

## Annexe

### Programme initiale P0:

```
STEP init0
  IF                                NOP
  THEN  CMP 0                      'IP_TABLE
        WITH V1                    "set
        WITH V1                    "index ip
        WITH V193                  "@ ip
        WITH V50
        WITH V179
        WITH V217
        LOAD FU32                  "on regarde si il y a une erreur
durant le CMP
      TO erreur

  IF                                erreur
      = V0
  THEN SET P2                      "on lance les programmes: envoi et
      SET P4                      " réception
      SET P5

STEP fond
  IF                                NOP
  THEN JMP TO fond
```

# Programme de réceptions

```

STEP reçoit
  IF                                NOP
  THEN CMP 1                        'UDP_HAND : installation du handler
      WITH                          "numéro du port local
      WITH                          "réception a partir de fw0
      LOAD                          FU32
      TO                            erreur

STEP ok1
  IF                                erreur
      =                             V0
  THEN                              NOP

STEP lire
  IF                                N (   modif   " on surveille l'arrivée de paquets
      =                             FW0   )
  THEN LOAD                          FW0
      TO                            modif
      LOAD                          FW10   "fw10: 1° octet de données
      TO                            entête  " entête: lettre + case

STEP traiter
  IF                                NOP
  THEN LOAD (   entête
      AND      V$00FF   )   "on récupère le 2° octet de entête:
      TO        lettre   " lettre

STEP choix
  IF                                lettre
      =                             V79   "on regarde si on reçoit bien un 'O'
  THEN LOAD (   entête
      /          V256   )   "on récupère le 1° octet de entête:
      TO        case0   " case
      LOAD      FW11   " on met la valeur au bon endroit
      TO        valeurO
      CMP 11          'OutPut: on lance la tache de
                      " rangement des sorties

STEP fin
  IF                                NOP
  THEN JMP TO lire   " on exécute sans arrêt la réception

```

## Module output

```
STEP cherche
IF      =      case0      "si la case est la 0
THEN    LOAD   valeur0    " on charge la donnée en ow0 et dans
      TO      OW0        " son fw correspondant
      LOAD   valeur0
      TO      FW5513

IF      =      case0
THEN    LOAD   valeur0
      TO      OW1
      LOAD   valeur0
      TO      FW5515

.
.
.
.
.

IF      =      case0
THEN    LOAD   valeur0
      TO      OW11
      LOAD   valeur0
      TO      FW5535

IF      NOP      " ces deux lignes servent à désactiver
THEN    NOP      " le module quand il est fini.
```

## Programme d'initialisation des flag word:

```
STEP init4_1
  IF      NOP
  THEN LOAD V73      "73: code ascii de I
    TO I
    LOAD V0      "on met 0 pour commencer au numéro 0
    TO numéro
    LOAD V5000   "on commence la recopie des I/O a
    TO flag     " partir de fw5000

STEP charge_1
  IF      NOP
  THEN LOAD numéro      "on décale numéro de un octet
    * V256
    OR I
    TO total      " total: deux premier octet envoyé:
    CFM 0         " lettre + case
    WITH flag     'WINDEXMW: cette fonction remplit les
    WITH total    "flag word grâce a deux paramètres:
                  "le numéro du flag et sa valeur(total)

STEP incre_1
  IF      NOP
  THEN INC numéro      "on passe au numéro d'entrée suivant
    INC flag           "on incrémente deux fois le flag
    INC flag           "pour laisser la place au données

STEP saut_1
  IF      N ( numéro      "on a recommencé que 20 fois car on
    >= V20 )           "n'avait pas besoin de plus dans
  THEN JMP TO charge_1 "notre application

"*****

STEP init4_2
  IF      NOP
  THEN LOAD V79      "79: code ascii de 'O'
    TO O
    LOAD V0
    TO numéro
    LOAD V5512
    TO flag

STEP charge_2
  IF      NOP
  THEN LOAD numéro
    * V256
    OR V73
    TO total
    CFM 0      'WINDEXMW
    WITH flag
    WITH total

STEP incre_2
  IF      NOP
  THEN INC numéro
    INC flag
    INC flag

STEP saut_2
```

```
IF          N (    numéro
                >=    V20      )
THEN  JMP TO charge_2
```

## Programme de recopie des I/O

```
STEP iw0
IF          N (      IW0      "on regarde si la copie (fw5001) est
      =      FW5001      )  " différente de l'entrée 0 si oui on
THEN LOAD   IW0      "la recopie on l'envoi au serveur
      TO      FW5001
      CMP 10
      WITH      V5000      'envoi: module que nous avons crée et
IF          NOP      "qui envoi un flag word
THEN        NOP      "ces deux lignes sont nécessaire pour
                        "sortir du step
```

\*\*\*\*\*

```
STEP iw1
IF          N (      IW1      "on recommence avec l'entrée suivante
      =      FW5003      )
THEN LOAD   IW1
      TO      FW5003
      CMP 10
      WITH      V5002      'envoi
IF          NOP
THEN        NOP
```

.  
.
.  
.
.

\*\*\*\*\*

```
STEP ow0
IF          N (      OW0      "on fait pareil avec les sorties
      =      FW5513      )
THEN LOAD   OW0
      TO      FW5513
      CMP 10
      WITH      V5512      'envoi
IF          NOP
THEN        NOP
```

\*\*\*\*\*

.  
.
.  
.
.

```
STEP ow15
IF          N (      IW15
      =      FW5543      )
THEN LOAD   IW15
      TO      FW5543
      CMP 10
      WITH      V5542      'envoi
IF          NOP
THEN        NOP
```

```
STEP fin
  IF
  THEN  JMP TO iw0
```

```
NOP
```

```
"on revient au début et on recommence  
"indéfiniment
```

## Module d'envoi

STEP charger		"on récupèrent les paramètres du
IF	NOP	"module
LOAD	FU32	
TO	R11	
STEP envoi		
IF	NOP	
THEN CMP 2		'UDP_SEND
WITH	V2000	"local port
WITH	V1	"index dans table IP
WITH	V600	"remote port
WITH	V4	"nombre d'octets
WITH	R11	"a partir du fw donné par r11
LOAD	FU32	
TO	erreur	
STEP renvoi		" on vérifie les erreurs et on renvoi
IF	erreur	" les paramètres
=	V0	
THEN LOAD	R11	
TO	FU32	