

Calcul de Q:

Description du programme "QCALCUL" codé en Fortran

C. Metzger

1. Introduction
2. Equipement
  - Ordinateur et périphérique
  - Equipement CAMAC
3. Description du programme
  - Guide de l'utilisateur
  - Fonctions et sous-routines
  - Source et chargement du programme

## 1. Introduction

La réalisation du système de mesure "Q-CALCUL" a été entreprise en 1976 sur une proposition de K. Schindl.<sup>1)</sup>

La première version du programme fut emplantée sur un PDP 11/45, alors ordinateur d'exploitation du dispositif<sup>2)</sup> de saisie et de traitement des données du CPS. A la suite d'améliorations apportées à ce dispositif, le PDP 11/45 fut remplacé par un Nord-10 (Temporary Beam Measurement Computer) et le programme fut adapté et transformé pour utiliser au mieux les avantages offerts par les terminaux graphiques du nouveau complexe. Cette deuxième version du programme, "QIFID" sous "START-APPLICATION"\* , est encore à la disposition des utilisateurs.

Après quelques mois d'utilisation, on s'aperçut que la validité de l'algorithme utilisé se limitait à la partie centrale de la zone accessible du diagramme de Q. Il fut donc proposé d'écrire une nouvelle version du programme qui permette de comparer différents algorithmes.

L'objet principal de cette note est la description de cette troisième version du programme.

## 2. Equipement

Le complexe T-BMC est un dispositif, de saisie et de traitement de données, indépendant du système de contrôle du CPS. Son but est de permettre la mise au point de nouveaux équipements de mesure et de procéder à des expériences ou des mesures de caractère temporaire sans perturber l'opération normale des accélérateurs.

### - Ordinateur et périphérique

Le pilier de l'installation est un Nord 10S dans une configuration

---

\* START-APPLICATION est un programme d'appel qui permet d'appeler les programmes d'application depuis n'importe quel visuel de l'installation.

dite "Standard"<sup>3)</sup> dont les éléments sont représentés sur le schéma du dispositif (fig. 1).

Côté logiciel, cet ordinateur dispose d'un système d'exploitation SINTRAN III, du FORTRAN comme langage principal, du NODAL et du Nord-PL comme langage standard. Ces logiciels sont compatibles avec ceux de l'ordinateur de développement PRDEV.

L'installation (fig. 1) dispose d'une imprimante série avec clavier, d'un visuel "Newbury" pour le chargement et la maintenance du logiciel, et de deux visuels graphiques TEKTRONIX 4014 pour l'utilisation en salle de contrôle (MCR) et dans la salle d'observation du BR (BOR). Les visuels graphiques sont munis de:

- un module EGM (option 34: Enhanced Graphics Module) portant le nombre de points adressables à 4096 x 4096.
- une mémoire interactive (CM018-0120-00: Interactive Buffer) permettant de visualiser, en mode rafraîchi sans scintillement de l'image, 160 caractères ou 480 cm de vecteurs.
- un copieur thermique (TEKTRONIX 4031: Hardcopy unit) pour copier les images ou le texte affiché sur l'écran.

Pour une utilisation rationnelle de ces visuels, une librairie graphique (source en FORTRAN IV) contenant les trois ensembles de routines suivants:

- PLOT 10 "Terminal Control System (TCS)".
- PLOT 10 "Advanced Graphing-II (AGII)"
- PLOT 10 "Refresh Card Software"

a été installée dans le système. Quelques routines de l'ensemble AGII, spécialisées pour des problèmes de gestion, n'ont pas été implantées.

- Equipement CAMAC

Le système d'acquisition de cette installation est une boucle CAMAC série. La structure et les modules adoptés sont décrits dans le

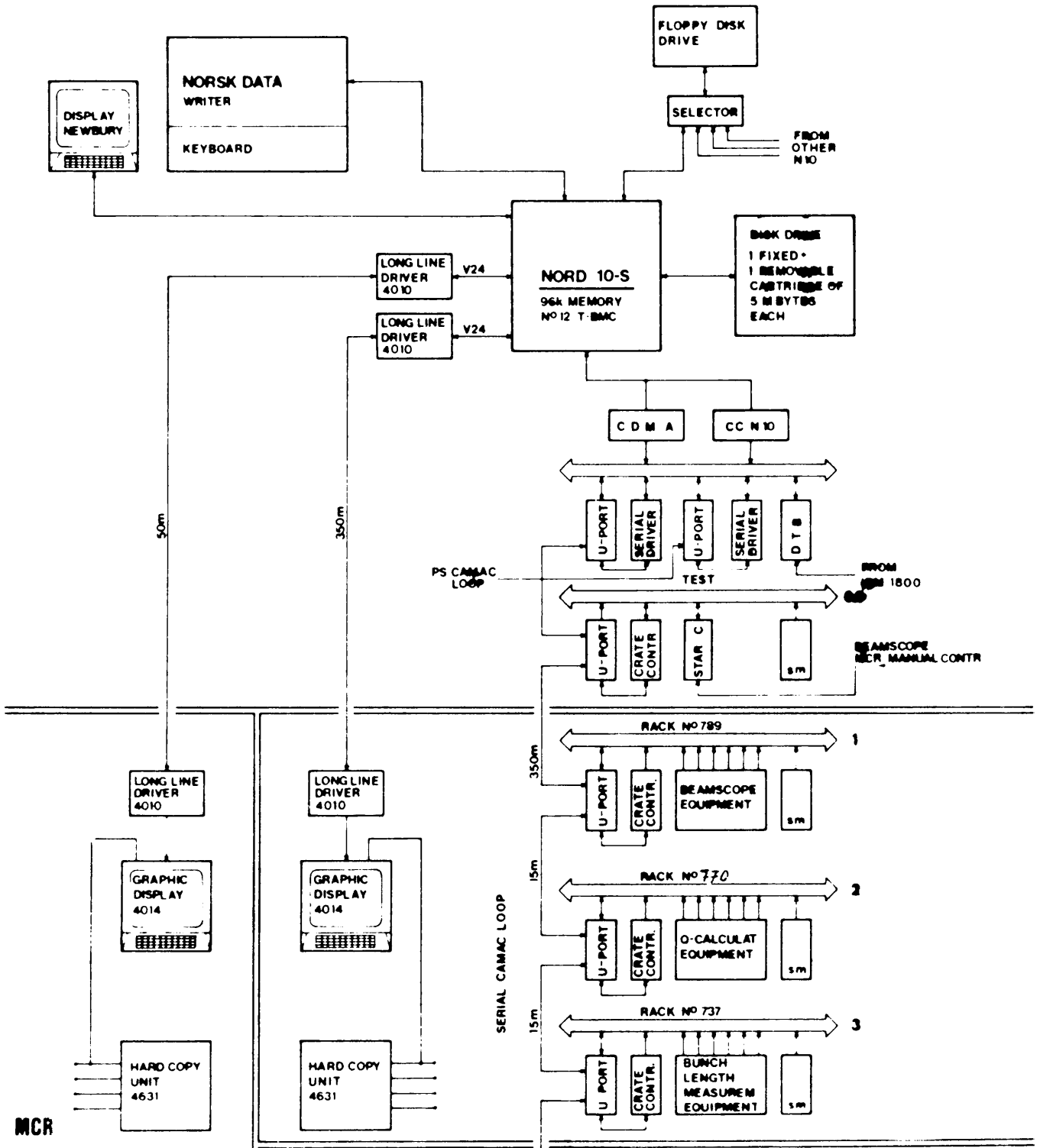


fig. 1. Schéma du dispositif de saisie et traitement des données, T-BMC

"Interface Handbook"<sup>4</sup>). Soulignons que la modularité du standard CAMAC répond parfaitement aux besoins des utilisateurs du T-BMC. Elle leur permet de modifier rapidement les équipements sans perturber l'exploitation de l'installation.

Les modules CAMAC ainsi que l'interface spécifique du "Q-CALCUL" sont installés dans le châssis No. 2 de la boucle, en haut de la baie 770 du BOR. Modules et interfaces correspondent aux spécifications initiales<sup>1</sup>) à l'exception du sélecteur de ligne de programme. Ce module non-standard (appendice A), positionné dans les créneaux 2 et 3 du châssis (fig. 2) permet de déclencher les mesures en fonction de la ligne de programme sélectionnée par l'utilisateur.

### 3. Description du programme

#### 3.1 Guide de l'utilisateur

Le nom du programme est QCALCUL. Actuellement il ne peut être appelé que du visuel graphique No. 9 de la MCR.

##### 3.1.1 Séquence d'appel

Action de l'utilisateur	Réponse de l'ordinateur
<u>presser</u> ESC	ENTER
<u>taper</u> T	PASSWORD
<u>presser</u> RETURN	(a)
<u>taper</u> RT QCALCUL	(a)
<u>taper</u> LOG	(Le programme est chargé dans la mémoire active)

##### 3.1.2 Sélection des paramètres

Question de l'ordinateur	Réponse de l'utilisateur
IPL (1 to 8) =	un entier de 1 à 8
CLOCK (B- or D-TRAIN) =	B ou D
DELAY =	un entier de 1 à 6500 (unité B ou D)

Serial Crate Controller K-3952	25
	24
U-Port Adapter K-3936	23
	22
	21
	20
	19
	18
	17
	16
	15
Memory (2) 256 Words SPS 2124	14
Memory (1) 256 Words SPS 2124	13
Preset 2 MPS 2001	12
Preset 1 MPS 2001	11
Specific Interface	10
	9
	8
	7
A/D Converter 80159 CC	6
	5
Multiplexer 80124 CC	4
Program Line Selector	3
	2
Data Way Service Module	1

fig. 2. Disposition des modules dans le châssis CAMAC No. 2 de la boucle série T-BMC.

Sélection des paramètres (suite)

<u>Question de l'ordinateur</u>	<u>Réponse de l'utilisateur</u>
DT1 =	un entier de 1 à 600 (unité B ou D)
DT2 =	un entier de 1 à 600 (unité B ou D)
RING (1 to 4) =	un entier de 1 à 4
NUMBER OF SHOT N =	un entier de 1 à 32767

Lorsqu'une réponse ambiguë est détectée , le programme affiche le message

\*\*\* AMBIGUOUS COMMAND \*\*\* .

Lorsqu'une réponse est en-dehors des limites acceptables, le programme affiche le message

\*\*\* INVALID CHOICE \*\*\*

Dans les deux cas la question est reposée.

