

# PROGRAMME SOURCE POUR MO5

MAIN. CR6809/11 version 20.54.23  
ASEM.SRC

21-Feb-86 11:37:24 Page 1

Error	Addr	Code	Seq	Source statement
	=0000		1	SECT ASEM
			2	INCLUDE HANDLER
			3	*****
			4	*
			5	* Ce module est destine a assurer l'interface entre le Basic 1.0
			6	* Ou le basic 128 et le contrôleur RS232 dont le listing est fourni
			7	* separemement. Et même MO5
			8	*
			9	* Ce module ainsi que le handler sera charge dans le T07 ou le
			10	* T07-70 via un chargeur écrit en basic
			11	*
			12	* Pour les personnes ne disposant pas d'assembleur, un programme
			13	* écrit en Basic est fourni conjointement qui permettra de créer le
			14	* fichier objet nécessaire.
			15	*
			16	* Ce module ne gère pas les demandes de paramètres divers, tels
			17	* la vitesse ou le nombre de bits. Ces paramètres sont supposés positionnés
			18	* par le programme Basic à l'aide de POKE aux 2 cases mémoires spécifiées.
			19	*
			20	* Ce programme doit être le premier des trois programmes.
			21	*
			22	*****
			23	
			24	
			25	*
			26	* Début du programme
			27	*
	=9C00		28	ORG \$9C00
			29	
	=9C00'		30	R232 EQU *
	9C00' 52 32 33 32		31	FCC /R232/ Non du périphérique sous basic
	9C04' 0010		32	FDB R232OPN-R232 Ouverture
	9C06' 0023		33	FDB R232CLS-R232 Fermeture
	9C08' 0027		34	FDB R232INP-R232 Lecture
	9C0A' 0034		35	FDB R232OUT-R232 Ecriture
	9C0C' 003D		36	FDB R232PRM-R232 Paramètres
	9C0E' E233*		37	FDB FMERR EOF
			38	
	=9C10'		39	R232OPN EQU *
	9C10' C6 01		40	LDB #RS.OPR Demande d'ouverture
	9C12' F1 2083		41	CMPB RS.STA Est-ce déjà ouvert ?
	9C15' 27 1C-		42	BEQ RETOUR Oui, on ne fait rien
			43	
	=9C17'		44	EXECUTE EQU *
	9C17' F7 2082		45	STB RS.OPC
	=9C1A'		46	EXEC2 EQU *
	9C1A' 8D 26		47	BSR RS232
	9C1C' 24 15		48	BCC RETOUR
	9C1E' 8D 03		49	BSR R232CLS On ferme la liaison
	9C20' 7E E88F		50	JMP IOERR On va faire IO Error dans le basic.

MAIN. CR6809/11 version 20.54.23  
ASEM.SRC

21-Feb-86 11:37:24 Page 1-1

Error	Addr	Code	Seq	Source statement
			51	
	=9C23'		52	R232CLS EQU *
	9C23' C6 10		53	LDB #RS.CLS Demande de fermeture
	9C25' 20 F0		54	BRA EXECUTE Et on l'exécute
			55	
	=9C27'		56	R232INP EQU *
	9C27' C6 02		57	LDB #RS.RDC Demande de lecture
	9C29' F7 2082		58	STB RS.OPC
	9C2C' BD 14		59	BSR RS232
	9C2E' 24 01		60	BCC LECTOK Pas d'erreur, il y a un caractère
	9C30' SF		61	CLRB Rien, on rend 0
	=9C31'		62	LECTOK EQU *
	9C31' 1F 98		63	TFR B.A On met le résultat dans A
	9C33' 39		64	RETOJR RTS
			65	
	=9C34'		66	R232OUT EQU *
	9C34' C6 09		67	LDB #RS.WRC Demande d'écriture
	9C36' F7 2082		68	STB RS.OPC
	9C39' 1F 89		69	TFR A,B Le caractère était dans A
	9C3B' 20 DD		70	BRA EXEC2
			71	
	=9C3D'		72	R232PRM EQU *
	9C3D' 4F		73	CLRA
	9C3E' SF		74	CLRB
	9C3F' 1F 01		75	TFR D,X Aucune position de spécifiée sur la voie série
	9C41' 39		76	PTS
			77	

