

START Job MODEM Req #912 for KSPROUL Date 11-Sep-82 20:42:46 Monitor: Rutgers/LCSR DEC-20 (Red), TOPS-20 Monit *START*

START Job MODEM Req #912 for KSPROUL Date 11-Sep-82 20:42:46 Monitor: Rutgers/LCSR DEC-20 (Red), TOPS-20 Monit *START*

```
K K SSSS PPPP RRRR 000 U U L
K K S P P R R 0 0 U U L
K K S P P R R 0 0 U U L
KKK SSS PPPP RRRR 0 0 U U L
K K S P R R 0 0 U U L
K K S P R R 0 0 U U L
K K SSSS P R R 000 UUUUU LLLLL
```

START Job MODEM Req #912 for KSPROUL Date 11-Sep-82 20:42:46 Monitor: Rutgers/LCSR DEC-20 (Red), TOPS-20 Monit *START*

START Job MODEM Req #912 for KSPROUL Date 11-Sep-82 20:42:46 Monitor: Rutgers/LCSR DEC-20 (Red), TOPS-20 Monit *START*

START Job MODEM Req #912 for KSPROUL Date 11-Sep-82 20:42:46 Monitor: Rutgers/LCSR DEC-20 (Red), TOPS-20 Monit *START*

```
N N SSSS 999 1 000 H H 000 L EEEEE SSSS
N N t S 9 9 11 0 0 H H 0 0 L E S
NN N ooo ttt eee ## S 9 9 1 0 00 H H 0 0 L E S
N N N o o t e e ## SSS 9999 1 0 0 0 ----- HHHHH 0 0 L EEEE SSS
N NN o o t eeee S 9 1 00 0 H H 0 0 L E S
N N o o t e ## S 9 1 0 0 H H 0 0 L E S
N N ooo tt eeee ## SSSS 999 111 000 H H 000 LLLLL EEEEE SSSS
```

START Job MODEM Req #912 for KSPROUL Date 11-Sep-82 20:42:46 Monitor: Rutgers/LCSR DEC-20 (Red), TOPS-20 Monit *START*

START Job MODEM Req #912 for KSPROUL Date 11-Sep-82 20:42:46 Monitor: Rutgers/LCSR DEC-20 (Red), TOPS-20 Monit *START*

START Job MODEM Req #912 for KSPROUL Date 11-Sep-82 20:42:46 Monitor: Rutgers/LCSR DEC-20 (Red), TOPS-20 Monit *START*

*
* * * * * * N O T I C E * * * * *
*
* ATTENTION!! NEW BIN NUMBERS AT HILL CENTER!
* -STARTS AUGUST 29- CHECK CAMPUS MAIL
* FOR DESCRIPTIVE MEMO OR PICK ONE UP AT HILL
* I/O COUNTER OR AID STATION-
* RESET BIN IN WYLBUR TO CONFORM!
*

THE UNITED STATES DEPARTMENT OF THE INTERIOR
 BUREAU OF LAND MANAGEMENT
 WASHINGTON, D. C. 20250

Section	Acres	Area	Notes
1	100	100	
2	200	200	
3	300	300	
4	400	400	
5	500	500	
6	600	600	
7	700	700	
8	800	800	
9	900	900	
10	1000	1000	
11	1100	1100	
12	1200	1200	
13	1300	1300	
14	1400	1400	
15	1500	1500	
16	1600	1600	
17	1700	1700	
18	1800	1800	
19	1900	1900	
20	2000	2000	
21	2100	2100	
22	2200	2200	
23	2300	2300	
24	2400	2400	
25	2500	2500	
26	2600	2600	
27	2700	2700	
28	2800	2800	
29	2900	2900	
30	3000	3000	

