

(BASIC 128) PROGRAMME SOURCE POUR T07-70 ET T07

.MAIN. CR6809/11 version 20.54.23
ASEM128.SRC

18-Feb-86 15:36:32 Page 1

```

Error Addr  Code          Seq  Source statement
          =0000          1      SECT   ASEN
                          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 controleur RS232 dont le listing est fourni
                          7      *          separement.
                          8      *
                          9      *          Ce module, ainsi que le handler sera charge dans le T07 ou le
10     *          T07-70 via un chargeur ecrit en basic
11     *
12     *          Pour les personnes ne disposant pas d'assembleur, un programme
13     *          ecrit en Basic est fourni conjointement qui permettra de creer le
14     *          fichier objet necessaire.
15     *
16     *          Ce module ne gere pas les demandes de parametres divers, tels
17     *          la vitesse ou le nombre de bits. Ces parametres sont supposes positionnes
18     *          par le programme Basic a l'aide de POKE aux 2 cases memoires specifiees.
19     *
20     *          Ce programme doit etre le premier des trois programmes.
21     *
22     *****
23
24
25     *
26     *          Debut du programme
27     *
          =8C00          28     ORG     $8C00
29
          =8C00'         30     R232   EQU     *
BC00'  52 32 33 32     31     FCC     /R232/          Nom du peripherique sous basic
BC04'  0010           32     FDB     R232OPN-R232      Ouverture
BC06'  0023           33     FDB     R232CLS-R232      Fermeture
BC08'  0027           34     FDB     R232INP-R232      Lecture
BC0A'  0034           35     FDB     R232OUT-R232      Ecriture
BC0C'  003D           36     FDB     R232PRM-R232      Parametres
BC0E'  23A1           37     FDB     FMERR          EOF
38
          =8C10'         39     R232OPN EQU     *          Ouverture de la voie serie
BC10'  C6 01          40     LDB     #RS.CPR      Demande d'ouverture
BC12'  F1 6083        41     CMPB    RS.STA       Est-ce deja ouvert ?
BC15'  27 1C          42     BEQ     RETOUR      Oui, on ne fait rien
43
          =8C17'         44     EXECUTE EQU     *
BC17'  F7 6082        45     STB     RS.OPC
          =8C1A'         46     EXEC2   EQU     *
BC1A'  8D 26          47     BSR     RS232
BC1C'  24 15          48     BCC     RETOJR
BC1E'  8D 03          49     BSR     R232CLS      On ferme la liaison
BC20'  7E 2ED2        50     JMP     IOERR        On va faire IO Error dans le basic.

```

.MAIN. CR6809/11 version 20.54.23
ASEM128.SRC

18-Feb-86 15:36:32 Page 1-1

```

Error Addr  Code          Seq  Source statement
          =8C23'         51
BC23'  C6 10          52     R232CLS EQU     *
BC25'  20 F0          53     LDB     #RS.CLS          Demande de fermeture
                          54     BRA     EXECUTE      Et on l'execute
55
          =8C27'         56     R232INP EQU     *
BC27'  C6 02          57     LDB     #RS.RDC          Demande de lecture
BC29'  F7 6082        58     STB     RS.OPC
BC2C'  8D 14          59     BSR     RS232          On demande la lecture
BC2E'  24 01          60     BCC     LECTOK        Pas d'erreur, il y a un caractere
BC30'  5F              61     CLRB
                          62     LECTOK EQU     *
BC31'  1F 98          63     TFR     B,A          On met le resultat dans A
BC33'  39              64     RETOJR   RTS
65
          =8C34'         66     R232OUT EQU     *
BC34'  C6 09          67     LDB     #RS.WRC          Demande d'ecriture
BC36'  F7 6082        68     STB     RS.OPC
BC39'  1F 89          69     TFR     A,B          Le caractere etait dans A
BC3B'  20 0D          70     BRA     EXEC2
71
          =8C3D'         72     R232PRM EQU     *
BC3D'  4F              73     CLRA
BC3E'  5F              74     CLRB
BC3F'  1F 01          75     TFR     D,X          aucune position de specifiee sur la voie serie
BC41'  39              76     RTS
77