Error Addr Code Seq Source statement

```

79      INCLUDE RS232
80 ****
81 *
82 * Ce programme permet de piloter les entrees-sorties serie
83 * de type RS232C de la nouvelle extension RS 57-932 sur les M65.
84 *
85 * Pour les personnes ne disposant pas de l'assembleur, un
86 * programme écrit en Basic, permettra de créer le fichier correspondant.
87 * Ce fichier doit être situé en deuxième, en suivant le handler, et en
88 * précédant le loader.
89 *
90 * Le registre RS.OPC permet de savoir le type d'opération demandée :
91 *
92 * - RS.OPR : ouverture en lecture écriture
93 * - RS.RDC : lecture d'un caractère (dans B)
94 * - RS.WRC : écriture d'un caractère (contenu dans B)
95 * - RS.CLS : fermeture
96 *
97 * A la fin de l'opération, le CY a 0 indique que tout s'est bien
98 * passé, le CY a 1 qu'il y a eu une erreur. Le registre RS.STA contient
99 * le type de l'erreur :
100 *
101 * - RS.OPR : Pas de caractère disponible
102 * - RS.CLS : Fermée
103 * - RS.NRD : Unité non prête
104 *
105 * Le registre VITESSE permet de choisir la vitesse de transmission :
106 *
107 * Vitesse          Valeur dans BAUDS
108 * 50               $1
109 * 75               $2
110 * 110              $3
111 * 134              $4
112 * 150              $5
113 * 300              $6
114 * 600              $7
115 * 1200             $8
116 * 1800             $9
117 * 2400             $A
118 * 3600             $B
119 * 4800             $C
120 * 7200             $D
121 * 9600             $E
122 * 19200            $F
123 *
124 * Le registre MODE permet de définir différents paramètres :
125 *
126 * Bit 7-6           Bit 5           Bit 4-3-2        Bit 1           Bit 0
127 * Longueur          Horloge         Parité          Mode           Stop bits
128 *

```

Error Addr Code Seq Source statement

```

129 *    11 5 bits      0 interne      111 bit à 0      0 modem      0 2 bits
130 *    10 8 bits      1 externe      101 bit à 1      1 terminal   1 1 bit
131 *    01 7 bits      011 parité paire
132 *    00 6 bits      001 parité impaire
133 *                000 sans parité
134 *
135 ****
136 *
137 *    Page 0 du moniteur
138 *
139 IRQPT EQU $2064
140 OFFSET EQU $2082
141 RS.OPC EQU $2082
142 RS.STA EQU $2083
143 MODE EQU $2084
144 VITESSE EQU $2085
145 *
146 *
147 *    Opérations demandées
148 *
149 RS.OPR EQU $01      Ouverture en lecture/écriture
150 RS.RDC EQU $02      Lecture d'un caractère
151 RS.WRC EQU $09      Ecriture d'un caractère
152 RS.CLS EQU $10      Fermeture
153 *
154 *
155 *    Cas d'erreur
156 *
157 RS.NRD EQU $80      Peripherique non prêt
158 *
159 *
160 *    Registres du contrôleur
161 *
162 SIOTRANSM EQU $A7E8
163 SIORECEPT EQU $A7E8
164 SIOSTATUS EQU $A7E9
165 SIOCMDE EQU $A7EA
166 SIOCNTRL EQU $A7EB
167 *
168 *
169 *
170 SETDP $A7          Page 0 sur les registres de commande
171 *
172 RS232 EQU *
173 PSHS A+B,CC,DP,X,Y,J
174 LDA #$A7
175 TFR A+DP DP est initialisé
176 LDY #OFFSET Y pointe sur les registres de la page 0
177 LDA RS.OPC-OFFSET,Y Operation souhaitée
178 TFR A+CC On va se débrancher selon le cas