Section	Acres	Area	Notes
31	3100	3100	
32	3200	3200	
33	3300	3300	
34	3400	3400	
35	3500	3500	
36	3600	3600	
37	3700	3700	
38	3800	3800	
39	3900	3900	
40	4000	4000	
41	4100	4100	
42	4200	4200	
43	4300	4300	
44	4400	4400	
45	4500	4500	
46	4600	4600	
47	4700	4700	
48	4800	4800	
49	4900	4900	
50	5000	5000	
51	5100	5100	
52	5200	5200	
53	5300	5300	
54	5400	5400	
55	5500	5500	
56	5600	5600	
57	5700	5700	
58	5800	5800	
59	5900	5900	
60	6000	6000	

```
1 .TITLE MODEM Communications Program
2
3 A000 VM.ORG = $A000
4
5 ; RDRF == Receiver Data Register Full
6 ; TDRE == Transmit Data Register Empty
7
8 0000 OFFSET = $0000
9
10 00E8 ADDR1 = $00E8
11 00EC ADDR2 = $00EC ; ADDRESS OF ERROR
12
13 00EF PCL = $00EF
14 00FO PCH = $00FO
15 00F6 CHKSUM = $00F6
16 00FC TEMP = $00FC ; temporary save
17 00FD TTLMEM = $00FD
18 00FE MEMERR = $00FE
19 00FF ERRCNT = $00FF ; ERROR COUNT
20
21 EF00 CTY = $EF00
22 EF10 TTY = $EF10
23
```

```

25
26          9000          . =          $9000
27
28 9000  4C  22  90  START:  JMP  INIT          ; JUMP TABLE SO THAT DOWN-LOADING
29 9003  4C  06  FO  CTYIN:  JMP  $FO06          ; CAN BE DONE WITH OTHER DEVICES
30 9006  4C  09  FO  CTYSPA: JMP  $FO09
31 9009  4C  0C  FO  CTYOUT: JMP  $FO0C
32 900C  4C  0F  FO  GETBYT: JMP  $FO0F          ; CTY GET BYTE
33 900F  4C  12  FO  PRTBYT: JMP  $FO12          ; SYSTEM PRINT BYTE
34 9012  4C  15  FO  CRLF:   JMP  $FO15          ; NEW LINE
35 9015  4C  1E  FO  TTYIN:  JMP  $FO1E
36 9018  4C  21  FO  TTYOUT: JMP  $FO21
37 901B  4C  99  91  UPDRVR: JMP  UPLOAD          ; UP-LOAD TO A MAIN-FRAME
38 901E  00  00  00          .BYTE  0,0,0,0
39
40 9022  A9  00          INIT:   LDA  #0
41 9024  85  00          STA  OFFSET
42 9026  AD  00  EF  MODEM:  LDA  CTY          ; TEST CTY FOR INCOMING CHAR
43 9029  4A          LSR  A          ; TEST RDRF (BIT 0)
44 902A  90  29          BCC  NO.CTY
45 902C  AD  01  EF  LDA  CTY+1          ; GET CHAR FROM CTY
46 902F  29  7F          AND  #$7F
47 9031  DO  1F          BNE  NO.NUL
48 9033  20  03  90  JSR  CTYIN          ; GET NEXT CHAR
49 9036  29  7F          AND  #$7F
50 9038  FO  18          BEQ  SNDNUL          ; SEND A NUL IF NUL
51 903A  C9  60          CMP  #$60          ; CHECK IF UPPER CASE
52 903C  90  02          BLT  UPPER
53 903E  29  5F          AND  #$5F          ; CONVERT TO UPPER IF LOWER
54 9040  C9  4D  UPPER:  CMP  #'M          ; 'M' ?
55 9042  FO  20          BEQ  MONITR          ; EXIT TO MONITOR LEVEL
56 9044  C9  44          CMP  #'D          ; 'D' ?
57 9046  FO  1F          BEQ  JCODOS
58 9048  C9  4F          CMP  #'O          ; 'O' ?
59 904A  FO  1F          BEQ  SETOFF
60 904C  C9  4C          CMP  #'L          ; 'L' ?
61 904E  FO  28          BEQ  DNLOAD          ; DOWN LOAD
62 9050  DO  03          BNE  NO.CTY          ; SKIP IF INVALID CHAR
63
64 9052          SNDNUL:          ; FALL THROUGH
65 9052  20  18  90  NO.NUL: JSR  TTYOUT          ; PRINT TO TTY:
66
67 9055  AD  10  EF  NO.CTY: LDA  TTY          ; TEST TTY FOR INCOMING CHAR
68 9058  4A          LSR  A          ; TEST RDRF (BIT 0)
69 9059  90  06          BCC  NO.TTY
70 905B  AD  11  EF  LDA  TTY+1          ; GET CHAR FROM TTY
71 905E  20  09  90  JSR  CTYOUT          ; PRINT TO CTY:
72
73 9061  4C  26  90  NO.TTY: JMP  MODEM
74
75 9064  6C  FC  FF  MONITR: JMP  ($FFFC)          ; JUMP TO RESET
76
77 9067  4C  03  C6  JCODOS: JMP  $C603          ; EXIT BACK TO CODOS
78
79 906A  60          RETURN: RTS

