

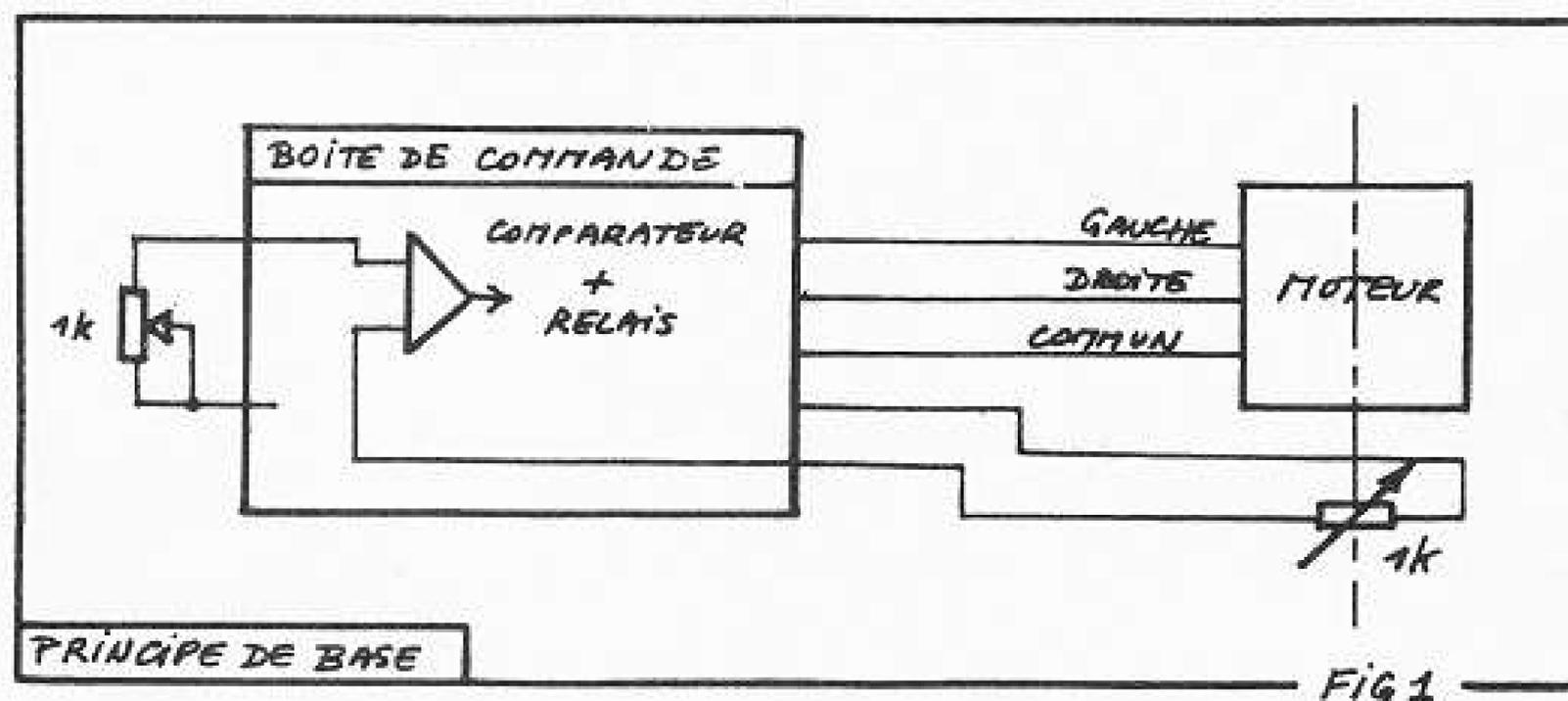
# OSCAR LE PETIT CANARD

Par Michel Vonlanthen, HB9AFO

Certains d'entre vous auront vu mon système de pointage des antennes à l'exposition COMPUTER 80. Voilà donc maintenant la description du hardware de cette installation. Précisons quand-même, pour faire écho aux questions posées, que l'antenne n'était là que pour la démonstration car celles utilisées pour le trafic sur satellite radioamateur étaient trop grosses pour tenir sur le stand (danger de décoiffer les visiteurs...). L'antenne exposée était une QUAGI 1296 