```

```

Error Addr Code      Seq Source statement
79      INCLUDE RS232
80      *****
81      *
82      *
83      *      Ce programme permet de piloter les entrees-sorties serie
84      *      de type RS232C de la nouvelle extension RS 57-932 sur les T07 et T07-70.
85      *
86      *      Pour les personnes ne disposant pas de l'assembleur, un
87      *      programme ecrit en Basic, permettra de creer le fichier correspondant.
88      *      Ce fichier doit etre situe en deuxieme, en suivant le handler, et en
89      *      precedant le loader.
90      *
91      *      Le registre RS.OPC permet de savoir le type d'operation demandee :
92      *
93      *      - RS.OPR : ouverture en lecture ecriture
94      *      - RS.RDC : lecture d'un caractere (dans B)
95      *      - RS.WRC : ecriture d'un caractere (contenu dans B)
96      *      - RS.CLS : Fermeture
97      *
98      *      A la fin de l'operation, le CY a 0 indique que tout s'est bien
99      *      passe, le CY a 1 qu'il y a eu une erreur. Le registre RS.STA contient
100     *      le type de l'erreur :
101     *
102     *      - RS.OPR : Pas de caractere disponible
103     *      - RS.CLS : Fermee
104     *      - RS.NRD : Unite non prete
105     *
106     *      Le registre VITESSE permet de choisir la vitesse de transmission :
107     *
108     *      Vitesse          Valeur dans BAUDS
109     *      50                $1
110     *      75                $2
111     *      110               $3
112     *      134               $4
113     *      150               $5
114     *      300               $6
115     *      600               $7
116     *      1200              $8
117     *      1800              $9
118     *      2400              $A
119     *      3600              $B
120     *      4800              $C
121     *      7200              $D
122     *      9600              $E
123     *      19200             $F
124     *
125     *      Le registre MODE permet de definir different parametres :
126     *
127     *      Bit 7-6          Bit 5          Bit 4-3-2          Bit 1          Bit 0
128     *      Longueur        Horloge        Parite             Mode           Stop bits

```

```

Error Addr Code      Seq Source statement
129     *      11 5 bits      0 interne      111 bit a 0      0 modem      0 2 bits
130     *      10 8 bits      1 externe      101 bit a 1      1 terminal    1 1 bit
131     *      01 7 bits
132     *      00 6 bits
133     *
134     *      001 parite paire
135     *      000 sans parite
136     *      *****
137     *
138     *      Page 0 du moniteur
139     *
140     *      =6021      139  IRQPT EQU      $6021
141     *      =6082      140  OFFSET EQU     $6082
142     *      =6083      141  RS.OPC EQU     $6082
143     *      =6084      142  RS.STA EQU     $6083
144     *      =6085      143  MODE EQU      $6084
145     *      144  VITESSE EQU    $6085
146     *
147     *      Operations demandees
148     *
149     *      =0001      149  RS.OPR EQJ     $01      Ouverture en lecture/ecriture
150     *      =0002      150  RS.RDC EQU     $02      Lecture d'un caractere
151     *      =0009      151  RS.WRC EQJ     $09      Ecriture d'un caractere
152     *      =0010      152  RS.CLS EQJ     $10      Fermeture
153     *
154     *
155     *      Cas d'erreur
156     *
157     *      =0080      157  RS.NRD EQU     $80      Peripherique non pret
158     *
159     *
160     *      Registres du controleur
161     *
162     *      =E7E8      162  SIOTRANSM EQU   $E7E8
163     *      =E7E8      163  SIODRECEPT EQU  $E7E8
164     *      =E7E9      164  SIOSTATUS EQU   $E7E9
165     *      =E7EA      165  SIOCMBDE EQU   $E7EA
166     *      =E7EB      166  SIODCNTRL EQU  $E7EB
167     *
168     *
169     *
170     *      =E7      170  SETDP EQU      $E7      Page 0 sur les registres de commande
171     *
172     *      =8C42*     172  RS232 EQU      *
173     *      BC42*     173  PS+5 EQU      A,B,CC,DP,X,Y,U
174     *      BC44*     174  LDA EQU        #$E7
175     *      BC46*     175  TFR EQU        A,DP      DP est initialise
176     *      BC48*     176  LDY EQU        #OFFSET    Y pointe sur les registres de la page 0
177     *      BC4C*     177  LDA EQU        RS.OPC-OFFSET.Y Operation souhaitee
178     *      BC4E*     178  TFR EQU        A,CC      On va se debrancher selon le cas