3.1.3 Liste des fonctions du programme

Lorsque la séquence de sélection des paramètres est terminée, les fonctions suivantes peuvent être appelées dès que le signe "\*\*\*" (double étoiles) apparaît sur l'écran.

<u>Nom d'appel</u>	<u>Action</u>
<u>Fonction de Sélection</u>	
IPL	Sélectionne la ligne de programme
RING	Sélectionne l'anneau
CLOCK	Sélectionne l'horloge (train B ou D)
DELAY	Fixe le delai entre WIO et le déclenchement de la mesure
DT1	Fixe l'intervalle entre les 16 premiers points de mesure
DT2	Fixe l'intervalle entre les 16 autres points de mesure
NSTAT	Fixe le nombre de mesure à effectuer (moyennes et variances)

Fonction de Calcul et d'Affichage

POLYNOM	Effectue NSTAT mesures et affiche les résultats des calculs (algorithme: approximation polynômiale)
---------	---

Nom d'appel	Action
<u>Fonction de Calcul et d'Affichage (suite)</u>	
LATTICE	Effectue NSTAT mesures et affiche les resultats des calculs (algorithme: avance de phase bêtatronique).
QVAL	Affiche les valeurs moyennes et les variances des $Q_H$ et $Q_V$ calculés pour POLYNOM ou LATTICE

<u>Fonction de Test</u>	
WCURRENT	Affiche les dernières mesures de IB,IF,ID
WIFCOR	Affiche les dernières mesures de correction IF
WIDCOR	Affiche les dernières mesures de correction ID
WBDLCOR	Affiche les dernières mesures de correction intégral BDL
TEST	Effectue une mesure, affiche et copie les tables suivantes:  a) contenu des mémoires tampons en octal b) table des courants IB,IF,ID c) table des courants de correction IF (DIF) d) table des courants de correction ID (DID) e) table des courants de correction IB (BDL) f) table des valeurs de Q g) contenu des mémoires tampons sans normalisation
HTEST	Effectue une mesure et affiche le contenu des mémoires tampons en octal.

Sortie du Programme

STOP	Met le terminal en mode alphanumérique et exécute l'instruction FORTRAN STOP
------	--

La routine de recherche des noms accepte les abréviations, exemple:

HTEST, HTES, HT ou simplement H.

Lorsqu'un groupe de noms commence par un ou plusieurs caractères identiques, la routine de recherche les reconnaît par le premier caractère différent, ex:

WCURRENT → WC  
WIFCOR → WIF  
WIDCOR → WID



#### 3.1.4 STOP

Comme le mode d'appel se reboucle sur lui-même il est  
IMPORTANT  
de ne pas oublier de terminer le programme en exécutant un STOP.

### 3.2 Programme, sous-routines et listes des instructions

#### 3.2.1 Programme principal

Le programme "QCALCUL" est structuré de manière à donner à l'utilisateur la possibilité de modifier la valeur des paramètres en cours de travail, de choisir entre différents algorithmes de calcul et de faire des tests pour s'assurer du bon fonctionnement de l'installation.

Le programme principal n'est qu'une séquence d'appel de différentes routines ou groupe de routines exécutant une fonction donnée. Lorsqu'une fonction est terminée, la routine d'interrogation et d'identification est rappelée automatiquement. L'organigramme et la liste des instructions Fortran montrent plus clairement l'organisation du programme.

#### 3.2.2 Initialisation

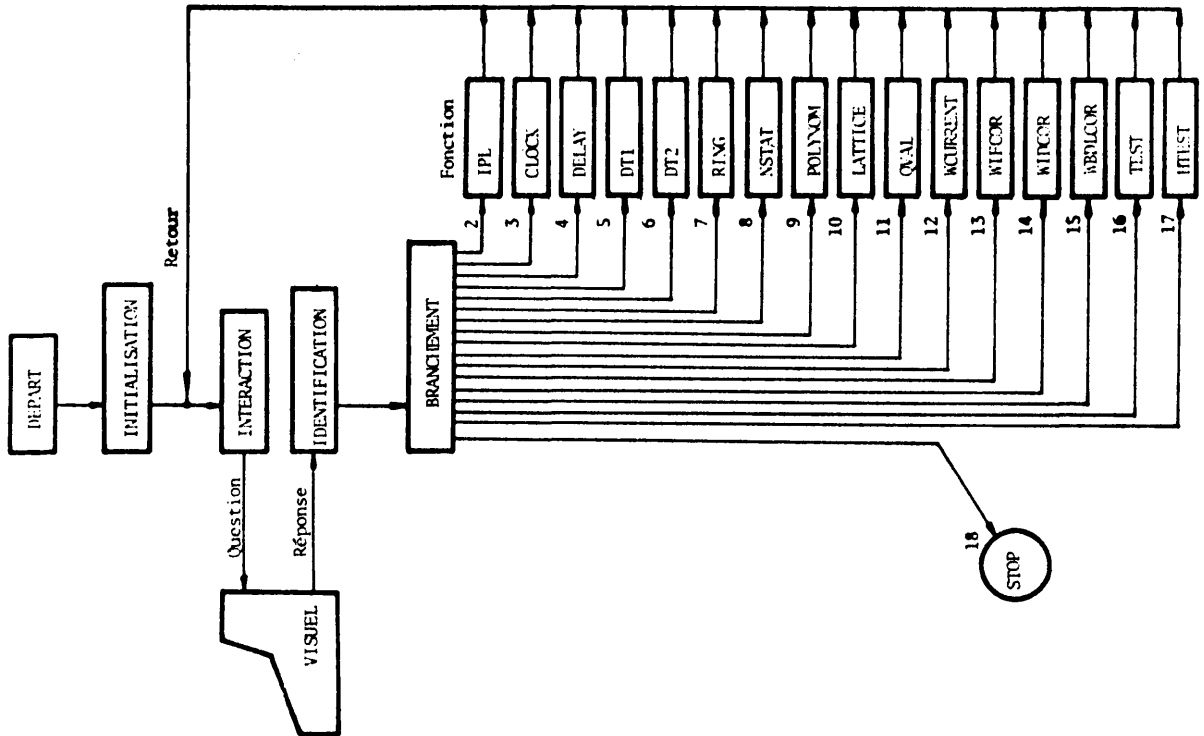
Le programme principal démarre en effectuant les trois séquences suivantes:

- Connexion des moyens d'entré/sortie par les routines du système (Sintran III). Pour cette application seul le visuel graphique 9 est connecté.
- Initialisation du visuel graphique par des routines de la librairie graphique.
- Sélection des paramètres et initialisation des modules CAMAC par des routines de ce programme, appelées par la séquence d'instruction SET.

```

1* C-----PROGRAM GCALCUL-----
2* PROGRAM GCALCUL,70 METZGER CLAUDE 15/06/1979
3* ORIGINAL VERSION: METZGER CLAUDE
4* COMMON/IODEV/INP,IOUTP,LAM
5* C-----CONNECTION TO I/O DEVICES
6* INP=9
7* IOUTP=9
8* CALL RESV (IOUTP,1,0)
9* CALL RESV (INP,0,0)
10* C-----INITIALIZE 4014 TERMINAL
11* CALL INIT (960)
12* CALL RINIT
13* CALL CINIT
14* CALL TER4 (3,4096)
15* CALL DMINDO (0,1200,0,1100,.)
16* CALL CHRSTZ (3)
17* C-----INITIALIZE CAMAC MODULES
18* CALL SET
19* C-----BRANCH TO SUBROUTINES FOR INTERACTIVE USE
20* NAIG=0
21* 1 CALL INTER (NAIG)
22* GO TO NAIG
23* 2 CALL SETPL
24* GO TO 1
25* 3 CALL TRAIM
26* GO TO 1
27* 4 CALL DELAY
28* GO TO 1
29* 5 CALL SAMPLE1
30* GO TO 1
31* 6 CALL SAMPLE2
32* GO TO 1
33* 7 CALL RING
34* GO TO 1
35* 8 CALL NSTAT
36* GO TO 1
37* 9 CALL DISPLAY (1)
38* GO TO 1
39* 10 CALL DISPLAY (2)
40* GO TO 1
41* 11 CALL WMEAN
42* GO TO 1
43* 12 CALL WRCUR (1)
44* GO TO 1
45* 13 CALL WRCUR (2)
46* GO TO 1
47* 14 CALL WRCUR (3)
48* GO TO 1
49* 15 CALL WFCUR (4)
50* GO TO 1
51* 16 CALL TEST
52* GO TO 1
53* 17 CALL HTEST
54* GO TO 1
55* 18 CALL BYE
56* END

```



```
57* C-----SURROUTINE SET-----
58* SUBROUTINE SET
59* COMMON/SETTING/LOOP,IPL,KLOCK,IT,IT1,IT2,NMEASUR,NRING
60* COMMON/VAL/VALUE
61* COMMON/FLAG/JFLAG
62* CHARACTER VALUE*10,KLOCK*7
63* LOOP=1
64* C---ENABLE LAM-----
65* CALL CAMA (LOOP,2,14,0,26,1,0)
66* 1 VALUE=' '
67* IFLAG=0
68* CALL SETIPL
69* IF(IFLAG.EQ.1) GO TO 1
70* 2 VALUE=' '
71* JFLAG=0
72* CALL TRAIN
73* IF(JFLAG.EQ.1) GO TO 2
74* 3 VALUE=' '
75* JFLAG=0
76* CALL DELAY
77* IF (JFLAG.EQ.1) GO TO 3
78* 4 VALUE=' '
79* JFLAG=0
80* CALL SAMPLE1
81* IF (JFLAG.EQ.1) GO TO 4
82* 5 VALUE=' '
83* JFLAG=0
84* CALL SAMPLE2
85* IF (JFLAG.EQ.1) GO TO 5
86* 6 VALUE=' '
87* JFLAG=0
88* CALL RING
89* IF (JFLAG.EQ.1) GO TO 6
90* 7 VALUE=' '
91* JFLAG=0
92* CALL NSTAT
93* IF (JFLAG.EQ.1) GO TO 7
94* RETURN
95* END
-----
```