MHz, prototype en phase de construction. Le fait qu'il n'y avait aucun câble coaxial connecté n'avait donc rien de magique car c'était justement ce qu'il me restait à réaliser sur ce prototype (qui fonctionne actuellement). La vitesse de rotation de l'antenne ne correspondait pas non plus à la réalité. Pour un passage de satellite, les antennes font de 20 à 25 minutes pour passer d'un horizon à l'autre. Je n'avais accéléré le mouvement que pour la démonstration.

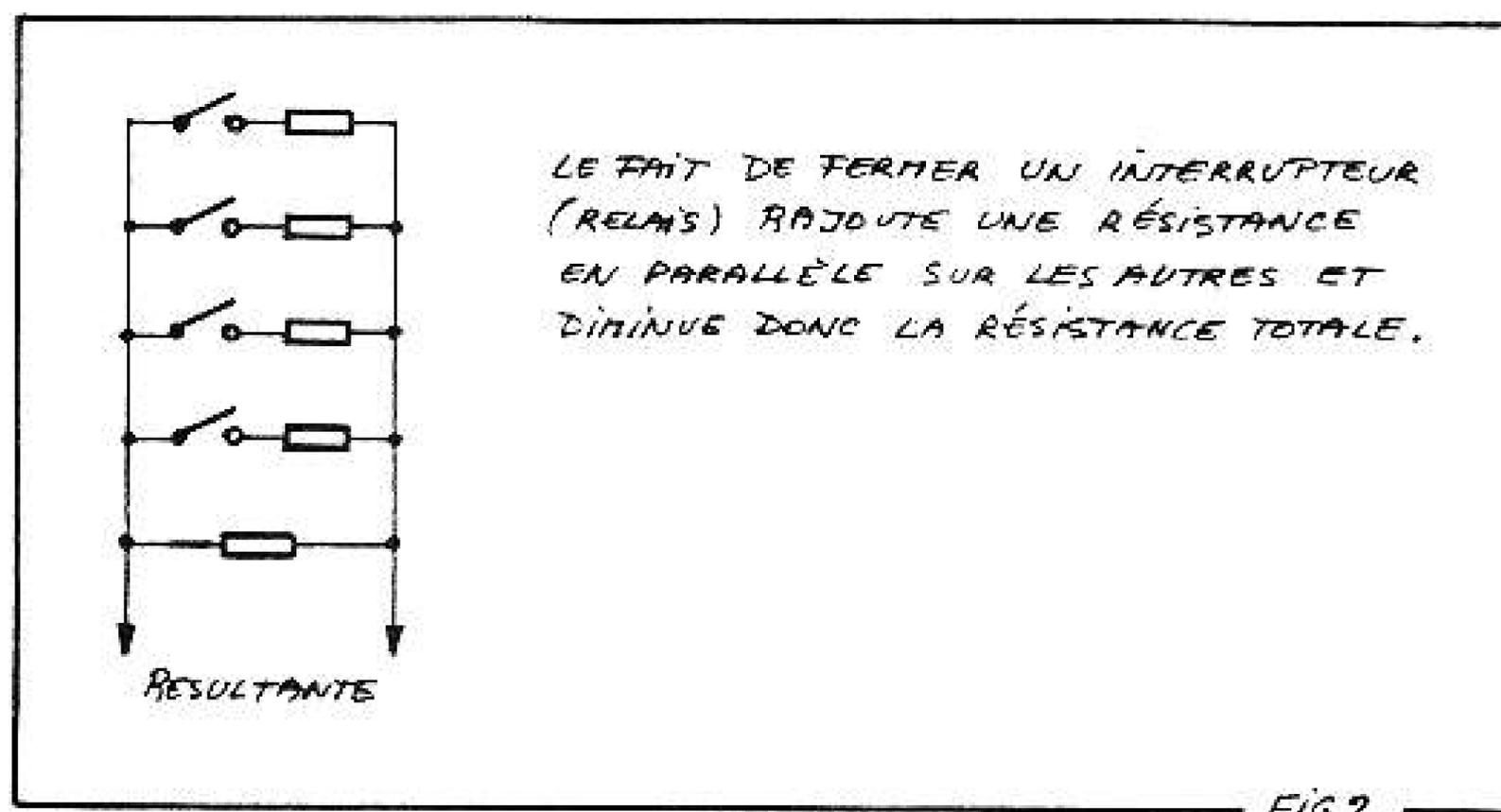
## Système de pointage automatique des antennes: hardware