```

Error	Addr	Code	Seq	Source	statement	
	9C50'	28 1C	179	BMI	I0CH	Ecriture d'un caractere
	9C52'	29 1A	180	BVS	I0CH	Lecture d'un caractere
	9C54'	25 0E	181	BCS	OPEN	Ouverture en lecture/ecriture
			182	*		
			183	*	Dans tous les autres cas on ferme	
			184	*		
	9C56'	B6 10	185	LDA	#RS.CLS	
	9C58'	A7 21	186	STA	RS.STA-OFFSET,Y	On range l'etat ferme
			187			
	9C5A'	96 EA	188	LDA	S10CMDE	
	9C5C'	8A 02	189	ORA	#2	Devalidation des IRQ
	9C5E'	97 EA	190	STA	S10CMDE	
	9C60'	96 E9	191	LDA	S10STATUS	Annulation d'une eventuelle interruption
			192	*		recus avant devalidation des IRQ
	9C62'	20 12	193	BRA	OUTRW	
			194			
	=9C64'		195	OPEN	EQU	*
	9C64'	86 01	196	LDA	#RS.OPR	
	9C66'	A5 21	197	BITA	RS.STA-OFFSET,Y	
	9C68'	26 15	198	BNE	RSERR	On etait deja ouvert, d'où erreur
	9C6A'	A7 21	199	STA	RS.STA-OFFSET,Y	On note l'état ouvert
	9C6C'	20 16	200	BRA	OUVRIR	
			201			
	=9C6E'		202	I0CH	EQU	*
	9C6E'	86 01	203	LDA	#RS.OPR	Lecture d'un caractere
	9C70'	A5 21	204	BITA	RS.STA-OFFSET,Y	On verifie que l'on est ouvert
	9C72'	27 0B	205	BEQ	RSERR	Non, d'où erreur
	9C74'	80 4C	206	BSR	EXECIO	On effectue la lecture
			207			
	=9C76'		208	OUTRW	EQU	*
	9C76'	35 7F	209	PULS	A,B,CC,DP,X,Y,U	
	9C78'	1C FE	210	ANDCC	#\$FE	Operation correcte ==> CY=0
	9C7A'	39	211	RTS		
			212			
	=9C7B'		213	RSNRD	EQU	*
	9C7B'	86 80	214	LDA	#RS.NRD	
	9C7D'	A7 21	215	STA	RS.STA-OFFSET,Y	Peripherique non pret
	=9C7F'		216	RSERR	EQU	*
	9C7F'	35 7F	217	PULS	A,B,CC,DP,X,Y,U	
	9C81'	1A 01	218	URCC	#01	Operation incorrecte ==> CY=1
	9C83'	39	219	RTS		
			220			
			221	*		
			222	*	Initialisation de la voie serie	
			223	*		
			224			
	=9C84'		225	OUVRIR	EQU	*
	9C84'	1A 10	226	DRCC	#\$10	Masquage des IRQ pour mettre dans l'adresse
			227	*		de derouatge l'adresse de reception RS232
	9C86'	4F	228	CLRA		