### 3.2.3 Routine d'appel

Cette routine questionne l'utilisateur, puis compare la variable entrée avec une liste de noms (variables de 8 caractères) et la traduit en code numérique pour l'instruction de branchement 22 du programme principal. L'algorithme choisi:

```
Input W
M=0
DO I=1, Limit
IF W=DATA(I) or DATA(I)(ABREVIATION),N(I)=I
IF N(I)≠0, M=M+1, NAIG=N(I)
End DO
IF M=1
Then return NAIG
Else display message, repeat
End
```

est simple mais suffisamment rapide vu le nombre limité de noms contenus dans la table d'identification DATA/A/.

```

96* C-----SUBROUTINE INTERP-----
97*      SUBROUTINE INTERP (NAIG)
98*      COMMON/IODEV/INP,IOUPT,LAM
99*      COMMON/VAL/VALUE
100*     DIMENSION N(17)
101*     CHARACTER *8,A(17),W,VALUE*10
102*     DATA A/'IPL','CLOCK','DELAY','DT1','DT2','RING','NSTAT',
103*           * 'POLYNOM','LATTICE','QVAL','WCURRENT','WIFCOR','WIDCOR',
104*           * 'WDLCOR','TEST','HTEST','STOP'/
105*     CALL ANMODE
106*     LIMIT=17
107*     1   DO 5 I=1,LIMIT
108*     5   N(I)=0
109*     NAIG=0
110*     M=0
111*     VALUE=' '
112*     WRITE (IOUPT,1000)
113*     INPUT (INP) W
114*     DO 10 I=1,LIMIT
115*     IF (W.EQ.A(I)(1:8).OR.W.EQ.A(I)(1:1)) N(I)=I
116*     IF (W.EQ.A(I)(1:7).OR.W.EQ.A(I)(1:2)) N(I)=I
117*     IF (W.EQ.A(I)(1:6).OR.W.EQ.A(I)(1:3)) N(I)=I
118*     IF (W.EQ.A(I)(1:5).OR.W.EQ.A(I)(1:4)) N(I)=I
119*     IF (N(I).NE.0) M=M+1
120*     10  IF (N(I).NE.0) NAIG=N(I)+1
121*     IF (M.EQ.1) GO TO 30
122*     CALL INVWORD
123*     GO TO 1
124*     30  RETURN
125*     1000 FORMAT (/,15***)
126*     END
-----

```

Remarque. Précédemment, l'instruction 113 était INPUT(INP) W, VALUE ce qui permettait de passer, par le truchement d'un common, la variable VALUE à d'autres routines afin d'accélérer le processus de sélection; Exemple: Sélection de l'anneau 4.

```

** RING,4 ✓ équivalait à
** RING ✓
   RING (1 to 4) = 4 ✓

```

Malheureusement, l'instruction FORTRAN "Input" impose un double "Return" (✓) lorsque la variable "value" est omise (\*\*RING ✓) et de ce fait introduit une indésirable confusion. En conséquence, la variable "value" a été supprimée dans l'instruction Input, mais la structure de passage de la variable a été conservée en vue d'une future amélioration (remplacement de l'instruction Input par une routine de lecture).

### 3.2.4 Routine de sélection de paramètre

Sept paramètres peuvent être modifiés indépendamment en cours d'utilisation. Ils sont mémorisés dans le COMMON/SETTING/ (voir liste des COMMON, page ) qui permet de passer ces paramètres aux quinze routines qui les utilisent.

Les paramètres sont entrés sous la forme d'une chaîne de dix caractères (character String). Une routine les convertis, si nécessaire, en nombre entier ou en réel. Le paramètre "RETURN" ou "R" permet de sortie des routines sans modifier la liste "SETTING".

```

127* C-----SUBROUTINE SETIPL-----
128*     SUBROUTINE SETIPL
129*     COMMON/IODEV/INP,IOUTP,LAM
130*     COMMON/SETTING/LOOP,IPL,KLOCK,IT,IT1,IT2,NMEASUR,NRING
131*     COMMON/VAL/VALUE
132*     CHARACTER VALUE*10,KLOCK*7
133*     CALL CONVER (VALUE,IPL,AA,IFLAG)
134*     GO TO (40,10,15,30,10),IFLAG
135* 10   CALL INVWORD
136* 15   WRITE (IOUTP,1000)
137*     INPUT (INP) VALUE
138*     CALL CONVER (VALUE,IPL,AA,IFLAG)
139*     GO TO (40,10,15,30,10),IFLAG
140* 30   IF (VALUE(1:1).EQ.'R') RETURN
141*     GO TO 10
142* 40   IF (IPL.GT.0.AND.IPL.LT.9) GO TO 50
143*     CALL INVALID
144*     GO TO 15
145* 50   RETURN
146* 1000 FORMAT (/,,'S IPL (1 TO 8) =')
147*     END

```

```

148* C-----SUBROUTINE RING-----
149*     SUBROUTINE RING
150*     COMMON/IODEV/INP,IOUTP,LAM
151*     COMMON/SETTING/LOOP,IPL,KLOCK,IT,IT1,IT2,NMEASUR,NRING
152*     COMMON/VAL/VALUE
153*     CHARACTER VALUE*10,KLOCK*7
154*     CALL CONVER (VALUE,NRING,AA,IFLAG)
155*     GO TO (40,10,15,30,10),IFLAG
156* 10   CALL INVWORD
157* 15   WRITE (IOUTP,1000)
158*     INPUT (INP) VALUE
159*     CALL CONVER (VALUE,NRING,AA,IFLAG)
160*     GO TO (40,10,15,30,10),IFLAG
161* 30   IF (VALUE(1:1).EQ.'R') RETURN
162*     GO TO 10
163* 40   IF (NRING.GT.0.AND.NRING.LE.4) GO TO 50
164*     CALL INVALID
165* 50   RETURN
166* 1000 FORMAT (/,,'S RING (1 TO 4) =')
167*     END

```

```

168* C-----SUBROUTINE TRAIN-----
169*     SUBROUTINE TRAIN
170*     COMMON/IODEV/INP,IOUTP,LAM
171*     COMMON/SETTING/LOOP,IPL,KLOCK,IT,IT1,IT2,NMEASUR,NRING
172*     COMMON/VAL/VALUE
173*     CHARACTER VALUE*10,KLOCK*7,CL*7,A(3)*7
174*     DIMENSION N(3)
175*     DATA A/'B-TRAIN','D-TRAIN','RETURN'/
176*     IF (VALUE.NE.' ') GO TO 10
177* 10   WRITE (IOUTP,1000)
178*     INPUT (INP) VALUE
179* 10   DO 15 I=1,3
180* 15   N(I)=0
181*     M=0
182*     CL=VALUE(1:7)
183*     DO 20 I=1,3
184*     IF (CL.EQ.A(I).OR.CL.EQ.A(I)(1:1)) N(I)=I
185*     IF (CL.EQ.A(I)(1:6).OR.CL.EQ.A(I)(1:2)) N(I)=I
186*     IF (CL.EQ.A(I)(1:5).OR.CL.EQ.A(I)(1:3).OR.CL.EQ.A(I)(1:4)) N(I)=I
187*     IF (N(I).GT.0) M=M+1
188* 20   IF (N(I).NE.0) NB=N(I)
189*     IF (M.EQ.1) GO TO 30
190*     CALL INVWORD
191*     GO TO 1
192* 30   IF (NB.EQ.3) RETURN
193*     KLOCK=A(NB)
194*     CALL CAMA (LOOP,2,9,0,16,1,(NB-1))
195*     RETURN
196* 1000 FORMAT (/,,'S CLOCK (B- OR D-TRAIN) =')
197*     END

```

```
198* C-----SUBROUTINE DELAY-----
199* SUBROUTINE DELAY
200* COMMON/IODEV/INP,IOUTP,LAM
201* COMMON/SETTING/LOOP,IPL,KLOCK,IT,IT1,IT2,NMEASUR,NRING
202* COMMON/VAL/VALUE
203* CHARACTER VALUE*10,KLOCK*7
204* CALL CONVER (VALUE,IT,AA,IFLAG)
205* GO TO (40,10,15,30,10),IFLAG
206* 10 CALL INVWORD
207* 15 WRITE (IOUTP,1000)
208* INPUT (INP) VALUE
209* CALL CONVER (VALUE,IT,AA,IFLAG)
210* GO TO (40,10,15,30,10),IFLAG
211* 30 IF (VALUE(1:1),EQ,'R') RETURN
212* GO TO 10
213* 40 IF (IT,GE,0,OR,IT,LE,6500) GO TO 50
214* CALL INVALID
215* GO TO 15
216* 50 CALL CAMA (LOOP,2,11,0,16,1,IT)
217* CALL TIMETAB
218* RETURN
219* 1000 FORMAT (/, 'S DELAY =')
220* END
```

```
221* C-----SURROUTINE SAMPLE1-----
222* SUBROUTINE SAMPLE1
223* COMMON/IODEV/INP,IOUTP,LAM
224* COMMON/VAL/VALUE
225* COMMON/SETTING/LOOP,IPL,KLOCK,IT,IT1,IT2,NMEASUR,NRING
226* CHARACTER VALUE*10,KLOCK*7
227* CALL CONVER (VALUE,IT1,AA,IFLAG)
228* GO TO (40,10,15,30,10),IFLAG
229* 10 CALL INVWORD
230* 15 WRITE (IOUTP,1000)
231* INPUT (INP) VALUE
232* CALL CONVER (VALUE,IT1,AA,IFLAG)
233* GO TO (40,10,15,30,10),IFLAG
234* 30 IF (VALUE(1:1),EQ,'R') RETURN
235* GO TO 10
236* 40 IF (IT1,GE,0,OR,IT1,LE,600) GO TO 50
237* CALL INVALID
238* GO TO 15
239* 50 CALL CAMA (LOOP,2,11,1,16,1,IT1)
240* CALL TIMETAB
241* RETURN
242* 1000 FORMAT (/, 'S DT1 =')
243* END
```

```
244* C-----SUBROUTINE SAMPLE2-----
245* SUBROUTINE SAMPLE2
246* COMMON/IODEV/INP,IOUTP,LAM
247* COMMON/VAL/VALUE
248* COMMON/SETTING/LOOP,IPL,KLOCK,IT,IT1,IT2,NMEASUR,NRING
249* CHARACTER VALUE*10,KLOCK*7
250* CALL CONVER (VALUE,IT2,AA,IFLAG)
251* GO TO (40,10,15,30,10),IFLAG
252* 10 CALL INVWORD
253* 15 WRITE (IOUTP,1000)
254* INPUT (INP) VALUE
255* CALL CONVER (VALUE,IT2,AA,IFLAG)
256* GO TO (40,10,15,30,10),IFLAG
257* 30 IF (VALUE(1:1),EQ,'R') RETURN
258* GO TO 10
259* 40 IF (IT2,GE,0,OR,IT2,LE,600) GO TO 50
260* CALL INVALID
261* GO TO 15
262* 50 CALL CAMA (LOOP,2,12,0,16,1,IT2)
263* CALL TIMETAB
264* RETURN
265* 1000 FORMAT (/, 'S DT2 =')
266* END
```

### 3.2.5 Calcul et affichage des valeurs de Q

La routine "Display" appelle les routines de mesure, de calculs et gère l'affichage des résultats. Selon l'option "IND = 1 ou 2" (voir organigramme, les valeurs de Q sont calculées avec l'un des deux algorithmes suivants:

a) "Polynom" (sous-routine QHQP de la librairie).

