

Voici le code pour écrire sur le port parallèle

```
Private Sub Commande0_Click()  
  
Dim Valeur As Long  
  
Valeur = 0  
DIPortWritePortUchar Val(888), Valeur  
  
End Sub
```

Voici le code pour lire sur le port parallèle (mais ça, je n'ai pas essayé...)

```
Dim Valeur As Long  
  
Valeur = DIPortReadPortUchar(Val(888))  
  
End If
```

La valeur 888 correspond à mon port imprimante, tu dois regarder dans propriété de configuration, général et gestionnaire de périphérique pour connaître l'adresse du port que tu désire utiliser.

Je crois qu'il faut le convertir en décimal

La variable valeur doit être entre 0 et 255 (en décimal), elle correspond à l'état de tes sorties (qui elles sont en binaire).

Pour que ça fonctionne, tu dois avoir un **module** (donne lui le nom que tu veux) avec le texte :

Option Compare Database

```
*****  
* @doc INTERNAL  
* @module dlportio.bas |  
*  
* DriverLINX Port I/O Driver Interface  
* <cp> Copyright 1996-1999 Scientific Software Tools, Inc.<nl>  
* All Rights Reserved.<nl>  
* DriverLINX is a registered trademark of Scientific Software Tools, Inc.  
*  
* Win32 Prototypes for DriverLINX Port I/O  
*  
* @comm  
* Author: RoyF<nl>  
* Date: 09/26/96 14:08:58  
*  
* @group Revision History  
* @comm  
* $Revision: 2 $  
* <nl>  
* $Log: /DLPortIO/API/DLPORTIO.BAS $  
,  
' 2 3/03/99 5:25p Kevind  
' Removed any reference for customer to call us when encountering bugs,  
' also removed our old address info.  
,  
' 1 9/27/96 2:03p Royf
```

' Initial revision.

'*

```
Public Declare Function DIPortReadPortUchar Lib "dlportio.dll" (ByVal Port As Long) As Byte
Public Declare Function DIPortReadPortUshort Lib "dlportio.dll" (ByVal Port As Long) As Integer
Public Declare Function DIPortReadPortUlong Lib "dlportio.dll" (ByVal Port As Long) As Long
```

```
Public Declare Sub DIPortReadPortBufferUchar Lib "dlportio.dll" (ByVal Port As Long, Buffer As Any, ByVal
Count As Long)
Public Declare Sub DIPortReadPortBufferUshort Lib "dlportio.dll" (ByVal Port As Long, Buffer As Any, ByVal
Count As Long)
Public Declare Sub DIPortReadPortBufferUlong Lib "dlportio.dll" (ByVal Port As Long, Buffer As Any, ByVal
Count As Long)
```

```
Public Declare Sub DIPortWritePortUchar Lib "dlportio.dll" (ByVal Port As Long, ByVal Value As Byte)
Public Declare Sub DIPortWritePortUshort Lib "dlportio.dll" (ByVal Port As Long, ByVal Value As Integer)
Public Declare Sub DIPortWritePortUlong Lib "dlportio.dll" (ByVal Port As Long, ByVal Value As Long)
```

```
Public Declare Sub DIPortWritePortBufferUchar Lib "dlportio.dll" (ByVal Port As Long, Buffer As Any, ByVal
Count As Long)
Public Declare Sub DIPortWritePortBufferUshort Lib "dlportio.dll" (ByVal Port As Long, Buffer As Any,
ByVal Count As Long)
Public Declare Sub DIPortWritePortBufferUlong Lib "dlportio.dll" (ByVal Port As Long, Buffer As Any, ByVal
Count As Long)
```

Ne sachant pas si tu l'as, voici le brochage du port série :

Le port parallèle des PC et compatibles

Cette interface est largement utilisée pour toutes les applications simples ce contrôle se branchant sur un PC. Ce petit texte de description est tiré du livre [Montages Avancés Pour PC](#). Vous pouvez le copier à condition de le garder intact. Afin d'améliorer ce document, toutes les [suggestions](#) sont bienvenues.

Géométrie

Le port parallèle des PC et compatibles se présente sous la forme d'une prise DB25 femelle dont voici la géométrie:

DB25 Femelle (vue de derrière)

\ 1 2 3 4 5 7 8 ... 13 /

\ 14 15 16 17 1825 /

Ce port ayant avant tout été pensé pour communiquer avec une imprimante, vous vous apercevrez que la plupart de ses signaux ont un rapport avec ce périphérique. Il possède 17 broches utilisables ainsi que 8 broches reliées à la masse. Leurs affectations physiques sont les suivantes:

BROCHE NOM Sortie du PC Entré dans le PC

1	/STROBE	X	X
2	D0	X	X si EPP
3	D1	X	X si EPP
4	D2	X	X si EPP
5	D3	X	X si EPP
6	D4	X	X si EPP
7	D5	X	X si EPP
8	D6	X	X si EPP
9	D7	X	X si EPP
10	/ACK		X
11	BUSY		X
12	PE		X
13	SELECT		X
14	/AUTOFEED	X	X
15	/ERROR		X
16	/INIT	X	X