Lorsqu'on étudie ce genre de problème, la première solution qui vient à l'esprit est celle du convertisseur digital-analogique avec comparateur de tension. Des descriptions de ce genre ont déjà été publiées, je n'entrerai donc pas dans les détails. L'inconvénient de ce genre de montage est qu'il faut construire de a à z une nouvelle commande d'antenne. La boîte de commande d'origine nous reste donc sur les bras, sans utilité aucune. C'est en pensant à cela que j'ai eu l'idée de ma propre solution. Mais avant d'y arriver, voyons comment se compose un système de rotation d'antenne. (figure 1)



Ce genre de "rotor d'antenne" (c'est leur nom habituel) est peu répandu chez nous pour la réception TV à cause de nos montagnes qui empêchent la réception d'émissions lointaines. Par contre, ils sont très courants dans les pays plats comme la Hollande ou la Belgique par exemple. La figure 1 en illustre le principe. Le pot. de 1k est gradué de 0 à 360 degrés et représente la direction de l'antenne. Un potentiomètre identique est logé dans le boîtier du moteur et tourne en synchronisme avec lui. Ils arrivent tous les

deux sur un comparateur de phase (on utilise plutôt le 50 Hz que le continu pour des questions de stabilité dans l'ampli-comparateur). S'ils sont dans la même position, il n'y a pas de tension d'erreur, donc le moteur ne bouge pas. Si on change maintenant la position du potentiomètre principal, on a une tension de différence ce qui fait généralement coller un relais qui, à son tour, fait tourner le moteur dans la bonne direction. Ce dernier s'arrête lorsque l'équilibre est rétabli. Voilà donc le principe de ce "rotor d'antenne". On voit donc que la rotation de l'antenne de 0 à 360 degrés correspond à une résistance variant de 0 à  $1024 \text{ Ohms}$ . L'idée est donc simple: trouver un montage qui transforme un mot de 8 bits du microprocesseur en une résistance variant de zéro à  $1024 \text{ Ohms}$ . Élémentaire mon cher Watson... 8 résistances et 8 relais (pour avoir une isolation galvanique totale entre le  $\mu\text{P}$  et la boîte de commande) et le tour est joué !



Avec les 8 bits du microprocesseur, nous avons un pas élémentaire de

$$360 \text{ degrés} / 256 \text{ bits} = 1,4 \text{ degré/bit}$$

D'autre part, je me suis aperçu avec étonnement que le rotor lui-même avait une hystérèse de 5 à 10 degrés. Je dis bien "hystérèse" et non "précision". Cela signifie que si l'on veut faire tourner le moteur, il faudra lui "donner" une différence de 10 degrés au minimum, sans cela, il ne tourne pas. Cet inconvénient est le lot des rotors bon-marché: potentiomètre de faible résolution et amplis de faible gain afin de diminuer les réglages. Pour mon application, suivre un satellite d'un horizon à l'autre, ces valeurs sont suffisantes car les antennes utilisées ont une directivité d'environ 30 degrés à -3 db. Inutile donc d'exiger une précision qui ne serait pas utilisée. Il n'en serait pas de même si l'on voulait faire tourner un groupe d'antennes pour des liaisons radio contre la lune comme vient de le faire un groupe de radioamateurs de Lausanne. Là il faut un très grand gain, donc une grande directivité. Le genre de moteur que j'utilise ne suffirait donc pas par manque de précision... et aussi par manque de robustesse car faire tourner 16 mètres cube, il faut le faire...