Les fonctions QH et QV sont approximées par un polynôme du 3ème degré:

$$Q(t) = C_0 + C_1F(t) + C_2D(t) + \\ C_3F^2(t) + C_4D^2(t) + C_5F(t).D(t) + \\ C_6F^3(t) + C_7D^3(t) + C_8F(t)D^2(t) + C_9F^2(t)D(t)$$

Les fonctions F (focalisation) et D (defocalisation) sont calculées à partir des courants acquis selon les spécifications du logiciel<sup>1)</sup>.

$$F(t) = \frac{IF(t) + DIF(t) - BDL(t)}{IB(t) + BDL(t)}$$

$$D(t) = \frac{ID(t) + DID(t) - BDL(t)}{IB(t) + BDL(t)}$$

Remarque: Dans ce programme, les variables DIF, DID et BDL sont normalisées et tiennent compte du rapport des nombres de spires (voir réf. 1, pages 2.2 et 2.3).

Les coefficients C ont été déterminé par la méthode des moindres carrés sur les résultats théoriques du programme de simulation BOOM<sup>5)</sup>. Ces coefficients, donnés dans la table 1, permettent de calculer les valeurs de Q avec une précision relative de 1 % sur toute la zone accessible du diagramme et avec une précision supérieure à 5% dans les régions hachurées. (fig. 3).

Coefficient	K	C(K,1) Q-Horizontal	C(K,2) Q-Vertical
$C_0$	1	4.5170	4.4491
$C_1$	2	7.1533	- 7.4356
$C_2$	3	- 3.3365	14.7728
$C_3$	4	- 3.2728	- 3.4119
$C_4$	5	.1574	- 7.7892
$C_5$	6	3.2939	13.5261
$C_6$	7	9.2598	-45.1301
$C_7$	8	- 1.2794	241.6152
$C_8$	9	-12.6745	127.8616
$C_9$	10	5.7572	303.3824

Table 1: Coefficients polynômiaux

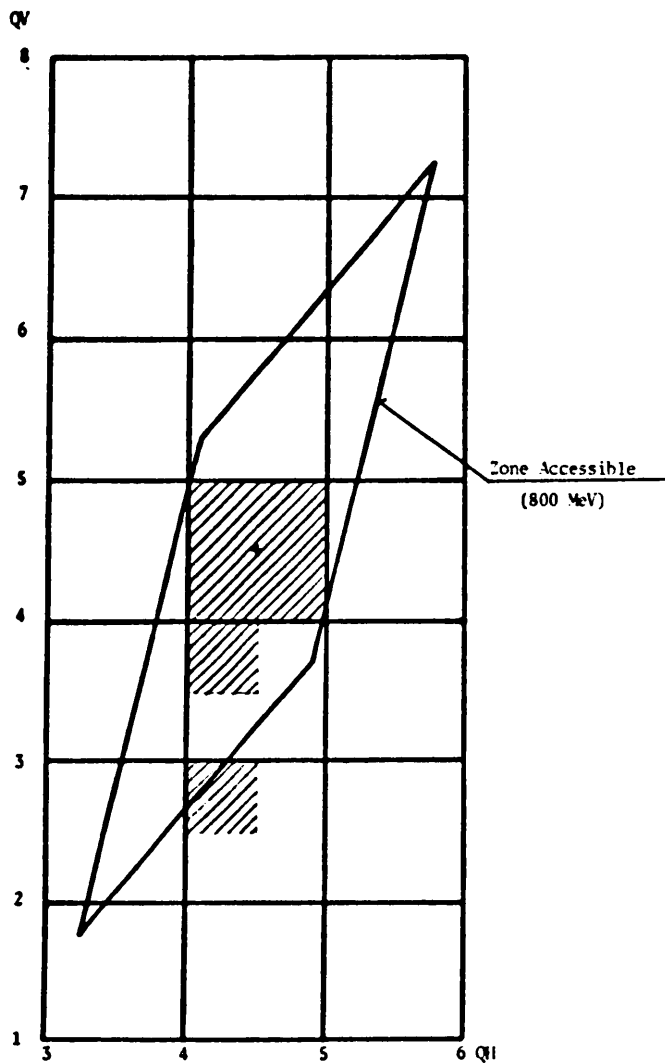
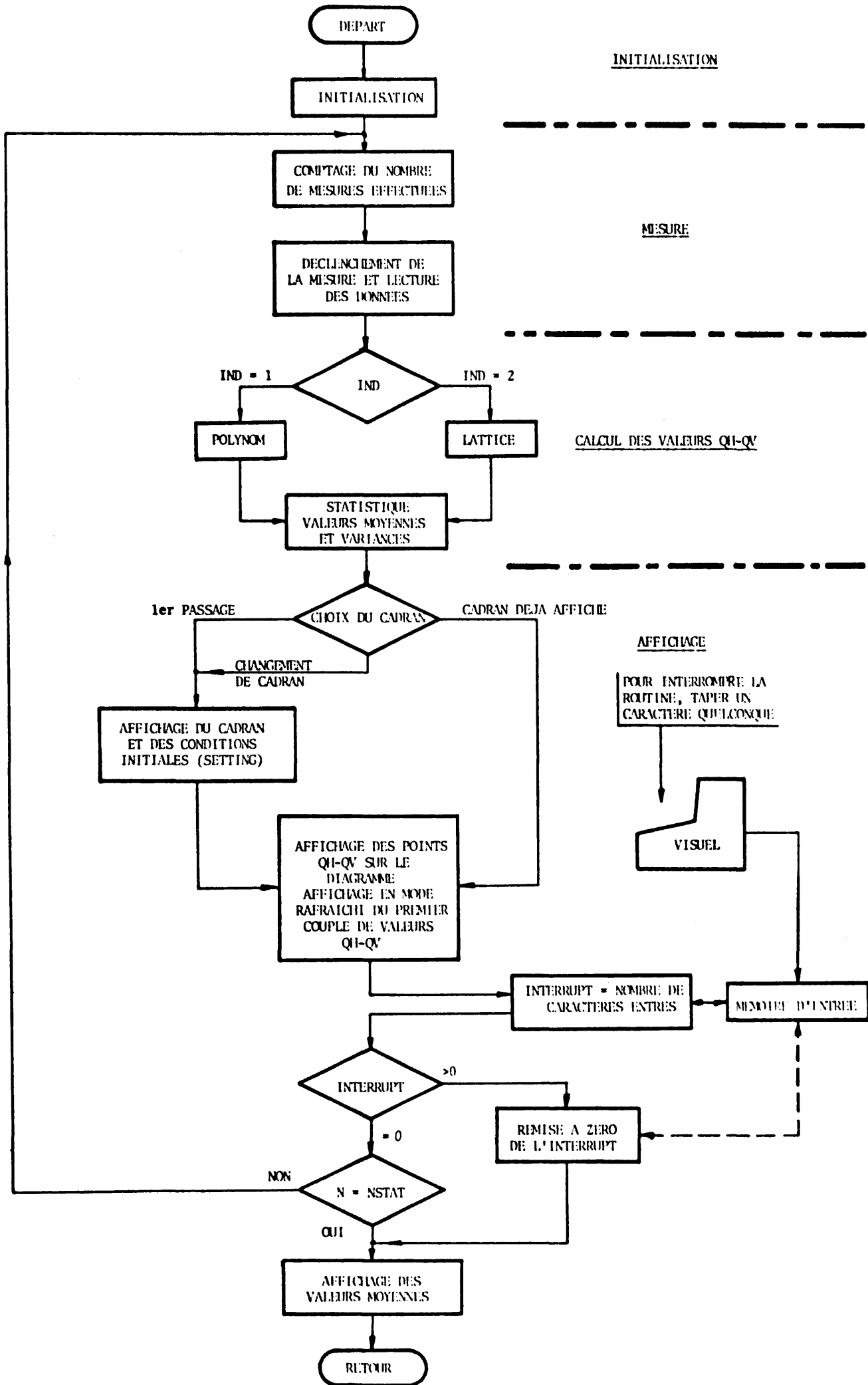


Fig. 3 : Diagramme de Q et limites à 800 MeV de la zone accessible avec les alimentations de IF, ID et les courants de correction actuels.





b) "Lattice" (subroutine BOOMQC de la librairie).

Les valeurs de Q sont calculées en fonction de l'avance de phase bêtatronique par maille<sup>6)</sup>.