```

Line	Code	Address	Value	Label
80	8000	00	00	START
81	8001	00	00	START
82	8002	00	00	START
83	8003	00	00	START
84	8004	00	00	START
85	8005	00	00	START
86	8006	00	00	START
87	8007	00	00	START
88	8008	00	00	START
89	8009	00	00	START
90	8010	00	00	START
91	8011	00	00	START
92	8012	00	00	START
93	8013	00	00	START
94	8014	00	00	START
95	8015	00	00	START
96	8016	00	00	START
97	8017	00	00	START
98	8018	00	00	START
99	8019	00	00	START
100	8020	00	00	START
101	8021	00	00	START
102	8022	00	00	START
103	8023	00	00	START
104	8024	00	00	START
105	8025	00	00	START
106	8026	00	00	START
107	8027	00	00	START
108	8028	00	00	START
109	8029	00	00	START
110	8030	00	00	START
111	8031	00	00	START
112	8032	00	00	START
113	8033	00	00	START
114	8034	00	00	START
115	8035	00	00	START
116	8036	00	00	START
117	8037	00	00	START
118	8038	00	00	START
119	8039	00	00	START
120	8040	00	00	START
121	8041	00	00	START
122	8042	00	00	START
123	8043	00	00	START
124	8044	00	00	START
125	8045	00	00	START
126	8046	00	00	START
127	8047	00	00	START
128	8048	00	00	START
129	8049	00	00	START
130	8050	00	00	START
131	8051	00	00	START
132	8052	00	00	START
133	8053	00	00	START
134	8054	00	00	START
135	8055	00	00	START
136	8056	00	00	START
137	8057	00	00	START
138	8058	00	00	START
139	8059	00	00	START
140	8060	00	00	START
141	8061	00	00	START
142	8062	00	00	START
143	8063	00	00	START
144	8064	00	00	START
145	8065	00	00	START
146	8066	00	00	START
147	8067	00	00	START
148	8068	00	00	START
149	8069	00	00	START
150	8070	00	00	START
151	8071	00	00	START
152	8072	00	00	START
153	8073	00	00	START
154	8074	00	00	START
155	8075	00	00	START
156	8076	00	00	START
157	8077	00	00	START
158	8078	00	00	START
159	8079	00	00	START
160	8080	00	00	START
161	8081	00	00	START
162	8082	00	00	START
163	8083	00	00	START
164	8084	00	00	START
165	8085	00	00	START
166	8086	00	00	START
167	8087	00	00	START
168	8088	00	00	START
169	8089	00	00	START
170	8090	00	00	START
171	8091	00	00	START
172	8092	00	00	START
173	8093	00	00	START
174	8094	00	00	START
175	8095	00	00	START
176	8096	00	00	START
177	8097	00	00	START
178	8098	00	00	START
179	8099	00	00	START
180	8100	00	00	START

```

83
84 906B 20 12 90   SETOFF: JSR   CRLF
85 906E A9 3F      LDA   #'?
