

INTERFACE PARALLÈLE ET ROBOTIQUE

Gérard MOLINENGAULT

REMPLENER L'IMPRIMANTE PAR UN AUTRE PÉRIPHÉRIQUE

Après tous les principes théoriques que nous avons vus dans l'article précédent, passons à la pratique. Une expérimentation avec une carte de leds vous est proposée. Vous pourrez retrouver des réalisations plus complètes dans l'ouvrage précédemment cité dont les références figurent dans ce bulletin.

Il est hors de question ici de donner toutes les indications nécessaires quant à la réalisation des circuits imprimés et nous estimons que le lecteur est capable de réaliser un circuit à partir d'un typon et d'un schéma d'implantation des composants. Si tel n'est pas le cas, le recours à des ouvrages d'initiation (il en existe de tous les styles...) ou à des personnes capables de cette réalisation est nécessaire. La carte ici proposée, d'un prix de revient minimum grâce à l'utilisation de composants courants et peu nombreux, permet de se confronter à différents problèmes rencontrés dans l'approche des entrées/sorties. Elle est réalisée sur circuit simple face, les composants à utiliser étant classiques. Le schéma, le typon, l'implantation des composants et la nomenclature sont donnés.

La méthode conseillée pour fabriquer une carte est la suivante :

A partir du typon qui vous est proposé, la méthode la plus simple est la méthode photographique avec utilisation de circuit présensibilisé (c'est celle utilisée par l'auteur) mais toute autre méthode menant à la réussite convient. Attention évidemment aux inversions de sens, mais qui a déjà tiré un circuit connaît ce problème. Les composants étant à votre disposition, ainsi qu'un morceau de plaque de circuit imprimé, il reste à réaliser.

La carte étant gravée et percée, les pistes sont soigneusement contrôlées à l'ohm-mètre pour déceler des courts-circuits éventuels ou des

ruptures de continuité des pistes. Les composants sont alors positionnés et soudés en respectant les sens pour les composants non symétriques (circuits intégrés, diodes, transistors et condensateurs). Il est prévu aux quatre coins de la carte la possibilité de mettre des pieds qui peuvent être des vis-écrous par exemple (c'est le choix de l'auteur). Une carte étant terminée, il reste à la tester.

Après positionnement sur le connecteur d'imprimante et mise sous tension, il ne doit rien se passer d'alarmant : on retrouve l'écran connu. La plupart des problèmes rencontrés au cours des fabrications sont dus le plus souvent à des soudures défectueuses. Mais dans notre cas, tout est O.K., nous pouvons continuer.

Notre première réalisation dont les schémas sont donnés ci-après sera une platine de 8 leds. Les borniers à vis au pas de 5,08 sont utilisés ici mais toute autre solution peut être envisagée en pensant que le minimum de contrainte sur la carte entraîne une longévité accrue. On peut par exemple utiliser des borniers sans vis à levier qui, bien que légèrement plus chers, ne nécessitent pas de tournevis pour la connexion. Il sera intéressant de repérer les bornes pour la suite.

