Carte mémoire système MC09®

Adresses		*Fonctions*	*Supports*	*Octets*
E000-FFFF		**EPROM 2732 moniteur**	Z6 Y7	8K
FFFE—>F020		Vecteur RESET		
FFFC—>FFEB		Vecteur NMI interruption non masquable		
FFF8—>FFE6		Vecteur IRQ interruption		
FFF6—>FFE1		Vecteur FIRQ interruption rapide		
FFF4—>FFDC		Vecteur SW2 interruption logicielle		
FFF2—>FFD7		Vecteur SW3 interruption logicielle		
FFB0-…		IO_ERR message d'erreur		
FF60-…		IO_OUT affichage		
FF00-…		IOSPRO clavier et affichage		
FE9B-…		SC_CV7 convert clavier/HEXA & 7-seg		
F6E0-…		MECV7 convert HEXA/7-segments		
C000-DFFF		**EPROM 2732 communication**	Z7 Y6	8K
C506-…		Réception 7 bits MINITEL		
C504-…		Émission 7 bits MINITEL		
C502-…		Réception 8 bits		
C500-…		Émission 8 bits		
A000-BFFF		**EPROM 2732 moteurs**	Z8 Y5	8K
A002-…		Moteur Pas à Pas		
A000-…		Moteur à Courant Continu		
8000-9FFF		**Entrées/Sorties**	Y4	8K
9C00-9FFF		Connecteur 2, face A, patte 20	Y'7	1K
9800-9BFF		Connecteur 2, face A, patte 19	Y'6	1K
9400-97FF		Connecteur 2, face A, patte 18	Y'5	1K
9400		CAN 7574 : 0/10V	B1	
9480		CAN 7574 : -10/+10V	B2	
9500		CNA 7224 : 0/10V	B4	
9580		CNA 7224 : -10/+10V	B3,5	
93FF-9000		PIA 6821 clavier+affich	Z11 Y'4	1K
9003		CRB (reg. de contrôle B)		
9002	CRB2=1	ORB (reg. de données B)		
	CRB2=0	DDRB(reg. sens de transfert données B)		
9001		CRA (reg. de contrôle A)		
9000	CRA2=1	ORA (reg. de données A)		
	CRA2=0	DDRA(reg. sens de transfert données A)		
8C00-8FFF		ACIA 6850	Z5 Y'3	1K
8C01		TDR, RDR (reg. d'émission/réception)		
8C00		reg. de contrôle, reg. d'état		
8800-8BFF		TIMER 6840	Z4 Y'2	1K
8806-8807		données timer 3		
8804-8805		données timer 2		
8802-8803		données timer 1		
8801		CR2 (reg. contrôle 2), INT (reg. d'état)		
8800	CR20=1	CR1 (reg. contrôle 1)		
	CR20=0	CR3 (reg. contrôle 3)		
8400-87FF		VIA 6522	Z3 Y'1	1K
840F		RB (reg. données B sans handshake)		
840E		IER (reg. d'interruption Enable)		
840D		IFR (reg. d'interruption Flag)		
840C		PCR (reg. contôle périphérique)		
840B		ACR (reg. contrôle auxiliaire)		
840A		SR (reg. à décalage)		

ISBN 2-9520781-1-4 http://www.syscope.net EAN 9782952078115

```
          8408-8409                T2C (timer 2 compteur)
          8406-8407                T1L (timer 1 latch)
          8404-8405                T1C (timer 1 compteur)
          8403                     DDRA (reg. transfert données A)
          8402                     DDRB (reg. transfert données B)
          8401                     RA (reg. données A)
          8400                     RB (reg. données B)
     8000-83FF                PIA 6821 utilisateur                          Z2  Y'0  1K
          8003                     CRB (reg. de contrôle B)
          8002  CRB2=1             ORB (reg. de données B)
                CRB2=0             DDRB( reg. sens de transfert données B)
          8001                     CRA (reg. de contrôle A)
          8000  CRA2=1             ORA (reg. de données A)
                CRA2=0             DDRA( reg. sens de transfert données A)

6000-7FFF              Connecteur 2, face A, patte 16                      Y3   8K

4000-5FFF              Connecteur 2, face A, patte 17                      Y2   8K

2000-3FFF              Connecteur 2, face A, patte 21                      Y1   8K

0000-1FFF                                                                  Y0   8K
     1000-1FFF                *RAM image*
     0800-0FFF                RAM 6116 haut                               Z9        2K
     0000-07FF                RAM 6116 bas                                Z10       2K
          0F60-0FFF                RAM moniteur                                     160
               0FF0-0FFB      Pile système (resp. CC,A,B,DP,X,Y,U,PC)
               0F71                Afficheur 6 (gauche)
               0F70                Afficheur 5
               0F6F                Afficheur 4
               0F6E                Afficheur 3
               0F6D                Afficheur 2
               0F6C                Afficheur 1 (droit)
               0F68                IRQ (interruption)
               0F66                FIRQ (")
               0F64                SW2  (")
               0F62                SW3  (")
               0F60                NMI  (")
          0000-0F5F                RAM utilisateur                                  3936
               0F00-0F5F      page $F (registre DP)                                96
               0E00-0EFF      page $E (DP)                                         256
               …/…                …/…
               0100-01FF      page $1 (DP)                                         256
               0000-00FF      page $0 (DP)                                         256
```

ISBN 2-9520781-1-4 http://www.syscope.net EAN 9782952078115

Brochage connecteurs système MC09®

	CONNECTEUR 1 (E/S)				CONNECTEUR 2 (µP)			
	FACE A		**FACE B**		**FACE A**		**FACE B**	
1	$\overline{\text{CTS}}$	ACIA	TxD	ACIA	$\overline{\text{NMI}}$	Bus contrôle	A15	Bus adresse
2	VCA2	VIA	$\overline{\text{RTS}}$	"	$\overline{\text{IRQ}}$	"	A14	"
3	VCA1	"	RxD	"	$\overline{\text{FIRQ}}$	"	A13	"
4	VA0	"	$\overline{\text{G2}}$	Timer	$\overline{\text{DMABREQ}}$		A12	"
5	VA1	"	O2	"	MRDY	"	A11	"
6	VA2	"	$\overline{\text{C2}}$	"	$\overline{\text{RST}}$	"	A10	"
7	VA3	"	$\overline{\text{G3}}$	"	$\overline{\text{HLT}}$	"	A9	"
8	VA4	"	O3	"	Q	"	A8	"
9	VA5	"	$\overline{\text{C3}}$	"	E	"	A0	"
10	VA6	"	PA0	PIA	R / $\overline{\text{W}}$	"	A1	"
11	VA7	"	PA1	"	BA	"	A2	"
12	VB0	"	PA2	"	BS	"	A3	"
13	VB1	"	PA3	"	Y7	Décodage	A4	"
14	VB2	"	PA4	"	Y6	"	A5	"
15	VB3	"	PA5	"	Y5	"	A6	"
16	VB4	"	PA6	"	Y3	"	A7	"
17	VB5	"	PA7	"	Y2	"	D7	Bus données
18	VB6	"	PB0	"	Y'7	"	D6	"
19	VB7	"	PB1	"	Y'6	"	D5	"
20	VCB1	"	PB2	"	Y'5	"	D4	"
21	VCB2	"	PB3	"	Y1	"	D3	"
22	CA2	PIA	PB4	"	Y0	"	D2	"
23	CA1	"	PB5	"	Vcc	Alim.	D1	"
24	Masse		PB6	"	Masse	"	D0	"
25	Vcc	Alim.	PB7	"	Masse	"	Masse	

ISBN 2-9520781-1-4 http://www.syscope.net EAN 9782952078115

Carte mémoire EPROM TP

Bibliothèque de sous-programmes

Note : le caractère "x" est égal à :
0 si les sous-prammes sont implantés en RAM (adresses $0000 à $0EFF)
A si les sous-prammes sont implantés en EPROM, support Z8 (adresses $A000 à $AEFF)
C si les sous-prammes sont implantés en EPROM, support Z7 (adresses $C000 à $CEFF)

Abréviations :

7-seg	code afficheurs 7 segments
hexa	code héxadécimal
bin	code binaire
clav	clavier
convert.	conversion de code
(…)	contenu de …
aff. 1…6	afficheurs n° 1 à 6 (de gauche à droite)
tc	touche clavier

Sous-programmes d'entrée/sortie clavier/afficheurs

Adresse	*Nom*	*Fonctionnalités*	*Paramètres d'entrée*	*Paramètres de sortie*
x100	**esclav**	-Affiche un message -Saisit une touche	(X) = adresse message 7-seg à afficher	(B) = code clavier ($0F50) idem
x140	**estemp**	Affiche un message pendant 1/100e s	idem	néant
x180	**erreur**	affiche message d'erreur pendant 3 secondes	néant	néant
x1C0	**esnum**	-Affiche un message -Saisit une touche num -Erreur si touche non num	idem esclav	(A) = val hexa touche (B) = code 7 segments
x200	**input1**	-Affiche un message -Saisit 1 octet hexa -Echo sur aff. 5&6	idem	(A) = octet saisi
x2C0	**printd**	affiche contenu hexa des reg A et B	(D) : octets à afficher	fonctionne comme esclav

Sous-programmes de conversions de codes

Adresse	*Nom*	*Fonctionnalités*	*Paramètres d'entrée*	*Paramètres de sortie*
x400	**abs1**	valeur absolue (8 bits)	(A) code compl. à 2	(A) code bin. naturel
x440	**abs2**	valeur absolue (16 bits)	(D) code compl. à 2	(D) code bin. naturel
x480	**clavhe**	convert. clav./hexa	(B) = digit code clav.	(A) = octet code hexa
x4C0	**heseg**	convert. hexa/7-seg	(A) = octet code hexa	(D) = 2 octets 7-seg
x500	**heasc**	convert hexa/ASCII	(A) = octet code hexa	(D) = 2 octets ASCII
x540	**asche**	convert ASCII/hexa	(A) = octet code ASCII	(A) = 1 digit hexa

ISBN 2-9520781-1-4 http://www.syscope.net EAN 9782952078115

x560	**ascseg**	convert ASCII/7-seg	(A) = octet code ASCII	(A) = octet code 7-seg
x600	**hebcd1**	convert hexa/BCD 8 bits	(A) = octet hexa ≤ $63	(A) = octet BCD ≤ $99
x640	**hebcd2**	convert hexa/BCD 16 bits	(D) = octets hexa ≤ $270F	(D) = octet BCD ≤ $9999
x6C0	**bcdhe1**	convert BCD/hexa	(A) = octet BCD ≤ $99	(A) = octet hexa ≤ $63
x700	**bcdhe2**	convert BCD/hexa	(D) = octets BCD ≤ $9999	(D) = octet hexa ≤ $270F