Nos moteurs bon-marché conviennent donc parfaitement à notre application et l'interface respecte le cahier des charges:

- utilise la boîte de commande existante
  - simple et économique à construire
- Le principe étant posé, voyons maintenant la

### Réalisation pratique (figure 3)

- On distingue 8 circuits identiques: 1 latch attaqué par le data bus, un driver de relais, 1 relais 5V et la résistance commutée correspondant au poids binaire considéré. L'entrée OUT 004 vient du décodeur d'adresses des périphériques. Lorsque le 8080 reçoit l'instruction OUT 004 il transfère les 8 bits correspondants à la direction de l'antenne du registre A au data bus. En même temps, le décodage du code de périphérique 004 et des signaux ADPERLOW et WRITELOW donnent une impulsion négative sur cette entrée ce qui transfère le contenu du data bus sur les latches et actionnent les relais correspondants. La résistance entre A et B varie comme si on avait tourné un potentiomètre. C.Q.F.D
- Le driver tri-state 74367 est un port d'entrée à 1 bit. Il répond à l'instruction IN 005 et transfère le signal "busy" du moteur sur le bit 0 du data bus. Si busy est à zéro, le moteur est en train de tourner. Ce signal est nécessaire car le moteur fait un tour complet en 1 minute et, comme je veux ré-orienter les antennes chaque minute, il faut savoir tout d'abord si l'orientation précédente est terminée. Rappelons qu'il faut 1 moteur vertical et 1 moteur horizontal mais patience, les détails dans environ 3 tasses de café...

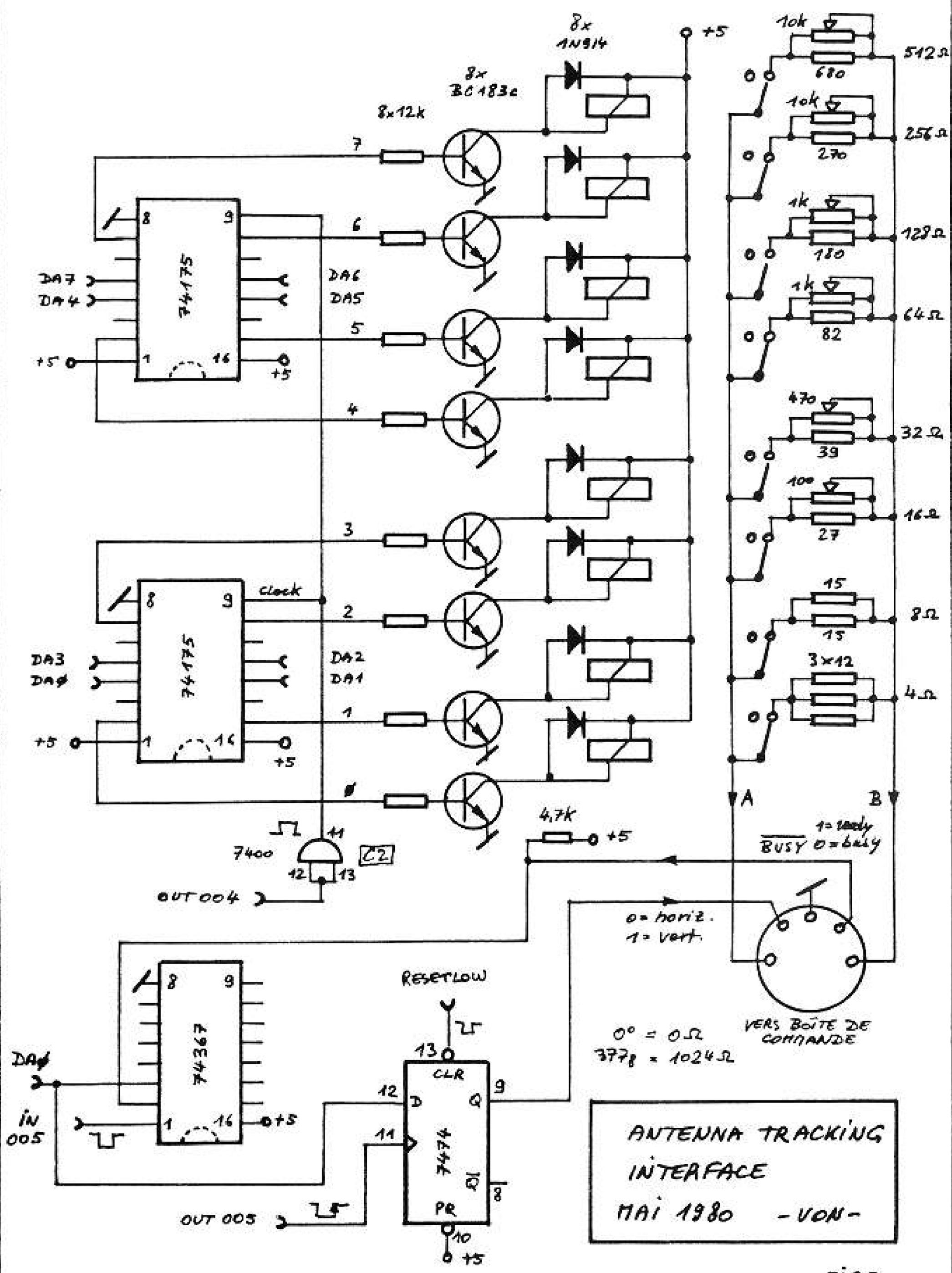
Le signal busy provient du comparateur de phase. Il s'agit d'une tension d'amplitude fixe à 50 Hz. Lorsque l'équilibre est réalisé, ce signal disparaît. Il suffit donc d'amplifier, de redresser et de transformer en TTL (avec optocoupleur en plus) cette tension alternative pour obtenir le "busy". Je n'entre pas dans les détails de cette partie car chaque boîte de commande est différente. A titre indicatif, le moteur horizontal et la boîte de commande sont de marque CDE, type AR33. Le moteur vertical est un STOLLE 2010 modifié. Attention aux couplages galvaniques entre l'interface et la boîte de commande. Il faut mettre des optocoupleurs car, dans mon cas, le fait de relier les masses ensemble provoquait un magnifique court-circuit avec le 24V AC du moteur! Ah le bon-marché...

