



Ne cherchez plus ! MEDOR le fait pour vous ! Cet utilitaire intéressera plus particulièrement les bidouilleurs d'octets invétérés. Ecrit en langage machine, il est très rapide : il examine la mémoire et affiche en quelques secondes l'adresse du mot ou du bloc d'octets que vous recherchez. Le même, en BASIC, vous laisserait le temps de prendre un café et de reposer vos yeux de programmeur acharné (à condition que vous sachiez les détacher de l'écran !). Ah, les vertus thérapeutiques du BASIC !

COMMENT ÇA MARCHE ?

D'abord taper le programme MEDOR.BAS et le lancer. Il créera et sauvegardera 2 programmes : Le premier en BASIC : MEDOR qui est chargé de lancer. Le deuxième en langage machine : MEDOR.BIN. Une fois MEDOR et MEDOR.BIN créés et sauvegardés, MEDOR.BAS devient inutile.

Deux exemples d'utilisation :

Tout d'abord RUN" MEDOR

Exemple 1 : vous recherchez la routine système &BD2B dans un programme en langage machine.
 1) Sélectionnez l'option BLOC HEXA en appuyant sur la barre d'espace-ment et validez par <ENTER>
 2) Précisez les 2 octets, sans préfixe : 2B BD (le plus faible d'abord !) et validez par <ENTER>
 3) Définissez dans l'ordre l'adresse initiale (4 caractères hexa obligatoirement, commencez par des zéros si nécessaire) et l'adresse finale (même procédure) sans valider.

Et la recherche s'effectue. Les résultats s'affichent à l'écran et sont imprimés (à condition que l'imprimante ait été mise en service avant l'utilisation de MEDOR).

Exemple 2 : on recherche le mot PLAYER dans un jeu en langage machine pour le remplacer par JOUEUR.

1) Sélectionnez l'option MOT(S) ASCII et validez par <ENTER>
 2) Précisez le mot : PLAYER (en majuscules dans ce cas, pensez-y avant !)

3) Procédez ensuite comme dans l'exemple précédent.

ATTENTION. Evitez absolument les erreurs de frappe. Afin de limiter la taille du programme, les routines de contrôle de saisie sont réduites au minimum.

STRUCTURE DU PROGRAMME

Elle est très simple :

- Choix de l'option MOT(S) ASCII / BLOC HEXA
- Saisie du MOT ou du BLOC
- Définition de l'adresse initiale
- Définition de l'adresse finale
- Recherche en mémoire
- Fin du programme ou nouvelle recherche.

Les programmeurs en assembleur trouveront dans le programme source des routines commentées : redéfinition de caractères, affichage à l'écran, impression, conversion ASCII/Hexa et Hexa/ASCII pour des octets à 1 ou 2 chiffres.

Le programme est logé dans la mémoire écran et scindé en plusieurs blocs correspondant chacun à deux lignes du bas de l'écran, ceci dans le but de libérer le maximum de place pour le programme à analyser.

GAGNER AU LOTO

un rêve qui peut devenir réalité avec

LOTO - MATIC

le programme qui vous révèle tout ce que vous devez savoir pour :

- trouver facilement les numéros qui ont le plus de chance de sortir
- établir scientifiquement les grilles les plus performantes grâce aux tests du Lotoscope
- contrôler sans peine les résultats de vos jeux Editions écran et imprimante