Utilitaires

Adresse	*Nom*	*Fonctionnalités*	*Paramètres d'entrée*	*Paramètres de sortie*
x800	**copabs**	transfert d'octets	(X) = adresse départ (Y) = adr fin du tableau (U) = adresse destination	néant
x820	**coprel**	transfert d'octets	(X) = adresse départ (Y) = nb d'octets à transférer (U) = adresse destination	
x840	**ipia**	initialisation PIA	(A) = sens des données (Y) = adresse du port	néant
x880	**itimer**	initialisation TIMER 1	(A) = format (X) = compteur	néant
x8C0	**iuart**	initialisation ACIA	(A) = format (X) = compteur timer 1	néant
x900	**emiss1**	émission d'un caractère	(A) = car. à émettre	néant
x940	**recep1**	réception d'un caractère	néant	(A) = caractère reçu (B) = bits PE,OVRN,FE
x980	**emibin**	émission d'une chaîne de caractères, format binaire	(X) = adresse chaîne (Y) = longueur chaîne	néant
x9A0	**emiasc**	émission d'une chaîne de caractères, format ASCII	(X) = adresse chaîne (Y) = longueur chaîne	néant
x9C0	**recstr**	réception d'une chaîne	(X) = adresse chaîne	chaîne pointée par X (Y) = longueur chaîne (B) = bits PE,OVRN,FE
xA00	**tempo**	temporisation	(X) = nb de $1/1000^{e}$ s	néant
xA40	**muls**	multiplication signée	(A), (B) : opérandes	(D) = résultat
xA80	**div**	division entière	(X) = dividende (D) = diviseur	(X) = quotient (D) = reste
xAC0	**divs**	division entière signée	idem	idem

ISBN 2-9520781-1-4 http://www.syscope.net EAN 9782952078115

Exécutables

xB00	**claexe**	Examen du code clavier
xB40	**segexe**	Examen du code 7 segments
xB80	**ascexe**	Examen du code ASCII et équivalent 7 segments
xC00	**prog0**	Génère un signal continu
xC0B	**prog1**	Génère un signal rectangulaire
xC24	**prog1B**	Génère un signal rectangulaire MLI
xC3E	**prog2**	Génère un signal en dent de scie
xC4B	**prog3**	Génère un signal quelconque à partir d'une table de valeurs
xC67	**prog4**	Recopie sur la CNA le signal présent à l'entrée du CAN
xC73	**prog5**	Idem avec contrôle de la période d'échantillonnage par interruption
xC91	**prog6**	Filtre non récursif, dérivateur pur
xCBF	**prog7**	Filtre non récursif, interpolateur (moyenne glissante sur 2 échantillons)
xCED	**prog8**	Filtre récursif passe-bas 1er ordre
xD21	**tor**	Comparateur à hystérésis
xD49	**pap**	Commande de moteur pas-à-pas, 2 phases, bipolaire
xD70	**daa1**	Acquisition de données non synchronisée
xDC0	**daa2**	Acquisition de données synchronisée
xE40	**pgm3**	Chronomètre, tempo logicielle
xE97	**chro_5**	Chronomètre, tempo matérielle
xF00	**prog5b**	Idem prog5 avec contrôle du délai d'acquisition
xF29	**prog5c**	Correcteur intégral pur
xF5C	**prog5d**	Correcteur proportionnel et intégral

ISBN 2-9520781-1-4 http://www.syscope.net EAN 9782952078115

Listing des sous-programmes EPROM TP

```
                        * RAM utilisateur
0F50                    clvier  EQU     $0F50   mem tampon clavier
0F51                    affich  EQU     $0F51   mem message 6 digits 7-seg
0F57                    buffer  EQU     $0F57   mém réservée pour les ss-prog

                        * RAM système
0F60                    vnmi    EQU     $0F60   vecteur d'interruption NMI
0F66                    vfirq   EQU     $0F66   vecteur d'interruption FIRQ

                        * Ports d'entrée/sortie
8000                    rapiau  EQU     $8000   reg données port A PIA util
8001                    capiau  EQU     $8001   reg contrôle port A PIA util
8002                    rbpiau  EQU     $8002   reg données port B PIA util
8003                    cbpiau  EQU     $8003   reg contrôle port B PIA util

8800                    crtim1  EQU     $8800   registre de contrôle timer 1
8801                    crtim2  EQU     $8801   registre de contrôle timer 2
8802                    drtim1  EQU     $8802   registre de données timer 1
8804                    drtim2  EQU     $8804   registre de données timer 2
8806                    drtim3  EQU     $8806   registre de données timer 3

8C00                    cracia  EQU     $8C00   registre de contrôle ACIA
8C01                    dracia  EQU     $8C01   registre de données ACIA

9400                    can_un  EQU     $9400   CAN unipolaire 0/10V (B1)
9480                    can_bi  EQU     $9480   CAN bipolaire -10/+10V (B2)
9500                    cna_un  EQU     $9500   CNA unipolaire 0/10V (B4)
9580                    cna_bi  EQU     $9580   CNA bipolaire -10/+10V (B5)

                        * Entrée/sortie clavier/afficheurs
                        * Affiche un message et saisit une touche
                        * E : (X) = adresse message 6 digits 7-seg
                        * S : (B) et ($OF50) = code clavier
                        *
A100                            ORG     $0100
A100 34 32              esclav  PSHS    A,X,Y
A102 17 00 3B           escl_1  LBSR    estemp
A105 4F                         CLRA
A106 10 8E 90 00                LDY     #rapias
A10A 17 07 33                   LBSR    ipia
A10D C6 05                      LDB     #$05
A10F 5C                 escl_2  INCB
A110 C1 09                      CMPB    #$09
A112 22 EE                      BHI     escl_1
A114 E7 22                      STB     2,Y
A116 A6 A4                      LDA     ,Y
A118 43                         COMA
A119 27 F4                      BEQ     escl_2
A11B C0 06                      SUBB    #$06
A11D F7 0F 50                   STB     clvier
A120 5F                         CLRB
A121 44                         LSRA
A122 CB 10              escl_3  ADDB    #$10
A124 44                         LSRA
A125 26 FB                      BNE     escl_3
A127 FA 0F 50                   ORB     clvier
A12A F7 0F 50                   STB     clvier
```

```
A12D 8E 07 FF           escl_4  LDX     #$07FF   anti-rebonds
A130 A6 A4              escl_5  LDA     ,Y
A132 43                         COMA
A133 26 F8                      BNE     escl_4
A135 30 1F                      LEAX    -1,X
A137 26 F7                      BNE     escl_5
A139 35 B2                      PULS    A,X,Y,PC


                        * Affiche message pendant 1/100e s
                        * E : (X) = adresse message 6 digits 7-seg
                        * S : néant
                        * tempo = 130 + 6(25 + 9x180) = 10000µs
                        *
A140                            ORG     $0140
A140 34 36              estemp  PSHS    A,B,X,Y         (JSR : 8µs +)    11
A142 10 8E 90 02                LDY     #rbpias                          +4
A146 86 0F                      LDA     #$0F                             +2
A148 17 06 F5                   LBSR    ipia                             +42
A14B 10 8E 90 00                LDY     #rapias                          +4
A14F 86 FE                      LDA     #$FE                             +2
A151 17 06 EC                   LBSR    ipia                             +42
A154 C6 05                      LDB     #$05                             +2
A156 A6 80              este_1  LDA     ,X+                       6
A158 43                         COMA                             +2
A159 E7 22                      STB     2,Y                      +5
A15B A7 A4                      STA     0,Y                      +5
A15D 86 B4                      LDA     #180                     +2
A15F 4A                 este_2  DECA                      2
A160 12                         NOP                      +2
A161 12                         NOP                      +2
A162 26 FB                      BNE     este_2           +3=9
A164 5A                         DECB                             +2
A165 2C EF                      BGE     este_1                   +3=25+9x180
A167 35 B6                      PULS    A,B,X,Y,PC                      +13=130

                        * Message d'erreur (pendant 3 secondes)
                        * E : néant
                        * S : néant
                        *
A180                            ORG     $0180
A180 34 16              erreur  PSHS    A,B,X
A182 C6 03                      LDB     #$03
A184 30 8D 00 19        erre_1  LEAX    afferr,PCR  message "Err-"...
A188 86 24                      LDA     #$24
A18A 17 FF B3           erre_2  LBSR    estemp  ...pendant 0,5s
A18D 4A                         DECA
A18E 26 FA                      BNE     erre_2
A190 30 8D 00 13                LEAX    affclr,PCR  message "     "
A194 86 24                      LDA     #$24
A196 17 FF A7           erre_3  LBSR    estemp  ...pendant 0,5s
A199 4A                         DECA
A19A 26 FA                      BNE     erre_3
A19C 5A                         DECB
A19D 26 E5                      BNE     erre_1   répéter 3 fois
A19F 35 96                      PULS    A,B,X,PC
A1A1 EC 88 88 80 00 00  afferr  FCB     $EC,$88,$88,$80,$00,$00
A1A7 00 00 00 00 00 00  affclr  FCB     $00,$00,$00,$00,$00,$00

                        * Saisie d'une touche numérique :
```

```
                        * -Affiche un message
                        * -Saisit une touche
                        * -Erreur si touche non numérique
                        * E : (X) = adresse du message 7-seg à afficher
                        * S : (A) = valeur hexadécimale de la touche
                        *     (B) = code 7 segments
                        *
A1C0                            ORG     $01C0
A1C0 17 FF 3D           esnum   LBSR    esclav
A1C3 C1 40                      CMPB    #$40
A1C5 24 05                      BHS     esnu_1
A1C7 17 FF B6                   LBSR    erreur
A1CA 20 F4                      BRA     esnum
A1CC 17 02 B1           esnu_1  LBSR    clavhe
A1CF 34 02                      PSHS    A
A1D1 17 02 EC                   LBSR    heseg
A1D4 35 02                      PULS    A
A1D6 39                         RTS

                        * Saisie d'un octet:
                        * -Affiche un message
                        * -Saisit un octet hexadécimal
                        * -Echo sur afficheurs 5 et 6
                        * E : (X) = adresse du message 7-seg à afficher
                        * S : (A) = octet saisi
                        *
A200                            ORG     $0200
A200 34 04              input1  PSHS    B
A202 17 FF BB                   LBSR    esnum
A205 48                         LSLA
A206 48                         LSLA
A207 48                         LSLA
A208 48                         LSLA
A209 34 02                      PSHS    A
A20B E7 04                      STB     4,X
A20D 6F 05                      CLR     5,X
A20F 17 FF AE                   LBSR    esnum
A212 AA E0                      ORA     ,S+
A214 E7 05                      STB     5,X
A216 35 84                      PULS    B,PC

                        * Affichage du contenu de A et B
                        * E : (D) : octets hexa à afficher
                        * S : ($OF50) : code clavier touche frappée
                        *
A2C0                            ORG     $02C0
A2C0 34 16              printd  PSHS    A,B,X
A2C2 8E 0F 51                   LDX     #affich
A2C5 17 01 F8                   LBSR    heseg
A2C8 ED 84                      STD     0,X
A2CA A6 61                      LDA     1,S
A2CC 17 01 F1                   LBSR    heseg
A2CF ED 02                      STD     2,X
A2D1 6F 04                      CLR     4,X
A2D3 6F 05                      CLR     5,X
A2D5 17 FE 28                   LBSR    esclav
A2D8 35 96                      PULS    A,B,X,PC

                        * Valeur absolue, 8 bits
                        * E : (A) = octet en code complément à deux
```