- Le flip-flop 7474 est un port de sortie. Il répond à l'instruction OUT 005 et correspond au bit 0 du data bus. Il sert à commuter la seule boîte de commande entre les deux moteurs.

Nous avons là la seconde originalité de notre système: il n'utilise qu'une boîte de commande pour deux moteurs. De ce fait, nous économisons 1 interface (donc 8 ic, 8 relais, etc...), une boîte de commande et surtout 5 fils sur le câble de liaison avec les moteurs. La contre-partie de cette économie est que nous ne pouvons pas faire tourner les 2 moteurs en même temps mais devons le faire séquentiellement. En pratique, cela n'a aucune importance car la vitesse du déplacement du satellite est suffisamment lente par rapport à celle des moteurs. La figure 4 est le schéma-bloc de la commande complète.

La boîte de commutation se trouve bien-sûr sur le toit, vers les moteurs. Afin de ne pas faire chauffer les relais, il faut que la position de repos des relais des moteurs corresponde au moteur horizontal car c'est celui qu'on tourne le plus.

A part les éléments de la figure 4, la boîte de commande, nouvelle version, comporte un commutateur manuel/automatique, un commutateur vertical/horizontal, 1 potentiomètre gradué de 0 à 360 degrés et un autre de 0 à 180 degrés. Chaque fonction est aussi visualisée par une lampe-témoin, ce qui est bien utile lors de la mise au point.