Error	Addr	Code	Seq	Source	statement	
	9C87'	E6 23	229	LOB	VITESSE-OFFSET,Y	B contient la vitesse
	9C89'	1F 03	230	TFR	0,U	
			231	*		
			232	*	Calcul du nombre de stop bits, de la longueur de l'octet et de	
			233	*	l'horloge de reception	
			234	*		
	9C88'	A6 22	235	LDA	MODE-OFFSET,Y	
	9C8D'	84 E1	236	ANDA	#\$E1	On ne garde que les bits interessants
	9C8F'	88 61	237	EORA	#\$61	
	9C91'	88 40	238	ADDA	#\$40	
	9C93'	1C FE	239	ANDCC	#\$FE	
	9C95'	46	240	RORA		
	9C96'	24 02	241	BCC	RS3	
	9C98'	8A 80	242	ORA	#\$80	
	=9C9A'		243	RS3	EQU	*
	9C9A'	33 C6	244	LEAU	A,U	
	9C9C'	1F 30	245	TFR	U,D	
	9C9E'	D7 EB	246	STB	S10CNTRL	Ecriture dans le registre de controle
			247	*		
			248	*	Creation du mot de commande : parite, IRQ sur reception	
			249	*		
	9CA0'	E6 22	250	LD5	MODE-OFFSET,Y	
	9CA2'	C4 1C	251	ANDB	#\$1C	
	9CA4'	58	252	LSLB		
	9CA5'	58	253	LSL3		
	9CA6'	58	254	LSLB		
	9CA7'	96 EA	255	LDA	S10CMDE	
	9CA9'	84 10	256	ANDA	#\$10	
	9CAB'	8A 09	257	ORA	#\$9	RTS etat 0 et DTR : RTS est relie a DSR et
	9CAD'	34 02	258	PSHS	A	DTR a CTS
	9CAF'	EA E0	259	DRB	+S+	
	9CB1'	D7 EA	260	STB	S10CMDE	Ecriture dans le registre de commande
			261	*		
			262	*	Test de presence de l'extension RS232	
			263	*		
	9CB3'	53	264	COMB		
	9CB4'	D8 EA	265	EDRB	S10CMDE	
	9CB6'	5C	266	INC8		
	9CB7'	26 C2	267	BNE	RSNRD	Elle n'est pas la
			268	*		
			269	*	Adresse de retour apres reception d'un octet	
			270	*		
	9CB9'	33 8D 0040	271	LEAU	RECU,PCR	
	9C8D'	FF 2064	272	STU	IRQPT	
	9CC0'	20 B4	273	BRA	OUTRW	
			274	*		
			275	*	Envoi ou lecture de caractere	
			275	*		
	=9CC2'		277	EXECIO	EQU	*
	9CC2'	A6 A4	278	LDA	RS.OPC-OFFSET,Y	

Error	Addr	Code	Seq	Source	statément	
	9CC4*	85 02	279	BITA	#RS.RDC	Est-ce une lecture ?
	9CC6*	26 11	280	BNE	RECEPT	Absolument
			281			
			282	*		
			283	*	On execute l'écriture	
			284	*		
		=9CC8*	285	SENDCH	EQU *	
	9CC8*	96 E9	285		LDA SIOSTATUS	
	9CCA*	85 10	287		BITA #\$10	
	9CCC*	27 FA	288		BEQ SENDCH	Des que le registre de transmission est vide
			289	*		Le bit correspondant est positionné
	9CCE*	86 3C	290		LDA #60	
		=9CD0*	291	NODSR	EQU *	
	9CD0*	4A	292		DECA	Boucle de 300 us pour assurer le positionnement
			293	*		de DSR par le récepteur
	9CD1*	96 E9	294		LDA SIOTSTATUS	
	9CD3*	4B	295		LSLA	
	9CD4*	28 F2	296		BMI SENDCH	Attente de DSR à 0
	9CD6*	D7 E8	297		STB SIOTRANSM	L'octet est mis dans le registre de transmission
	9CD8*	39	298		RTS	
			299			

	300	*		
	301	*	Reception d'un octet	
	302	*		
=9CD9*	303	RECEPT	EQU	*
9CD9*	304	LDA	SIOSTATUS	
9CDB*	305	BITA	#\$8	A-t-on recu un octet ?
9CDD*	306	BNE	RECUDCT	
	307			
=9CDF*	308	ERROR	EQU	*
9CDF*	309	ORCC	#\$10	Pour eviter des arrivees intempestives
9CE1*	310	LDB	MODE-OFFSET+Y	
9CE3*	311	LSRB		
9CE4*	312	LSRB		On teste si on est en modem ou terminal
9CE5*	313	BCS	RECEPT1	En terminal on saute la suite
9CE7*	314	LSLA		On attend la descente de DSR
9CE8*	315	BMI	RECEPT2	
	316			
=9CEA*	317	RECEPT1	EQU	*
9CEA*	318	LDA	SIOCMDE	
9CEC*	319	ORA	#8	RTS est mis a 0
9CEE*	320	STA	SIOCMDE	
	321			
=9CF0*	322	RECEPT2	EQU	*
9CF0*	323	LEAS	2+S	
9CF2*	324	BRA	RSERR	
	325			
=9CF4*	326	RECUDCT	EQU	*
9CF4*	327	LDB	STORECEPT	
9CF6*	328	ANDA	#3	Framing error ou parity error ?