$$Q = \frac{8}{\pi} \text{Arc cos} (m_{11} m_{22} + m_{12} m_{21})$$

où les coefficients  $m_{ij}$  sont ceux de la matrice de transfert de la  $\frac{1}{2}$  maille de la machine. Cet algorithme est valable par tout le domaine accessible du diagramme. Le détail des calculs ainsi que les valeurs des constantes utilisées sont donnés dans la note de J.P. Delahaye<sup>6)</sup>.

```

287* C-----SURROUTINE DISPLAY-----
288*   SUBROUTINE DISPLAY (INJ)
289*   COMMON/ICDEV/INP,IOUTP,LAM
290*   COMMON/SETTING/LOOP,IPL,KLOCK,IT,IT1,IT2,NMEASUR,NRING
291*   COMMON/QVALUE/QVAL(32,2)
292*   COMMON/CAD/STH,STV,XOR,YOR,XLARG,YLARG
293*   COMMON/STATIS/QMEAN(32,2),OSIGMA(32,2),SSQV(32,2),NESHOT
294*   COMMON/TABTIME/ITAB(32)
295*   CHARACTER KLOCK*7
296*   DATA XOR,YOR,XLARG,YLARG/40.,15.,800.,1000.,/
297* C---INITIALISATION
298*   NBSHOT=0
299*   DO 5 I=1,32
300*     DO 5 J=1,2
301*       QMEAN(I,J)=0.
302*       OSIGMA(I,J)=0.
303*     5   SSQV(I,J)=0.
304*     CALL ERASE
305* C---MEASUREMENT
306*     DO 500 N=1,NMEASUR
307*       NBSHOT=NBSHOT+1
308*       CALL READMEM (0)
309*       IF (IND.EQ.1) CALL QHGV
310*       IF (IND.EQ.2) CALL BOOMQC
311*       CALL STAT
312* C---SEARCH QHMIN AND QVMIN
313*       QHMIN=QVAL(32,1)
314*       QVMIN=QVAL(32,2)
315*       DO 10 K=1,31
316*         IF (QVAL(K,1).LT.QHMIN) QHMIN=QVAL(K,1)
317*       10 IF (QVAL(K,2).LT.QVMIN) QVMIN=QVAL(K,2)
318* C---SET DRAWING PARAMETERS
319*       STH=INT(QHMIN)
320*       STV=INT(QVMIN)
321*       PH=QHMIN-STH
322*       PV=QVMIN-STV
323*       IF (PH.GT..5) STH=STH+.5
324*       IF (PV.GT..5) STV=STV+.5
325*       CALL NEUTRL
326*       IF (NBSHOT.EQ.1) GO TO 20
327*       IF (ABS(RPH-STH).LT..4.AND.ABS(RPV-STV).LT..4) GO TO 50
328*       CALL NEWPAG
329* C---DISPLAY FRAME AND DATSET
330*   20   CALL ANMODE
331*       IF (IND.EQ.1) WRITE (IOUTP,1010)
332*       IF (IND.EQ.2) WRITE (IOUTP,1020)
333*       CALL QUAD (PH,PV)
334*       CALL DATSET (3)
335*       RPH=STH
336*       RPV=STV
337* C---DISPLAY POINTS QH-QV
338*   50   DO 100 J=1,32
339*         PX=XOR+(QVAL(J,1)-STH)*2.*XLARG
340*         PY=YOR+(QVAL(J,2)-STV)*2.*YLARG
341*       100 CALL POINTA (PX,PY)
342* C---REFRESH DISPLAY OF FIRST QH-QV POINT
343*       CALL NEUTRL
344*       CALL BUFFER
345*       CALL CHPSIZ (1)

```

```

346*      CALL MOVEA (820.,750.)
347*      CALL ANMODE
348*      WRITE (IOUTP,1006) QVAL(1,1),QVAL(1,2)
349*      CALL PFRSH
350*      CALL CHRSTZ (3)
351* C----TEST FOR INTERRUPT
352*      INTERRUPT=ISIZE(INP)
353*      IF (INTERRUPT.NE.0) GO TO 501
354*      500 CONTINUE
355* C---EMPTY INPUT BUFFER
356*      501 IF (INTERRUPT.EQ.0) GO TO 530
357*          DO 520 I=1,INTERRUPT
358*      520 ICHARC=INCH(INP)
359*      530 CALL STORE
360*          CALL NEUTRL
361*          CALL HOME
362*          CALL ANMODE
363*          WRITE (IOUTP,1000)
364*          DO 550 J=1,32
365*      550 WRITE (IOUTP,1001) ITTAB(J),(QMEAN(J,I),I=1,2)
366*          WRITE (IOUTP,1002) NBSHOT
367*          CALL HOME
368*          CALL BELL
369*          RETURN
370*      1000 FORMAT (19/,94X,'I',7X,'QH',8X,'QV',/)
371*      1001 FORMAT (90X,I6,2F10,3)
372*      1002 FORMAT (//,90X,'NR. PULSES =',I5)
373*      1006 FORMAT ('$',4X,2F8,3)
374*      1010 FORMAT (5X,'QCALCUL : POLYNOMIAL APPROXIMATION')
375*      1020 FORMAT (5X,'QCALCUL : LATTICE MATRIX')
376*      END

```

Les valeurs moyennes des QH, QV et leurs variances peuvent être obtenues sur une page séparée en appelant la routine WQMEAN (nom d'appel: QVAL) qui affiche le contenu du COMMON/ STATIS/.

```

377* C-----SUBROUTINE WQMEAN-----
378*      SUBROUTINE WQMEAN
379*      COMMON/IODEV/INP,IOUTP,LAM
380*      COMMON/QVALUE/QVAL(32,2)
381*      COMMON/STATIS/QMEAN(32,2),QSIGMA(32,2),SSQOV(32,2),NBSHOT
382*      COMMON/TABTIME/ITTAB(32)
383*      CALL ERASE
384*      CALL HOME
385*      CALL ANMODE
386*      WRITE (IOUTP,1000) NBSHOT
387*      DO 10 J=1,32
388*      10 WRITE(IOUTP,1001) ITTAB(J),(QMEAN(J,I),QSIGMA(J,I),I=1,2)
389*      CALL DATSET (2)
390*      RETURN
391*      1000 FORMAT (3/,10X,'MEAN VALUES OF CALCULATED Q ON',I6,' PULSES',
392*      * 3/,21X,'I',6X,'QH',8X,'QV',8X,'QV',8X,'QV',/)
393*      1001 FORMAT (15X,I7,2(4X,F6,3,4X,E9,3))
394*      END

```

### 3.2.6 Test et contrôle

La routine WRCUR permet d'afficher le contenu des COMMON/DATMAT/ QVAL, c'est-à-dire les valeurs des courants des aimants et les valeurs calculées des QH-QV.

Lorsque les résultats sont douteux, on peut appeler l'une des quatre pages suivantes:

<u>Nom d'appel</u>	<u>Affichage</u>
WCURRENT	IB, IF, ID en ampère
WIFCOR	Q-Strip IF en ampère
WIDCOR	Q-Strip ID en ampère
WBDLCOR	Intégrale BDL en ampère

```

395* C-----SUBROUTINE WRCUR(N)-----
396* SUBROUTINE WRCUR(N)
397* PEAL IB(32),IF(32),ID(32)
398* COMMON/IODEV/ INP,IOUPL,LAM
399* COMMON/SETTING/LOOP,IPL,KLOCK,IT,IT1,IT2,NMEASUR,NPING
400* COMMON/DATMAT/IBUF(512),DIF(32,4),RDL(32,4),DID(32,4),IR,IF,ID
401* COMMON/TABTIME/ITTAB(32)
402* COMMON/QVALUE/QVAL(32,2)
403* DIMENSION RUF(4)
404* CHARACTER KLOCK*7,A(2)*3
405* DATA A/'DIF','DID'/
406* CALL ERASE
407* CALL ANMODE
408* GO TO (10,20,30,40,60,50),N
409* C---WRITE IR,IF,ID
410* 10 WRITE (IOUPL,1010)
411* DO 15 J=1,32
412* 15 WRITE (IOUPL,1011) ITTAB(J),IB(J),IF(J),ID(J)
413* CALL DATSET(1)
414* GO TO 100
415* C---WRITE Q-SHIFT DELTA-IF
416* 20 WRITE (IOUPL,1020) A(1)
417* DO 25 J=1,32
418* DO 21 K=1,4
419* 21 RUF(K)=DIF(J,K)*2.
420* 25 WRITE (IOUPL,1021) ITTAB(J),(RUF(I),I=1,4)
421* CALL DATSET(1)
422* GO TO 100
423* C---WRITE Q-SHIFT DELTA-ID
424* 30 WRITE (IOUPL,1020) A(2)
425* DO 35 J=1,32
426* DO 31 K=1,4
427* 31 RUF(K)=DID(J,K)*2.
428* 35 WRITE (IOUPL,1021) ITTAB(J),(RUF(I),I=1,4)
429* CALL DATSET(1)
430* GO TO 100
431* C---WRITE INTEGRAL BDL
432* 40 WRITE (IOUPL,1040)
433* DO 45 J=1,32
434* DO 41 K=1,4
435* 41 RUF(K)=RDL(J,K)*6.
436* 45 WRITE (IOUPL,1021) ITTAB(J),(RUF(I),I=1,4)
437* CALL DATSET(1)
438* GO TO 100
439* C---WRITE BUFFER
440* 50 WRITE (IOUPL,1050)
441* K=0
442* DO 55 J=1,512*16
443* K=K+1
444* 55 WRITE (IOUPL,1051) ITTAB(K),(IBUF(J-1+I),I=1,16)
445* CALL DATSET(1)
446* GO TO 100
447* C---WRITE Q-VALUE
448* 60 WRITE (IOUPL,1060)
449* DO 65 J=1,32
450* 65 WRITE (IOUPL,1061) ITTAB(J),(QVAL(J,I),I=1,2)
451* CALL DATSET(2)
452* 100 RETURN
453* 1010 FORMAT(3//,10X,'VALUES OF CURRENT IB:IF:ID.',3//,21X,'T',7X,'IB',8X
454* , 'IF',8X,'ID',/)
455* 1011 FORMAT (15X,I7,3(2X,F8.2))
456* 1020 FORMAT (3//,10X,'VALUES OF QSTRIP-',A3,'-CURRENT',3//,21X,'T',4X,
457* , 'RING 1',4X,'RING 2',4X,'RING 3',4X,'RING 4',/)
458* 1021 FORMAT (15X,I7,4(2X,F8.2))
459* 1040 FORMAT (3//,10X,'VALUES OF INTEGRAL -BDL-',3//,21X,'T',4X
460* , 'RING 1',4X,'RING 2',4X,'RING 3',4X,'RING 4',/)
461* 1050 FORMAT (14//,10X,'CONTENT OF THE BUFFER =WHITOUT NORMALISATION=',
462* , 3/9X,'T',5X,'IB',5X,'IF',5X,'ID',3X,'DIF1',3X,'DID1',3X,'DIF2',3X
463* , 'DID2',3X,'DIF3',3X,'DID3',3X,'DIF4',3X,'DID4',3X,'BDL1',3X,'BDL2',
464* , 3X,'BDL3',3X,'BDL4',/)
465* 1051 FORMAT (3X,16I7)
466* 1060 FORMAT (3//,10X,'VALUE OF CALCULATED Q',3//,21X,'T',6X,'QH',8X,'QV',
467* , /)
468* 1061 FORMAT (15X,I7,2(4X,F6.3))
469* END

```

La routine TEST effectue une mesure et affiche les quatre pages des courants, les valeurs de Q calculées suivant l'algorithme "POLYNOM" et le contenu de la mémoire tampon IBUF en octal.

```
470* C-----SUBROUTINE TEST-----
471*     SUBROUTINE TEST
472*     CALL ERASE
473*     CALL READMEM (1)
474*     CALL QMOV
475*     DO 20 I=1,6
476*     CALL WRCUR (I)
477*     CALL HOCOPY
478* 20   CONTINUE
479*     RETURN
480*     END
```

La routine HTEST effectue une mesure et affiche les valeurs acquises en octal. Cette routine est prévue pour le dépannage de l'équipement.

```
481* C-----SUBROUTINE HTEST-----
482*     SUBROUTINE HTEST
483*     CALL ERASE
484*     CALL READMEM (3)
485*     RETURN
486*     END
```

La routine BYE permet de sortir le programme et de libérer le visuel. En appelant la routine FINITT de la librairie graphique, elle laisse le visuel en mode alphanumérique.

```
487* C-----SUBROUTINE BYE-----
488*     SUBROUTINE BYE
489*     COMMON/IODEV/IOUPT,LAM
490*     CALL NEWPAG
491*     CALL CHRISZ (1)
492*     WRITE (IOUPT,1000)
493*     CALL CHRISZ (4)
494*     CALL FINITT (0,3120)
495* 1000  FORMAT (/,100*PROGRAM QCALCUL TERMINATED!)
496*     END
```

### 3.2.7 Librairie de routines

En règle générale, les arguments (formal parameters) des sous-routines sont utilisés pour transmettre les options. Les variables utilisées par plusieurs routines sont transférées par le truchement des COMMON (voir table 2). Seules les routines QUAD et CONVER, provenant d'autre programmes, font exception à cette règle.