```

Error Addr	Code	Seq	Source statement
BC50*	28 1C	179	BMI IOCH Ecriture d'un caractere
BC52*	29 1A	180	BVS IOCH Lecture d'un caractere
BC54*	25 0E	181	BCS OPEN Ouverture en lecture/ecriture
		182	*
		183	*
		184	*
		185	Dans tous les autres cas on ferme
BC56*	86 10	185	LDA #RS.CLS
BC58*	A7 21	185	STA RS.STA-OFFSET,Y On range l'etat ferme
		187	
BC5A*	96 EA	188	LOA SIOCMDE
BC5C*	8A 02	189	ORA #2 Devalidation des IRQ
BC5E*	97 EA	190	STA SIOCMDE
BC60*	96 E9	191	LDA SIOSTATUS Annulation d'une eventuelle interruption
		192	*
BC62*	20 12	193	BRA OUTRW recue avant devalidation des IRQ
		194	
	=BC64*	195	OPEN EQU *
BC64*	86 01	195	LDA #RS.OPR
BC66*	A5 21	197	BITA RS.STA-OFFSET,Y
BC68*	26 15	198	BNE RSERR On etait deja ouvert, d'ou erreur
BC6A*	A7 21	199	STA RS.STA-OFFSET,Y On note l'etat ouvert
BC6C*	20 16	200	BRA OUVRIR
		201	
	=BC6E*	202	IOCH EQU *
BC6E*	86 01	203	LDA #RS.OPR Lecture d'un caractere
BC70*	A5 21	204	BITA RS.STA-OFFSET,Y On verifie que l'on est ouvert
BC72*	27 0B	205	BEQ RSERR Non, d'ou erreur
BC74*	80 4C	205	BSR EXECIO On effectue la lecture
		207	
	=BC76*	208	OUTRW EQU *
BC76*	35 7F	209	PULS A,B,CC,DP,X,Y,U
BC78*	1C FE	210	ANDCC #SFE Operation correcte ==> CY=0
BC7A*	39	211	RTS
		212	
	=BC7B*	213	RSNRD EQU *
BC7B*	86 80	214	LOA #RS.NRD
BC7D*	A7 21	215	STA RS.STA-OFFSET,Y Peripherique non pret
	=BC7F*	216	RSERR EQU *
BC7F*	35 7F	217	PULS A,B,CC,DP,X,Y,U
BC81*	1A 01	218	ORCC #01 Operation incorrecte ==> CY=1
BC83*	39	219	RTS
		220	
		221	*
		222	*
		223	*
		224	Initialisation de la voie serie
	=BC84*	225	OUVRIR EQU *
BC84*	1A 10	225	ORCC #S10 Masquage des IRQ pour mettre dans l'adresse
		227	*
BC86*	4F	228	CLR A de derouatge l'adresse de reception RS232

Error Addr	Code	Seq	Source statement
BC87*	E6 23	229	LDB VITESSE-OFFSET,Y B contient la vitesse
BC89*	1F 03	230	TFR D,U
		231	*
		232	*
		233	*
		234	*
BC8B*	A6 22	235	LDA MODE-OFFSET,Y
BC8D*	84 E1	235	ANDA #S11 On ne garde que les bits interessants
BC8F*	88 61	237	EORA #S61
BC91*	88 40	238	ADDA #S40
BC93*	1C FE	239	ANDCC #SFE
BC95*	46	240	RORA
BC96*	24 02	241	BCC RS3
BC98*	8A 80	242	ORA #S80
	=BC9A*	243	RS3 EQU *
BC9A*	33 C6	244	LEAU A,U
BC9C*	1F 30	245	TFR U,D
BC9E*	D7 EB	245	STB SIOCNTRL Ecriture dans le registre de controle
		247	*
		248	*
		249	*
BCA0*	E6 22	250	LDB MODE-OFFSET,Y
BCA2*	C4 1C	251	ANDB #S1C
BCA4*	58	252	LSLB
BCA5*	58	253	LSLB
BCA6*	58	254	LSLB
BCA7*	96 EA	255	LOA SIOCMDE
BCA9*	84 10	256	ANDA #S1U
BCAB*	8A 09	257	ORA #9 RTS etat 0 et DTR : RTS est relie a DSR et
BCAD*	34 02	258	PSHS A DTR a CTS
BCAF*	EA E0	259	ORB #S+
BCB1*	D7 EA	260	STB SIOCMDE Ecriture dans le registre de commande
		261	*
		262	*
		263	*
		264	Test de presence de l'extension RS232
BCB3*	53	264	COMB
BCB4*	D9 EA	265	EORB SIOCMDE
BCB6*	5C	266	INCB
BCB7*	26 C2	267	BNE RSNRD Elle n'est pas la
		268	*
		269	*
		270	*
		271	Adresse de retour apres reception d'un octet
BCB9*	33 8D 0040	271	LEAU RECU,PCR
BCBD*	FF 6021	272	STU IRQPT
BCC0*	20 B4	273	BRA OUTRW
		274	*
		275	*
		275	*
	=BCC2*	277	EXECIO EQU *
BCC2*	A6 A4	278	LDA RS.OPC-OFFSET,Y