.MAIN. CR6809/11 version 20.54.23      21-Feb-86 11:37:24      Page 2-5  
 ASEM.SRC

Error	Addr	Code	Seq	Source statement	
	9CF8*	26 E5	329	BNE      ERROR	
	9CFA*	E7 64	330	STB      4,S	On range B dans la pile, le PULS le restituera.
	9FCF*	39	331	RTS	
			332		
			333	*	
			334	*	Reception geree par interruption
			335	*	
	=9CFD*	39	336	* ECU      EQU      *	
	9CFD*	86 A7E9	337	LDA      >SIOSTATUS	Flag IRQ a 0
	9D00*	85 08	338	BITA      #8	Est-ce une reception ?
	9D02*	27 08	339	BEQ      DSR	
	9D04*	86 A7EA	340	LDA      >SIOCMDE	Generation d'une interruption annulee par
	9D07*	84 F7	341	*	lecture du registre status
	9D09*	87 A7EA	342	ANDA      #\$F7	RTS a l (peripheriques lents)
	=9DOC*		343	STA      >SIOCMDE	
	9DOC*	381	344	DSR      EQU      *	
			345	RTI	
			346		

.MAIN. CR6809/11 version 20.54.23      21-Feb-86 11:37:24      Page 3  
 ASEM.SRC

Error	Addr	Code	Seq	Source statement
			348	INCLUDE LOADER
			349	*****
			350	*
			351	*
			352	*
			353	Ce programme assure le lancement d'un handler (RS232) ecrit en code
			354	autorelogeable.
			355	Cette partie est charge en meme temps que le handler mais se
			356	remplace ensuite par des donnees du BASIC.
			357	Cette partie concerne le BASIC MOS version 1.0 version cassette,
			358	disquette 5"1/4 simple et double densite et CDD.
			359	Il doit etre situe en dernier lors de l'assemblage.
			360	*
			361	*****
			362	*
			363	EQUATES
			364	*
			365	*****
			366	*
			367	Liens au BASIC 1.0 MOS
			368	*
=E233	369	FMERR	EQU	\$E233
=EB8F	370	IOERR	EQU	\$EB8F
=E966	371	ENDINI	EQU	\$E966
	372			
=25A3	373	DEBDOS	EQU	\$25A3
	374			
=0099	375	FCBLEN+SD	EQU	\$99
=0119	376	FCBLEN+DD	EQU	\$119
=0118	377	FCBLEN+QD	EQU	\$118
	378			
=278A	379	BUFBOT+SD	EQU	DEBDOS+\$217
=2836	380	BUFBOT+DD	EQU	DEBDOS+\$293
=2749	381	BUFBOT+QD	EQU	DEBDOS+\$1A6
	382			
=25AD	383	NHFCB+SD	EQU	DEBDOS+\$A
=25AC	384	NHFCB+DD	EQU	DEBDOS+\$9
=25AF	385	NHFCB+QD	EQU	DEBDOS+\$C
	386			
=25AE	387	BUFSIZ+SD	EQU	DEBDOS+\$B
=25AD	388	BUFSIZ+DD	EQU	DEBDOS+\$A
=25B0	389	BUFSIZ+QD	EQU	DEBDOS+\$D
	390			
=270A	391	FCBOARD+SD	EQU	DEBDOS+\$167
=2709	392	FCBOARD+DD	EQU	DEBDOS+\$166
=25EC	393	FCBOARD+QD	EQU	DEBDOS+\$49
	394			
	395	DCBTBL	EQU	\$23C8
	397	SCRATCH	EQU	\$C424