LA CARTE DE 8 LEDS

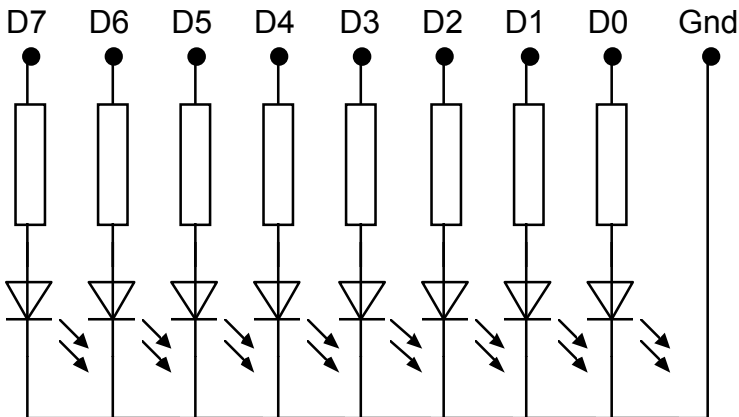
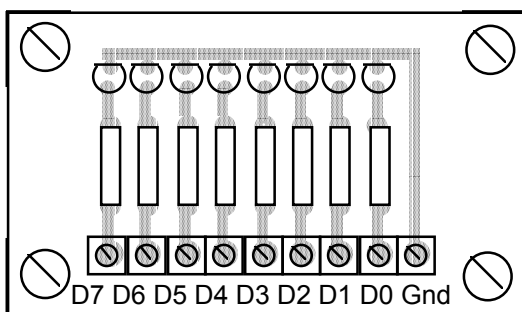
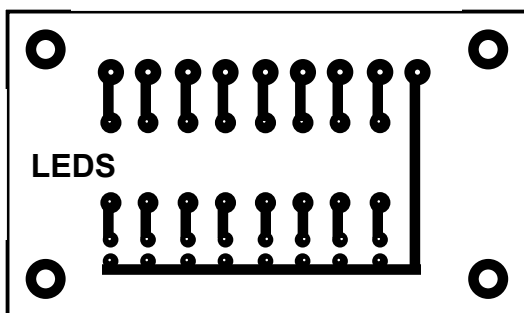


schéma de principe



Nomenclature

8 leds

8 résistances de $150 \Omega \frac{1}{4} W$

un bornier à 9 connecteurs

4 vis-écrous

C'est cette carte que nous allons utiliser pour notre première manipulation. Si des montages et démontages fréquents sont envisagés au niveau des borniers, il est souhaitable d'étamer les extrémités des fils de liaison. Les leds sont reliées au port parallèle, chaque fil établissant la correspondance entre un bit de ce port et une led.

On peut alors visualiser les états de chacune des sorties de l'interface. Un état haut correspond à une led allumée, un état bas à une led éteinte. L'intensité fournie par chacun des bits de données est suffisante pour visualiser l'allumage ou l'extinction des diodes.

On ne fera la mise sous tension de l'ordinateur que lorsque les branchements seront réalisés. A la mise en service de l'ensemble, les leds sont, dans la plupart des cas, toutes éteintes.

Le contrôle des leds s'effectue en basic, en utilisant celui qui est fourni avec le DOS (qbasic sous DOS 6). Taper la commande suivante dans qbasic

OUT &H378,1

La led correspondant au bit D0 du port de sorties s'allume.

Que s'est-il passé ?

&H378 est l'adresse du port de sorties (voir article précédent). Ce port de sorties contient un mot de huit bits. Nous y avons déposé la valeur 1 qui s'écrit 00000001 en binaire avec huit bits (chiffre binaire).

00	00	00	00	00	00	00	11
----	----	----	----	----	----	----	----

Comme une led s'allume sur un état 1 et s'éteint sur un état 0, nous avons bien une led allumée et les autres éteintes.

Pour l'éteindre, il suffit de taper

OUT &H378,0

Cela revient à déposer la valeur 0 sur chacun des bits du port de données, ce qui place toutes les leds en extinction. Ceci étant compris, on peut alors allumer n'importe lesquelles des huit leds. Pour ce faire, il suffit de mettre la combinaison binaire voulue en sortie, une led allumée étant représentée par 1 et 0 représentant une led éteinte. Ce nombre est mémorisé en sortie tant qu'une autre donnée n'est pas inscrite (latch).

Exemple : allumer une led sur deux

00	11	00	11	00	11	00	11
----	----	----	----	----	----	----	----

ce qui correspond à la valeur décimale :

$$1 * 1 + 0 * 2 + 1 * 4 + 0 * 8 + 1 * 16 + 0 * 32 + 1 * 64 + 0 * 128 = 85.$$

On écrit donc

OUT &H378,85

Pour faire défiler toutes les combinaisons possibles, voici un petit programme :

```
FOR P = 0 TO 255
```

```
OUT &H378,P
```

```
NEXT P
```

Si vous trouvez que le défilement est trop rapide, vous devez mettre une pause convenable où il faut. En qbasic, la commande **SLEEP** permet des pauses en secondes, ce qui permet de s'affranchir des boucles dont le temps d'exécution dépend de la fréquence de fonctionnement de l'ordinateur.