86 9070 20 09 90   JSR   CTYOUT
87 9073 20 0C 90   JSR   GETBYT
88 9076 85 00      STA   OFFSET
89
90 9078 D8         DNLOAD: CLD           ; HEALTH INSURANCE
91 9079 A9 0D      LDA   #$0D           ; <CR>
92 907B 20 18 90   JSR   TTYOUT          ; TERMINATE CMD TO HOST COMPUTER
93 907E A9 00      LDA   #$00           ; 'NO-ERROR' CODE
94 9080 48         PHA           ; PUSH ON STACK
95 9081 85 FF      STA   ERRCNT          ; SET # of ERRORS to ZERO
96 9083 85 FE      STA   MEMERR          ; SET # OF MEMORY ERRORS TO ZERO
97 9085 85 FD      STA   TTLMEM          ; TOTAL # OF MEMORY ERRORS
98 9087 85 EC      STA   ADDRE+0
99 9089 85 ED      STA   ADDRE+1
100
101 908B 20 AE 91   LOADPT: JSR   TTYINE          ; INPUT WITH ECHO
102 ;              CMP   #'.'          ; SYM END OF PAPER TAPE (PROMPT)
103 ;              BEQ   SYMEND
104 908E C9 3B      CMP   #'.'          ; WAIT FOR ';' START OF RECORD CHAR
105 9090 D0 F9      BNE   LOADPT
106 9092 A9 00      LDA   #$00           ; INIT CHECK SUMS
107 9094 85 F6      STA   CHKSUM
108 9096 85 F7      STA   CHKSUM+1
109 9098 85 FE      STA   MEMERR          ; INIT MEM ERRS FOR THIS LINE
110 909A 20 B5 91   JSR   TTYBYT
111 909D F0 5E      BEQ   LOADEX          ; IF ZERO THEN LAST RECORD
112 909F AA        TAX           ; PUT # OF BYTES IN %X
113 90A0 20 8D 91   JSR   CHECK
114 90A3 20 B5 91   JSR   TTYBYT          ; ADDR HI
115 90A6 85 E9      STA   ADDR1+1
116 90A8 20 8D 91   JSR   CHECK
117
118 90AB 18         CLC
119 90AC A5 E9      LDA   ADDR1+1
120 90AE 65 00      ADC   OFFSET
121 90B0 85 E9      STA   ADDR1+1
122
123 90B2 20 B5 91   JSR   TTYBYT          ; ADDR LO
124 90B5 85 E8      STA   ADDR1
125 90B7 20 8D 91   JSR   CHECK
126
127 90BA 20 B5 91   LDDATA: JSR   TTYBYT
128 90BD A0 00      LDY   #$00           ; STRAIGHT INDIRECT ADDRESSING
129 90BF 91 E8      STA   (ADDR1),Y
130 90C1 D1 E8      CMP   (ADDR1),Y
131 90C3 F0 04      BEQ   MEMOK          ; CHECK MEMORY
132 90C5 E6 FE      INC   MEMERR          ; IF BAD MEMORY, ABORT CURRENT LINE of DATA
133 90C7 E6 FD      INC   TTLMEM          ; BUMP UP TOTAL # OF MEMORY ERRORS TOO
134 90C9 20 8D 91   MEMOK: JSR   CHECK
135 90CC 20 11 92   JSR   INCADR
136 90CF CA        DEX
137 90D0 D0 E8      BNE   LDDATA