- Liste des routines de la librairie:

<u>Nom</u>	<u>Fonction</u>
INVALID	Affiche un message d'erreur
INVWORD	Affiche un message d'erreur
QHQV	Calcule les couples de valeurs QH-QV (approximation Polynomiale)
BOOMQC	Calcule les couples de valeurs QH-QV (avance de phase bêtatronique)
STAT	Calcule les valeurs moyennes et les variances des QH et QV
QUAD	Affiche le cadran du graphique en fonction des paramètres de sélection. Cette routine provient du programme QIFID écrit par M. Lelaizant.
CAMA	Routine de gestion des appels CAMAC
READMEN	Routine de gestion de la mesure. Elle acquiert les données et les regroupe après normalisation dans les mémoires appropriées.
CONVER	Convertit une chaîne de 10 caractères en nombre entier ou réel. Cette routine provient d'un ancien programme.
TIMETAB	Calcule et tient à jour la table des temps de mesure.
DATSET	Routine d'écriture des paramètres initiaux, de la date et de l'heure d'utilisation.
WOCTAL	Routine d'écriture pour afficher le contenu de IBUF en octal.

ROUTINE \ COMMON	IDDEV	SETTING	VAL	DATMAT	QVALUE	TABTIME	CAD	STATUS	FLAG
QCALCUL	X								
SET		X	X						X
INTERP	X		X						
SETIPL	X	X	X						
RING	X	X	X						
TRAIN	X	X	X						
DELAY	X	X	X						
SAMPLE 1	X	X	X						
SAMPLE 2	X	X	X						
NSTAT	X	X	X						
DISPLAY	X	X			X	X	X	X	
WQMEAN	X				X	X		X	
WRCUR	X	X		X	X	X			
TEST									
HTEST									
BYE	X								
INVALID	X								X
INVWORD	X								X
QH QV		X		X	X				
BOOMQC		X		X	X				
STAT					X			X	
QUAD	X						X		
CANA	X								
REDDHEM	X	X		X					
CONVER									
TINETAB	X	X				X			
DATSET	X	X			X				
WOCTAL	X			X		X			

Table 2. Connexions entre routines et COMMON.

```
497* C-----SUBROUTINE INVALID-----
498*     SUBROUTINE INVALID
499*     COMMON/IODEV/INP,IOUTP,LAM
500*     COMMON/FLAG/JFLAG
501*     WRITE(IOUTP,1000)
502*     JFLAG=1
503*     1000 FORMAT(/3X,'*** INVALID CHOICE ***'/)
504*     RETURN
505*     END
-----
506* C-----SUBROUTINE INVWORD-----
507*     SUBROUTINE INVWORD
508*     COMMON/IODEV/INP,IOUTP,LAM
509*     COMMON/FLAG/JFLAG
510*     WRITE(IOUTP,1000)
511*     JFLAG=1
512*     1000 FORMAT(/3X,'*** AMBIGUOUS COMMAND ***',/)
513*     RETURN
514*     END
-----
515* C-----SUBROUTINE QMGV-----
516*     SUBROUTINE QMGV
517*     REAL IB(32),IF(32),ID(32)
518*     COMMON/DATMAT/IBUF(512),DIF(32,4),BDL(32,4),DID(32,4),IB,IF,ID
519*     COMMON/SETTING/LOOP,IPL,KLOCK,IT,IT1,IT2,NMEASUR,NRING
520*     COMMON/OVALUE/QVAL(32,2)
521*     COMMON/IODEV/INP,IOUTP,LAM
522*     CHARACTER KLOCK*7
523*     DIMENSION C(10,2),ALFA(32,10)
524*     DATA C/4,5170*22,7,153J226,-3,336499,-3,272797,,1574102,3,273977
525*     * ,9,259836,-1,27938,-12,67454,5,75724,4,4490739,-7,435645
526*     * ,14,772799,-3,411908,-7,789181,13,526076,-45,13013,241,61517
527*     * ,127,86162,-303,3824/
528* C---DESIGN MATRIX
529*     DO 100 K=1,32
530*         ALFA(K,1)=1.
531*         ALFA(K,2)=(IF(K)+DIF(K,NRING)-BDL(K,NRING))/(IB(K)+BDL(K,NRING))
532*         ALFA(K,3)=(ID(K)+DID(K,NRING)-BDL(K,NRING))/(IB(K)+BDL(K,NRING))
533*         ALFA(K,4)=ALFA(K,2)*ALFA(K,2)
534*         ALFA(K,5)=ALFA(K,3)*ALFA(K,3)
535*         ALFA(K,6)=ALFA(K,2)*ALFA(K,3)
536*         ALFA(K,7)=ALFA(K,4)*ALFA(K,2)
537*         ALFA(K,8)=ALFA(K,3)*ALFA(K,5)
538*         ALFA(K,9)=ALFA(K,4)*ALFA(K,3)
539*         ALFA(K,10)=ALFA(K,5)*ALFA(K,2)
540*     100 CONTINUE
541*     DO 200 I=1,32
542*         DO 200 J=1,2
543*             QVAL(I,J)=0.
544*             DO 200 K=1,10
545*                 QVAL(I,J)=QVAL(I,J)+ALFA(I,K)*C(K,J)
546*             RETURN
547*     END
```



```

548* C-----SUBROUTINE BOOMQC-----
549*   SUBROUTINE BOOMQC
550*     REAL IR(32),IF(32),ID(32),K0,KF,KD,LD,LF,L3
551*     COMMON/DATMAT/IBUF(512),DIF(32,4),BDL(32,4),DID(32,4),IR,IF,ID
552*     COMMON/SETTING/LOOP,IPL,KLOCK,IT,IT1,IT2,NMEASUR,NRING
553*     COMMON/GVALUE/QVAL(32,2)
554*     CHARACTER KLOCK*7
555*     DIMENSION BENDH(2,2),BENDV(2,2),QDH(2,2),QDV(2,2)
556*     DIMENSION REH(2,2),REV(2,2)
557*     DATA BENDH/.999911,-7.57F-5,3.302179,.999839/,BENDV/.972872,
558*     + -.022903,3.269812,.950907/,K0/.78614/,LF/.50357/,LD/.43947/,
559*     + L3/.653/
560* C---COEFFICIENT
561*   CONST=8./3,1415926
562*   DO 500 K=1,32
563*     DUM0=IB(K)*BDL(K,NRING)
564*     KF=SQRT(K0*(1+(IF(K)*DIF(K,NRING)-BDL(K,NRING))/DUM0))
565*     KD=SQRT(K0*(1+(ID(K)*DID(K,NRING)-BDL(K,NRING))/DUM0))
566*     DUM1=KD*LD
567*     DUM2=KF*LF
568*     SHD=SINH(DUM1)
569*     SHF=SINH(DUM2)
570*     AH=COSH(DUM1)
571*     DV=COSH(DUM2)
572*     SD=SIN(DUM1)
573*     SF=SIN(DUM2)
574*     AV=COS(DUM1)
575*     DH=COS(DUM2)
576*     BH=SHD/KD
577*     BV=SD/KD
578*     CH=KD*SHD
579*     CV=KD*SD
580*     FH=SF/KF
581*     FV=SHF/KF
582*     GH=KF*SF
583*     GV=KF*SHF
584* C---VARIABLE MATRIX COEFFICIENTS
585*   QDH(1,1)=AH*DH+(AH*L3+BH)*GH
586*   QDH(1,2)=AH*FH+(AH*L3+BH)*DH
587*   QDH(2,1)=CH*DH+(CH*L3+AH)*GH
588*   QDH(2,2)=CH*FH+(CH*L3+AH)*DH
589*   QDV(1,1)=AV*DV+(AV*L3+BV)*GV
590*   QDV(1,2)=AV*FV+(AV*L3+BV)*DV
591*   QDV(2,1)=CV*DV+(CV*L3+AV)*GV
592*   QDV(2,2)=CV*FV+(CV*L3+AV)*DV
593*   DO 100 I=1,2
594*     DO 100 J=1,2
595*       REH(I,J)=0.
596*       REV(I,J)=0.
597*       DO 100 M=1,2
598*         REH(I,J)=REH(I,J)+QDH(I,M)*RENDH(M,J)
599*   100 REV(I,J)=REV(I,J)+QDV(I,M)*RENDV(M,J)
600*     TRH=REH(1,1)*REH(2,2)+REH(1,2)*REH(2,1)
601*     TRV=REV(1,1)*REV(2,2)+REV(1,2)*REV(2,1)
602*     QVAL(K,1)=CONST*ATAN(SQRT(1-TRH*TRH)/TRH)
603*     QVAL(K,2)=CONST*ATAN(SQRT(1-TRV*TRV)/TRV)
604*     IF (TRH.LT.0) QVAL(K,1)=QVAL(K,1)+8.
605*   500 IF (TRV.LT.0) QVAL(K,2)=QVAL(K,2)+8.
606*     RETURN
607*   END
-----
608* C-----SUBROUTINE STAT-----
609*   SUBROUTINE STAT
610*     COMMON/QVALUE/QVAL(32,2)
611*     COMMON/STATIS/QMEAN(32,2),QSIGMA(32,2),SSQOV(32,2),NBSHOT
612* C---MOVING AVERAGE
613*   DO 10 I=1,32
614*     DO 10 J=1,2
615*   10 QMEAN(I,J)=(QMEAN(I,J)*(NBSHOT-1)+QVAL(I,J))/NBSHOT
616* C---MOVING VARIANCE
617*   DO 20 I=1,32
618*     DO 20 J=1,2
619*     SSQOV(I,J)=SSQOV(I,J)+(QVAL(I,J)*QVAL(I,J))
620*     IF (NBSHOT.EQ.1) GO TO 20
621*     TAMP=SSQOV(I,J)-QMEAN(I,J)*QMEAN(I,J)*NBSHOT
622*     IF (TAMP.LE.0.) GO TO 15
623*     QSIGMA(I,J)=SQRT(TAMP/(NBSHOT-1)/NBSHOT)
624*     GO TO 20
625*   15 QSIGMA(I,J)=0.
626*   20 CONTINUE
627*   RETURN
628*   END
-----