Documentation détaillée
+ CADEAU contre 4 timbres

INFORMATIC Applications

B.P. 78 - 67800 BISCHHEIM
Tél. 88.33.58.85

LISTING

COPYRIGHT 1985 MICRO-APPLICATION
DAMS.

```
*** MEDOR.ASM ***
(J. M. CHEVALIER 1987)

ORG #7730 ; le prog est ensuite transfere
              ; en #d730

===== AFFICHAGE EN-TETE =====
symbol
LD h1,table ; redéfinit les caractères 251 à 255
LD de,#a6d4 ; précédemment définis à partir
LD bc,40 ; de l'adresse #a6d4
LDIR

LD de,titre ; affiche * RECHERCHE EN MEMOIRE *
CALL affiche

esp44 LD b,44 ; espace de 44 car
boucl1 LD a,32
CALL #bb5a
DJNZ boucl1

LD de,lojmciel ; affiche 80 chr$(154)
CALL affiche

ligne154 LD b,80
boucl2 LD a,154
CALL #bb5a
DJNZ boucl2

===== PROG PRINCIPAL =====
debutprg
LD h,0 ; window #0,1,80,2,22
LD d,79
LD 1,2
LD e,22
CALL #bb66

CALL #bb6c ; efface la fenêtre courante

LD h1,imp ; teste si l'imprimante est (ou non)
CALL #bd2e ; en service
JR nc,imprim ; si oui: imp contient #ff
LD (h1),0 ; sinon imp contient 0
JR quest1
imprim LD (h1),#ff

===== QUEST 1: MOT(S) ASCII / BLOC HEXA ? =====
quest1
LD h,1 ; curseur col:1 lig:1
LD 1,1
CALL #bb75
LD de,mes1
CALL affiche
CALL #bb06
CP 13

JR z,motasci ; si <enter> alors c'est un mot ascii
CP 32 ; si appui sur espace
JR nz,attend1 ; alors est-ce un bloc hexa?

attend1 LD h,1
LD 1,1
CALL #bb75
LD de,mes2
CALL affiche
CALL #bb06
CP 13
JP z,blochex
CP 32

; si <enter> alors c'est un bloc hexa
; si appui sur espace
```

```
JR z,quest1 ; test-ce un mot ascii?
JR attend2 ; si non fin du programme

;===== SAISIE MOT(S) ASCII =====
motasci
LD de,mes3
CALL affetimp
CALL #bb81
LD b,0
LD hi,mot
boucl3 CALL #bb06
CP 13
JP z,suite ; affiche le curseur
CALL affimp ; nb caractere
LD (hi),a ; mot contient la valeur ascii
INC hi ; de la 1ère lettre du mot
; si <enter>, alors fin du mot
INC b ; affiche et imprime le caractère
LD a,b ; courant du mot
LD (l1gmot),a
CP 208
JR nz,boucl3 ; 208 car max

JP #7f30 ; transfère ensuite en #df30
ORG #7f30

;===== SAISIE BLOC HEXA =====
blochex JR blochex1
blochex1 LD de,mes4 ; affiche et imprime:

CALL affetimp ; BLOC HEXA (208 OCT. MAX)
CALL #bb81 ; affiche le curseur
LD b,0 ; nb d'octets
LD hi,mot

boucl4 CALL #bb06 ; traitement poids fort de l'oct. #XY
CP 13 ; si <enter> : fin du bloc
JR z,suite ; alors: quest. 3 (ad. init.)
CALL testhexa ; test-ce un car. hexa?
JR c,boucl4 ; sinon on recommence
CALL affimp ; si oui affich et impr
CALL ascihexa ; conversion ascii/hexa
SLA a ; conversion de l'oct. #0X en #X
SLA a
SLA a