Error	Addr	Code	Sec	Source statement	
	=C370	398	READY	EQU	\$C370
	=2113	399	TXTTAB3	EQU	\$2113
	=21F6	400	FCERR	EQU	\$21F6
	=223F	401	EXECAD	EQU	\$223F
		402			
	=2080	403	DKFLG	EQU	\$2080
		404			*****
		405			*****
		406	*		
		407	*		
		408	*	I'execution commence ici	
		409	*		
		410	*	le DCB contient actuellement des adresses sans offset.	
		411	*	il faut y additionner une valeur pour avoir ses adresses	
		412	*	absolus	
		413	*		
		414			
	=010D	415	RSLEN	EQU	#-R232
		416			Longueur utile du handler.
		417			
	=21	418	SETOP	EQU	\$21
	=9D0D*	419	EXEC	EQU	*
		420	*		
		421	*		-----
		422	*	Calcul de l'endroit ou mettre le handler	
		423	*		-----
9000'	7F 2083/	424	CLR	RS STA	Ainsi on ne se croit pas ouvert !!!
		425			
9D10'	8E 21F6	426	LDX	#FCERR	
9D13'	8F 223F	427	STX	EXECAD	
		428			
9D16'	8E 2113	429	LDX	#TXTTAB3	La ou on charge pour les K7.
9D19'	7D 2080	430	TST	DKFLG	
9D1C'	27 14	431	BEQ	K7	Pas de disque.
		432			
9D1E'	8E 2836	433	LDX	#BUFBDT+DD	
9D21'	F6 25A5	434	LDB	DEB00S+2	On trouve ici \$38 pour le dos SD
		435	*		\$80 DD
		436	*		\$A8 QD
9D24'	C0 BD	437	SUBB	#\$8D	
9D26'	27 OA	438	BEQ	K7	
		439			
9D28'	2A 05	440	BPL	IS	B vaut ici \$1E pour le QD et \$A8 pour le SD
9D2A'	(8E 278A)	441	LDX	#BUFBDT+SD	
9D2D'	20 03	442	BRA	K7	
		443			
9D2F'	8E 2749	444	IS	LDX	#BUFBDT+QD
		445			
9D32'	34 14	446	K7	PSHS	8,X
9D34'	AE 84	447	LDX	*	[1,S] Adresse debut pour le handler reqX = cette adresse

Error	Addr	Code	Sec	Source statement	
			448	*	
			449	*	
			450	*	-----
			451	*	Calcul de la table du handler.
			452	LDU	TXTTAB
9D36'	DE 13	453	LEAU	4,U	On a charge en TXTTAB...
9D38'	33 44	454	TFR	X+0	On saute le nom (4 caracteres)
9D3A'	1F 10	455			Sert d'offset, pour la table du handler.
		456			
9D3C'	108E 0005	457	DCBLDP	LDY	#5
=9D40'		458	EQU	*	Compteur pour 6 adresses
9D40'	AE C4	459	LDX	*U	
9D42'	30 83	460	LEAX	0,X	On ajoute l'offset a tous les adresses du DCB
9D44'	AF C1	461	STX	*U++	
9D46'	31 3F	462	LEAY	-1,Y	Fin du DCB ?
9D48'	26 F6	463	BNE	DCBLDP	A la prochaine
		464	*		
		465	*		Recherche d'une place vide en DCBTBL
		466	*		-----
9D4A'	CE 23CA	467	LDJ	#DCBTBL	
9D4D'	C6 FF	468	LDB	#-1	Le numero du device
=9D4F'		469	CHERCH	EQU	*
9D4F'	5C	470	INC8		...commence par 0
9D50'	AE C1	471	LDX	*U++	Une entree non utilisee ?
9D52'	27 04	472	BEQ	TRouve	Oui
9D54'	C1 0F	473	CMPB	#15	Fin de la table ?
9D56'	26 F7	474	BNE	CHERCH	Non, essaie le prochain
	=9D58'	475	*		Si on ne trouve rien, alors on prend le dernier
9D58'	AE F8 01	476	TROUVE	EQU	*
9D58'	AF 5E	477	LDX	{1,S}	Pointeur sur notre DCB
		478	STX	-2,U	On l'accroche
		479			
		480	*		
		481	*	On regarde si on a un disque ou non	
		482	*		
		483			
9D5D'	7D 2080	484	TST	DKFLG	
9D60'	26 0F	485	BNE	DISQUE	
		486	*		
		487	*		Sur K7, pas de block-move.
9D62'	30 8C A8	488	LEAX	EXEC,PCR	On recuper la place prise par le loader
9D65'	6F 80	489	CLR	,X+	
9D67'	9F 13	490	STX	TXTTAB	On met a jour le pointeur de texte
		491	*		
		492	*		
9D69'	8E C37D	493	LDX	#READY	
9D6C'	34 10	494	PSHS	X	Et on retourne au point d'entree de BASIC
9D6E'	7E C424	495	JMP	SCRATCH	Pas JSR car SCRATCH va ecraser ce bout de code.
		496	*		
		497	*		