```


138	90D2	20	B5	91	JSR	TTYBYT	; GET CHKSUM
139	90D5	C5	F7		CMP	CHKSUM+1	; HIGH BYTE
140	90D7	D0	07		BNE	LDERR	; INDICATE ERROR
141	90D9	20	B5	91	JSR	TTYBYT	
142	90DC	C5	F6		CMP	CHKSUM+0	; LOW BYTE
143	90DE	FO	0E		BEQ	LDOKAY	
144	90E0	68			LDERR:	PLA	; PULL the \$00 (or last '?')
145	90E1	A9	3F		LDA	#'?	; PUSH a '?' INDICATING an ERROR
146	90E3	48			PHA		
147	90E4	E6	FF		INC	ERRCNT	; COUNT # of ERRORS
148	90E6	A5	E8		LDA	ADDR1+0	; SAVE BAD ADDR LOW\
149	90E8	85	EC		STA	ADDRE+0	; >of LAST ERROR
150	90EA	A5	E9		LDA	ADDR1+1	; SAVE BAD ADDR HIGH/
151	90EC	85	ED		STA	ADDRE+1	
152							; CHECK MEMORY ERRORS
153	90EE	A5	FE		LDOKAY:	LDA	MEMERR
154	90FO	FO	99		BEQ	LOADPT	
155	90F2	20	06	90	JSR	CTYSPA	
156	90F5	A9	3F		LDA	#'?	
157	90F7	20	09	90	JSR	CTYOUT	
158	90FA	4C	8B	90	JMP	LOADPT	; CONTINUE LOADING
159							
160							; IF ERROR FALL THROUGH
161							

```

163
164 90FD A9 07          LOADEX: LDA    #$07          ; ASCII BELL
165 90FF 20 09 90      JSR    CTYOUT        ; BEEP IT
166 9102 20 B5 91      JSR    TTYBYT        ; GET START ADDR HIGH
167 9105 85 F0          STA    PCH
168 9107 20 B5 91      JSR    TTYBYT        ; GET START ADDR LOW
169 910A 85 EF          STA    PCL
170 910C 20 B5 91      JSR    TTYBYT        ; GET THE CHECK SUM, BUT IGNORE (HIGH)
171 910F 20 B5 91      JSR    TTYBYT        ; (LOW)
172 9112 20 12 90      SYMEND: JSR   CRLF          ; NEW LINE
173 9115 68             PLA
174 9116 F0 25         BEQ    LEXIT          ; LOADED CORRECTLY IF A = $00
175
176 9118 48             PHA          ; RE-SAVE '?'
177 9119 20 12 90      JSR    CRLF          ; NEW LINE
178 911C 68             PLA          ; RESTORE '?'
179 911D 20 09 90      JSR    CTYOUT        ; PRINT '?'
180 9120 20 09 90      JSR    CTYOUT
181 9123 20 09 90      JSR    CTYOUT
182 9126 A5 FF          LDA    ERRCNT
183 9128 20 OF 90      JSR    PRTBYT        ; PRINT # of ERRORS
184 912B 20 12 90      JSR    CRLF          ; NEW LINE
185 912E A9 24          LDA    #'$
186 9130 20 09 90      JSR    CTYOUT
187 9133 A5 ED          LDA    ADDRE+1
188 9135 20 OF 90      JSR    PRTBYT
189 9138 A5 EC          LDA    ADDRE+0
190 913A 20 OF 90      JSR    PRTBYT
191
192 913D                LEXIT:
193 913D A5 FD          LDA    TTLMEM
194 913F F0 12          BEQ    LEXMEM
195 9141 20 12 90      JSR    CRLF
196 9144 A9 4D          LDA    #'M
197 9146 20 09 90      JSR    CTYOUT
198 9149 A9 3D          LDA    #'=
199 914B 20 09 90      JSR    CTYOUT
200 914E A5 FD          LDA    TTLMEM
201 9150 20 OF 90      JSR    PRTBYT
202
203 9153 20 12 90      LEXMEM: JSR   CRLF
204 9156 A9 53          LDA    #'S
205 9158 20 09 90      JSR    CTYOUT
206 915B A9 3D          LDA    #'=
207 915D 20 09 90      JSR    CTYOUT
208 9160 A5 F0          LDA    PCH
209 9162 20 OF 90      JSR    PRTBYT
210 9165 A5 EF          LDA    PCL
211 9167 20 OF 90      JSR    PRTBYT
212 916A 20 06 90      JSR    CTYSPA
213 916D 20 06 90      JSR    CTYSPA
214 9170 20 06 90      JSR    CTYSPA
215 9173 A9 45          LDA    #'E
216 9175 20 09 90      JSR    CTYOUT
217 9178 A9 3D          LDA    #'=

```

218	917A	20	09	90	JSR	CTYOUT	
219	917D	A5	E9		LDA	ADDR1+1	
220	917F	20	OF	90	JSR	PRTBYT	
221	9182	A5	E8		LDA	ADDR1+0	
222	9184	20	OF	90	JSR	PRTBYT	
223	9187	20	12	90	JSR	CRLF	; NEW-LINE
224	918A	4C	26	90	JMP	MODEM	; GO BACK TO MODEM PGM
225							
226							
227					.MCALL	CHECK	
228	918D				CHECK		; CHECK SUM ROUTINE
229							

231
232
233
234
235
236
237
238
239
240
241
242
243
244
245

9199 C9 OA
919B FO 10
919D 48
919E 20 18 90
91A1 68
91A2 C9 OD
91A4 DO O7
91A6 20 15 90
91A9 C9 OA
91AB DO F9
91AD 60

UPLOAD: CMP # \$OA
BEQ UPLRTN
PHA
JSR TTYOUT
PLA
CMP # \$OD
BNE UPLRTN
UPWAIT: JSR TTYIN
CMP # \$OA
BNE UPWAIT
UPLRTN: RTS