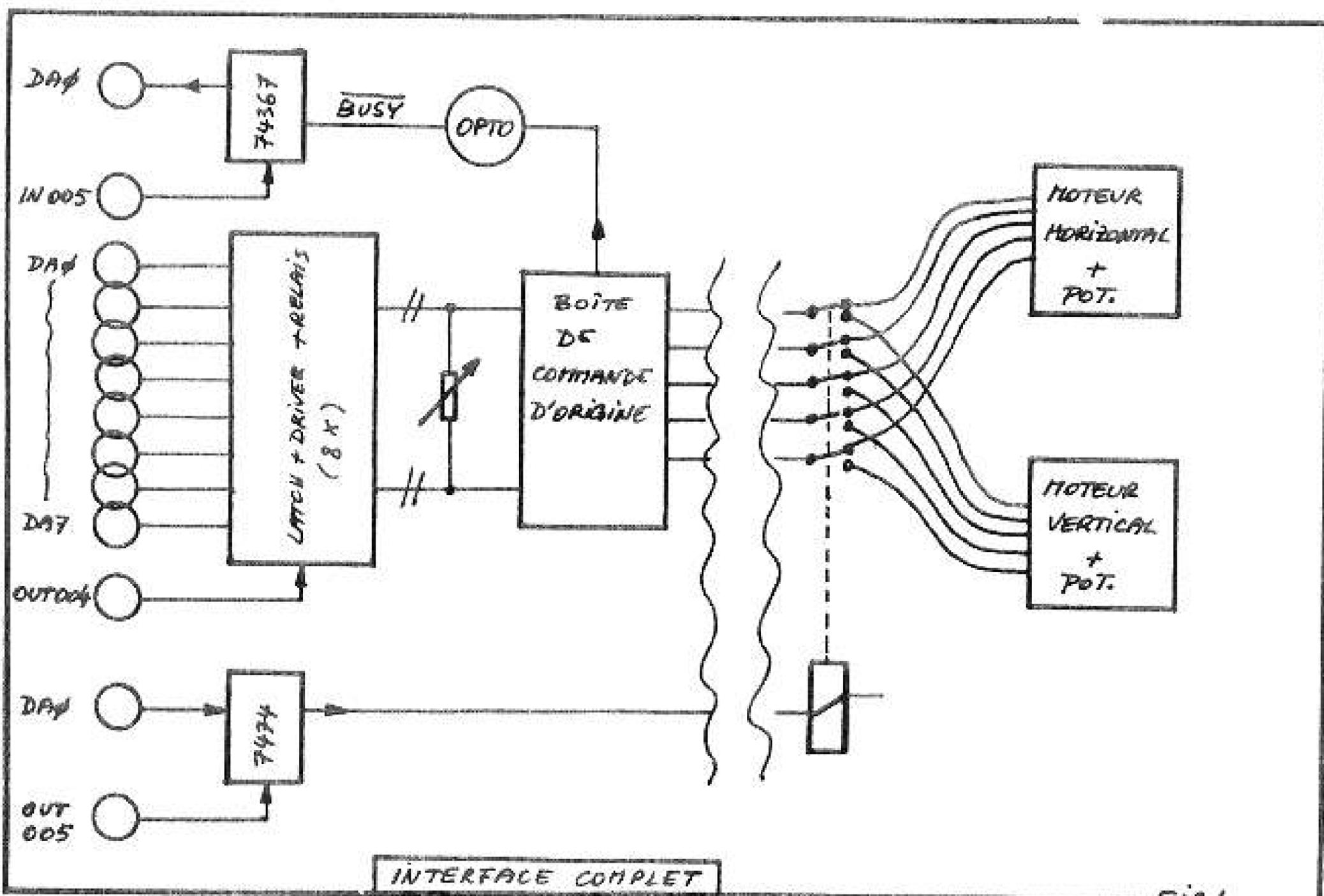


ANTENNA TRACKING  
 INTERFACE  
 MAI 1980 - VON -

FIG 3

## Le microcomputer

Comme vous l'aurez deviné, il utilise le MUBUS bus, du GESO, également utilisé par le Dauphin, Smaky, etc...). Le microprocesseur est un 8080, avec 3k de RAM, 8k d'EPROM, modem et divers interfaces. L'interface "rotor", comme vous pouvez le constater, est très simple et pourra être adapté à n'importe-quel système. Dans la version définitive (est-ce que ça existe pour un chercheur le mot "définitif"?...), le microordinateur sera réduit au minimum et incorporé dans la boîte de commande. Les prints sont déjà sur la planche à dessin.