LD c,a ; traitement du poids faible de #XY
CALL #bb06
CALL testhexa
JR c,spi
CALL affimp
CALL ascihexa
ADD a,c ; si oct. #0Y est ajouté au précédent #X
LD (hi),a ; oct. courant #XY dans la table MOT
INC b
LD a,b
LD (l1gmot),a
CP 208 ; 208 oct max
JR z,suite
INC hi
LD de,mes5
CALL affiche

JR boucl4 ; si oct. #0Y est ajouté au précédent #X
testhexa CALL minmaj ; mise en majuscule
CP "0"
RET c ; caractère entre 0 et 9 ou entre A et F
CP "F"+1
JR nc,suit1
CP "9"+1
JR c,suit2
CP "A"
RET CP "#FF" ; si oui pas de carry
RET ; sinon met le "carry"

suit1 AND a ; enlève le carry
RET

minmaj CP "a" ; met les caractères entre a et f
RET c CP "f"+1
RET nc ; en majuscule
SUB #20
```



```

        RET
suite
; ===== QUEST.3:ADRESSE INITIALE =====
quest3
    CALL #bb84           ;enleve le curseur
    LD de,mes6            ;affiche et impr. ADRESSE INITIALE:
    CALL affetimp
    CALL #bb81           ;remet le curseur
    CALL sp2              ;affiche et saisit un mot de 2 oct.
    LD (adinit),de

; ===== QUEST.4:ADRESSE FINALE =====
quest4
    CALL #bb84           ;commentaires ci-dessus
    LD de,mes7            ;affiche et impr. ADRESSE FINALE:
    CALL affetimp
    CALL #bb81
    CALL sp2
    LD (adfin),de
    LD de,mes8
    CALL affetimp

    JR suitpgi           ;saut a la routine de recherche

sp2
    CALL sp3
    LD d,a
    CALL sp3
    LD e,a
    RET
sp3
    CALL affichex
    SLA a
    SLA a
    SLA a
    SLA a
    LD c,a
    CALL affichex
    ADD a,c
    RET
affichex
    CALL #bb06           ;saisit,affiche et impr. le caractere
    CALL testhexa          ;hexa courant
    JR c,affichex
    CALL affimp
    CALL ascihexa
    RET
.COPYRIGHT 1985 MICRO-APPLICATION.
.DAMS.

; suitpgi CALL #bb84           ;enleve le curseur
;         JR suitpg2
; suitpg2 LD h1,(adinit)
;         JP #8730
; ***** *****
;         ORG #8730           ;transfere ensuite a #8730
; ===== RECHERCHE EN MEMOIRE =====
debut
    LD (adcour),h1
    LD de,mot
testmot
    LD a,(lgmot)
    LD b,a
testcar
    LD a,(de)
    CP (h1)
    JR nz,testad
    INC de
    INC h1
    DJNZ testcar

    LD h1,(adcour)
P2
    PUSH h1             ;convertit un mot de 2 oct. XYZT
    SRL h               ;en 4 val.ascii "X","Y","Z" et "T"
    SRL h
    SRL h
    SRL h
    LD a,h
    CALL hexasci
    CALL affimp
    POP h1

    PUSH h1             ;traitement de Y
    LD a,h
    AND #0f
    CALL hexasci
    CALL affimp
    POP h1

    PUSH h1             ;traitement de Z
    SRL 1
    SRL 1

