

Et bien notre nouvelle rubrique démarre! Notre ambition est d'entreprendre un travail semblable à celui de Fabrice Broche pour la ROM de l'Oric. Mais nous ne parviendrons peut-être pas à un tel résultat! L'essentiel pour le moment serait de bien saisir toutes les routines du DOS, particulièrement celles des mots-clés. Ceci nous permettrait de remédier aux bogues du LINPUT par exemple, ou bien d'insérer de nouvelles routines (sauvegarde de tableaux par MSAVE...!). Disons, que ce style d'article s'adresse évidemment à des personnes soucieuses d'améliorer leurs connaissances du Sédoric, mais aussi du langage machine. Nous aimerions recevoir de l'aide de tous ceux qui se sentiront concernés et qui de leur côté se sont déjà frottés au Sédoric.

ANALYSE DES COMMANDES SEDORIC

Lors de l'initialisation, le noyau du code Sédoric est implanté en **RAMOVERLAY** (RAMOV, C000 à FFFF) et quelques modifications sont apportées aux pages 0, 2, et 4 de la RAM afin de permettre l'analyse des commandes Sédoric et d'accéder à cette RAMOV tout en préservant l'accès aux commandes BASIC (mêmes adresses, C000 à FFFF, mais en ROM).

Le coeur de l'analyseur de commande se trouve en ROM. Sans vouloir entrer dans le détail (voir "L'Oric à Nu" de Fabrice Broche), cet analyseur fait appel à une routine située en page zéro (routine CHARGET, 00E2 à 00F2 avec entrée secondaire en 00E8). Cette routine actualise le pointeur de texte (**TXTPTR**, 00E9/00EA) sur la ligne de commande (**TIB**, Terminal Input Buffer) ou bien sur la ligne de programme, lit le caractère présent au pointeur, exécute un saut en ROM à l'adresse d'entrée de l'interpréteur (JSR ECB9) suivit d'un retour au point d'appel initial (RTS). Sous Sédoric, le JSR en ROM et le RTS sont remplacés par un saut en page 4 (JMP 0400). C'est là qu'intervient la fameuse page 4 du Sédoric.

La routine **CHARGET** (00E2 ou 00E8) peut être appelée à plusieurs endroits à partir de la ROM, notamment en C90C ou en CA88. Dans ces deux cas, l'adresse de retour-1 est empilée. Si cette adresse est C90E, il s'agit de l'adresse de retour en C90F appartenant au s/p "exécuter une ligne" de l'interpréteur BASIC. Si cette adresse est CA8A, il s'agit de l'adresse de retour en CA8B appartenant au s/p "IF".

La routine en 0400 effectue plusieurs tâches:

Elle analyse si le caractère lu (et situé maintenant dans l'accumulateur A) est un chiffre. Si c'est le cas retour au cours normal des choses (c'est à dire à l'interpréteur en ECB9).

De même si A contient un code égal ou supérieur à #80 (c'est à dire un mot-clé BASIC) on retourne à l'interpréteur en ECB9.

Si ni l'un ni l'autre n'est le cas, les registres A et X sont sauvegardés en 000E et 000F avant de recevoir l'adresse présente sur la pile afin de savoir d'où CHARGET avait été appelé. Si aucune des deux adresses indiquées plus haut n'est trouvée, l'adresse de retour est remise en place sur la pile, les valeurs initiales des registres A et X sont restaurées et on retourne à l'interpréteur en ECB9 (tout se passe comme si le détour par la page 4 du Sédoric n'avait pas eu lieu).

Si l'adresse C90E indiquée plus haut est trouvée, le bit n°7 du drapeau 04FC est mis à zéro. Si c'est l'adresse CA8A ce

bit est mis à 1.

Puis la routine recherche si un signe "=" est présent sur la ligne de commande ou de programme et ce jusqu'au prochain "0" ou ":" marquant la fin de l'instruction. Si un signe "=" est rencontré, il s'agissait d'affecter une variable, l'adresse de retour est remise en place sur la pile, les valeurs initiales des registres A et X sont restaurées et on retourne à l'interpréteur en ECB9.

S'il n'y a pas de signe "=", on est en présence d'une commande Sédoric on continue sans restaurer l'adresse de retour sur la pile. S'agit-il d'un mot-clé utilisateur? (voir manuel Sédoric p106). Un JSR 04E9 est effectué, qui conduit à un JMP à l'adresse de l'interpréteur utilisateur ou à un simple RTS (cas général).

Un JSR 0467 est alors effectué (entrée vecteur "!"). Une bascule sur la RAMOV est opérée, puis un saut au sous-programme D3AE (**INTERPRETEUR SEDORIC** d'où l'on reviendra par un RTS), enfin une bascule sur la ROM permet de reprendre le cours normal de la routine en 0447 où le flag 04FC est testé.