Error Addr	Code	Seq	Source statement	
BCC4*	85 02	279	BITA #RS.2DC	Est-ce une lecture ?
BCC6*	26 11	280	BNE RECEP	Absolument
		281		
		282	*	
		283	*	On execute l'écriture
		284	*	
	=BCC8*	285	SENDCH EQU *	
BCC8*	96 E9	285	LDA SIOSTATUS	
BCCA*	85 10	287	BITA #10	
BCCC*	27 FA	288	BEQ SENDCH	Des que le registre de transmission est vide
		289	*	Le bit correspondant est positionne
BCCF*	86 3C	290	LDA #60	
	=BCD0*	291	NOOSR EQU *	
BCD0*	4A	292	DECA	Boucle de 300 us pour assurer le positionnement
		293	*	de DSR par le recepteur
BCD1*	96 E9	294	LDA SIOSTATUS	
BCD3*	48	295	LSLA	
BCD4*	28 F2	296	BMI SENDCH	Attente de DSR a 0
BCD6*	D7 E8	297	STB SIOTRASH	L'octet est mis dans le registre de transmission
BCD8*	39	298	RTS	
		299		
		300	*	
		301	*	Reception d'un octet
		302	*	
	=BCD9*	303	RECEP EQU *	
BCD9*	96 E9	304	LDA SIOSTATUS	
BCDB*	85 08	305	BITA #8	A-t-on recu un octet ?
BCDD*	26 15	306	BNE RECUJCT	
		307		
	=BCDF*	308	ERRR EQU *	
BCDF*	1A 10	309	ORCC #10	Pour eviter des arrivees intempestives
BCE1*	E6 22	310	LDB MODE-OFFSET,Y	
BCE3*	54	311	LSRB	
BCE4*	54	312	LSRB	On teste si on est en modem ou terminal
BCE5*	25 03	313	BCS RECEP1	En terminal on saute la suite
BCE7*	48	314	LSLA	On attend la descente de DSR
BCE8*	2B 06	315	BMI RECEP2	
		316		
	=BCEA*	317	RECEP1 EQU *	
BCEA*	96 EA	318	LDA SIOCMDE	
BCEC*	8A 08	319	ORA #8	RTS est mis a 0
BCEE*	97 EA	320	STA SIOCMDE	
		321		
	=BCFD*	322	RECEP2 EQU *	
BCFD*	32 62	323	LEAS 2,S	
BCF2*	20 8B	324	BRA RSERR	
		325		
	=BCF4*	326	RECUJCT EQU *	
BCF4*	D6 E8	327	LDB SIORECEP	
BCF6*	84 03	328	ANDA #3	Framing error ou parity error ?

Error Addr	Code	Seq	Source statement	
BCF8*	26 E5	329	BNE ERROR	
BCFA*	E7 64	330	STB 4,S	On range B dans la pile, le PULS le restituera.
BCFC*	39	331	RTS	
		332		
		333	*	
		334	*	Reception geree par interruption
		335	*	
	=BCFD*	336	RECU EQU *	
BCFD*	86 E7E9	337	LDA >SIOSTATUS	Flag IRQ a 0
BD00*	85 08	338	BITA #3	Est-ce une reception ?
BD02*	27 08	339	BEQ DSR	
BD04*	86 E7EA	340	LDA >SIOCMDE	Generation d'une interruption annulee par
		341	*	lecture du registre status
BD07*	84 F7	342	ANDA #5F7	RTS a 1 (peripheriques lents)
BD09*	87 E7EA	343	STA >SIOCMDE	
	=BDOC*	344	DSR EQU *	
BDOC*	38	345	RTI	
		346		

