

UNE UTILISATION PÉDAGOGIQUE DE MICRO-TOUR

Michel DEVAUD

PRÉAMBULE

Avertissement

Cet article se contente de relater l'utilisation de MICRO-TOUR (cf. N° Spécial 1985) que j'ai pratiquée avec des élèves de collège. Il ne s'agit pas de prétendre que c'est la meilleure utilisation, ni la seule possible.

Motivations

La démarche qui m'a conduit à concevoir, dans un premier temps, puis à utiliser, par la suite, MICRO-TOUR est une démarche informatique. Il s'agit pour moi de montrer que l'on peut connecter à un ordinateur des périphériques d'entrées/sorties autres que les traditionnels clavier/écran/ imprimante et ainsi faire découvrir aux élèves les principes de la robotique et de la conduite informatisée de processus industriel.

MICRO-TOUR n'est pas une machine à commande numérique et point n'est besoin de connaître les techniques de tournage pour l'utiliser. Alors, pourquoi un tour ?

Tout d'abord, je voulais un robot. Le "traditionnel" bras manipulateur ne me satisfaisait pas: cher à l'achat, il est difficilement pilotable par l'élève moyen de collège. En effet, son système de fonctionnement par articulations demande une parfaite maîtrise de la notion d'angle (dans l'espace) et du système de repérage polaire. De plus, s'il comporte des actionneurs, il n'a généralement pas de capteurs et ne mérite guère le nom de robot. Enfin, très différent de ses homologues de l'industrie, son utilité n'est pas évidente à expliquer.

J'avais utilisé avec mes élèves la maquette de pont roulant réalisée par Michel BONNET (I.N.R.P.) et j'étais convaincu qu'il fallait se limiter

à un système plan (2D) avec repérage cartésien (en "XY"). Il fallait que le fait que le robot serve "à quelque chose" soit évident et qu'il offre une mage relativement fidèle du modèle industriel. D'où un tour.

Les élèves concernés

MICRO-TOUR étant un périphérique supplémentaire, je l'utilise avec des élèves qui ont déjà reçu quelques petites notions d'informatique.

Les pré-requis nécessaires en informatique sont peu importants en gros, ils consistent en savoir écrire quelques lignes de programme en un quelconque langage, le seul élément indispensable étant la notion de test (SI ... ALORS ...).

Il n'est pas question pour moi d'aborder (ce que MICRO-TOUR permet facilement) le côté technologie du tour et de son électronique de commande ni la technique du tournage. Ces points relèvent du cours de technologie. A ce sujet, il est intéressant de remarquer que MICRO-TOUR permet une interdisciplinarité informatique (pilotage), technologie (électronique et mécanique) et mathématique (définition d'un profil et repérage dans le plan) ainsi que des réflexions et des recherches au sujet de l'automatisation du travail.

UTILISATION PRATIQUE

Présentation du matériel

Qu'est-ce-que c'est ? Un tour !

A quoi cela sert-il ? A fabriquer des objets comme celui-ci (une quille) !

A ce moment là, le prof de Math que je suis, parle de solides de révolution.

Comment cela fonctionne-t-il ?

Le plus simple, pour répondre sommairement à cette question est d'utiliser le programme de démonstration pour usiner une pièce. On peut également faire définir des pièces par les élèves et les réaliser (cela prend un peu de temps mais a l'avantage de "mobiliser" les élèves).

Détail du matériel

Un tour doit pouvoir déplacer son outil suivant deux axes: parallèlement à l'axe de rotation de la broche (ce qui tient la pièce à usiner) par déplacement du traînard et transversalement par déplace

ment du chariot. Les déplacements obliques peuvent se faire par combinaison de ces deux mouvements. Ces déplacements sont réalisés par des moteurs qui entraînent les éléments mobiles à l'aide de vis.

Ces déplacements doivent être précis: les moteurs utilisés sont des "moteurs pas à pas".

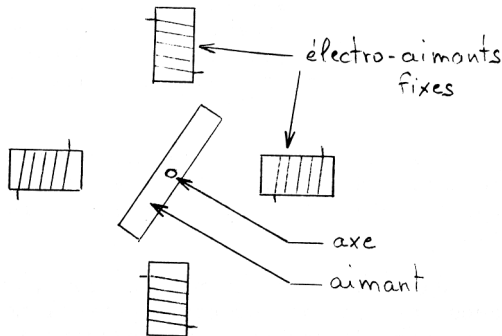
Qu'est-ce-qu'un moteur pas à pas ? C'est un moteur que l'on peut faire tourner d'un angle précis !

Il est intéressant, afin de permettre à l'élève de comprendre ce qu'est un moteur pas à pas de réaliser le petit bricolage suivant (moteur pas à pas, 4 pas au tour).