ISBN 2-9520781-1-4 http://www.syscope.net EAN 9782952078115

```
                        * S : (A) = octet en code binaire naturel
                        *
A400                            ORG     $0400
A400 4D                 abs1    TSTA
A401 2A 01                      BPL     abs_1
A403 40                         NEGA
A404 39                 abs_1   RTS

                        * Valeur absolue, 16 bits
                        * E : (D) = octet en code complément à deux
                        * S : (D) = octet en code binaire naturel
                        *
A440                            ORG     $0440
A440 4D                 abs2    TSTA
A441 2A 05                      BPL     abs_2
A443 43                         COMA
A444 53                         COMB
A445 C3 00 01                   ADDD    #$0001
A448 39                 abs_2   RTS

                        * Conversion code clavier/code hexadécimal
                        * E : (B) = code clavier à convertir
                        * S : (A) = octet $0x code héxadécimal
                        *
A480                            ORG     $0480
A480 34 04              clavhe  PSHS    B       Exemple : "B" = $72
A482 C0 40                      SUBB    #$40    $72-$40=$32=%0011 0010
A484 1F 98                      TFR     B,A     (code touche hexa >= $40)
A486 84 03                      ANDA    #$03    $32.$03 = %0000 0010
A488 48                         LSLA
A489 48                         LSLA            -> %0000 1000 = $08
A48A 54                         LSRB            %0011 0010 ->
A48B 54                         LSRB            ->
A48C 54                         LSRB            ->
A48D 54                         LSRB            -> %0000 0011
A48E C4 03                      ANDB    #$03    -> %0000 0011
A490 34 04                      PSHS    B
A492 AB E0                      ADDA    ,S+     -> %0000 1011 = $0B
A494 35 84                      PULS    B,PC

                        * Conversion hexadécimal/7 segments
                        * E : (A) = octet hexadécimal
                        * S : (D) = 2 octets 7-seg, poids fort, poids faible
                        *
A4C0                            ORG     $04C0
A4C0 34 10              heseg   PSHS    X
A4C2 30 8D 00 0E                LEAX    hese_1,PCR
A4C6 1F 89                      TFR     A,B
A4C8 44                         LSRA
A4C9 44                         LSRA
A4CA 44                         LSRA
A4CB 44                         LSRA
A4CC A6 86                      LDA     A,X
A4CE C4 0F                      ANDB    #$0F
A4D0 E6 85                      LDB     B,X
A4D2 35 90                      PULS    X,PC
A4D4 7E 12 BC B6 D2 E6  hese_1  FCB     $7E,$12,$BC,$B6,$D2,$E6
A4DA EE 32 FE F6 FA CE          FCB     $EE,$32,$FE,$F6,$FA,$CE
A4E0 6C 9E EC E8                FCB     $6C,$9E,$EC,$E8
```

ISBN 2-9520781-1-4 http://www.syscope.net EAN 9782952078115

```
                        * Conversion hexadécimal/ASCII
                        * E : (A) = octet hexadécimal
                        * S : (D) = 2 octets ASCII, poids fort, poids faible
                        * rappels : 0,...,9 = $30,...,$39
                        *           A,...,F = $41,...,$46
                        *
A500                            ORG     $0500
A500 1F 89              heasc   TFR     A,B
A502 C4 0F                      ANDB    #$0F     poids faible
A504 CB 30                      ADDB    #$30
A506 C1 39                      CMPB    #$39
A508 23 02                      BLS     heas_1
A50A CB 07                      ADDB    #$07     ajouter 7 si A,B,C,D,E,F
A50C 44                 heas_1  LSRA             poids fort
A50D 44                         LSRA
A50E 44                         LSRA
A50F 44                         LSRA
A510 8B 30                      ADDA    #$30
A512 81 39                      CMPA    #$39
A514 23 02                      BLS     heas_2
A516 8B 07                      ADDA    #$07
A518 39                 heas_2  RTS

                        * Conversion ASCII/hexa
                        * E : (A) = code ASCII d'un nb x = $0,...,$F
                        * S : (A) = digit hexadécimal $0x
                        *
A540                            ORG     $0540
A540 80 30              asche   SUBA    #$30
A542 81 09                      CMPA    #$09
A544 23 02                      BLS     asch_1
A546 80 07                      SUBA    #$07
A548 39                 asch_1  RTS

                        * Conversion ASCII/7-seg
                        * NB : la conversion est parfois approximative...
                        * et ne fait pas de différence entre minuscules
                        * et majuscules.
                        * E : (A) = octet code ASCII
                        * S : (A) = code 7-segment équivalent
                        *
A560                            ORG     $0560
A560 34 10              ascseg  PSHS    X
A562 30 8D 00 04                LEAX    ascs_1,PCR
A566 A6 86                      LDA     A,X
A568 35 90                      PULS    X,PC
A56A 00 00 00 00 00 00  ascs_1  FCB     $00,$00,$00,$00,$00,$00  car contrôle
A570 00 00 00 00 00 00          FCB     $00,$00,$00,$00,$00,$00
A576 00 00 00 00 00 00          FCB     $00,$00,$00,$00,$00,$00
A57C 00 00 00 00 00 00          FCB     $00,$00,$00,$00,$00,$00
A582 00 00 00 00 00 00          FCB     $00,$00,$00,$00,$00,$00
A588 00 00                      FCB     $00,$00
A58A 00 40 50 F0 E6 A4          FCB     $00,$40,$50,$F0,$E6,$A4   !"#$%
A590 7A 10 6C 36 F0 92          FCB     $7A,$10,$6C,$36,$F0,$92  &'()*+
A596 08 80 8E 98                FCB     $08,$80,$8E,$98          ,-./
A59A 7E 12 BC B6 D2 E6          FCB     $7E,$12,$BC,$B6,$D2,$E6  012345
A5A0 EE 32 FE F6                FCB     $EE,$32,$FE,$F6          6789
A5A4 24 22 8C 84 86 B8          FCB     $24,$22,$8C,$84,$86,$B8  :;<=>?
A5AA FC                         FCB     $FC                      @
A5AB FA CE 6C 9E EC E8          FCB     $FA,$CE,$6C,$9E,$EC,$E8  ABCDEF
```

ISBN 2-9520781-1-4 http://www.syscope.net EAN 9782952078115

```
A5B1 6E DA 12 16 D8 4C          FCB     $6E,$DA,$12,$16,$D8,$4C  GHIJKL
A5B7 7A 8A 7E F8 F2 88          FCB     $7A,$8A,$7E,$F8,$F2,$88  MNOPQR
A5BD E6 C8 5E 0E 0E DA          FCB     $E6,$C8,$5E,$0E,$0E,$DA  STUVWX
A5C3 D6 BC                      FCB     $D6,$BC                  YZ
A5C5 6C C2 36 70 C0 80          FCB     $6C,$C2,$36,$70,$C0,$80  [\]^_`
A5CB FA CE 6C 9E EC E8          FCB     $FA,$CE,$6C,$9E,$EC,$E8  abcdef
A5D1 6E DA 12 16 D8 4C          FCB     $6E,$DA,$12,$16,$D8,$4C  ghijkl
A5D7 7A 8A 7E F8 F2 88          FCB     $7A,$8A,$7E,$F8,$F2,$88  mnopqr
A5DD E6 C8 5E 0E 0E DA          FCB     $E6,$C8,$5E,$0E,$0E,$DA  stuvwx
A5E3 D6 BC                      FCB     $D6,$BC                  yz
A5E5 6C 80 36 80 00             FCB     $6C,$80,$36,$80,$00      {|}~


                        * Conversion hexadécimal/BCD, 8 bits
                        * E : (A) = octet hexa <= $63 (sinon, erreur)
                        * S : (A) = octet BCD <= 99
                        *
A600                            ORG     $0600
A600 34 04              hebcd1  PSHS    B
A602 81 64                      CMPA    #100    vérification si < 99
A604 25 05                      BLO     heb1_1
A606 17 FB 77                   LBSR    erreur
A609 20 16                      BRA     heb1_2
A60B 1F 89              heb1_1  TFR     A,B
A60D 4F                         CLRA
A60E 1F 01                      TFR     D,X
A610 C6 0A                      LDB     #$0A   X/10
A612 17 04 6B                   LBSR    div
A615 34 04                      PSHS    B      reste = poids faible
A617 1F 10                      TFR     X,D    quotient = poids fort
A619 58                         ASLB
A61A 58                         ASLB
A61B 58                         ASLB
A61C 58                         ASLB
A61D EA E0                      ORB     ,S+    digits poids 0+1
A61F 1F 98                      TFR     B,A
A621 35 84              heb1_2  PULS    B,PC

                        * Conversion hexadécimal/BCD 16 bits
                        * E : (D) = nb hexa <= $270F (sinon, erreur)
                        * S : (D) = nb BCD <= 9999
                        *
A640                            ORG     $0640
A640 10 83 27 10        hebcd2  CMPD    #10000  vérification si nb<=9999
A644 25 04                      BLO     heb2_1
A646 17 FB 37                   LBSR    erreur
A649 39                         RTS
A64A 1F 01              heb2_1  TFR     D,X
A64C 4F                         CLRA
A64D C6 0A                      LDB     #$0A   X/10, digit poids 0
A64F 17 04 2E                   LBSR    div
A652 34 04                      PSHS    B
A654 C6 0A                      LDB     #$0A   X/10, digit poids 1
A656 17 04 27                   LBSR    div
A659 58                         ASLB
A65A 58                         ASLB
A65B 58                         ASLB
A65C 58                         ASLB
A65D EA E4                      ORB     0,S   digits poids 0+1
A65F E7 E4                      STB     0,S
```

