

Utilisation du port // des PC en mode SPP

Présentation

Le port parallèle des PC initialement dédié à la communication entre l'ordinateur et une imprimante peut être mis à contribution pour réaliser une interface entre le PC et un dispositif électronique. L'avantage d'un tel port est sa facilité de mise en œuvre, tant sur le plan matériel que logiciel.

Il existe 3 modes de fonctionnement pour les ports // :

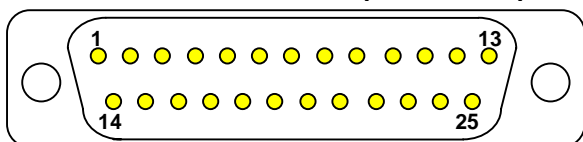
SPP (Standart Parallel Port) : Il s'agit du mode le plus simple et décrit sur cette fiche pratique.

EPP (Enhanced Parallel Port) : Ce mode permet de rendre le port // bidirectionnel et augmente le débit à 2Mo/s.

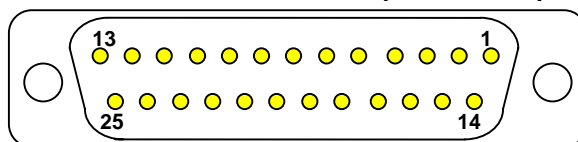
ECP (Extended Capacity Port) : Il s'agit du mode le plus récent et possède la particularité d'être Plug& Play.

Signaux & Connectique

Connecteur DB25 Mâle (vue de face)



Connecteur DB25 Femelle (vue de face)



N° Broche	Nom	Type	N° Broche	Nom	Type
1	/STROBE	Sortie	10	/ACK	Entrée
2	D0	Sortie	11	BUSY	Entrée
3	D1	Sortie	12	PE	Entrée
4	D2	Sortie	13	SELECT	Entrée
5	D3	Sortie	14	/AUTOFEED	Sortie
6	D4	Sortie	15	/ERROR	Entrée
7	D5	Sortie	16	/INIT	Sortie
8	D6	Sortie	17	/SELECT IN	Sortie
9	D7	Sortie	18 - 25	GND	GND

NB : On retrouve des noms de broches en rapport avec la communication vers une imprimante (ex PE : Paper Error).

Les registres

Pour accéder à la lecture ou l'écriture des lignes du port // on dispose simplement de 3 registres de 8 bits comme l'indique la figure ci contre :

Les registres 0x378 (registre de données) et 0x37A (registre de contrôle) sont uniquement accessibles en écriture. On peut lire le contenu du registre, mais on ne peut pas lire les niveaux physiques sur les signaux correspondants.

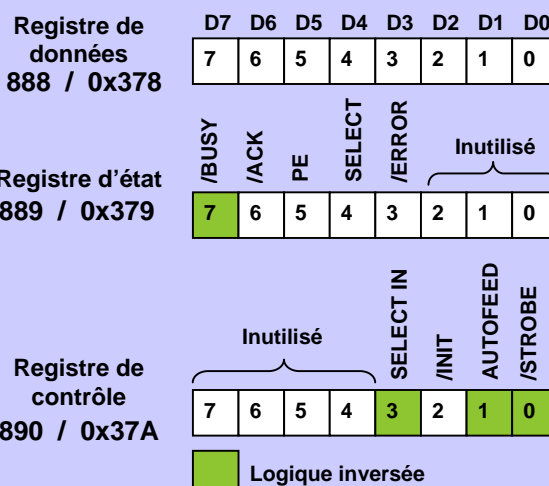
Le registre 0x379 (registre d'état) est uniquement accessible en lecture.

La programmation avec CVI

Seules 2 fonctions sont à connaître :

Pour lire le contenu du registre : valeur=**inp**(adresse).

Pour écrire le contenu d'un registre : **outp**(adresse,valeur).



Exemples

```
outp(0x378,0xAA); /* Ecrit la valeur 0xAA sur les lignes de données) */
unsigned char x; /* Déclaration d'une variable x */
x=inp(0x379)&0x10; /* si x = 0x10 la ligne SELECT se trouve à un niveau logique haut */
outp(0x378,inp(0x378)|0x80); /* Mise à un de la ligne D7 sans changer les autres bits */
outp(0x378,inp(0x378)&0xFE); /* Mise à zéro de la ligne D0 sans changer les autres bits */
```