17	/SELECT IN	X	X
18-25	MASSE	X	X

Description des signaux

- STROBE: cette ligne active basse indique à l'imprimante que des données sont présentes sur les lignes D0 à D7 et qu'il faut les prendre en compte.
- D0 à D7: c'est le bus de données sur lequel véhicule la valeur du caractère à imprimer. On ne peut écrire sur ce port, à moins d'avoir un port parallèle étendu (c'est le cas pour les ports de type ECP/EPP).
- ACK: l'imprimante met à 0 cette ligne pour indiquer à l'ordinateur qu'elle a bien reçu le caractère transmis et que ce dernier peut continuer la transmission.
- BUSY: cette ligne est mise à 0 par l'imprimante lorsque son buffer de réception est plein. L'ordinateur est ainsi averti que celle-ci ne peut plus recevoir de données. Il doit attendre que cette ligne revienne à 1 pour recommencer à émettre.
- PE: signifie " paper error ". L'imprimante indique par cette ligne à l'ordinateur que l'alimentation en papier a été interrompue.
- SELECT: cette ligne indique à l'ordinateur si l'imprimante est "on line" ou "off line".
- AUTOFEED: lorsque ce signal est à 1, l'imprimante doit effectuer un saut de ligne à chaque caractère "return" reçu. En effet, certaines imprimantes se contentent d'effectuer un simple retour du chariot en présence de ce caractère.
- ERROR: indique à l'ordinateur que l'imprimante a détecté une erreur.
- INIT: l'ordinateur peut effectuer une initialisation de l'imprimante par l'intermédiaire de cette ligne.
- SELECT IN: l'ordinateur peut mettre l'imprimante hors ligne par l'intermédiaire de ce signal.
- MASSE: c'est la masse du PC.

Performances

La vitesse de transmission maximale que l'on peut espérer obtenir avec un tel port est de l'ordre de 150ko/s. Les ports plus récents, de type EPP (pour Enhanced Parallel Port, développé par Xircom, Zenith et Intel en 1991), permettent d'atteindre un débit de 2Mo/s soit un débit environ treize fois supérieur. Si il reste inférieur au débit du bus ISA (8Mo/s) il permet néanmoins la réception de périphériques tels que des lecteurs de CD-ROM ou des disques durs. En plus d'un débit supérieur, les ports EPP sont bidirectionnels. Encore plus récent, le port ECP (Extended Capacity Port) possède la particularité d'être Plug and Play (traduire par: branchez et utilisez). Il dérive du port EPP et en possède les même caractéristiques. Sa particularité est qu'il permet au périphérique branché sur celui-ci de s'identifier à la machine dès le démarrage. Ainsi l'imprimante peut s'autoconfigurer lors du lancement du système d'exploitation (à condition que ce dernier et le BIOS soient eux aussi compatibles Plug and Play). Mais du point de vue de l'électronicien amateur, l'avantage de ces ports est sans conteste leur bidirectionnalité.

Néanmoins, que ceux qui n'ont pas de port étendu se rassurent, il est tout à fait possible de

rendre bidirectionnel le port grâce à une petite astuce. L'interface parallèle possède en effet 4 bits en entrée (ACK, BUSY, PE et SELECT). Il suffit de multiplexer les 8 bits dont on veut effectuer l'acquisition en deux quartets. La sélection du quartet haut ou du quartet bas s'effectue par une ligne libre du port (SELECT IN par exemple). Ainsi un simple multiplexeur du type 74LS157 résout le problème...

Programmation

Cette interface se programme de manière très simple: trois registres seulement sont nécessaire au contrôle total des signaux. En fait, si vous possédez plusieurs ports parallèles il y aura autant de groupe de trois registres que de nombre de port. Cette précision peut paraître évidente mais cela ne coûte rien de le dire...

Lignes de données (378h)

A moins que vous ne possédiez une interface parallèle étendue, ce registre n'est accessible qu'en écriture. L'écriture d'un octet dans ce registre place immédiatement les lignes D0-D7 aux niveaux demandés sur la prise. Il est une croyance populaire qui dit que les données ne deviennent valables en sortie que si la ligne STROBE est à l'état bas, mais il n'en est rien. Il n'y a pas de logique interne de ce type dans les interfaces parallèles.

7 6 5 4 3 2 1 0

* D7

. * D6

. . * D5

. . . * D4

. . . . * D3

. * . . . D2

. * . . D1

. * D0

Etat de l'imprimante (379h)

Ce registre, accessible uniquement en lecture, est l'image de l'état des lignes ERROR, SELECT, PE, ACK et BUSY. Il faut cependant remarquer que le bit 7 du registre (correspondant à l'image de la ligne BUSY) est inversé par rapport à l'entrée. Un inverseur est en effet présent entre la prise et le registre. Attention donc à ne pas se tromper.

7 6 5 4 3 2 1 0

* /BUSY (Logique inversée)

```

. * . . . . . /ACK

. . * . . . . PE

. . . * . . . . SELECT

. . . . * . . . /ERROR

. . . . . * * * Indéfini

```

Commande de l'imprimante (37Ah)

Ce dernier registre est accessible à la fois en lecture et en écriture. Les 4 premiers bits sont l'image des lignes STROBE, AUTOFEED, INIT et SELECT IN (attention ici aussi aux inverseurs placés entre la prise et le registre). Le bit 4 autorise ou non le déclenchement d'une interruption lorsque ACK passe à l'état bas (c'est à dire quand l'imprimante valide la réception d'un caractère).

Le bit BIT CTRL n'a de sens que pour les ports bidirectionnels (EPP, ECP). En mettant ce bit à 1, le port est accessible en écriture.

```
7 6 5 4 3 2 1 0
```

```

* * . . . . . Inutilis&eactues

. . * . . . . . BID CTRL

. . . * . . . . IRQ INABLE

. . . . * . . . . SELECT IN (Logique inversée)

. . . . . * . . . /INIT

. . . . . . * . . AUTOFEED (Logique inversée)

. . . . . . . * /STROBE (Logique inversée)

```