Le matériel nécessaire :

- un barreau aimanté
- quatre électro-aimants 4,5 V (on peut assez facilement en récupérer ou en trouver chez des marchands de jouets)

Plan du moteur



Montage électrique

Les quatre bobines sont câblées avec un fil en commun de façon à ce qu'elles présentent, lorsqu'elles sont alimentées, toutes la même polarité (N ou S peu importe) vers le noyau tournant. On dispose donc ainsi d'un "commun" et de 4 fils d'alimentation.

Fonctionnement

Lorsque j'alimente une bobine, l'aimant vient se positionner face à elle (N contre S). Si je coupe son alimentation et alimente la suivante, l'aimant tourne d'un quart de tour. Je peux donc ainsi faire tourner mon moteur par pas de 90°. Si j'alimente les bobines dans l'ordre 1, 2, 3, 4, 1,

2... le moteur tourne dans un sens, si l'ordre est 1, 4, 3, 2, 1, 4... le sens de rotation est inversé.

Remarque : Ce moteur, comme un vrai, peut-être alimenté de plusieurs façons: celle déjà décrite, bien sûr, mais aussi une alimentation double (pour plus de puissance - c'est ce que l'on fait avec MICRO-TOUR): 1 et 2, 2 et 3, 3 et 4, 4 et 1, 1 et 2... (à remarquer qu'alors les "pas" sont décalés de 45^0 par rapport au système d'alimentation précédent) ainsi qu'une alimentation par demi pas (pour avoir 8 pas au tour) : 1, 1 et 2, 2, 2 et 3, 3, 3 et 4, 4, 4 et 1, 1...

Passage à la réalité

Les moteurs utilisés sur MICRO-TOUR possèdent plus de bobinages et leur noyau tournant est composé de plusieurs aimants de façon à avoir 200 pas au tour (pas de $1,8^0$) mais leur principe de fonctionnement est identique.

Les déplacements ne sont pas illimités et il est indispensable de savoir qu'un élément mobile est arrivé en fin de course. D'où le rôle des quatre interrupteurs de fin de course.

Sur la face avant du tour, un poussoir d'arrêt d'urgence permet de couper instantanément l'alimentation du tour en cas de problème. Un poussoir dit "d'arrêt programmé" permet de demander à l'ordinateur pilote d'interrompre momentanément le traitement en cours (pour pouvoir intervenir sur le tour pour dégager un copeau ou lubrifier l'outil, par exemple).

Le moteur de broche est piloté par un relais commandé par l'ordinateur. Cela permet de le mettre en marche ou de l'arrêter par programme.

L'électronique

L'électronique de commande peut se décomposer en deux parties: les circuits d'interface (sur l'ordinateur) et les circuits de puissance (dans le tour).

Je précise un peu le sens des mots interface et circuit de puissance mais n'entre pas dans le détail (je n'enseigne pas l'électronique).

Le pilotage informatique

Il se compose de plusieurs parties distinctes :

1) la programmation du circuit d'interface

Il s'agit d'un circuit d'interface parallèle comparable à celui utilisé pour la connexion d'une imprimante. Le pilotage des moteurs pas à pas demande 8 sorties (2 fois 4 bobines), la commande du moteur de broche, 1 sortie et le contrôle des fin de course et du poussoir d'arrêt programmé, 5 entrées. On a donc au total 9 sorties et 5 entrées. On peut accéder à l'image de ces entrées/sorties par l'intermédiaire de deux adresses mémoire. Chacune de ces mémoires contient un octet soit 8 bits d'information. Quand on écrit dans une de ces mémoires on positionne les sorties, quand on la lit, on peut connaître l'état des entrées (et des sorties aussi!).

Une de ces mémoires, appelée port A, sera programmée en entrée pour ses bits 0 (poussoir d'arrêt programmé), 1, 2, 3 et 4 (interrupteurs de fin de course) et en sortie pour son bit 7 (commande du moteur de broche). L'autre appelée port B sera entièrement programmée en sorties (pour la commande des moteurs pas à pas).

Je ne détaille pas plus et me contente de donner aux élèves les procédures d'initialisation du PIA.

Initialisation de l'interface

- En BASIC

```
20000 REM Initialisation de l'interface
20010 PORTA=&HE7CC:REM pour T07, A7CC pour M05
20020 PORTB=&HE7CE:REM pour T07, A7CE pour M05
20030 POKE PORTA+1,0:POKE PORTA,128:POKE PORTA+1,4
20040 POKE PORTB+1,0:POKE PORTB,255:POKE PORTB+1,4
20050 POKE PORTA,255:POKE PORTB,255:REM Pour tout arrêter
20060 RETURN
```

- En LOGO

(Adresse de base: 59340 pour T007, 42956 pour M05)

```
POUR INIT
  .DEP 59341,0 DEP 59340,128
  .DEP 59341,4 .DEP 59340,255
  .DEP 59343,0 DEP 59342,255
  .DEP 59343,4 .DEP 59342,255
FIN
```