```

```

    SRL 1
    SRL 1
    LD a,1
    CALL hexasci
    CALL affimp
    POP h1

    PUSH h1             ;traitement de T
    LD a,1
    AND #0f
    CALL hexasci
    CALL affimp
    POP h1

    PUSH bc             ;affiche 4 espaces
    LD b,4
boucl5
    LD a,32
    CALL aff
    CALL affimp
    DJNZ boucl5
    POP bc

testad
    LD h1,(adcour)       ;adcour+1=adfin?
    INC h1
    LD (adcour),h1
    LD a,1
    LD ix,adfin
    CP (ix+0)
    JR nz,debut
    LD a,h
    CP (ix+1)
    JR nz,debut
    ; ===== QUEST.5:NOUVELLE RECHERCHE =====
    LD de,mes9            ;affiche et impr.:
    CALL affiche
    LD de,mes10           ;NOUVELLE RECHERCHE:APPUYER SUR [ENTER]
    CALL affetimp          ;retour a la ligne et separation

    CALL #bb06
    CP 13
    JP z,debutprg

fin
    RET
; ===== ROUTINES =====
hexasci
    ADD a,#90             ;convertit une valeur hexa
    DAA
    ADC a,#40             ;entre 0 et 9 ou entre A et F
    DAA
    RET                   ;en la val.ascii correspondant
                        ;hex:A ->#41 , B ->#42 etc...
affimp
    CALL #bb5a
    PUSH af
    LD a,(imp)
    CP #ff
    JR nz,nonimp
    POP af
    CALL #bd2b
    RET
nonimp
    POP af
    RET
affiche
    LD a,(de)
    CP 0
    RET z
    CALL #bb5a
    INC de
    JR affiche
affetimp
    LD a,(de)
    CP 0
    RET z
    CALL affimp
    INC de
    JR affetimp
ascihexa
    SUB "0"
    CP #10
    RET c
    SUB 7
    RET
***** *****