ISBN 2-9520781-1-4 http://www.syscope.net EAN 9782952078115

```
A661 C6 0A                      LDB     #$0A  X/10, digit poids 2
A663 17 04 1A                   LBSR    div
A666 34 04                      PSHS    B
A668 C6 0A                      LDB     #$0A  X/10, digit poids 3
A66A 17 04 13                   LBSR    div
A66D 58                         ASLB
A66E 58                         ASLB
A66F 58                         ASLB
A670 58                         ASLB
A671 EA E4                      ORB     0,S   digits poids 2+3
A673 E7 E4                      STB     0,S
A675 35 86              heb2_2  PULS    A,B,PC

                        * Conversion BCD/hexadécimal 8 bits
                        * E : (A) = nb BCD < 99
                        * S : (A) = équivalent hexadécimal
                        *
A6C0                            ORG     $06C0
A6C0 34 04              bcdhe1  PSHS    B
A6C2 1F 89                      TFR     A,B
A6C4 C4 0F                      ANDB    #$0F    sélection des unités
A6C6 34 04                      PSHS    B       et mise en mémoire
A6C8 44                         LSRA            sélection des dizaines
A6C9 44                         LSRA
A6CA 44                         LSRA
A6CB 44                         LSRA
A6CC C6 0A                      LDB     #$0A
A6CE 3D                         MUL             Ax10->D=$00XX
A6CF EB E0                      ADDB    ,S+     centaines+dizaines
A6D1 1F 98                      TFR     B,A     sortir le résultat ds A
A6D3 35 84                      PULS    B,PC

                        * Conversion BCD/hexadécimal 16 bits
                        * E : (D) = nb BCD < 9999
                        * S : (D) = équivalent hexadécimal
                        *
A700                            ORG     $0700
A700 34 10              bcdhe2  PSHS    X
A702 1F 01                      TFR     D,X     pds faible ds X (Lsb)
A704 17 FF B9                   LBSR    bcdhe1  pds fort (ds A) : BCD ->hexa
A707 C6 64                      LDB     #$64    x 100
A709 3D                         MUL
A70A 34 06                      PSHS    D       sauvegarde résultat
A70C 1F 10                      TFR     X,D     pds faible ds A
A70E 1F 98                      TFR     B,A
A710 17 FF AD                   LBSR    bcdhe1  pds faible : BCD -> hexa
A713 1F 89                      TFR     A,B     pds faible ds B
A715 4F                         CLRA            effacer A
A716 E3 E1                      ADDD    ,S++
A718 35 90                      PULS    X,PC

                        * Copie d'un tableau (transfert d'octets)
                        * E : (X) = adresse début du tableau
                        *     (Y) = adresse fin du tableau
                        *     (U) = adresse de destination (début du tableau)
                        * S : néant
                        *
A800                            ORG     $0800
A800 34 72              copabs  PSHS    A,X,Y,U   Y est sauvegardé ds la pile
A802 A6 80              copa_1  LDA     ,X+     copier et incrémenter X
```

ISBN 2-9520781-1-4 http://www.syscope.net EAN 9782952078115

```
A804 A7 C0                      STA     ,U+
A806 AC 63                      CMPX    3,S     comparer X à Y
A808 23 F8                      BLS     copa_1
A80A 35 F2                      PULS    A,X,Y,U,PC

                        * Copie d'un tableau (transfert d'octets)
                        * E : (X) = adresse début du tableau
                        *     (Y) = nb d'octets à copier
                        *     (U) = adresse de destination (début du tableau)
                        * S : néant
A820                            ORG     $0820
A820 34 72              coprel  PSHS    A,X,Y,U
A822 A6 80              copr_1  LDA     ,X+
A824 A7 C0                      STA     ,U+
A826 31 3F                      LEAY    -1,Y
A828 26 F8                      BNE     copr_1
A82A 35 F2                      PULS    A,X,Y,U,PC

                        * Initialisation PIA
                        * E : (A) = registre DDRx (sens de transfert données)
                        *     (Y) = adresse port PIA (ORx : reg de données)
                        * S : néant
                        *
A840                            ORG     $0840
A840 34 04              ipia    PSHS    B
A842 5F                         CLRB
A843 E7 21                      STB     1,Y     CRx2=0 => accès à DDRx
A845 A7 A4                      STA     0,Y     programmation de DDRx
A847 C6 04                      LDB     #$04
A849 E7 21                      STB     1,Y     CRx2=1 => accès à ORx
A84B 35 84                      PULS    B,PC

                        * Initialisation du timer 1
                        * E : (A) = registre de contrôle (CR) timer 1
                        *     (X) = compteur timer 1
                        * S : néant
                        *
A880                            ORG     $0880
A880 34 02              itimer  PSHS    A
A882 86 01                      LDA     #$01    bit 0 du CR timer 2 mis à 1
A884 B7 88 01                   STA     crtim2  => accès CR timer 1
A887 A6 E4                      LDA     ,S      programmation du timer 1
A889 B7 88 00                   STA     crtim1
A88C BF 88 02                   STX     drtim1  init compteur timer 1
A88F 35 82                      PULS    A,PC

                        * Initialisation de l'ACIA
                        * Fonctionne à la vitesse programmée sur le timer 1
                        * selon la relation :
                        *  2(X+1) = 1.000.000/f    avec f bauds
                        * E : (A) = format de communication (registre CR)
                        *     (X) = compteur timer 1
                        * S : néant
                        *
A8C0                            ORG     $08C0
A8C0 34 02              iacia   PSHS    A
A8C2 86 82                      LDA     #$82    programmation timer 1
A8C4 17 FF B9                   LBSR    itimer  avec (X) = compteur
A8C7 86 03                      LDA     #$03    initialisation de l'ACIA
A8C9 B7 8C 00                   STA     cracia
```

ISBN 2-9520781-1-4 http://www.syscope.net EAN 9782952078115

```
A8CC A6 E4                      LDA     ,S      programmation de l'ACIA
A8CE B7 8C 00                   STA     cracia
A8D1 35 82                      PULS    A,PC

                        * Emission d'un caractère
                        * NB : initialiser l'ACIA auparavant
                        * E : (A) = octet à émettre
                        * S : néant
                        *
A900                            ORG     $0900
A900 34 04              emiss1  PSHS    B
A902 C6 02                      LDB     #$02    reg transmission vide ?
A904 F5 8C 00           emis_1  BITB    cracia
A907 27 FB                      BEQ     emis_1  si oui, émettre la donnée
A909 B7 8C 01                   STA     dracia
A90C 35 84                      PULS    B,PC

                        * Réception d'un caractère
                        * NB : initialiser l'ACIA auparavant
                        * E : néant
                        * S : (A) = octet reáu
                        *     (B) = état des bits FE,OVRN,PE
                        *
A940                            ORG     $0940
A940 B6 8C 00           recep1  LDA     cracia  reg réception plein ?
A943 85 01                      BITA    #$01    si non, recommencer tant que
A945 27 F9                      BEQ     recep1          RDR est vide
A947 1F 89                      TFR     A,B
A949 C4 70                      ANDB    #%01110000      bits PE,OVRN,FE
A94B B6 8C 01                   LDA     dracia          donnée reáue
A94E 39                         RTS

                        * Emission d'une chaîne de caractères, format binaire
                        * NB : initialiser l'ACIA auparavant
                        * E : (X) = adresse chaîne
                        *     (Y) = longueur chaîne (nb d'octets à émettre)
                        * S : néant
                        *
A980                            ORG     $0980
A980 34 32              embin   PSHS    A,X,Y
A982 A6 80              embi_1  LDA     ,X+     charger un caractère...
A984 17 FF 79                   LBSR    emiss1   ...et l'émettre
A987 31 3F                      LEAY    -1,Y    décrémenter le compteur (Y)
A989 26 F7                      BNE     embi_1  jusqu'à la fin de la chaîne
A98B 35 B2                      PULS    A,X,Y,PC

                        * Emission d'une chaîne de caractères, format ASCII
                        * NB : initialiser l'ACIA auparavant
                        * E : (X) = adresse chaîne
                        *     (Y) = longueur chaîne (nb d'octets à émettre)
                        * S : néant
                        *
A9A0                            ORG     $09A0
A9A0 34 36              emiasc  PSHS    A,B,X,Y
A9A2 A6 80              emas_1  LDA     ,X+     charger un caractère...
A9A4 17 FB 59                   LBSR    heasc
A9A7 17 FF 56                   LBSR    emiss1
A9AA 1F 98                      TFR     B,A
A9AC 17 FF 51                   LBSR    emiss1   ...et l'émettre
A9AF 31 3F                      LEAY    -1,Y    décrémenter le compteur (Y)
```