Error Addr	Code	Seq	Source statement		
		498	*		Dans le cas d'un disque
		499	*		il faut faire autre chose
=9071'		500	*		
		501	DISQUE EQU *		
		502	*		
9071'	DE 13	503	LDX [1,S]		On le recopie dans les tampons disques.
9073'	108E 010D	504	LDU TXTTAB		Début du handler
		505	LDY #RSLEN		Longueur du handler.
9077'	A6 C0	506			
9079'	A7 80	507	RECOP	LDA U+	Recopie Y octets de U vers X
907B'	31 3F	508	STA X+		
907D'	26 F8	509	LEAY -1,Y		
		510	BNE RECOP		
		511			
907F'	10AE 61	512			
		513	LDY 1,S		Adresse de BUFBOT
9082'	30 8C 88	514			
9085'	AF A4	515	LEAX EXEC,PCR		On recuper la place prise par le loader
9087'	AF 22	516	STX ,Y		Dans BUFBOT
		517	STX Z,Y		Dans BUFFRE
9089'	FC 25AD	518			
909C'	60 E4-	519	LDD BUFSIZ+00		Taille pour les FIELDS
908E'	[27 0A	520	TST ,S		
9090'	2A 05	521	BEQ 1\$		
9092'	FC 25AE	522	BPL 2\$		
9095'	20 03	523	LDD BUFSIZ+SD		
		524	BRA 1\$		
9097'	FC 25B0	525	2\$ LDD BUFSIZ+QD		
909A'	30 88	526	LEAX D,X		
909C'	AF 24-	527	STX 4,Y		Dans BUFTOP
909E'	60 E4-	528	TST ,S		
90A0'	27 15	529	BEQ GOOD		Double densite
90A2'	2B 26-	530	BMI GSOD		Simple densite
		531			
		532			
		533			
		534			
		535			
		536	*		QD
90A4'	F6 25AF-	537	LDB NHFCB+QD		
90A7'	CE 25EC-	538	LDU #FCBOARD+QD		Adresse table des FCB
		539			
90AA'	6F 84	540	LOOPQD CLR ,X		CLR 1er octet de chaque FCB
90AC'	AF C1	541	STX ,U++		Table des pointeurs vers les FCB
90AE'	30 89 0118	542	LEAX FCBLLEN+QD,X		Adresse prochain FCB
90B2'	5A	543	DEC8		
90B3'	2A F5	544	BPL LOOPQD		
90B5'	20 24	545	BRA QUF		
		546	*		
=9087'		547	GOOD EQU *		Double densite !