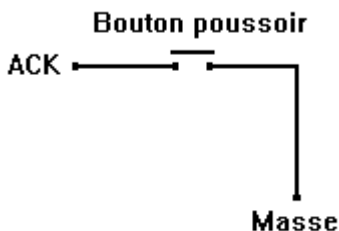
Exemple :

```
FOR P = 0 TO 255
OUT &H378,P
SLEEP 1
NEXT P
```

D'autres exercices :

- une led allumée se déplace dans un sens puis dans l'autre ;
- les huit leds s'allument une à une (la première puis les deux premières puis les trois premières ; elles s'éteignent ensuite une à une... ;
- les led s'allument de manière aléatoire.

Il est possible de commander l'allumage des leds avec un bouton poussoir par exemple. Il faut alors utiliser une entrée dont le changement d'état commandera l'incrémentation du nombre binaire présent sur le port de sorties. L'entrée choisie dans notre exemple est ACK. En se référant à l'article précédent, on constate que l'état de cette entrée s'obtient en utilisant le port de status et un masque. ACK correspond au bit 6 du port d'adresse &H379, donc l'état de ACK est donné par **INP(&H379) and 64**. Le câblage du bouton est le suivant :



A l'état de repos, ACK est à l'état haut. Un appui sur le bouton relie ACK à la masse et donne donc un 0 qui est détecté. Le programme suivant devrait donc permettre l'allumage des leds comme désiré.

affiche :

```
top = INP(&H379) AND 64
IF top = 0 THEN r = r + 1
IF r = 256 THEN r = 0
OUT &H378, r
GOTO affiche
```

En fait, il semble que le résultat tienne plus de l'aléatoire que du contrôlé. Cela est dû aux rebondissements lors de l'établissement du contact du bouton poussoir. Une solution électronique consiste à utiliser un circuit anti-rebonds. Nous préférons une solution logicielle qui consiste à attendre un certain temps pour que le contact soit franchement établi (sous-programme attend dont on peut faire varier la durée). L'ensemble fonctionne alors correctement.

Voici le programme correspondant :

affiche :

```
top = INP(&H379) AND 64
IF top = 0 THEN r = r + 1 : GOSUB attend
IF r = 256 THEN r = 0
OUT &H378, r
GOTO affiche
```

attend :

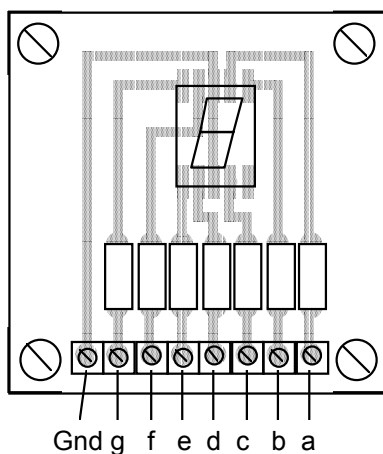
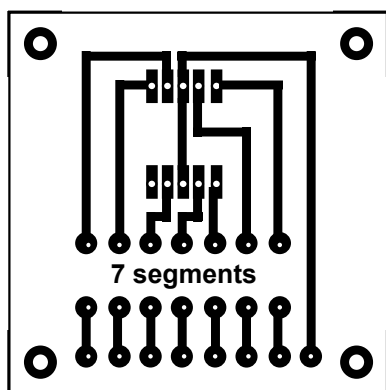
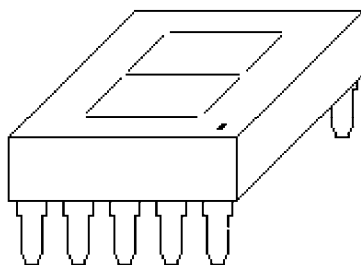
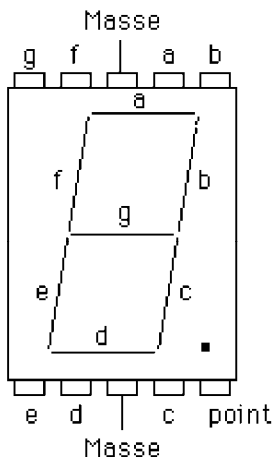
```
FOR t = 0 TO 2000 : NEXT
RETURN
```