ISBN 2-9520781-1-4 http://www.syscope.net EAN 9782952078115

```
A9B1 26 EF                      BNE     emas_1  jusqu'à la fin de la chaîne
A9B3 35 B6                      PULS    A,B,X,Y,PC


                        * Réception d'une chaîne de caractères
                        * Si l'émission cesse, la réception s'arrête automa-
                        *    tiquement après un temps de pause ~ 5 secondes.
                        * Erreurs de transmission signalées par S-P "erreur"
                        * NB : initialiser l'ACIA auparavant
                        * E : (X) = adresse de stockage de la chaîne
                        * S :  - chaîne pointée par X
                        *     (Y) = longueur de la chaîne (nb d'octets reáus)
                        *     (B) = état des bits FE,OVRN,PE en cas d'erreur
                        *
A9C0                            ORG     $09C0
A9C0 34 12              recstr  PSHS    A,X
A9C2 10 8E 00 00                LDY     #$0000  initialisation long. chaîne
A9C6 C6 04              recs_1  LDB     #$04    initialisation temps de pause
A9C8 CE FF FF                   LDU     #$FFFF
A9CB 33 5F              recs_2  LEAU    -1,U
A9CD 11 83 00 00                CMPU    #$0000
A9D1 26 06                      BNE     recs_3
A9D3 CE FF FF                   LDU     #$FFFF
A9D6 5A                         DECB
A9D7 27 17                      BEQ     recs_4
A9D9 B6 8C 00           recs_3  LDA     cracia      reg réception plein ?
A9DC 85 01                      BITA    #$01
A9DE 27 EB                      BEQ     recs_2
A9E0 1F 89                      TFR     A,B
A9E2 B6 8C 01                   LDA     dracia
A9E5 A7 80                      STA     ,X+         stocker la donnée reáue
A9E7 31 21                      LEAY    1,Y         incrémenter compteur
A9E9 C4 70                      ANDB    #%01110000      bits PE,OVRN,FE
A9EB 5D                         TSTB                test erreurs transm.
A9EC 26 04                      BNE     recs_5
A9EE 20 D6                      BRA     recs_1
A9F0 35 92              recs_4  PULS    A,X,PC
A9F2 17 F7 8B           recs_5  LBSR    erreur
A9F5 35 92                      PULS    A,X,PC


                        * Temporisation logicielle
                        * E : (X) = nb de 1/1000e sec (dont appel par JSR)
                        * minimum : 1 ms (X = $0001)
                        * maximum : 65535 ms (X = $FFFF)
                        * Sortie : néant
                        * Structure du programme : t(µs) = 1000 + 1000x(X-1)
                        *
AA00                            ORG     $0A00
AA00 34 14              tempo   PSHS    B,X                      8 µs
AA02 12                         NOP                             +2
AA03 12                         NOP                             +2
AA04 12                         NOP                             +2
AA05 5A                 temp_1  DECB            2               +96x10
AA06 10 21 F5 F6                LBRN    $0000   +5              ...
AA0A 26 F9                      BNE     temp_1  +3=10µs         ...
AA0C 30 1F              temp_2  LEAX    -1,X             5      +5
AA0E 27 10                      BEQ     temp_4          +3      +3
AA10 10 21 F5 EC                LBRN    $0000           +5
AA14 C6 62                      LDB     #98             +2
```

```
AA16 5A                 temp_3  DECB             2      +98x10
AA17 10 21 F5 E5                LBRN    $0000    +5     ...
AA1B 26 F9                      BNE     temp_3   +3=10µs ...
AA1D 16 FF EC                   LBRA     temp_2         +5=1000x(X-1)µs
AA20 35 94              temp_4  PULS    B,X,PC                  +10
                        *                                       + 8 (JSR)
                        *                                       = 1000µs = 1ms

                        * Multiplication signée
                        * E : (A) et (B) : opérandes en code complément à 2
                        * S : (D) : résultat en code complément à 2
                        *
AA40                            ORG     $0A40
AA40 7F 0F 57           muls    CLR     buffer   RAZ drapeau nb<0
AA43 4D                         TSTA             A >=0 ?
AA44 2A 04                      BPL     muls_1   non : continuer
AA46 40                         NEGA             oui : A -> -A
AA47 73 0F 57                   COM     buffer
AA4A 5D                 muls_1  TSTB             idem B
AA4B 2A 04                      BPL     muls_2
AA4D 50                         NEGB
AA4E 73 0F 57                   COM     buffer
AA51 3D                 muls_2  MUL
AA52 7D 0F 57                   TST     buffer
AA55 27 05                      BEQ     muls_3   si drapeau = 1,
AA57 43                         COMA             inverser le signe du produit
AA58 53                         COMB
AA59 C3 00 01                   ADDD    #$0001
AA5C 39                 muls_3  RTS


                        * Division 16 bits non signée
                        * E :   (X) = dividende
                        *       (D) = diviseur (erreur et fin si D=0)
                        * S :   (X) = quotient
                        *       (D) = reste
                        * structure de la pile :
                        *                                        <- S avant
                        *       compteur bits   YL
                        *                       YH
                        *       dividende       XL      3,S
                        *                       XH      2,S
                        *       diviseur        B       1,S
                        *                       A       0,S     <- S après
                        * dividende concaténé avec quotient sur 32 bits :
                        *   A : B : 2,S : 3,S  (avec décalage à gauche)
                        *
AA80                            ORG     $0A80
AA80 34 36              div     PSHS    A,B,X,Y
AA82 10 83 00 00                CMPD    #$0000   test division par zéro
AA86 26 05                      BNE     div_1
AA88 17 F6 F5                   LBSR    erreur
AA8B 20 19                      BRA     div_4
AA8D 10 8E 00 10        div_1   LDY     #16      compteur nb bits (<=16) divid
AA91 4F                         CLRA
AA92 5F                         CLRB
AA93 68 63              div_2   ASL     3,S      dec à g de dividende:quotient
AA95 69 62                      ROL     2,S      (32 bits à décaler)
AA97 59                         ROLB
AA98 49                         ROLA
```

ISBN 2-9520781-1-4 http://www.syscope.net EAN 9782952078115

```
AA99 10 A3 E4                   CMPD    0,S     comparer dividende à diviseur
AA9C 25 04                      BLO     div_3
AA9E A3 E4                      SUBD    0,S     si > : soustraire ...
AAA0 6C 63                      INC     3,S             et incrémenter quotient
AAA2 31 3F              div_3   LEAY    -1,Y    décrémenter nb bits
AAA4 26 ED                      BNE     div_2   et continuer si > 0
AAA6 32 62              div_4   LEAS    2,S
AAA8 35 B0                      PULS    X,Y,PC  restituer les résultats

                        * Division signée 16 bits
                        * E :   (X) = dividende en code complément à 2
                        *       (D) = diviseur en code complément à 2
                        * S :   (X) = quotient en code complément à 2
                        *       (D) = reste en code complément à 2
                        * NB : quotient < 0 si dividende x diviseur < 0
                        *      reste < 0 si dividende < 0
                        * exemples :    (+13)/(+4) <=> 13 = (+4)(+3)+(+1)
                        *               (-13)/(+4) <=> -13 = (+4)(-3)+(-1)
                        *               (+13)/(-4) <=> 13 = (-4)(-3)+(+1)
                        *               (-13)/(-4) <=> -13 = (-4)(+3)+(-1)
                        *
AAC0                            ORG     $0AC0
AAC0 7F 0F 57           divs    CLR     buffer  RAZ drapeaux nb < 0
AAC3 7F 0F 58                   CLR     buffer+1
AAC6 4D                         TSTA            diviseur < 0 ?
AAC7 2A 08                      BPL     divs_1  non : continuer
AAC9 43                         COMA            oui : D -> -D
AACA 53                         COMB
AACB C3 00 01                   ADDD    #$0001
AACE 73 0F 57                   COM     buffer  diviseur < 0 => buffer=$FF
AAD1 1E 01              divs_1  EXG     D,X
AAD3 4D                         TSTA            idem dividende
AAD4 2A 0B                      BPL     divs_2
AAD6 43                         COMA
AAD7 53                         COMB
AAD8 C3 00 01                   ADDD    #$0001
AADB 73 0F 57                   COM     buffer    divis x divid <0 => buf=$FF
AADE 73 0F 58                   COM     buffer+1  dividende<0 => buffer+1=$FF
AAE1 1E 01              divs_2  EXG     D,X
AAE3 17 FF 9A                   LBSR    div
AAE6 7D 0F 58                   TST     buffer+1  si dividende < 0,
AAE9 27 05                      BEQ     divs_3
AAEB 43                         COMA            inverser le signe du reste...
AAEC 53                         COMB
AAED C3 00 01                   ADDD    #$0001
AAF0 7D 0F 57           divs_3  TST     buffer  si divis OU (excl) divid <0
AAF3 27 09                      BEQ     divs_4
AAF5 1E 01                      EXG     D,X     inverser le signe du quotient
AAF7 43                         COMA
AAF8 53                         COMB
AAF9 C3 00 01                   ADDD    #$0001
AAFC 1E 01                      EXG     D,X
AAFE 39                 divs_4  RTS


*************************************************************************
                        * PROGRAMMES EXECUTABLES
*************************************************************************
```

ISBN 2-9520781-1-4 http://www.syscope.net EAN 9782952078115

```
                        * Code clavier
                        * Frapper une touche (sauf RST et NMI) : son identifi-
                        * cation n¯colonne/n¯ligne apparaît sur les afficheurs
                        * Arràt du programme par RST
                        *
AB00                            ORG     $0B00
AB00 8E 0F 51           claexe  LDX     #affich
AB03 6F 84                      CLR     0,X
AB05 6F 01                      CLR     1,X
AB07 6F 02                      CLR     2,X
AB09 6F 03                      CLR     3,X
AB0B 6F 04                      CLR     4,X
AB0D 6F 05                      CLR     5,X
AB0F 17 F5 EE           clax_1  LBSR    esclav
AB12 B6 0F 50                   LDA     clvier
AB15 17 F9 A8                   LBSR    heseg
AB18 ED 04                      STD     4,X
AB1A 20 F3                      BRA     clax_1

                        * Code 7 segments
                        * Entrer un octet, valider par EXC (FIN si erreur)
                        * L'équivalent 7 segments apparaît à droite
                        * Entrer un 2ème octet : un 2ème afficheur est activé
                        * Etc. Arràt par RST.
                        *
AB40                            ORG     $0B40
AB40 8E 0F 51           segexe  LDX     #affich
AB43 6F 84                      CLR     0,X
AB45 6F 01                      CLR     1,X
AB47 6F 02                      CLR     2,X
AB49 6F 03                      CLR     3,X
AB4B 6F 04                      CLR     4,X
AB4D 6F 05                      CLR     5,X
AB4F 17 F6 AE           segx_1  LBSR    input1
AB52 E6 01                      LDB     1,X
AB54 E7 84                      STB     0,X
AB56 E6 02                      LDB     2,X
AB58 E7 01                      STB     1,X
AB5A E6 03                      LDB     3,X
AB5C E7 02                      STB     2,X
AB5E A7 03                      STA     3,X
AB60 17 F9 5D                   LBSR    heseg
AB63 ED 04                      STD     4,X
AB65 20 E8                      BRA     segx_1

                        * Code ASCII
                        * Affichage de : code ASCII, symbole, code 7 segments
                        * équivalent
                        * Appuyer sur INC ou DEC pour faire défiler les codes
                        * Les afficheurs 7 segments ne permettent pas de
                        * distinguer les majuscules des minuscules.
                        *
AB80                            ORG     $0B80
AB80 8E 0F 51           ascexe  LDX     #affich
AB83 86 20                      LDA     #$20
AB85 B7 0F 57           asce_1  STA     buffer
AB88 17 F9 35                   LBSR    heseg   afficher code ASCII
AB8B ED 84                      STD     0,X
AB8D 6F 02                      CLR     2,X     blanc
AB8F B6 0F 57                   LDA     buffer
```