```



```

767* C-----SURROUTINE READMEM-----
768* SUBROUTINE READMEM (INU)
769* REAL IP(32),IF(32),ID(32),NORM1,NORM2,NORM3,NORM4
770* COMMON/IODEV/IMP,IOUTP,LAM
771* INTEGER C,A,F,C,DATA(1),CELL(6),COCO
772* C-----CAMAC CELL
773* CELL(1)=L*64+C
774* CELL(2)=(%16+A)*32+F
775* CELL(5)=0
776* CELL(6)=1
777* C-----CHECK ON SINGLE WORD TRANSFER. JUMP IF NOT.
778* IF (C.GT.1) GO TO 10
779* C-----SINGLE WORD TRANSFER
780* CELL(3)=0
781* CELL(4)=0
782* IF (F.GE.16.AND.F.LT.24) CELL(4)=DATA(1)
783* CALL CAMAC (CELL,COCO)
784* IF (F.LT.8) DATA(1)=CELL(4)
785* GO TO 20
786* C-----BLOCK TRANSFER
787* 10 CELL(3)=C
788* CELL(4)=IREFA(DATA)
789* CALL CAMAC (CELL,COCO)
790* C-----RESPONSE ANALYSIS
791* 20 IF (COCO.NE.0) GO TO 50
792* RESPONSE ANALYSIS. SET LAM
793* IF (CELL(5).EQ.600B) LAM=1
794* RETURN
795* 50 WRITE (IOUTP,1000) L,C,N,A,F,CELL(5)
796* STOP
797* 1000 FORMAT (2,' CAMAC TRANSMISSION - FATAL ERROR',/2X,'LOOP #',I2,'I'
798* , CRATE #,I2,'I' STATION #,I2,'I' A #,I2,'I' F #,I2,'I' STATUS
799* , #,I6,'I' (OCTAL),/2X,'PROGRAM ABORTED,')
800* END
801*
802* DO 20 K=1,512
803* IDUM=ISHFT(IDUM,12)
804* IBUF(K)=IBUF(K)-IDUM
805* IF (IDUM.LT.0) IBUF(K)=IBUF(K)
806* CONTINUE
807* C-----SCALING
808* NORM1=300./4095.
809* NORM2=300./4095.
810* NORM3=60./4095./2.
811* NORM4=25./4095./5.
812* C-----POP DIVISOR 2 AND 6 SEE PS/BR NOTE 76-6
813* C-----APPENDIX 2,DFORMULAE 5
814* K=0
815* DO 100 I=1,512,16
816* K=K+1
817* IB(K)=IBUF(I)*NORM1
818* IF (K)=IBUF(I+1)*NORM2
819* ID(K)=IBUF(I+2)*NORM2
820* DO 100 J=1,4
821* DIP(K,J)=IPUF(I+3+(J-1)*2)*NORM3
822* DID(K,J)=IDUF(I+4+(J-1)*2)*NORM3
823* BOL(K,J)=IBUF(I+10+J)*NORM4
824* CONTINUE
825* 100 RETURN
826* 1000 FORMAT (/,'10X','LAM NOT ARRIVED',/08)

```

```

734* C-----SURROUTINE CAMA-----
735* SUBROUTINE CAMA (L,C,N,A,F,C,DATA)
736* COMMON/IODEV/IMP,IOUTP,LAM
737* INTEGER C,A,F,C,DATA(1),CELL(6),COCO
738* C-----CAMAC CELL
739* CELL(1)=L*64+C
740* CELL(2)=(%16+A)*32+F
741* CELL(5)=0
742* CELL(6)=1
743* C-----CHECK ON SINGLE WORD TRANSFER. JUMP IF NOT.
744* IF (C.GT.1) GO TO 10
745* C-----SINGLE WORD TRANSFER
746* CELL(3)=0
747* CELL(4)=0
748* IF (F.GE.16.AND.F.LT.24) CELL(4)=DATA(1)
749* CALL CAMAC (CELL,COCO)
750* IF (F.LT.8) DATA(1)=CELL(4)
751* GO TO 20
752* C-----BLOCK TRANSFER
753* 10 CELL(3)=C
754* CELL(4)=IREFA(DATA)
755* CALL CAMAC (CELL,COCO)
756* C-----RESPONSE ANALYSIS
757* 20 IF (COCO.NE.0) GO TO 50
758* RESPONSE ANALYSIS. SET LAM
759* IF (CELL(5).EQ.600B) LAM=1
760* RETURN
761* 50 WRITE (IOUTP,1000) L,C,N,A,F,CELL(5)
762* STOP
763* 1000 FORMAT (2,' CAMAC TRANSMISSION - FATAL ERROR',/2X,'LOOP #',I2,'I'
764* , CRATE #,I2,'I' STATION #,I2,'I' A #,I2,'I' F #,I2,'I' STATUS
765* , #,I6,'I' (OCTAL),/2X,'PROGRAM ABORTED,')
766* END

```

```

827* C-----SUBROUTINE CONVER-----
828* SUBROUTINE CONVER (CHAR,I1,AA,IFLAG)
829* CHARACTER CHAN*10,A(13)*1
830* DIMENSION I(13)
831* DATA A/1,'2','3','4','5','6','7','8','9','0','0',' ',' ','-'/
832* C-----INITIALISATION
833*
834* DO 5 K=1,13
835* I(K)=30
836* C-----CHARACTER IDENTIFICATION
837* DO 10 J=1,13
838* IF (CHAR(I(1),EQ,A(J)) I(1)=J
839* IF (I(1),EQ,30) GO TO 120
840* IF (I(1),EQ,11) GO TO 110
841* IF (I(1),EQ,10) I(N)=0
842* DO 30 N=2,10
843* DO 29 J=1,13
844* IF (CHAR(NIN),EQ,A(J)) I(N)=J
845* IF (I(N),EQ,10) I(N)=0
846* DO 31 J=2,10
847* IF (I(J),EQ,30) GO TO 110
848* IF (I(J),EQ,11,AND,(I(J-1),NE,11,AND,I(J+1),NE,11)) GO TO 110
849* C-----POINT,FIRST BLANK,SIGN
850* IPP=0
851* IPFB=0
852* DO 50 K=1,10
853* IF (IPP,NE,0) GO TO 35
854* IF (I(K),EQ,12) IPP=K
855* IF (IPFB,NE,0) GO TO 50
856* IF (I(K),EQ,11) IPFB=K
857* IF (IPP,NE,0,AND,IPFB,NE,0) GO TO 55
858* N=1
859* IF (I(1),EQ,13) N=2
860* IF (IPP,NE,0) GO TO 80
861* IF (I(5+N),NE,11) GO TO 110
862* C-----INTEGER CONVERSION
863* IFLAG=1
864* II=0
865* DO 60 K=N,(5*N)
866* Q=10,**(IPFB-(K*1))
867* IF (Q,LT,1.) GO TO 61
868* II=II+I(K)*Q
869* IF (I(1),EQ,13) II=-II
870* RETURN
871* C-----REAL CONVERSION
872* IFLAG=2
873* AA=0.
874* L=1
875* DO 90 K=N,10
876* IF (I(K),EQ,11) GO TO 100
877* IF (I(K),EQ,12) L=0
878* Q=10,**(IPP-(K*L))
879* IF (I(K),EQ,12) GO TO 90
880* AA=AA+I(K)*Q
881* CONTINUE
882* IF (I(1),EQ,13) AA=-AA
883* RETURN
884* CONTINUE
885* IFLAG=3

```



```

903* -----SURROUTINE DATSET-----
904* SUBROUTINE DATSET (N)
905* COMMON/IODEV/INP, IOUTP, LAM
906* COMMON/SETT/NGLOOP, IPL, KLOCK, IT, IT1, IT2, NMEASUR, NRING
907* INTEGER DATIM (7)
908* CHARACTER KLOCK*7
909* CALL CLOCK (DATIM)
910* CALL HOME
911* CALL ANMODE
912* WRITE (IOUTP,1000) (DATIM(I),I=5,7)
913* WRITE (IOUTP,1001) (DATIM(5-I),I=1,3)
914* WRITE (IOUTP,1002) IPL,IT,IT1,KLOCK,IT2
915* IF (N.GT.1) WRITE (IOUTP,1003) NRING
916* IF (N.GT.2) GO TO 10
917* RETURN
918* 5
919* 10
920* CALL CHRSIZ (1)
921* CALL MOVEA (R20,,750.)
922* CALL ANMODE
923* J=1
924* WRITE (IOUTP,1004) J
925* CALL CHRSIZ (3)
926* GO TO 5
927* 1000 FORMAT (5/,90X,DATE '1,2X,2(I2,,),I4)
928* 1001 FORMAT (90X,TIME '1,2X,12, H,13, M,13, S,1)
929* 1002 FORMAT (/,90X,IPL '1,12,9X,IT '1,15,/,90X,TRIGGER ' W10',
930* 7X,IT1 '1,15,/,90X,CLOCK '1, A7,3X,IT2 '1,15)
931* 1003 FORMAT (/,90X,NRING '1,12)
932* 1004 FORMAT ('S',I4)
933* END
-----SURROUTINE WOCIAL-----
934* SUBROUTINE WOCIAL
935* COMMON/IODEV/INP, IOUTP, LAM
936* COMMON/DATMAT/IBUF (512), DIF (32,4), BDL (32,4), DID (32,4), I9, IF, ID
937* COMMON/TABTIME/ITTAB(32)
938* C-----WRITE CONTENT OF BUFFER
939* WRITE (IOUTP,1000)
940* K=0
941* DO 10 J=1,512,16
942* K=K+1
943* 10 WRITE (IOUTP,1010) ITTAB(K), (IBUF(J-1),I=1,15)
944* CALL DATSET (1)
945* RETURN
946* 1000 FORMAT (14/,10X,CONTENT OF THE BUFFER =OCTAL VALUE=,
947* 3/9X,IT,5X,IT1,5X,IF,5X>ID,3X'DIF1',3X'DID1',3X'DIF2',3X
948* 3X'DID2',3X'DIF3',3X'DID3',3X'DIF4',3X'DID4',3X'DIF1',3X'DIF2',3X
949* 3X'DIF3',3X'DIF4',/)
950* 1010 FORMAT (3X,I7,1507)
951* END
-----SURROUTINE DATSET-----
952* SUBROUTINE DATSET (N)
953* COMMON/IODEV/INP, IOUTP, LAM
954* COMMON/SETT/NGLOOP, IPL, KLOCK, IT, IT1, IT2, NMEASUR, NRING
955* INTEGER DATIM (7)
956* CHARACTER KLOCK*7
957* CALL CLOCK (DATIM)
958* CALL HOME
959* CALL ANMODE
960* WRITE (IOUTP,1000) (DATIM(I),I=5,7)
961* WRITE (IOUTP,1001) (DATIM(5-I),I=1,3)
962* WRITE (IOUTP,1002) IPL,IT,IT1,KLOCK,IT2
963* IF (N.GT.1) WRITE (IOUTP,1003) NRING
964* IF (N.GT.2) GO TO 10
965* RETURN
966* 5
967* 10
968* CALL CHRSIZ (1)
969* CALL MOVEA (R20,,750.)
970* CALL ANMODE
971* J=1
972* WRITE (IOUTP,1004) J
973* CALL CHRSIZ (3)
974* GO TO 5
975* 1000 FORMAT (5/,90X,DATE '1,2X,2(I2,,),I4)
976* 1001 FORMAT (90X,TIME '1,2X,12, H,13, M,13, S,1)
977* 1002 FORMAT (/,90X,IPL '1,12,9X,IT '1,15,/,90X,TRIGGER ' W10',
978* 7X,IT1 '1,15,/,90X,CLOCK '1, A7,3X,IT2 '1,15)
979* 1003 FORMAT (/,90X,NRING '1,12)
980* 1004 FORMAT ('S',I4)
981* END