```

CPC n°36 - Juillet/Août 1988

```

;===== ORG #8F30 =====
;===== DONNEES =====
; table
; DEFB #0C, #18, #3C, #66 ;CHR$(251)
; DEFB #7E, #60, #3C, #00 ;CHR$(262)
; DEFB #CF, #EC, #E0, #ED ;CHR$(253)
; DEFB #ED, #ED, #C4, #FF ;CHR$(254)
; DEFB #F1, #3A, #BA, #BB ;CHR$(255)
; DEFB #BB, #AB, #23, #FF
; DEFB #0F, #AA, #BE, #BA
; DEFB #BA, #AA, #8A, #FF
; DEFB #E7, #37, #B7, #37
; DEFB #F7, #B7, #23, #FF

; titre
; DEFB 24, 32
; DEFM MEDOR
; DEFB 32, 24, 32
; DEFM recherche
; DEFB 32
; DEFM en m
; DEFB 251
; DEFM moire
; DEFB 0

; lojmciel DEFB 252, 253, 254, 255 ;lojmciel
; DEFB 0

; mes1
; DEFB 13
; DEFB 24, 32
; DEFM MOT(S) ASCII
; DEFB 32, 24
; DEFB 32, 32, 32
; DEFM BLOC HEXA
; DEFB 32
; DEFB 0

; mes2
; DEFB 13
; DEFB 32
; DEFM MOT(S) ASCII
; DEFB 32, 32, 32
; DEFB 24, 32
; DEFM BLOC HEXA
; DEFB 32, 24
; DEFB 0

; mes3
; DEFB 13, 10, 10
; DEFM MOT(S) ASCII
; DEFB 32
; DEFM (208 CAR. MAXI):
; DEFB 32, 0
; =====*
;
```

```

;===== ORG #9730 =====
;===== transfere ensuite a #f730 =====
; mes4
; DEFB 13, 10, 10
; DEFM BLOC HEXA
; DEFB 32
; DEFM (208 OCTETS MAX):
; DEFB 32, 0
; DEFB 32, 32, 0

; mes5
; mes6
; DEFB 13, 10, 10
; DEFM ADRESSE INITIALE:
; DEFB 32, 0

; mes7
; DEFB 13, 10, 10
; DEFM ADRESSE FINALE:
; DEFB 32, 0
; DEFB 13, 10, 10, 0

; mes8
; mes9
; DEFB 13, 10, 10
; DEFM NOUVELLE RECHERCHE:
; DEFB 32, 24
; DEFM [ENTER]
; DEFB 24, 32
; DEFM SINON AUTRE TOUCHE
; DEFB 0

; mes10
; DEFB 13, 10, 10, 0
; =====*
;===== ORG #9F30 =====
;===== transfere ensuite a #ff30 =====
; tampons
;
```

```

Text:#11485 End:17804 6319 Bytes
Hmem:28461
symbol #7730 esp44 #7741 boucl1 #7743 ligne154 #7750
boucl2 #7752 debutprg #7759 imprim #7773 quest1 #7775
attend1 #7782 attend2 #779A motasci #77A8 boucl3 #77B6
blochex #7F32 blochex1 #7F34 boucl4 #7F42 spi #7F50
testhexa #7F7F suit1 #7F90 suit2 #7F93 minmaj #7F95
suite #7F9E quest3 #7F9E quest4 #7FB1 sp2 #7FCC
sp3 #7FD5 affiche #7FE6 suitpg1 #7FF5 suitpg2 #7FFA
debut #8730 testmot #8736 testcar #873A p2 #8745
boucl5 #8780 testad #8788 fin #87B3 hexasci #87B4
affimp #87BB nonimp #87CB affiche #87C0 afftemp #87D7
ascihexa #87E1 table #8F30 titre #8F58 lojmciel #8F78
mes1 #8F80 mes2 #8F9F mes3 #8FBE mes4 #9730
mes5 #9750 mes6 #9753 mes7 #9769 mes8 #9770
mes9 #9781 mes10 #97B5 adinit #9F30 adfin #9F32
lgmot #9F34 adcour #9F35 imp #9F37 mot #9F38

Text:#2CDD End:#458C #16AF Bytes
Hmem:#6F2D

; * * MEDOR *
10 GOTO 55 >FA
15 ===== routine de chargement ===== >LF
20 s=0:j=j+1 >DE
25 FOR i=0 TO 7:READ a$:POKE ad,VAL("&" + a$):ad=ad+1 >TZ
30 s=s+VAL("&" + a$) >LD
35 NEXT >YJ
40 READ s2:IF s <> s2 THEN PRINT CHR$(7); "erreur dans la ligne:""200+((j-1)*5)":END >NZ
45 IF s <> s3 THEN 20 >MU
50 RETURN >TD
55 MODE 2:PRINT CHR$(24)" MEDOR "CHR$(24)" se charge .." >UY
." :LOCATE 1,1 >YE
100 : >XJ
105 ad=&D730:s3=1194:GOSUB 20 >WX
110 ad=&DF30:s3=843:GOSUB 20 >WT
115 ad=&E730:s3=956:GOSUB 20 >WZ
120 ad=&EF30:s3=357:GOSUB 20 >WK
125 ad=&F730:s3=326:GOSUB 20 >RE
130 ===== sauvegardes ===== >PJ
135 OPENOUT " !MEDOR "
140 PRINT #9,"10 MODE 2" >RU
145 PRINT #9,"20 LOAD";CHR$(34);" !MEDOR.BIN" >MB
150 PRINT #9,"30 CALL &D730" >VE
155 CLOSEOUT >TB
160 SAVE" !MEDOR.BIN",B,&D730,&2907 >BY
165 CALL &D730 >LC
170 END >TB