Error Addr	Code	Seq	Source statement
		349	INCLUDE LOADER
		349	*****
		350	*
		351	* Ce programme assure le lancement d'un handler (RS232) ecrit en code
		352	* autorelogeable.
		353	* Cette partie est charge en meme temps que le handler mais se
		354	* remplace ensuite par des donnees du BASIC.
		355	*
		355	* Cette partie ne concerne que le BASIC version 1.0 version cassette.
		357	* disquette 5 1/4 simple et double densite.
		356	*
		357	* Il doit etre situe en dernier lors de l'assemblage.
		360	*
		361	*****
		362	*
		363	* EQUATES
		364	*
		365	*****
		365	*
		367	* liens au BASIC 1.0
		368	*
=255E		369	FMERR EQU \$255E
=39A1		370	IDERR EQU \$39A1
=686C		371	BUFBOTS EQU \$686C
=68E9		372	BUFBOTD EQU \$68E9
=63B1		373	DCBTBL EQU \$63B1
=04B8		374	SCRTOH EQU \$4B8
=03EE		375	READY EQU \$3EE
=623F		375	EXECAD EQU \$623F
=611C		377	TXTTAB EQU \$611C
=058C		379	FCERR EQU \$058C
=607F		379	RUNFLG EQU \$607F
=8512		380	INITSD EQU \$8512
=879C		381	INITDD EQU \$879C
=67EB		382	SIGNSD EQU \$67EB
=67E8		383	SIGNDD EQU \$67E8
=6080		384	DKFLG EQU \$6080
		385	*
		386	*
		387	* l'execution commence ici
		388	*
		389	* le DCB contient actuellement des adresses sans offset,
		390	* il faut y additionner une valeur pour avoir les adresses
		391	* absolus
		392	*
=0100		393	RSLEN EQU *-R232 Longueur utile du handler.
		394	*
		395	*
=61		396	SETOP \$61
=8000'		397	EXEC EQU *

Error Addr	Code	Seq	Source statement
		398	*
		399	*
		400	* Calcul de l'endroit ou mettre le handler
		401	-----
		401	*
BD00'	7F 6083	402	CLR RS,STA On n'est pas ouvert
		403	*
BD10'	9E 1C	404	LDX TXTTAB La ou on charge pour les K7.
BD12'	7D 6080	405	TST DKFLG
BD15'	27 00	406	BEQ K7 Pas de disque.
		407	*
BD17'	BE 68E9	408	LDX BUFBOTD On le recopie dans les tampons disques.
		409	*
BD1A'	F6 65F5	410	LDB \$65F5 En \$65F5 il y a \$85 pour le dos simple densite
BD1D'	CD 86	411	SUBB \$86 et \$87 pour le dos double densite.
		412	*
BD1F'	2A 03	413	BPL K7
BD21'	BE 686C	414	LDX BUFBOTS BUFBOT n'est pas au meme endroit selon
		415	* les versions de DOS.
BD24'	34 14	416	K7 PSMS B*X
		417	*
		418	*
		419	*
		420	*
		421	* Calcul de la table du handler.
		422	-----
		422	*
BD26'	DE 1C	423	LDU TXTTAB On a charge en TXTTAB...
BD28'	33 44	424	LEAU 4,U On saute le nps (4 caracteres)
BD2A'	1F 10	425	TFR X,D Sert d'offset, pour la table du handler.
		426	*
BD2C'	108E 0005	427	LDY #5 Compteur pour 6 adresses
=BD30'		428	DCBLCP EQU *
BD30'	AE C4	429	LDX ,U
BD32'	30 88	430	LEAX D,X On ajoute l'offset a tous les adresses du DCB
BD34'	AF C1	431	STX ,U++
BD36'	31 3F	432	LEAY -1,Y Fin du DCB ?
BD38'	26 F6	433	BNE DCBLCP A la prochaine
		434	*
		435	*
		435	* Recherche d'une place vide en DCBTBL
		437	-----
		437	*
BD3A'	CE 63B1	438	LDJ #DCBTBL
BD3D'	C6 FF	439	LDB #-1 Le numero du device
=BD3F'		440	CHERCH EQU *
BD3F'	5C	441	INCB
BD40'	AE C1	442	LDX ,U++ Une entree non utilisee ?
BD42'	27 04	443	BEQ TROUVE Oui
BD44'	C1 0F	444	CMPB #15 Fin de la table ?
BD46'	26 F7	445	BNE CHERCH Non, essaie le prochain
		446	*
=BD48'		447	TROUVE EQU * Si on ne trouve rien, alors on prend le dernier