; ROUTINE TO UP-LOAD TEXT FILES
; TO A MAIN-FRAME SYSTEM
; IGNORE LFs
; SAVE CHARACTER
; OUTPUT TO MODEM
; RESTORE CHARACTER
; IF NOT THEN DONE
; GET CHAR FROM MODEM
; ? LF ?
; WAIT FOR A LINE-FEED
; UP-LOAD RETURN

```

247
248 91AE 20 15 90 TTYINE: JSR TTYIN ; TTYIN WITH ECHO
249 91B1 20 09 90 JSR CTYOUT
250 91B4 60 RTS
251
252 91B5 20 AE 91 TTYBYT: JSR TTYINE ; GET HIGH NIBBLE
253 91B8 20 CD 91 JSR HEX
254 91BB 0A ASL A
255 91BC 0A ASL A
256 91BD 0A ASL A
257 91BE 0A ASL A
258 91BF 85 FC STA TEMP ; SAVE HIGH NIBBLE
259 91C1 20 AE 91 JSR TTYINE ; GET LOW NIBBLE
260 91C4 20 CD 91 JSR HEX
261 91C7 05 FC ORA TEMP
262 91C9 8D 00 A0 STA VM.ORG ; DISPLAY ON TOP OF VM
263 91CC 60 RTS ; RTS X = X Y = Y A = BYTE C = 1
264
265
266 91CD C9 30 HEX: CMP #'0 ; LESS THAN 0 ?
267 91CF 90 11 BLT NONHEX
268 91D1 C9 3A CMP #<'9 +1> ; ASCII 9 +1
269 91D3 90 0A BLT HEXOK
270 91D5 C9 47 CMP #<'F +1> ; ASCII F +1
271 91D7 B0 09 BGE NONHEX
272 91D9 E9 06 SBC #$06
273 91DB C9 3A CMP #$3A ; 9 < CHR < A ?
274 91DD 90 03 BLT NONHEX
275 91DF 29 0F HEXOK: AND #$0F
276 91E1 60 RTS
277
278 91E2 A9 07 NONHEX: LDA #$07 ; BELL
279 91E4 20 09 90 JSR CTYOUT
280 91E7 20 12 90 JSR CRLF
281 91EA A9 48 LDA #'H
282 91EC 20 09 90 JSR CTYOUT
283 91EF A9 45 LDA #'E
284 91F1 20 09 90 JSR CTYOUT
285 91F4 A9 58 LDA #'X
286 91F6 20 09 90 JSR CTYOUT
287 91F9 20 06 90 JSR CTYSPA
288 91FC A9 45 LDA #'E
289 91FE 20 09 90 JSR CTYOUT
290 9201 A9 52 LDA #'R
291 9203 20 09 90 JSR CTYOUT
292 9206 A9 52 LDA #'R
293 9208 20 09 90 JSR CTYOUT
294 920B 20 12 90 JSR CRLF
295 920E 4C 26 90 JMP MODEM
296
297
298 9211 E6 E8 INCADR: INC ADDR1+0
299 9213 DO 02 BNE .+4
300 9215 E6 E9 INC ADDR1+1
301 9217 60 RTS ; RTS X = X Y = Y A = A