Error Addr	Code	Seq	Source statement		
9087'	F6 25AC	543	LDB NHFCB+QD		
9084'	CE 2709	544	LDU #FCBOARD+QD		Adresse table des FCB
		545			
908D'	6F 84	546	LOOPQD CLR ,X		CLR 1er octet de chaque FCB
90BF'	AF C1	547	STX ,U++		Table des pointeurs vers les FCB
90C1'	30 89 0119	548	LEAX FCBLLEN+QD,X		Adresse prochain FCB
90C5'	5A	549	DEC8		
90C6'	2A F5	550	BPL LOOPQD		
90C8'	20 11	551	BRA QUF		
		552			
		553			
		554			
		555			
		556			
		557			
		558			
		559			
		560			
90CA'	=90CA'	561	GSOD FOJ *		Simple densite !
90CD'	F6 25AD	562	LDB NHFCB+SD		
90CD'	CE 270A	563	LDU #FCBOARD+SD		Adresse table des FCB
		564			
90D0'	6F 84	565	LOOPSD CLR ,X		CLR 1er octet de chaque FCB
90D2'	AF C1	566	STX ,U++		Table des pointeurs vers les FCB
90D4'	30 89 0099	567	LEAX FCBLLEN+SD,X		Adresse prochain FCB
90D8'	5A	568	DEC8		
90D9'	2A F5	569	BPL LOOPSD		
		570			
90DB'	6F 80	571	JUF CLR ,X+		
90D0'	9F 13	572	STX TXTTAB		
90DF'	30 8D 0002	573	LEAX MESS-1,PCR		
90E3'	7E E966	574	JMP ENDIVI		
		575			
90E6'	0C	576			
90E7'	40 6F 64 75 6C 65	577	MESS FCB \$0C		
90ED'	20 52 32 33 32 20	578	FCC /Module R232 install/		
90F3'	69 66 23 74 61 6C				
90F9'	6C				
90FA'	16	579	FCB \$16		
90FB'	42 65	580	FCC /Re/		
90FD'	0D 0A 00	581	FCB \$0D,\$0A,\$00		
		582			
		583			
		584	END		

## \*\* Symbol Table \*\*

BUFBOT+DD	2836	BUFBOT+QD	2749	BUFBOT+SD	2784	BUFSIZ+DD	25AD	BUFSIZ+QD	2580
BUFSIZ+SD	25AE	CHERCH	904F'	DCBLDP	904D'	DCBTBL	23C9	DEBDOS	25A3
DISQUE	9071'	DKFLG	2080	DSR	900C'	ENDINI	E966	ERROR	9C0F'
EXEC	9000'	EXEC2	9C1A'	EXECAD	223F	EXECIO	9CC2'	EXECUTE	9C17'
FCBOARD+DD	2709	FCBOARD+QD	25EC	FCBOARD+SD	270A	FCBLLEN+DD	0119	FCBLLEN+QD	0118
FCBLLEN+SD	0099	FERR	21F6	FMERR	E233	GOOD	90B7'	GOSO	90CA'
TOCH	9C6E'	I0ERR	E88F	IRQPT	2064	K7	9D32'	LECTOK	9C31'
LOOPDD	9DBD'	LOOPQD	90AA'	LOOPSD	90D0'	MESS	9DE6'	MODE	2084
NHFCB+DD	25AC	NHFCB+QD	25AF	NHFCB+SD	25AD	NOOSR	9C0D'	OFFSET	2082
OPEN	9C64'	OUF	90D8'	OUTRW	9C76'	OUVRIR	9C84'	R232	9C00'
R232CLS	9C23'	R232INP	9C27'	R232OPN	9C10'	R232OUT	9C34'	R232PRM	9C30'
READY	C370	RECEPT	9C09'	RECEPT1	9CEA'	RECEPT2	9CF0'	RECOP	9077'
RECU	9CFD'	RECUJCT	9CF4'	RETUR	9C33'	RSCLS	0010	RS.NRD	0080
RS.OPC	2082	RS.OPR	0001	RS.RDC	0002	RS.STA	2083	RS.WRC	0009
RS232	9C42'	RS3	9C9A'	RSERR	9C7F'	RSLEN	0100	RSNRD	9C78'
SCRATCH	C424	SENDCH	9CC8'	SIOCMDE	A7EA	SIOCNTL	A7EB	SIORECEPT	A7E8
SIOSTATUS	A7E9	SIOTRANSM	A7E8	TROUVE	905B'	TXTTAB	2113	VITESSE	2085