ISBN 2-9520781-1-4 http://www.syscope.net EAN 9782952078115

```
AB92 17 F9 CB                   LBSR    ascseg  afficher symbole
AB95 A7 03                      STA     3,X
AB97 17 F9 26                   LBSR    heseg   afficher code 7 segments
AB9A ED 04                      STD     4,X
AB9C 17 F5 61                   LBSR    esclav  entrer une touche
AB9F B6 0F 57                   LDA     buffer
ABA2 C1 30                      CMPB    #$30    INC
ABA4 27 0A                      BEQ     asce_2
ABA6 C1 31                      CMPB    #$31    DEC
ABA8 27 0F                      BEQ     asce_3
ABAA C1 11                      CMPB    #$11    FIN
ABAC 27 14                      BEQ     asce_4
ABAE 20 D5                      BRA     asce_1  sinon, recommencer
ABB0 4C                 asce_2  INCA            symbole suivant
ABB1 81 80                      CMPA    #$80    si code ASCII = fin table...
ABB3 26 D0                      BNE     asce_1
ABB5 86 20                      LDA     #$20    ...afficher début table
ABB7 20 CC                      BRA     asce_1
ABB9 4A                 asce_3  DECA            symbole précédant
ABBA 81 1F                      CMPA    #$1F    si code ASCII = début table...
ABBC 26 C7                      BNE     asce_1
ABBE 86 7F                      LDA     #$7F    ...afficher fin table
ABC0 20 C3                      BRA     asce_1
ABC2 3F                 asce_4  SWI


                        * Fin...
                        *
ABFF                            ORG     $0BFF
ABFF 3F                         SWI

                        * TP initiation au traitement numérique du signal
                        *

AC00                            ORG     $AC00
                        * NB : au début de chaque programme figure la liste
                        * des adresses des paramètres d'entrée.
                        * Entrer ces paramètres au clavier avant exécution


                        * 2-0 : - génère un signal continu
                        *       - échelle : [$00,$FF]  <->  [+10V,-10V]
                        *       - sortie CNA : borne 5
                        *
                        * $0F02 : octet hexadécimal à convertir
                        *
AC00 B6 0F 02           prog0   LDA     $0F02           donnée -> A
AC03 B7 95 80                   STA     cna_bi          (A) -> CNA
AC06 3F                         SWI

                        * Sous-programme de temporisation (pour 2-1 & sq)
                        *       - durée tempo = 5µs x (B)
                        *
AC07 5A                 delai   DECB                    (B)-1 -> B
AC08 26 FD                      BNE     delai           B = 0 ?
AC0A 39                         RTS                     si oui, retour

                        * 2-1 : - génère un signal rectangulaire
                        *       - échelle : [$00,$FF]  <->  [+10V,-10V]
                        *       - sortie CNA : borne 5
```

ISBN 2-9520781-1-4 http://www.syscope.net EAN 9782952078115

```
                        *
                        * $0F00 : durée niveau bas (B)
                        * $0F01 : durée niveau haut (B)
                        * $0F02 : niveau bas
                        * $0F03 : niveau haut
                        *
AC0B B6 0F 02           prog1   LDA     $0F02           niveau bas -> CNA
AC0E B7 95 80                   STA     cna_bi
AC11 F6 0F 00                   LDB     $0F00           tempo niveau bas
AC14 8D F1                      BSR     delai
AC16 B6 0F 03                   LDA     $0F03           niveau haut -> CNA
AC19 B7 95 80                   STA     cna_bi
AC1C F6 0F 01                   LDB     $0F01           tempo niveau haut
AC1F 8D E6                      BSR     delai
AC21 20 E8                      BRA     prog1           boucle
AC23 3F                         SWI

                        * 2-1bis:- génère un signal rectangulaire de rapport
                        * cyclique proportionnel à une tension continue
                        *       - entrée : éch : [$00,$FF]  <->  [0,+10V]
                        *                  CAN : borne 1
                        *       - sortie : éch : [$00,$FF]  <->  [+10V,-10V]
                        *                  CNA : borne 5
                        *
                        * $0F02 : niveau bas
                        * $0F03 : niveau haut

AC24 B6 0F 02           prog1b  LDA     $0F02           niveau bas -> CNA
AC27 B7 95 80                   STA     cna_bi
AC2A F6 94 00                   LDB     can_un          durée niveau bas =
AC2D 8D D8                      BSR     delai           valeur lue dans le CAN
AC2F B6 0F 03                   LDA     $0F03           niveau haut -> CNA
AC32 B7 95 80                   STA     cna_bi
AC35 F6 94 00                   LDB     can_un          durée niveau haut =
AC38 53                         COMB                    256 - durée niveau bas
AC39 8D CC                      BSR     delai
AC3B 20 E7                      BRA     prog1b          boucle
AC3D 3F                         SWI

                        * 2-2 : - génère une dent de scie
                        *       - sortie CNA : borne 5
                        *
                        * $0F00 : durée d'un échantillon (B)
                        *
AC3E 4F                 prog2   CLRA                    RAZ de A
AC3F B7 95 80           debut2  STA     cna_bi          (A) -> CNA
AC42 F6 0F 00                   LDB     $0F00           durée d'un échantillon
AC45 8D C0                      BSR     delai
AC47 4C                         INCA                    (A)+1 -> A
AC48 20 F5                      BRA     debut2
AC4A 3F                         SWI

                        * 2-3 : - génère un signal quelconque à partir
                        *         d'une table de valeurs
                        *       - sortie CNA : borne 5
                        *
                        * $0F00 : durée d'un échantillon (B)
                        * $0F01 : nombre d'échantillons/période (1 à 256)
                        * $0000 : début de la table des échantillons
                        * (variable interne : $0F02 : compteur)
```

ISBN 2-9520781-1-4 http://www.syscope.net EAN 9782952078115

```
                        *
AC4B 10 8E 00 00        prog3   LDY     #$0000          adr base table -> Y
AC4F B6 0F 01                   LDA     $0F01           nb d'échant./période
AC52 B7 0F 02                   STA     $0F02           décompteur en $0F02
AC55 A6 A0              lect3   LDA     0,Y+            lecture table
AC57 B7 95 80                   STA     cna_bi
AC5A F6 0F 00                   LDB     $0F00           durée d'un échantillon
AC5D 8D A8                      BSR     delai
AC5F 7A 0F 02                   DEC     $0F02           décrémenter compteur
AC62 26 F1                      BNE     lect3           si pas = 0, continuer
AC64 20 E5                      BRA     prog3
AC66 3F                         SWI

                        * 3-1 : - recopie le signal d'entrée
                        *       - entrée : éch : [$00,$FF]  <->  [+10V,-10V]
                        *                  CAN : borne 2
                        *       - sortie : éch : [$00,$FF]  <->  [+10V,-10V]
                        *                  CNA : borne 5
                        *
AC67 B6 94 80           prog4   LDA     can_bi
AC6A B7 95 80                   STA     cna_bi
AC6D 12                         NOP
AC6E 12                         NOP
AC6F 12                         NOP
AC70 20 F5                      BRA     prog4
AC72 3F                         SWI

                        * 3-2 : - recopie le signal d'entrée avec contrôle de
                        *         la période d'échantillonnage Te
                        *       - Te = $Δt+1 µs , $Δt valeur hexa écrite sur
                        *         2 octets, poids fort, poids faible
                        *         => 1µs ($Δt = $0000) < Te < 65ms ($FFFF)
                        *       - entrée : éch : [$00,$FF]  <->  [+10V,-10V]
                        *                  CAN : borne 2
                        *       - sortie : éch : [$00,$FF]  <->  [+10V,-10V]
                        *                  CNA : borne 5
                        *
                        * $0F00:0F01 : période d'échantillonnage ($Δt)
                        *
AC73 30 8D 00 10        prog5   LEAX    spg5,PCR        adr s-p d'interruption
AC77 BF 0F 66                   STX     vfirq           vecteur d'interruption
AC7A 86 F2                      LDA     #$F2            initialisation TIMER 1
AC7C BE 0F 00                   LDX     $0F00
AC7F BD A8 80                   JSR     itimer
AC82 3C BF              loop5   CWAI    #%10111111      autorisation et attente
AC84 20 FC                      BRA     loop5           d'interruption
AC86 3F                         SWI
AC87 BF 88 02           spg5    STX     drtim1          démarrage décompteur
AC8A B6 94 80                   LDA     can_bi          transfert CAN -> CNA
AC8D B7 95 80                   STA     cna_bi
AC90 3B                         RTI

                        * 3-3 : - filtre non récursif, soustraction
                        *       - Te, entrée/sortie, adr mémoire : idem 3-2
                        * (variable interne : $0F02 : table des u(k))
                        *
AC91 30 8D 00 10        prog6   LEAX    spg6,PCR        adr s-p d'interruption
AC95 BF 0F 66                   STX     vfirq           vecteur d'interruption
AC98 86 F2                      LDA     #$F2            initialisation TIMER 1
AC9A BE 0F 00                   LDX     $0F00
```

ISBN 2-9520781-1-4 http://www.syscope.net EAN 9782952078115

```
AC9D BD A8 80                   JSR     itimer
ACA0 3C BF              loop6   CWAI    #%10111111      autorisation et attente
ACA2 20 FC                      BRA     loop6           d'interruption
ACA4 3F                         SWI
ACA5 BF 88 02           spg6    STX     drtim1          démarrage décompteur
ACA8 CE 0F 02                   LDU     #$0F02          adr table des u(k)
ACAB B6 94 80                   LDA     can_bi          acquisition u(k)
ACAE 88 80                      EORA    #%10000000      bin décalé -> comp 2
ACB0 A7 41                      STA     1,U             sauve u(k) à adr U+1
ACB2 A0 C4                      SUBA    ,U              u(k) - u(k-1)
ACB4 47                         ASRA                    div par 2, avec signe
ACB5 E6 41                      LDB     1,U             copie u(k) à adr U
ACB7 E7 C4                      STB     ,U
ACB9 88 80                      EORA    #%10000000      comp 2 -> bin décalé
ACBB B7 95 80                   STA     cna_bi
ACBE 3B                         RTI

                        * 3-3bis: - filtre non récursif, addition
                        *         - Te, entrée/sortie, adr mémoire : idem 3-2
                        *
ACBF 30 8D 00 10        prog7   LEAX    spg7,PCR        adr s-p d'interruption
ACC3 BF 0F 66                   STX     vfirq           vecteur d'interruption
ACC6 86 F2                      LDA     #$F2            initialisation TIMER 1
ACC8 BE 0F 00                   LDX     $0F00
ACCB BD A8 80                   JSR     itimer
ACCE 3C BF              loop7   CWAI    #%10111111      autorisation et attente
ACD0 20 FC                      BRA     loop7           d'interruption
ACD2 3F                         SWI
ACD3 BF 88 02           spg7    STX     drtim1          démarrage décompteur
ACD6 CE 0F 02                   LDU     #$0F02          adr table des u(k)
ACD9 B6 94 80                   LDA     can_bi          acquisition u(k)
ACDC 88 80                      EORA    #%10000000      bin décalé -> comp 2
ACDE A7 41                      STA     1,U             sauve u(k) à adr U+1
ACE0 AB C4                      ADDA    ,U              u(k) - u(k-1)
ACE2 47                         ASRA                    div par 2, avec signe
ACE3 E6 41                      LDB     1,U             copie u(k) à adr U
ACE5 E7 C4                      STB     ,U
ACE7 88 80                      EORA    #%10000000      comp 2 -> bin décalé
ACE9 B7 95 80                   STA     cna_bi
ACEC 3B                         RTI

                        * 3-4 : - filtre récursif
                        *       - Te, entrée/sortie, adr mémoire : idem 3-2
                        *
ACED 30 8D 00 10        prog8   LEAX    spg8,PCR        adr s-p d'interruption
ACF1 BF 0F 66                   STX     vfirq           vecteur d'interruption
ACF4 86 F2                      LDA     #$F2            initialisation TIMER 1
ACF6 BE 0F 00                   LDX     $0F00
ACF9 BD A8 80                   JSR     itimer
ACFC 3C BF              loop8   CWAI    #%10111111      autorisation et attente
ACFE 20 FC                      BRA     loop8           d'interruption
AD00 3F                         SWI
AD01 BF 88 02           spg8    STX     drtim1          démarrage décompteur
AD04 CE 0F 02                   LDU     #$0F02          adr table des u(k)
AD07 6F 41                      CLR     1,U             u(0) = 0
AD09 B6 94 80                   LDA     can_bi          acquisition u(k)
AD0C A7 42                      STA     2,U             sauve u(k) à adr U+1
AD0E A6 C4                      LDA     ,U              7 x v(k-1) + u(k)
AD10 C6 07                      LDB     #$07
AD12 3D                         MUL                     opérations sur 16 bits
```