Si le bit n°7 de ce flag est nul un JMP C8C1 (s/p exécuter une ligne) est effectué. Sinon (bit n°7 à 1), un "IF" est en cours et on met à 1 le bit n°7 du flag 0252 (drapeau "IF en cours").

Le RTS final achève ce s/p 0400 et permet de retourner au programme appelant.

DESASSEMBLAGE PAGE 4 SEDORIC

COMPLEMENT INTERPRETEUR BASIC:

(en fait s'intercale entre fin s/p 00E2 et début s/p ECB9)

0400-	C9 30	CMP "0"	si A est un chiffre (de 0 à 9)
0402-	90 04	BCC 0408	on retourne à ECB9 en ROM
0404-	C9 3A	CMP ":"	sinon on continue en 0408
0406-	90 35	BCC 043D	
0408-	86 0F	STX 0F	{ les registres A et X sont
040A-	AA	TAX	{ sauvegardés en 0E et 0F
040B-	30 2E	BMI 043B	{ si A est négatif (token BASIC)
040D-	85 0E	STA 0E	{ récupère A et X et -> ECB9
040F-	68	PLA	} prend adr sur pile
0410-	AA	TAX	} -> A (HH) et X (LL)
0411-	68	PLA	}
0412-	48	PHA	} remet HH sur la pile
0413-	E0 0E	CPX #0E	
0415-	D0 04	BNE 041B	si adr était C90E met à 0
0417-	C9 C9	CMP #C9	le b7 du drapeau 04FC
0419-	F0 09	BEQ 0424	et continue en 0428
041B-	E0 8A	CPX #8A	{
041D-	D0 18	BNE 0437	{ si adr était CA8A met à 1
041F-	C9 CA	CMP #CA	{ le b7 du drapeau 04FC
0421-	D0 14	BNE 0437	{ et continue en 0428
0423-	24 18	BIT 18	-> continue en 0425
0424-	18	CLC	
0425-	6E FC 04	ROR 04FC	si ni C90E ni CA8A -> 0437
0428-	A0 FF	LDY #FF	
042A-	C8	INY	s'il y a un "=" d'ici la fin
042B-	B1 E9	LDA (E9),Y	de la commande (marquée par
042D-	F0 11	BEQ 0440	"0" ou par ":") il s'agit
042F-	C9 3A	CMP "0"	d'une affectation basic
0431-	F0 0D	BEQ 0440	on remet LL sur la pile
0433-	C9 D4	CMP ":"	(HH y est déjà)
0435-	D0 F3	BNE 042A	on récupère les valeurs

```

0437- 8A      TXA      | d'origine de A et X
0438- 48      PHA      | et on retourne en ECB9
0439- A5 0E   LDA      0E   |
043B- A6 0F   LDX      0F   | sinon on continue en 0440
043D- 4C B9 EC JMP      ECB9 |
0440- 68      PLA      } on retire HH de la pile
0441- 20 E9 04 JSR      04E9 { vers vecteur utilisateur
0444- 20 67 04 JSR      0467 } vers vecteur "!" Sédoric
0447- 0E FC 04 ASL      04FC | teste b7 du flag 04FC
044A- B0 03   BCS      044F | s'il est à 0 continue en C8C1
044C- 4C C1 C8 JMP      C8C1 | (sous-prog ROM exécuter ligne)
044F- 6E 52 02 ROR      0252 | sinon met à 1 le b7 du flag
0452- 60      RTS      | "IF" (0252) et fin s/p 0400

```

NB: Lorsque le s/p 00E2 est appelé au point C90C, l'adresse de retour-1 C90E est empilée, s'il est appelé en C4C1, C4C3 est empilé. Enfin, lorsque 00E8 est appelé en CA88, à partir du s/p "IF", l'adresse CA8A est empilée.

GESTION DU VECTEUR D'EXECUTION

```

0453- 20 77 04 JSR      0477 { entrée appelée du s/p D81C
0456- B1 16   LDA      (16),Y { bascule ROM/RAMOV, lecture en
0458- 4C 77 04 JMP      0477 { page zéro, bascule ROM/RAMOV
045B- A9 45   LDA      #45   | s/p EB78 modifié
045D- A0 D8   LDY      #D8   | initialise pour exécution
045F- D0 0A   BNE      046B | en D845
0461- A9 8E   LDA      #8E   { entrée vecteur "&()"
0463- A0 F8   LDY      #F8   { initialise pour exécution
0465- D0 04   BNE      046B { en F88E
0467- A9 AE   LDA      #AE   | entrée vecteur "!" initialise
0469- A0 D3   LDY      #D3   | pour exécution en D3AE
046B- 8D F0 04 STA      04F0 { mise à jour du l'adresse du
046E- 8C F1 04 STY      04F1 { s/p à exécuter (EXEVEC+1)
0471- 20 77 04 JSR      0477 | entrée s/p EXERAM: bascule
0474- 20 EF 04 JSR      04EF | ROM/RAMOV, vecteur EXEVEC
0477- 08      PHP      | entrée s/p bascule ROM/RAMOV
0478- 48      PHA      | (n'affecte aucun registre)
0479- 78      SEI      |
047A- AD FB 04 LDA      04FB | il existe une autre version:
047D- 49 02   EOR      #02   | LDA 0321 EOR #06
047F- 8D FB 04 STA      04FB | STA 0321
0482- 8D 14 03 STA      0314 | PLA PLP RTS
0485- 68      PLA      | qui est plus courte et se
0486- 28      PLP      | termine en 0484, la zone
0487- 60      RTS      | 0485 à 0487 est donc libre

