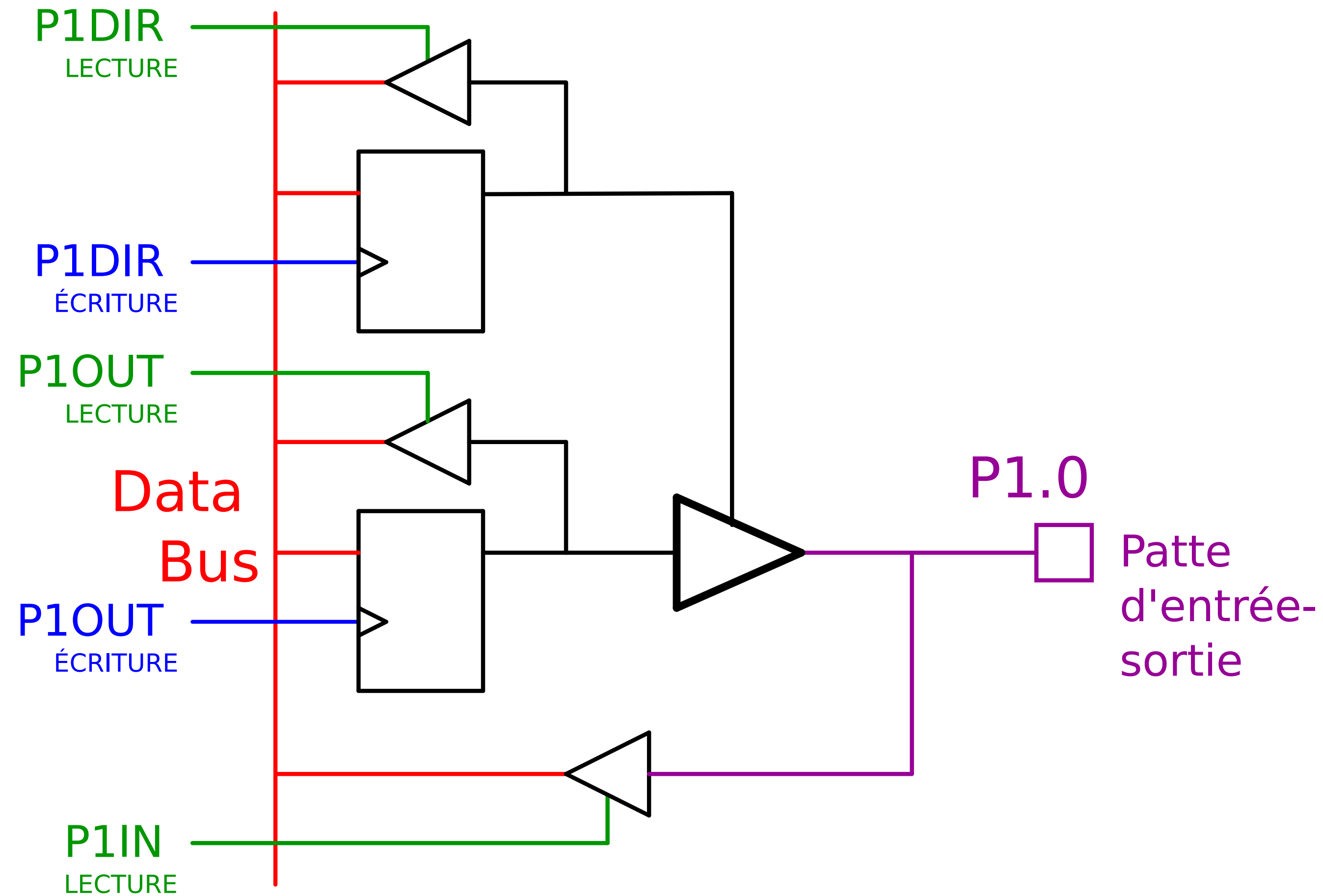




# Le schéma interne



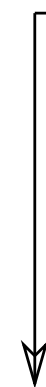
# Le schéma interne



# Représentation logicielle

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

Bit 6



```
P1DIR = 0b01000001;
```

```
P1OUT = 0b01000000;
```

```
if (P1IN == 0b00001000) ...
```

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

Il nous manque l'indépendance des bits... et une bonne lisibilité !

- Le schéma interne :
  - 2 bascule : sortie et direction
  - 2 passeurs : sortie et entrée
- Registre de direction (**P1DIR**)
- Registre de sortie (**P1OUT**)
- Valeur de la patte (**P1IN**)