ISBN 2-9520781-1-4 http://www.syscope.net EAN 9782952078115

```
AD13 E3 41                      ADDD    1,U
AD15 44                         LSRA                    division par 8 (/2/2/2)
AD16 56                         RORB
AD17 44                         LSRA
AD18 56                         RORB
AD19 44                         LSRA
AD1A 56                         RORB
AD1B E7 C4                      STB     ,U              copie de v(k)
AD1D F7 95 80                   STB     cna_bi          seul le poids faible
AD20 3B                         RTI                     est recopié ds le CNA

                        * exercice de synthèse n°1 : comparateur à hystérésis
                        *       - entrée : éch : [$00,$FF]  <->  [0,+100°C]
                        *                  CAN : borne 1
                        *       - sortie : 1 = arrêt moteur ($FF)
                        *                  0 = marche moteur ($00)
                        *                  VIA : connecteur 1
                        *
                        * $0F00 : température seuil bas
                        * $0F01 : température seuil haut
                        *
AD21 86 FF              tor     LDA     #$FF            port B VIA en sortie
AD23 B7 84 02                   STA     cbvia
AD26 86 FF                      LDA     #$FF            init: moteur à l'arrêt
AD28 B7 84 00                   STA     rbvia
AD2B B6 94 00           trloop  LDA     can_un          boucle
AD2E B1 0F 00                   CMPA    $0F00
AD31 25 07                      BLO     off
AD33 B1 0F 01                   CMPA    $0F01
AD36 22 09                      BHI     on
AD38 20 F1                      BRA     trloop
AD3A 86 FF              off     LDA     #$FF            arrêt moteur
AD3C B7 84 00                   STA     rbvia
AD3F 20 EA                      BRA     trloop
AD41 86 00              on      LDA     #$00
AD43 B7 84 00                   STA     rbvia
AD46 20 E3                      BRA     trloop
AD48 3F                         SWI

                        * Exercice de synthèse n°2 : commande moteur pas-à-pas
                        *       - entrée : mots de commande : $0x = %0000abcd
                        *                  a,b : connexions 1 et 2 phase I
                        *                  c,d : connexions 3 et 4 phase II
                        *                  a,b,c ou d = 0 : 0 V
                        *                  a,b,c ou d = 1 : +12 V
                        *       - sortie : PIA : connecteur 1
                        *
                        * $0F00->$0F03 : table de commande moteur PAP (4 octs)
                        * $0F04:0F05 : valeur hexa de la durée d'un pas (en ms)
                        *           comprise entre 1ms ($0001) et 0,65s ($FFFF)
                        *
AD49 10 8E 80 00        pap     LDY     #rapiau         init PIA port A
AD4D 86 FF                      LDA     #$FF
AD4F BD A8 40                   JSR     ipia
AD52 CE 0F 00                   LDU     #$0F00          init table
AD55 BE 0F 04                   LDX     $0F04           durée d'un pas
AD58 A6 C0              pploop  LDA     ,U+             lecture table
AD5A B7 80 00                   STA     rapiau          et commande moteur
AD5D BD AA 00                   JSR     tempo
AD60 11 83 0F 04                CMPU    #$0F04          fin de table ?
```

ISBN 2-9520781-1-4 http://www.syscope.net EAN 9782952078115

```
AD64 26 F2                      BNE     pploop          non : continuer
AD66 CE 0F 00                   LDU     #$0F00          oui : raz pointeur
AD69 20 ED                      BRA     pploop
AD6B 3F                         SWI

                *       TP : acquisition et analyse des signaux

                        *
0000                    table   EQU     $0000   adresse table d'échantillons
0F00                    nbech   EQU     $0F00   adr nb d'échantillons 2 oct
0F02                    deltat  EQU     $0F02   adr période d'échant. 2 oct
0F04                    seuil   EQU     $0F04   adr seuil de déclench.1 oct
0F05                    suptab  EQU     $0F05   adresse borne sup de table


                        * Programme DAA1 : acquisition simple
                        * Entrer préalablement, sur le clavier du MC09 :
                        * -le nombre d'échantillons N,
                        *    N en hexadécimal, 2 octets, adresses $0F00:0F01,
                        *    poids fort, poids faible
                        * -la période d'échantillonnage Te, avec :
                        *    deltat = Te - 1 microsecondes
                        *    deltat en héxadécimal, 2 octets, adr $0F02:0F03,
                        *    pds fort, pds faible
                        *    NB : valeur minimale de Te : 74 microsecondes
                        *
AD70                            ORG     $AD70
AD70 1A 40              daa1    ORCC    #%01000000    masquage des interrupt
AD72 CC 00 00                   LDD     #table        calcul borne sup table
AD75 F3 0F 00                   ADDD    nbech
AD78 FD 0F 05                   STD     suptab

                        * initialisation échnatillonnage
AD7B 30 8D 00 2E        debut1  LEAX    intrp1,PCR    init interruptions
AD7F BF 0F 66                   STX     vfirq
AD82 86 F2                      LDA     #$F2
AD84 BE 0F 02                   LDX     deltat
AD87 BD A8 80                   JSR     itimer

                        * acquisition
AD8A CE 00 00           acq1    LDU     #table
AD8D 3C BF              acqlp1  CWAI    #%10111111  autorisation des interrup
AD8F B6 94 80                   LDA     can_bi
AD92 A7 C0                      STA     ,U+
AD94 11 B3 0F 05                CMPU    suptab
AD98 23 F3                      BLS     acqlp1

                        * émission série
AD9A 86 15                      LDA     #%00010101    8 bits, 1 stop, 1:16
AD9C 8E 00 19                   LDX     #$0019        1200 bauds
AD9F BD A8 C0                   JSR     iuart         initialisation ACIA
ADA2 8E 00 01                   LDX     #table+1
ADA5 10 BE 0F 00                LDY     nbech
ADA9 BD A9 80                   JSR     emibin

                        * fin
ADAC 3F                         SWI

                        * Sous-programme d'interruptions
                        *
```

```
ADAD BF 88 02           intrp1  STX     drtim1        réinitialisation timer
ADB0 1A 40                      ORCC    #%01000000    masquage des interrupt
ADB2 3B                         RTI


                        * Programme DAA2 : acquisition avec déclenchement
                        * Entrer préalablement, sur le clavier du MC09 :
                        * -le nombre d'échantillons N,
                        *    N en hexadécimal, 2 octets, adresses $0F00:0F01,
                        *    poids fort, poids faible
                        * -la période d'échantillonnage Te, avec :
                        *    deltat = Te - 1 microsecondes
                        *    deltat en héxadécimal, 2 octets, adr $0F02:0F03,
                        *    pds fort, pds faible
                        *    NB : valeur minimale de Te : 74 microsecondes
                        * -le seuil de déclenchement Vs,
                        *    avec n = INT(-12,8.Vs + 128)
                        *    n en hexadécimal, 1 octet, adresse $0F04
                        *
ADC0                            ORG     $ADC0
ADC0 1A 40              daa2    ORCC    #%01000000      masquage interpt FIRQ
ADC2 CC 00 00                   LDD     #table          calcul borne sup table
ADC5 F3 0F 00                   ADDD    nbech
ADC8 FD 0F 05                   STD     suptab

                        * initialisation échnatillonnage
ADCB 30 8D 00 66        debut2  LEAX    intrp2,PCR      init interruptions
ADCF BF 0F 66                   STX     vfirq
ADD2 86 F2                      LDA     #$F2
ADD4 BE 0F 02                   LDX     deltat
ADD7 BD A8 80                   JSR     itimer

                        * déclenchement
ADDA CE 00 00           acq2    LDU     #table
ADDD 10 8E 10 00                LDY     #$1000
ADE1 3C BF              y0      CWAI    #%10111111   v(t) > 0 ?
ADE3 31 3F                      LEAY    -1,Y         tempo si tension continue
ADE5 27 2E                      BEQ     acqlp2
ADE7 B6 94 80                   LDA     can_bi
ADEA B1 0F 04                   CMPA    seuil
ADED 22 F2                      BHI     y0
ADEF 10 8E 10 00                LDY     #$1000
ADF3 3C BF              y1      CWAI    #%10111111   v(t) < 0 ?
ADF5 31 3F                      LEAY    -1,Y         tempo si tension continue
ADF7 27 1C                      BEQ     acqlp2
ADF9 B6 94 80                   LDA     can_bi
ADFC B1 0F 04                   CMPA    seuil
ADFF 25 F2                      BLO     y1
AE01 10 8E 10 00                LDY     #$1000
AE05 3C BF              y2      CWAI    #%10111111   v(t) > 0 ?
AE07 31 3F                      LEAY    -1,Y         tempo si tension continue
AE09 27 0A                      BEQ     acqlp2
AE0B B6 94 80                   LDA     can_bi
AE0E B1 0F 04                   CMPA    seuil
AE11 22 F2                      BHI     y2            si oui, on y va !
AE13 A7 C0                      STA     ,U+

                        * acquisition
AE15 3C BF              acqlp2  CWAI    #%10111111  autorisation des interrup
AE17 B6 94 80                   LDA     can_bi
```

```
AE1A A7 C0                      STA     ,U+
AE1C 11 B3 0F 05                CMPU    suptab
AE20 23 F3                      BLS     acqlp2

                        * émission série
AE22 86 15                      LDA     #%00010101    8 bits, 1 stop, 1:16
AE24 8E 00 19                   LDX     #$0019        1200 bauds
AE27 BD A8 C0                   JSR     iuart         initialisation ACIA
AE2A 8E 00 00                   LDX     #table
AE2D 10 BE 0F 00                LDY     nbech
AE31 BD A9 80                   JSR     emibin

                        * fin
AE34 3F                         SWI

                        * Sous-programme d'interruptions
                        *
AE35 BF 88 02           intrp2  STX     drtim1        réinitialisation timer
AE38 1A 40                      ORCC   #%01000000     masquage des interrupt
AE3A 3B                         RTI

                        * Chronometre
                        * version 1 : tempo logicielle
                        *
                        * Sous-programmes predefinis en EPROM
                        * Tempo 10 ms
A140                    estemp  EQU     $A140   message pte par X pdt 1/100e s
                        * Conversion de codes
A4C0                    heseg   EQU     $A4C0   hexa / 7 segments
A600                    hebcd1  EQU     $A600   hexa / BCD 8 bits
                        * variables :
0F00                    temps   EQU     $0F00

AE40                            ORG     $AE40
AE40 CE 0F 00           pgm3    LDU     #temps
AE43 6F C4                      CLR     0,U         RAZ 1/100e, sec, min
AE45 6F 41                      CLR     1,U
AE47 6F 42                      CLR     2,U
AE49 8D 0B              chro_1  BSR     chro_2      boucle principale
AE4B 8E 0F 51                   LDX     #affich
AE4E BD A1 40                   JSR     estemp
AE51 8D 25                      BSR     chro_3
AE53 20 F4                      BRA     chro_1
AE55 3F                         SWI

AE56 A6 C4              chro_2  LDA     0,U      preparation de l'affichage
AE58 BD A6 00                   JSR     hebcd1
AE5B BD A4 C0                   JSR     heseg
AE5E FD 0F 51                   STD     affich
AE61 A6 41                      LDA     1,U
AE63 BD A6 00                   JSR     hebcd1
AE66 BD A4 C0                   JSR     heseg
AE69 FD 0F 53                   STD     affich+2
AE6C A6 42                      LDA     2,U
AE6E BD A6 00                   JSR     hebcd1
AE71 BD A4 C0                   JSR     heseg
AE74 FD 0F 55                   STD     affich+4
AE77 39                         RTS

AE78 6C 42              chro_3  INC     2,U      calcul 1/100e, sec, mn
```