## Le software

Il est petit car les calculs ne sont pas faits en temps réel. Voilà donc encore une amélioration à faire: faire le calcul en temps réel en basic ce qui éviterait d'utiliser une table fixe comme je le fais. Malheureusement, mon système n'est pas encore équipé d'un programme de langage évolué et, comme tous les calculs font appel aux fonctions trigonométriques (arcsin, arccos, etc...) il est quasiment impossible de les faire en langage machine à moins d'y passer le restant de sa vie...

Soit dit en passant, il est tout-à-fait possible de travailler en langage machine sans assembleur. Bien-sûr, c'est un peu "faire de la mécanique avec une lime" (Pr Nicoud dixit !) mais cela permet de connaître à fond les instructions de son microprocesseur. Le 8080 est très facile pour ce genre de programmation car il n'a pas de type d'adressage compliqué. Ce qui serait un inconvénient pour

autre application est ici un avantage. A cet égard, le 8080 est donc idéal pour débiter. Nous reparlerons peut-être dans un autre article et dans un tas de tasses de café... de cette manière de programmer.

Pour en revenir à la programmation du système, il se fait en trois étapes:

1. Calculer l'heure et la longitude du passage à l'équateur pour chaque orbite utilisable. Ce programme a été décrit dans l'article précédent.
2. Calculer les directions site et azimut tout au long de l'orbite, de minute en minute, et ceci pour tous les 5 degrés de longitude. Je n'ai pas fait ce programme moi-même pour l'instant. Voici celui réalisé par Jean-François Zurcher, HB9MCZ.
3. Les valeurs obtenues avec le programme ci-dessus sont entrées dans des eeproms que le programme parcourt. Il suffit ensuite de donner la longitude au clavier et de starter le processus à l'heure du passage à l'équateur du satellite. Ensuite les antennes tournent toutes seules.

```

10 PRINT "PROGRAMME DE L'HEURE, MINUTE, SECONDE"
20 PRINT ""
30 PRINT ""
40 Qn=46.45  Latitude du PNT en degrés Nord
50 Qw=353.1  Longitude en degrés Ouest du PNT
60 Ew=longitude en degrés au moment de passage à l'équateur
70 T=nombre de minutes après le passage à l'équateur
80 R=6378  Rayon terrestre
90 H=1460  altitude d'oscar ?
100 A=101.701  inclinaison d'oscar ?
110 P=114.943980  période d'oscar ?
120 DEG
130 PRINT LIN(1)
140 Heq=0
150 FOR Ew=0 TO 360 STEP 5
160 IF (Ew>29) AND (Ew<120) THEN Qw=120
161 IF FRACT(Ew/3)≠0 THEN Qw=0
162 GOTO 170
163 PRINT CHR$(12)
170 PRINT "ANGLE D'INTERFERENCE: ",Ew;" DEGRES"
180 PRINT LIN(2)
190 PRINT " T [min]";TAB(40);"AZIMUT";TAB(41);"ELEVATION"
200 FOR T=4 TO 54 STEP 1
210 K=360*(T/P)  calcul
220 Sn=ASN(SIN(A)*SIN(K))  ldu
230 Sw=Ew+ACS(COS(K)*COS(Qn)+1)  lssr
240 V=Sw-Qw
250 C=ACS(SIN(Sn)*SIN(Qn)+COS(Sn)*COS(V))
260 B=ACS(COS(Sn)-SIN(Sn)*COS(Qn)+COS(Qn)*SIN(V))
270 S0=SIN(V)
280 IF S0=0 THEN Az=B
290 IF S0<0 THEN Az=360-B
300 E1=ATN(COS(C)-R/(R+H)*SIN(C))
310 IF E1<0 THEN GOTO 330
320 PRINT T;TAB(20);DROUND(Az,3);TAB(40);DROUND(E1,3)
330 NEXT T
340 PRINT LIN(2)
350 NEXT Ew
360 END

```

HB9MCZ

Il est bien clair que le microcomputer doit être équipé d'une horloge en temps réel pour avoir des temps exacts. Dans mon cas, je divise le 50 Hz du réseau par 50 pour générer un interrupt par seconde. Il suffit alors d'en compter 60 pour démarrer l'orientation suivante.