```

NB: ce s/p permet d'exécuter notamment les routines F590, D3AE, D136, EC17, F88E, D845

NOUVEL IRQ (REMPLECE EE22 DE ROM)

```

0488- 2C 0D 03 BIT      030D (s/p utilisé à partir de la ROM)
048B- 50 0F   BVC      049C
048D- 48      PHA      |
048E- A9 04   LDA      #04   |
0490- 2D 6A 02 AND      026A |
0493- F0 03   BEQ      0498 |
0495- EE 74 02 INC      0274 |
0498- 68      PLA      |
0499- 4C 22 EE JMP      EE22 on est bien sur la ROM
049C- 68      PLA      |
049D- 68      PLA      |
049E- 85 F2   STA      F2   |
04A0- 68      PLA      |

```

```

04A1- AA          TAX
04A2- A9 36      LDA #36      | initialisation pour execution
04A4- A0 D1      LDY #D1      | du s/p D136 en RAMOV
04A6- D0 C3      BNE 046B     | avec retour sur la ROM

```

NOUVEAU COLDSTART (REMPLECE FFFC et donc F88F)

```

04A8- 20 F2 04 JSR 04F2      bascule ROM/RAMOV
04AB- 68          PLA          dépile
04AC- 40          RTI          retour d'interruption
04AD- 8D 14 03 STA 0314      flag ROM/RAMOV
04B0- 6C FC FF JMP (FFFC)     vecteur colstart ROM

```

S/P IRQRAM et NMIRAM

```

04B3- 18          CLC entrée IRQRAM (04B4 = entrée NMIRAM)
04B4- 20 77 04 JSR 0477      bascule ROM/RAMOV
04B7- 48          PHA          empile A
04B8- A9 04      LDA #04
04BA- 48          PHA
04BB- A9 A8      LDA #A8
04BD- 48          PHA          empile adr 04A8 (coldstart)
04BE- 08          PHP          empile indicateurs d'état
04BF- B0 03      BCS 04C4     si NMI saute en 04C4
04C1- 4C 44 02 JMP 0244     si IRQ -> 0244 soit en 0488
04C4- 20 B8 F8 JSR F8B8     nouvel NMI: en ROM F8B8 au lieu
04C7- A9 17      LDA #17     de F8B2 (saute warmstart BASIC)
04C9- A0 EC      LDY #EC
04CB- 20 6B 04 JSR 046B     exécute s/p EC17 sur RAMOV
04CE- 4C 71 C4 JMP C471     exécute enfin warmstart BASIC

```

????????????????

```

04D1- A9 04      LDA #04
04D3- 48          PHA
04D4- A9 F1      LDA #F1
04D6- 48          PHA          empile adr F104
04D7- 8A          TXA
04D8- 48          PHA          empile X
04D9- 98          TYA
04DA- 48          PHA          empile Y
04DB- 20 F2 04 JSR 04F2     bascule ROM/RAMOV
04DE- 4C 06 D3 JMP D306     (trouver le tableau s/p ROM?)

```

VECTEURS et DRAPEAUX

```

04E1- EA EA EA EA EA EA EA EA NOP NOP NOP NOP NOP NOP NOP NOP
04E9- 4C 87 04 JMP 0487     DETE2C vecteur utilisateur
04EC- 4C 71 04 JMP 0471     EXERAM execution de s/p EXEVEC+1 sur
          RAMOV (si ROM active) ou sur ROM (si RAMOV active)
04EF- 4C 00 00 JMP 00E8     EXEVEC vecteur execution, dont..
04F0- 00 00          adresse execution
04F2- 4C 77 04 JMP 0477     RAMROM bascule RAMOV/ROM
04F5- 4C B3 04 JMP 04B3     IRQRAM execution IRQ sur RAM
04F8- 4C B4 04 JMP 04B4     NMIRAM execution NMI sur RAM
04FB- 94          flag ROM/RAMOV
04FC- 00          BRK          flag C90E/CA8A
04FD- 00          BRK          numéro de l'erreur
04FE- 00          BRK          numéro de la ligne de l'erreur
0500- 00          BRK          début BASIC

```

Yann Legrand, 608, rue de l'Eglise, "Les Templiers", 62610
LANDRETHUN LES ADRES et André Chéramy, 54, rue de Sours, 28000
CHARTRES