ISBN 2-9520781-1-4 http://www.syscope.net EAN 9782952078115

```
AE7A 86 64                      LDA     #100
AE7C A1 42                      CMPA    2,U
AE7E 26 16                      BNE     chro_4
AE80 6F 42                      CLR     2,U
AE82 6C 41                      INC     1,U
AE84 86 3C                      LDA     #60
AE86 A1 41                      CMPA    1,U
AE88 26 0C                      BNE     chro_4
AE8A 6F 41                      CLR     1,U
AE8C 6C C4                      INC     0,U
AE8E 86 3C                      LDA     #60
AE90 A1 C4                      CMPA    0,U
AE92 26 02                      BNE     chro_4
AE94 6F C4                      CLR     0,U
AE96 39                 chro_4  RTS


                        * Chronometre, suite.
                        * version 2 : tempo materielle par FIRQ
                        *             + interpt prioritaire par NMI
                        *

AE97                            ORG     $AE97
AE97 CE 0F 00           chro_5  LDU     #temps
AE9A 6F C4                      CLR     0,U             RAZ 1/100e, sec, min
AE9C 6F 41                      CLR     1,U
AE9E 6F 42                      CLR     2,U
AEA0 30 8D 00 1D                LEAX    chro_7,PCR      vecteur interpt FIRQ
AEA4 BF 0F 66                   STX     vfirq
AEA7 30 8D 00 25                LEAX    chro_8,PCR      vecteur interpt NMI
AEAB BF 0F 60                   STX     vnmi
AEAE 86 C2                      LDA     #%11000010      compteur mode continu
AEB0 8E 27 0F                   LDX     #9999           9999µs+1=1/100e sec
AEB3 BD A8 80                   JSR     itimer          init compteur
AEB6 1C BF                      ANDCC   #%10111111      autorisation FIRQ
AEB8 8E 0F 51           chro_6  LDX     #affich         boucle principale
AEBB BD A1 00                   JSR     esclav
AEBE 20 F8                      BRA     chro_6
AEC0 3F                         SWI

AEC1 34 36              chro_7  PSHS    A,B,X,Y         sauvegarde registres
AEC3 8D B3                      BSR     chro_3          calculer 1/100e, s, m
AEC5 8D 8F                      BSR     chro_2          preparer affichage
AEC7 B6 88 01                   LDA     crtim2          RAZ de l'indicateur
AECA FC 88 02                   LDD     drtim1          d'interuption du timer
AECD 35 36                      PULS    A,B,X,Y         restituer registres
AECF 3B                         RTI

AED0 34 36              chro_8  PSHS    A,B,X,Y
AED2 8E 0F 51                   LDX     #affich
AED5 BD A1 00                   JSR     esclav
AED8 35 36                      PULS    A,B,X,Y
AEDA 3B                         RTI


                        * TP : trnasformée en Z
                        * 3-2 : - recopie le signal d'entrée avec contrôle de
                        *         la période d'échantillonnage Te
                        *       - Te = $Δt+1 µs , $Δt valeur hexa écrite sur
                        *         2 octets, poids fort, poids faible
```

ISBN 2-9520781-1-4 http://www.syscope.net EAN 9782952078115

```
                        *         => 1µs ($Δt = $0000) < Te < 65ms ($FFFF)
                        *       - entrée : éch : [$00,$FF]  <->  [+10V,-10V]
                        *                  CAN : borne 2
                        *       - sortie : éch : [$00,$FF]  <->  [+10V,-10V]
                        *                  CNA : borne 5
                        *
                        * $0F00:0F01 : période d'échantillonnage ($Δt)
                        *
AF00                            ORG     $AF00
AF00 30 8D 00 10        prog5b  LEAX    spg5,PCR        adr s-p d'interruption
AF04 BF 0F 66                   STX     vfirq           vecteur d'interruption
AF07 86 F2                      LDA     #$F2            initialisation TIMER 1
AF09 BE 0F 00                   LDX     $0F00
AF0C BD A8 80                   JSR     itimer
AF0F 3C BF              loop5   CWAI    #%10111111      autorisation et attente
AF11 20 FC                      BRA     loop5           d'interruption
AF13 3F                         SWI

                        * sous-programme d'acquisition modifié :

AF14 BF 88 02           spg5    STX     drtim1          démarrage décompteur
AF17 B6 94 80                   LDA     can_bi          lancement conversion CAN
AF1A 12                         NOP                   attente fin conversion CAN
AF1B 12                         NOP
AF1C 12                         NOP
AF1D 12                         NOP
AF1E 12                         NOP
AF1F 12                         NOP
AF20 12                         NOP
AF21 12                         NOP                     5 + 8x2 = 21µs
AF22 B6 94 80                   LDA     can_bi          lecture CAN
AF25 B7 95 80                   STA     cna_bi          transfert CAN -> CNA
AF28 3B                         RTI


                        * TP : transformée en Z, correcteur I

AF29 30 8D 00 10        prog5c  LEAX    spg5,PCR        adr s-p d'interruption
AF2D BF 0F 66                   STX     vfirq           vecteur d'interruption
AF30 86 F2                      LDA     #$F2            initialisation TIMER 1
AF32 BE 0F 00                   LDX     $0F00
AF35 BD A8 80                   JSR     itimer
AF38 3C BF              loop5   CWAI    #%10111111      autorisation et attente
AF3A 20 FC                      BRA     loop5           d'interruption
AF3C 3F                         SWI

                        * sous-programme d'acquisition modifié pour intégrateur:

AF3D BF 88 02           spg5    STX     drtim1          démarrage décompteur
AF40 B6 94 80                   LDA     can_bi          lancement conversion CAN
AF43 12                         NOP                   attente fin conversion CAN
AF44 12                         NOP
AF45 12                         NOP
AF46 12                         NOP
AF47 12                         NOP
AF48 12                         NOP
AF49 12                         NOP
AF4A 12                         NOP                     5 + 8x2 = 21µs
AF4B B6 94 80                   LDA     can_bi          lecture CAN
```

ISBN 2-9520781-1-4 http://www.syscope.net EAN 9782952078115

```
AF4E 88 80                      EORA    #%10000000  conv bin décalé -> 2's comp
AF50 BB 0F 02                   ADDA    $0F02           variable interne
AF53 B7 0F 02                   STA     $0F02
AF56 88 80                      EORA    #%10000000  conv 2's comp -> bin décalé
AF58 B7 95 80                   STA     cna_bi          transfert CAN -> CNA
AF5B 3B                         RTI


                        * TP : transformée en Z, correcteur PI
0F02                    vk_1    EQU     $0F02           adresse de Vi(k-1)
0F03                    ek      EQU     $0F03           adresse de e(k)
0F04                    ek_1    EQU     $0F04           adresse de e(k-1)
0F05                    uk_1    EQU     $0F05           adresse de u(k-1)

AF5C 30 8D 00 10        prog5d  LEAX    spg5,PCR        adr s-p d'interruption
AF60 BF 0F 66                   STX     vfirq           vecteur d'interruption
AF63 86 F2                      LDA     #$F2            initialisation TIMER 1
AF65 BE 0F 00                   LDX     $0F00
AF68 BD A8 80                   JSR     itimer
AF6B 3C BF              loop5   CWAI    #%10111111      autorisation et attente
AF6D 20 FC                      BRA     loop5           d'interruption
AF6F 3F                         SWI

                        * sous-programme d'acquisition modifié pour intégrateur:

AF70 BF 88 02           spg5    STX     drtim1          démarrage décompteur
AF73 B6 94 80                   LDA     can_bi          lancement conversion CAN
AF76 12                         NOP                   attente fin conversion CAN
AF77 12                         NOP
AF78 12                         NOP
AF79 12                         NOP
AF7A 12                         NOP
AF7B 12                         NOP
AF7C 12                         NOP
                                NOP                                          21
                                LDB     can_bi      lecture CAN de e(k)      +5
                                EORB    #%10000000  bin decale -> compl. a 2 +2
                                STB     ek          sauv. e(k)               +5
                                ADDB    uk_1        e(k) + u(k-1)            +5
                                EORB    #%10000000  compl. a 2 -> bin decale +2
                                STB     cna_bi  transfert CAN -> CNA : Vi(k) +5
                                LDA     vk_1  Vi(k-1)                 Δt = 50µs
                                EORB    #%10000000  binaire decale -> compl. a 2
                                STB     vk_1  sauv. Vi(k) à la place de Vi(k-1)
                                SUBA    ek_1  Vi(k-1) - e(k-1)
                                ASR     ek_1  /2 avec conservation du signe
                                ASR     ek_1  /2
                                ASR     ek_1  /2
                                ADDA    ek_1  Vi(k-1) - e(k-1) + e(k-1)/8
                                STA     uk_1  sauv. u(k-1)
                                LDA     ek
                                STA     ek_1  sauv. e(k) à la place de e(k-1)
                                RTI
```

ISBN 2-9520781-1-4 http://www.syscope.net EAN 9782952078115