Déroulement simplifié du programme:

- chaque seconde, la rtc décrémente la loc rtc, initialisée à 60
- si loc rtc=0, ré-initialisation à 60 et début rotation
- incrémente le pointeur de table. si contenu=0 =fin de passage = stop
- prend la valeur de l'axe vertical et horizontal dans la table et les met dans deux registres.
- contrôle que ces deux valeurs soient comprises dans certaines limites de sécurité afin d'éviter les mouvements dangereux (par exemple faire pointer les antennes vers le sol). Si hors tolérances, les moteurs ne bougent pas. Si tout ok:
- actionne moteur vertical après avoir commuté
- attend retour du busy + 1.5 seconde de délai supplémentaire
- commute en horizontal et même processus que le moteur vertical
- retour en position d'attente de la prochaine minute (pendant tout cela, la rtc (real time clock) fonctionne et le programme contrôle sans arrêt que la loc rtc n'a pas atteint le zéro)

Dans sa version actuelle, le système doit être programmé avant chaque passage de satellite, c'est-à-dire qu'il faut lui donner la table contenant les positions pour le passage concerné. Dans la version suivante, toutes les tables seront en eeprom et il suffira alors de ne donner que la longitude à l'équateur du satellite.

La table se présente de la façon suivante:

```
10008 : AAA      valeur azimut à la minute 1
        SSS      "      site           "
        AAA      "      azimut        "      2
        SSS      "      site           "      2
        ...      etc...
        000      fin de passage
```

ANGLE D'INTERSECTION À L'ÉQUATEUR: 295 DEGRÉS

T (min)	AZIMUT	ELEVATION
12	89.1	+14.1
13	85.2	2.56
14	80.8	4.98
15	75.9	7.34
16	70.2	9.53
17	63.9	11.6
18	56.9	13.3
19	49.3	14.5
20	41.3	15.1
21	33.2	15.1
22	25.3	14.4
23	17.8	13.1
24	10.8	11.4
25	4.61	9.32
26	359	7.05
27	354	4.67
28	350	2.24

HB97C2

Pour faire la table définitive, il suffit de transformer les degrés ci-dessus en mots binaires correspondant au byte à donner à l'interface pour que l'antenne aille dans la direction voulue. Cette transformation est nécessaire car la caractéristique de transfert totale du système n'est pas linéaire, spécialement à cause des

potentiomètres. A la mise en service de la commande, il faut donc mesurer la correspondance mot binaire-direction de l'antenne, et ceci dans les 2 axes. Nous saurons donc quelle valeur binaire doit donner le  $\mu P$  pour que le moteur tourne dans la bonne direction.

### Améliorations

Je suis en train de refaire un hardware spécifique qui permettra d'avoir une commande indépendante de mon microcomputer. Cette solution est de loin préférable à la situation actuelle qui voit de grosses boîtes connectées entre elles par de nombreux câbles, système trop compliqué, trop gros et générateur/récepteur de parasites hf.

Système spécifique donc, avec un petit clavier octal, affichage 7 segments, le tout devant être incorporé dans la boîte actuelle qui contient l'interface. Une horloge avec batterie back-up permettrait un calcul en temps réel à l'aide de programmes résidents. Il suffirait alors de donner le nom du satellite pour que le programme calcule l'heure du prochain passage et donne, en temps réel, le pointage des aériens. ...le rêve...

A l'heure où paraîtront ces lignes, nous saurons si le lancement d'OSCAR 9 à l'aide du vol Ariane L02 sera réussi. Il sera alors temps de parler de ce super satellite radioamateur, le premier de la phase III.

Michel Vonlanthen, HB9AFO, Préverenges