```

MODEM,M65

302							
303	9000				.END	START	

1000
 1001
 1002
 1003
 1004
 1005
 1006
 1007
 1008
 1009
 1010
 1011
 1012
 1013
 1014
 1015
 1016
 1017
 1018
 1019
 1020
 1021
 1022
 1023
 1024
 1025
 1026
 1027
 1028
 1029
 1030
 1031
 1032
 1033
 1034
 1035
 1036
 1037
 1038
 1039
 1040
 1041
 1042
 1043
 1044
 1045
 1046
 1047
 1048
 1049
 1050
 1051
 1052
 1053
 1054
 1055
 1056
 1057
 1058
 1059
 1060
 1061
 1062
 1063
 1064
 1065
 1066
 1067
 1068
 1069
 1070
 1071
 1072
 1073
 1074
 1075
 1076
 1077
 1078
 1079
 1080
 1081
 1082
 1083
 1084
 1085
 1086
 1087
 1088
 1089
 1090
 1091
 1092
 1093
 1094
 1095
 1096
 1097
 1098
 1099
 1100
 1101
 1102
 1103
 1104
 1105
 1106
 1107
 1108
 1109
 1110
 1111
 1112
 1113
 1114
 1115
 1116
 1117
 1118
 1119
 1120
 1121
 1122
 1123
 1124
 1125
 1126
 1127
 1128
 1129
 1130
 1131
 1132
 1133
 1134
 1135
 1136
 1137
 1138
 1139
 1140
 1141
 1142
 1143
 1144
 1145
 1146
 1147
 1148
 1149
 1150
 1151
 1152
 1153
 1154
 1155
 1156
 1157
 1158
 1159
 1160
 1161
 1162
 1163
 1164
 1165
 1166
 1167
 1168
 1169
 1170
 1171
 1172
 1173
 1174
 1175
 1176
 1177
 1178
 1179
 1180
 1181
 1182
 1183
 1184
 1185
 1186
 1187
 1188
 1189
 1190
 1191
 1192
 1193
 1194
 1195
 1196
 1197
 1198
 1199
 1200
 1201
 1202
 1203
 1204
 1205
 1206
 1207
 1208
 1209
 1210
 1211
 1212
 1213
 1214
 1215
 1216
 1217
 1218
 1219
 1220
 1221
 1222
 1223
 1224
 1225
 1226
 1227
 1228
 1229
 1230
 1231
 1232
 1233
 1234
 1235
 1236
 1237
 1238
 1239
 1240
 1241
 1242
 1243
 1244
 1245
 1246
 1247
 1248
 1249
 1250
 1251
 1252
 1253
 1254
 1255
 1256
 1257
 1258
 1259
 1260
 1261
 1262
 1263
 1264
 1265
 1266
 1267
 1268
 1269
 1270
 1271
 1272
 1273
 1274
 1275
 1276
 1277
 1278
 1279
 1280
 1281
 1282
 1283
 1284
 1285
 1286
 1287
 1288
 1289
 1290
 1291
 1292
 1293
 1294
 1295
 1296
 1297
 1298
 1299
 1300
 1301
 1302
 1303
 1304
 1305
 1306
 1307
 1308
 1309
 1310
 1311
 1312
 1313
 1314
 1315
 1316
 1317
 1318
 1319
 1320
 1321
 1322
 1323
 1324
 1325
 1326
 1327
 1328
 1329
 1330
 1331
 1332
 1333
 1334
 1335
 1336
 1337
 1338
 1339
 1340
 1341
 1342
 1343
 1344
 1345
 1346
 1347
 1348
 1349
 1350
 1351
 1352
 1353
 1354
 1355
 1356
 1357
 1358
 1359
 1360
 1361
 1362
 1363
 1364
 1365
 1366
 1367
 1368
 1369
 1370
 1371
 1372
 1373
 1374
 1375
 1376
 1377
 1378
 1379
 1380
 1381
 1382
 1383
 1384
 1385
 1386
 1387
 1388
 1389
 1390
 1391
 1392
 1393
 1394
 1395
 1396
 1397
 1398
 1399
 1400
 1401
 1402
 1403
 1404
 1405
 1406
 1407
 1408
 1409
 1410
 1411
 1412
 1413
 1414
 1415
 1416
 1417
 1418
 1419
 1420
 1421
 1422
 1423
 1424
 1425
 1426
 1427
 1428
 1429
 1430
 1431
 1432
 1433
 1434
 1435
 1436
 1437
 1438
 1439
 1440
 1441
 1442
 1443
 1444
 1445
 1446
 1447
 1448
 1449
 1450
 1451
 1452
 1453
 1454
 1455
 1456
 1457
 1458
 1459
 1460
 1461
 1462
 1463
 1464
 1465
 1466
 1467
 1468
 1469
 1470