COMMANDE D'AFFICHEUR 7 SEGMENTS

Au cours de cette manipulation, nous utilisons un afficheur 7 segments à cathode commune.

L'afficheur 7 segments

Rappelons qu'il existe deux grands types d'afficheurs 7 segments : celui à anode commune (AC) et celui à cathode commune (CC). Nous choisissons un type cathode commune, tout ce que nous dirons pouvant facilement s'adapter à l'autre catégorie. Le modèle employé est représenté ci-dessous avec son brochage.



Nomenclature

un afficheur 7 segments cathode commune avec éventuellement un support

7 résistances $150 \Omega \frac{1}{4} W$

un bornier 8 connecteurs

4 vis-écrous

Dans ce type d'afficheur, un segment s'allume sur un 1 logique (c'est le contraire dans le cas d'un afficheur à anode commune, d'où l'existence des deux types). Ainsi pour afficher 6, les segments c, d, e, f et g doivent être allumés, les segments a et b étant éteints, ce qui conduit à la situation suivante :

- a et b sont au niveau logique 0
- c, d, e, f et g sont au niveau logique 1

L'ensemble des cas possibles est établi dans le tableau suivant. Les colonnes CC et AC donnent les nombres décimaux à utiliser pour obtenir, suivant le type d'afficheur utilisé, le chiffre indiqué dans la première colonne. On remarquera que la somme des deux nombres est toujours 127.

	D6(64)	D5(32)	D4(16)	D3(8)	D2(4)	D1 (2)	D0 (1)		
Chiffre	g	f	e	d	c	b	a	CC	AC
0	0	1	1	1	1	1	1	63	64
1	0	0	0	0	1	1	0	6	121
2	1	0	1	1	0	1	1	91	36
3	1	0	0	1	1	1	1	79	48
4	1	1	0	0	1	1	0	102	25
5	1	1	0	1	1	0	1	109	18
6	1	1	1	1	1	0	0	124	3
7	0	0	0	0	1	1	1	7	120
8	1	1	1	1	1	1	1	127	0
9	1	1	0	0	1	1	1	103	24

Programme de commande de l'afficheur

Affiche les nombres de 0 à 9 avec une temporisation de 1s
début :

```
FOR I = 0 TO 9 : READ A(I) : NEXT
DATA 63, 6, 91, 79, 102, 109, 124, 7, 127, 103
```

affiche :

```
FOR R = 0 TO 9
OUT &H378, A (R)
SLEEP 1
NEXT R
GOTO affiche
```

Comme pour la platine de leds, nous pouvons utiliser un bouton-poussoir pour le changement de chiffres. Avec le même montage (commande sur ACK), le programme de commande devient alors :

```
FOR i = 0 TO 9 : READ a(i) : NEXT
DATA 63,6,91,79,102,109,124,7,127,103
```

affiche :

```
top = INP(&H379) AND 64
IF top = 0 THEN r = r + 1 : GOSUB attend
IF r = 10 THEN r = 0
OUT &H378, a(r)
GOTO affiche
```

attend :

```
FOR t = 0 TO 2000 : NEXT
RETURN
```

Dans le cas présent, nous n'avons pas la possibilité d'utiliser deux afficheurs de la même manière car chacun d'eux nécessite sept bits. Une solution serait dans l'utilisation de multiplexage, ce qui est développé dans la publication : interface parallèle et robotique disponible auprès de l'EPI.

Gérard MOLINENGAULT