Error Addr	Code	Seq	Source statement	
BD48*	AE 61	448	LDX 1,S	Pointeur sur notre DCB
BD4A*	AF 5E	449	STX -2,U	On l'accroche
		450		
		451		
		452		
BD4C*	8E 088C	453	LDX #FCERR	
BD4F*	BF 623F	454	STX EXECAD	Pour eviter un EXEC sans adresse
		455		
		456	*	
		457	*	On regarde si on a un disque ou non
		458	*	
BD52*	7D 6080	459	TST DKFLG	
BD55*	26 0F	460	BNE DISQUE	
		461	*	
BD57*	30 8C B3	462	LEAX EXEC,PCR	On recupere la place prise par le loader
BD5A*	6F 80	463	CLR ,X+	
BD5C*	9F 1C	464	STX TXTTAB	On met a jour le pointeur de texte
		465	*	
		466	*	
BD5E*	8E 03EE	467	LDX #READY	
BD61*	34 10	468	PSHS X	Et on retourne au point d'entree de Basic
BD63*	7E 0488	469	JMP SCRATCH	Pas JSR car SCRATCH va ecraser ce bout de code.
		470	*	
		471	*	
		472	*	Dans le cas d'un disque
		473	*	il faut faire autre chose
		474	*	
	=BD66*	475	DISQUE EQU *	
		476	*	
BD65*	35 14	477	PULS B,X	On le recopie dans les tampons disques.
BD68*	DE 1C	478	LDU TXTTAB	Debut du handler
BD6A*	108E 010D	479	LDY #RSLN	Longueur du handler.
		480		
BD6E*	A6 C0	481	RECOP LDA U+	Recopie Y octets de U vers X
BD70*	A7 80	482	STA X+	
BD72*	31 3F	483	LEAY -1,Y	
BD74*	26 F8	484	BNE RECOP	
		485	*	Ici regX pointe sur la 1ere case libre
		486	*	apres le handler.
		487		
BD75*	7F 607F	488	CLR RUNFLG	Pas de AUTO.BAT
		489		
BD79*	5D	490	TSTB	
BD7A*	2B 06	491	BMI SIMPLE	
		492		
BD7C*	7F 67E8	493	CLR SIGNDD	On met a zero le message de debut
BD7F*	7E 879C	494	JMP INITDD	
		495		
	=BD82*	496	SIMPLE EQU *	
		497		

Error Addr	Code	Seq	Source statement	
BD82*	7F 67E8	498	CLR SIGNSD	On met a zero le message de debut
BD85*	7E 8512	499	JMP INITSD	
		500		
		501	END	

No errors detected

* * Symbol Table * *

HUFBOTD	68E9	HUFBOTS	686C	CHERCH	BD3F*	DCBLOP	BD30*	DCBTBL	63B1
DISQUE	BD66*	DKFLG	6080	DSR	BDOC*	ERROR	BC0F*	EXEC	BD0D*
EXECZ	BC1A*	EXECAD	623F	EXECIO	BCC2*	EXECUTE	BC17*	FCERR	0B8C
FMERR	255E	INITDD	879C	INITSD	8512	IOCM	BC6E*	IDERR	39A1
IROPT	6021	K7	BD24*	LECTOK	BC31*	MODE	6084	+JDSR	BC00*
OFFSET	6082	OPEN	BC64*	OUTRW	BC76*	OUVRIR	BC84*	R232	BC00*
R232CLS	BC23*	R232IMP	BC27*	R232OPN	BC10*	R232OUT	BC34*	R232PRM	BC30*
READY	03EE	RECEP	BCD9*	RECEP1	BCEA*	RECEP2	BCFD*	RECOP	BD6E*
RECU	9CFD*	RECUJCT	BCF4*	RETOUR	BC33*	RS.CLS	0010	RS.NRD	0080
RS.DPC	6082	RS.DPR	0001	RS.RDC	0002	RS.STA	6083	RS.WRC	0009
RS232	BC42*	RS3	BC9A*	RSERR	BC7F*	RSLEN	010D	RSNRD	BC7B*
RUNFLG	607F	SCRATCH	0488	SENDCH	BCC8*	SIGNDD	67E8	SIGNSD	67E8
SIMPLE	BD82*	SIOCME	E7EA	SIOCNTL	E7E8	SIORECP	E7E8	SIOSTATUS	E7E9
SIOTRANSM	E7E8	TROUVE	BD49*	TXTTAB	611C	VITESSE	6085		