```

195 ===== codes hexa.=====	>TF	480 DATA E7,E1,E5,7D,E6,0F,CD,5A, 1446	DT
200 DATA 21,30,EF,11,D4,Ae,01,28, 756	>BG	485 DATA E7,CD,C1,E7,E1,C5,05,04, 1292	DM
205 DATA 00,ED,B0,11,58,EF,CD,D3, 1173	>DH	490 DATA 3E,20,CD,C1,E7,10,F9,C1, 1181	DY
210 DATA E7,06,2C,3E,20,CD,5A,BB, 857	>EB	495 DATA 2A,35,FF,23,22,35,FF,7D, 351	CF
215 DATA 10,F9,11,7B,EF,CD,D3,E7, 1291	>CZ	500 DATA DD,21,32,FF,DD,BE,00,20, 1002	DF
220 DATA 06,50,3E,9A,CD,5A,BB,10, 800	>DU	505 DATA 97,7C,DD,BE,01,20,91,11, 891	CR
225 DATA F9,26,00,16,4F,2E,02,1E, 466	>BY	510 DATA 81,F7,CD,D3,E7,11,B5,F7, 1468	DT
230 DATA 16,CD,66,BB,CD,6C,BB,21, 1049	>DC	515 DATA CD,DD,E7,CD,06,BB,FE,0D, 1322	EC
235 DATA 37,FF,CD,2E,BD,30,04,36, 856	>CZ	520 DATA C4,59,D7,C9,11,B5,F7,CD, 1357	CH
240 DATA 00,18,02,36,FF,26,01,2E, 420	>BJ	525 DATA D0,E7,C6,90,27,DE,40,27, 1142	DT
245 DATA 01,CD,75,BB,11,80,EF,CD, 1099	>DW	530 DATA C9,CD,5A,BB,F5,3A,37,FF, 1296	DR
250 DATA D3,E7,CD,06,BB,FE,0D,28, 1147	>DM	535 DATA FE,FF,20,05,F1,CD,2B,BD, 1224	DZ
255 DATA 1F,FE,20,20,F5,26,01,2E, 679	>CK	540 DATA C9,F1,C9,1A,FE,00,08,CD, 1228	DF
260 DATA 01,CD,75,BB,11,9F,EF,CD, 1130	>DC	545 DATA 5A,BB,13,18,F6,1A,FE,00, 846	CR
265 DATA D3,E7,CD,06,BB,FE,0D,CA, 1309	>DX	550 DATA C8,CD,C1,E7,13,1B,F6,D6, 1332	DI
270 DATA 32,DF,FE,20,28,CF,1B,F2, 1072	>DE	555 DATA 30,FE,10,D8,D6,07,C9,00, 956	CU
275 DATA 11,BE,EF,CD,DD,E7,CD,81, 1437	>EA	560 :	ZE
280 DATA BB,06,00,21,38,FF,CD,06, 748	>CX	565 DATA 0C,18,3C,66,7E,60,3C,00, 480	BA
285 DATA BB,FE,0D,CA,9E,DF,CD,C1, 1435	>ED	570 DATA CF,EC,ED,ED,ED,04,FF, 1842	EC
290 DATA E7,77,23,04,7B,32,34,FF, 866	>BH	575 DATA F1,3A,BA,BB,BB,AB,23,FF, 1320	DX
295 DATA FE,D0,20,EA,C3,30,DF,00, 1194	>DL	580 DATA 0F,AA,BE,BA,BA,AA,BA,FF, 1310	EV
300 :	>YG	585 DATA E7,37,B7,37,F7,B7,23,FF, 1244	DQ
305 DATA 18,6C,18,00,11,30,F7,CD, 673	>BU	590 DATA 18,20,4D,45,44,4F,52,20, 463	EQ
310 DATA DD,E7,CD,81,BB,06,00,21, 1012	>DB	595 DATA 18,20,72,65,63,68,65,72, 589	BT
315 DATA 38,FF,CD,06,BB,FE,0D,28, 1016	>DN	600 DATA 63,68,65,20,65,6E,20,6D, 698	BH
320 DATA 55,CD,7F,DF,38,F4,CD,C1, 1338	>DZ	605 DATA FB,60,6F,69,72,65,20,2E, 869	CM
325 DATA E7,CD,E7,E7,CB,27,CB,27, 1382	>DC	610 DATA 2E,2E,00,FC,FD,FE,FF,00, 1106	DU
330 DATA CB,27,CB,27,4F,CD,06,BB, 961	>CY	615 DATA 0D,18,20,4D,4F,54,28,53, 432	BG