- En LSE

On pourrait procéder comme en BASIC avec la P.B.I. POKE, mais je donne directement aux élèves la P.B.I. INIT

```
20000 PROCEDURE BINAIRE INIT
```

2) *Reconnaître l'état des entrées*

Nous allons essayer de savoir quel est l'état des entrées. Pour cela, nous allons écrire le petit programme suivant :

- En BASIC

```
10 GOSUB 20000:REM initialisation interface
20 PRINT PEEK(PORTA):GOTO 20
RUN
```

- En LOGO

```
POUR LIRE1
EC EXA 59340
LIRE1
FIN
INIT
LIRE1
```

- EN LSE

```
10 &INIT()
20 AFFICHER PEEK(59340);ALLER EN 20
20001 PROCÉDURE BINAIRE PEEK
EXÉCUTER A PARTIR DE 10
```

Les élèves lancent leur programme puis appuient manuellement sur les butées de fin de course et sur le poussoir d'arrêt programmé et observent la variation de la valeur affichée à l'écran.

Pour des raisons de sécurité évidentes, pour cette manipulation, le moteur de broche doit être mis hors tension à l'aide de l'interrupteur situé sur le tour.

Nous allons maintenant écrire des procédures plus sophistiquées nous permettant, pour chacune des entrées, de rendre le résultat VRAI ou FAUX suivant qu'elle est enfoncée ou non.

- En BASIC

```

21000 REM état des butées et du poussoir d'arrêt prog.
21010 X=PEEK(&HE7CC)
21020 PAG=X AND 1
21030 BGA=X AND 2:BDR=X AND 4
21040 BAV=X AND 8:BAR=X AND 16
21050 RETURN

```

- En LOGO

```

POUR ARRET?
RENDS ÉGAL? 1 RESTE EXA 59340 2
FIN

POUR BUTÉE-GAUCHE?
RENDS ÉGAL? 1 RESTE QUOT EXA 59340 2 2
FIN

POUR BUTÉE-DROITE?
RENDS ÉGAL? 1 RESTE QUOT EXA 59340 4 2
FIN

POUR BUTÉE-AVANT?
RENDS ÉGAL? 1 RESTE QUOT EXA 59340 8 2
FIN

POUR BUTÉE-ARRIERE?
RENDS ÉGAL? 1 RESTE QUOT EXA 59340 16 2
FIN

```

- En LSE

```

21000 PROCEDURE &ARRET() LOCAL X
21010 X PEEK(59340);RESULTAT ENT(X/2)4X/2
21020 PROCEDURE &BGA() LOCAL X
21030 X ENT(PEEK(59340)/2);RESULTAT ENT(X/2)4X/2
21040 PROCEDURE &BDR() LOCAL X
21050 X ENT(PEEK(59340)/4);RESULTAT ENT(X/2)0X/2
21060 PROCEDURE &BAV() LOCAL X
21070 X ENT(PEEK(59340)/8);RESULTAT ENT(X/2)X/2
21080 PROCEDURE &BAR() LOCAL X
21090 X ENT(PEEK(59340)/16);RESULTAT ENT(X/2)4X/2

```

On peut à l'aide de ces procédures écrire un petit programme donnant en clair l'état des entrées.

- En BASIC

```

10 GOSUB 20000:REM initialisation interface
20 CLS
30 GOSUB 21000:REM saisie des entrées
40 PRINT CHR$(30):REM HOME
50 PRINT "Arrêt: ";:IF PAG THEN PRINT "VRAI" ELSE PRINT "FAUX"
60 PRINT "Bu Ga- ";:IF BGA THEN PRINT "VRAI" ELSE PRINT "FAUX"
70 PRINT "Bu Dr: ";:IF BDR THEN PRINT "VRAI" ELSE PRINT "FAUX"
80 PRINT "Bu Av: ";:IF BAV THEN PRINT "VRAI" ELSE PRINT "FAUX"
90 PRINT "Bu Ar: ";:IF BAR THEN PRINT "VRAI" ELSE PRINT "FAUX"
100 GOTO 30 RUN

```

- En LOGO

```

POUR LIRE
VT LIRE2
FIN

```

```

POUR LIRE2
EC CAR 30
EC LISTE "Arrêt: ARRET?"
EC LISTE "Bu-Ga: BUTEE-GAUCHE?"
EC LISTE "Bu-Dr: BUTEE-DROITE?"
EC LISTE "Bu-Av: BUTEE-AVANT?"
EC LISTE "Bu-Ar: BUTEE-ARRIERE?"
LIRE2
FIN

```

- En LSE

```

10 AFFICHER [.12.' PAG: BGA: BDR: BAV: BAR:',2/]
20 FAIRE 30 TANT QUE VRAI.
30 AFFICHER [C] &ARRET(),&BGA(),&BDR(),&BAV(),&BAR()
EXECUTER A PARTIR DE 10

```

L'élève peut ainsi vérifier la possibilité de connaître à tout instant l'état des différentes entrées.

Michel DEVAUD
(à suivre)