```

### 3.3 Source et chargement du programme

Pour une raison non élucidée, ce programme ne fonctionne pas lorsqu'il est compilé avec le compilateur installé sur le T-BMC. Par contre, il tourne sans problème lorsqu'on utilise la nouvelle version NORD du compilateur qui est installé sur le PRDEV. Comme, pour le moment, il n'est pas possible de changer de compilateur sur le T-BMC (problème de longueur avec le programme BEAMSCOPE), la source FTN-QCALCUL, le programme compilé QCALCUL et la routine de chargement M-Q-54 sont stockés sur un disque flexible.

```
FILE NAME: (M-C:)  
OUTPUT FILE:  
FILE 0 : (METZGER-CLAUDE:TEMP-BMC)FTN-PS-CHANGE:SYMB#1  
FILE 1 : (METZGER-CLAUDE:TEMP-BMC)M-Q-54:SYMB#1  
FILE 2 : (METZGER-CLAUDE:TEMP-BMC)FTN-CURRENT:SYMB#1  
FILE 3 : (METZGER-CLAUDE:TEMP-BMC)FTN-MATLIB:SYMB#1  
FILE 4 : (METZGER-CLAUDE:TEMP-BMC)CURRENT:BRF#1  
FILE 5 : (METZGER-CLAUDE:TEMP-BMC)FTN-QCALCUL:SYMB#1  
FILE 6 : (METZGER-CLAUDE:TEMP-BMC)QCALCUL:BRF#1  
FILE 10 : (METZGER-CLAUDE:TEMP-BMC)FTNLIBR-2091C:BRF#1
```

La routine de chargement qui fonctionne sous l'utilisateur RT (Password BMC) fait appel à la nouvelle librairie FORTRAN qui, elle aussi, est stockée sur le disque flexible.

```
@CC          REBUILT  QCALCUL  
@CC  
@RT-LOADER  
DELET-PROGRAM QCALCUL  
YES  
CLEAR-SEGMENT 54  
YES  
NEW-SEGMENT,54,2,DM,,,,  
FIX-SYMBOLS 173  
LOAD (M-C:T-BMC)QCALCUL,,  
LOAD USERLIB,,  
LOAD GRAPHLIB,,  
LOAD (M-C:T-BMC)FTNLIB,,  
WRITE-REFERENCES,,  
END-LOAD  
EXIT-LOADER  
*
```

Dès que les problèmes de compilation seront résolus, ce programme sera installé sur le T-BMC et remplacera QIFID dans la librairie de START-APPLICATION.

Bibliographie

1. G. Baribaud et al. PS/BR Note 76-6. Calcul de Q à partir des courants IB, IF, ID,  $\Delta IF$ ,  $\Delta ID$  et  $f_{Bd\ell}$  du PSB - Spécifications du matériel et du logiciel.
2. G. Baribaud et al. PS/BR Note 78-13. Diapositifs de saisie et de traitement des données de mesures du CPS à l'aide d'un ordinateur PDP 11/45.
3. W. Remmer. PS/CCI/Note 78-4. Proposal for future exploitation of the Beam Measurement Computer.
4. S. Battisti et H. Kugler. A paraître (preprint 8.12.77). Interface Handbook.
5. C. Bovet. SI/Note DL/69-17. BOOM: Un programme pour l'optique du PSB.
6. J.P. Delahaye. Communication privée (11.06.79). Calcul de Q. Une formule sans approximation.



Sélecteur de ligne de programme (IPL)

Description

Ce module sélectionne une IPL parmi 8 et actionne 8 contacts indépendants pour continuité de 8 signaux analogiques distincts.

Présentation

Tiroir Camac 2V avec en face avant visualisation par Leds des IPL choisies et de l'IPL sélectionnée.

Contacts sortis sur prises Lemo.

Fonctionnement

Une commande NA(0) F(16) permet de mémoriser les IPL utilisées.

La sélection des IPL est réalisée par un registre à 8 bits, soit 1 bit pour chaque IPL.

00000001 = IPL 1  
01000011 = IPL 1,2 et 7 etc...

Une commande NA(0) F(0) permet de relire ces IPL. La sélection de l'IPL désirée, visualisée en face avant par Led verte, se produira chaque fois que cette dernière sera présente sur le connecteur d'entrée Emihus.

Un OU cable permet de fermer les contacts des huit chaînes analogiques.

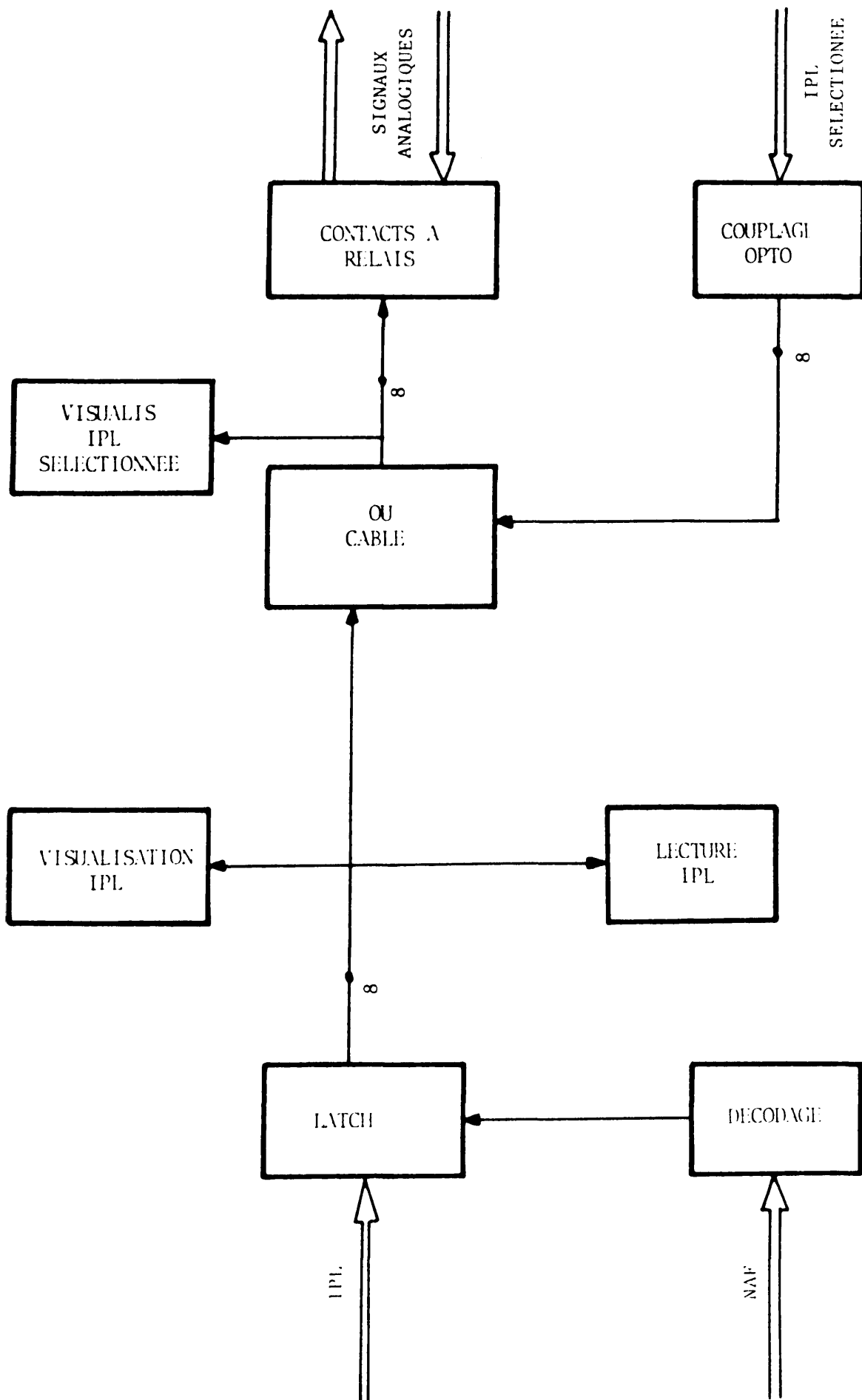
Réalisation

Circuits intégrés TTL série normale, montés sur carte Camac standard.

Cablage Wire Wrap

Current absorbe par un opto coupleur

Valeur nominale  $\frac{24}{830}$  # 28mA



Sélecteur de ligne de programme