335 DATA CD,7F,DF,38,F8,CD,C1,E7, 1488	>EC	620 DATA 29,20,41,53,43,49,49,20, 466	BJ
340 DATA CD,E7,E7,81,77,04,78,32, 1089	>CH	625 DATA 18,20,20,20,42,4C,4F,43, 408	BC
345 DATA 34,FF,FE,D0,28,28,23,11, 901	>CK	630 DATA 20,48,45,58,41,20,00,0D, 371	BG
350 DATA 50,F7,CD,D3,E7,18,C3,CD, 1398	>DK	635 DATA 20,4D,4F,54,28,53,29,20, 468	BB
355 DATA 95,DF,FE,30,D8,FE,47,30, 1263	>DY	640 DATA 41,53,43,49,49,20,20,20, 457	BB
360 DATA 07,FE,3A,38,06,FE,41,C9, 901	>CF	645 DATA 18,20,42,4C,4F,43,20,48, 448	BV
365 DATA FE,FF,C9,A7,C9,FE,61,D8, 1645	>EH	650 DATA 45,58,41,20,18,00,0D,0A, 301	BF
370 DATA FE,67,D0,D6,20,C9,CD,84, 1349	>DW	655 DATA 0A,4D,4F,54,28,53,29,20, 446	BQ
375 DATA BB,11,53,F7,CD,DD,E7,CD, 1396	>DV	660 DATA 41,53,43,49,49,20,28,32, 483	BP
380 DATA 81,BB,CD,CC,DF,ED,53,30, 1316	>DB	665 DATA 30,38,20,43,41,52,2E,4D, 473	BF
385 DATA FF,CD,84,BB,11,69,F7,CD, 1353	>DH	670 DATA 41,58,49,29,3A,20,00,00, 357	BR
390 DATA DD,E7,CD,81,BB,CD,CC,DF, 1605	>EE	675 :	AB
395 DATA ED,53,32,FF,11,7D,F7,CD, 1219	>DM	680 DATA 0D,0A,0A,42,4C,4F,43,20, 353	BX
400 DATA DD,E7,18,29,CD,D5,DF,57, 1245	>DJ	685 DATA 48,45,58,41,20,28,32,30, 464	BR
405 DATA CD,D5,DF,5F,C9,CD,E6,DF, 1595	>EH	690 DATA 38,20,4F,43,54,45,54,53, 554	BL
410 DATA CB,27,CB,27,CB,27,CB,27, 968	>CN	695 DATA 20,4D,41,58,29,3A,20,00, 393	BN
415 DATA 4F,CD,E6,DF,81,C9,CD,06, 1278	>DH	700 DATA 20,20,00,0D,0A,0A,41,44, 230	BG
420 DATA BB,CD,7F,DF,38,F8,CD,C1, 1444	>DF	705 DATA 52,45,53,53,45,20,49,4E, 569	BP
425 DATA E7,CD,E7,E7,C9,CD,84,BB, 1623	>DB	710 DATA 49,54,49,41,4C,45,3A,20, 530	BL
430 DATA 18,00,2A,30,FF,C3,30,E7, 843	>BE	715 DATA 00,0D,0A,0A,41,44,52,45, 317	BH
435 :	>ZF	720 DATA 53,53,45,20,46,49,4E,41, 553	BC
440 DATA 22,35,FF,11,38,FF,3A,34, 780	>CK	725 DATA 4C,45,3A,20,00,0D,0A,0A, 268	BR
445 DATA FF,47,1A,BE,20,4A,13,23, 702	>CN	730 DATA 00,0D,0A,0A,4E,4F,55,56, 361	BA
450 DATA 10,FB,2A,35,FF,E5,CD,3C, 1106	>DP	735 DATA 45,4C,4C,45,20,52,45,43, 540	BP
455 DATA CB,3C,CB,3C,CB,3C,7C,CD, 1118	>DH	740 DATA 48,45,52,43,48,45,3A,20, 521	BX
460 DATA BA,E7,CD,C1,E7,E1,E5,7C, 1624	>DL	745 DATA 18,5B,45,4E,54,45,52,5D, 590	BA
465 DATA E6,0F,CD,BA,E7,CD,C1,E7, 1496	>EF	750 DATA 18,20,53,49,4E,4F,4E,20, 479	BW
470 DATA E1,E5,CB,3D,CB,3D,CB,3D, 1246	>DR	755 DATA 41,55,54,52,45,20,54,4F, 580	BH
475 DATA CB,3D,7D,CD,BA,E7,CD,C1, 1409	>EL	760 DATA 55,43,48,45,00,0D,0A,0A, 326	BT