 1471
 1472
 1473
 1474
 1475
 1476
 1477
 1478
 1479
 1480
 1481
 1482
 1483
 1484
 1485
 1486
 1487
 1488
 1489
 1490
 1491
 1492
 1493
 1494
 1495
 1496
 1497
 1498
 1499
 1500
 1501
 1502
 1503
 1504
 1505
 1506
 1507
 1508
 1509
 1510
 1511
 1512
 1513
 1514
 1515
 1516
 1517
 1518
 1519
 1520
 1521
 1522
 1523
 1524
 1525
 1526
 1527
 1528
 1529
 1530
 1531
 1532
 1533
 1534
 1535
 1536
 1537
 1538
 1539
 1540
 1541
 1542
 1543
 1544
 1545
 1546
 1547
 1548
 1549
 1550
 1551
 1552
 1553
 1554
 1555
 1556
 1557
 1558
 1559
 1560
 1561
 1562
 1563
 1564
 1565
 1566
 1567
 1568
 1569
 1570
 1571
 1572
 1573
 1574
 1575
 1576
 1577
 1578
 1579
 1580
 1581
 1582
 1583
 1584
 1585
 1586
 1587
 1588
 1589
 1590
 1591
 1592
 1593
 1594
 1595
 1596
 1597
 1598
 1599
 1600
 1601
 1602
 1603
 1604
 1605
 1606
 1607
 1608
 1609
 1610
 1611
 1612
 1613
 1614
 1615
 1616
 1617
 1618
 1619
 1620
 1621
 1622
 1623
 1624
 1625
 1626
 1627
 1628
 1629
 1630
 1631
 1632
 1633
 1634
 1635
 1636
 1637
 1638
 1639
 1640
 1641
 1642
 1643
 1644
 1645
 1646
 1647
 1648
 1649
 1650
 1651
 1652
 1653
 1654
 1655
 1656
 1657
 1658
 1659
 1660
 1661
 1662
 1663
 1664
 1665
 1666
 1667
 1668
 1669
 1670
 1671
 1672
 1673
 1674
 1675
 1676
 1677
 1678
 1679
 1680
 1681
 1682
 1683
 1684
 1685
 1686
 1687
 1688
 1689
 1690
 1691
 1692
 1693
 1694
 1695
 1696
 1697
 1698
 1699
 1700
 1701
 1702
 1703
 1704
 1705
 1706
 1707
 1708
 1709
 1710
 1711
 1712
 1713
 1714
 1715
 1716
 1717
 1718
 1719
 1720
 1721
 1722
 1723
 1724
 1725
 1726
 1727
 1728
 1729
 1730
 1731
 1732
 1733
 1734
 1735
 1736
 1737
 1738
 1739
 1740
 1741
 1742
 1743
 1744
 1745
 1746
 1747
 1748
 1749
 1750
 1751
 1752
 1753
 1754
 1755
 1756
 1757
 1758
 1759
 1760
 1761
 1762
 1763
 1764
 1765
 1766
 1767
 1768
 1769
 1770
 1771
 1772
 1773
 1774
 1775
 1776
 1777
 1778
 1779
 1780
 1781
 1782
 1783
 1784
 1785
 1786
 1787
 1788
 1789
 1790
 1791
 1792
 1793
 1794
 1795
 1796
 1797
 1798
 1799
 1800
 1801
 1802
 1803
 1804
 1805
 1806
 1807
 1808
 1809
 1810
 1811
 1812
 1813
 1814
 1815
 1816
 1817
 1818
 1819
 1820
 1821
 1822
 1823
 1824
 1825
 1826
 1827
 1828
 1829
 1830
 1831
 1832
 1833
 1834
 1835
 1836
 1837
 1838
 1839
 1840
 1841
 1842
 1843
 1844
 1845
 1846
 1847
 1848
 1849
 1850
 1851
 1852
 1853
 1854
 1855
 1856
 1857
 1858
 1859
 1860
 1861
 1862
 1863
 1864
 1865
 1866
 1867
 1868
 1869
 1870
 1871
 1872
 1873
 1874
 1875
 1876
 1877
 1878
 1879
 1880
 1881
 1882
 1883
 1884
 1885
 1886
 1887
 1888
 1889
 1890
 1891
 1892
 1893
 1894
 1895
 1896
 1897
 1898
 1899
 1900
 1901
 1902
 1903
 1904
 1905
 1906
 1907
 1908
 1909
 1910
 1911
 1912
 1913
 1914
 1915
 1916
 1917
 1918
 1919
 1920
 1921
 1922
 1923
 1924
 1925
 1926
 1927
 1928
 1929
 1930
 1931
 1932
 1933
 1934
 1935
 1936
 1937
 1938
 1939
 1940
 1941
 1942
 1943
 1944
 1945
 1946
 1947
 1948
 1949
 1950
 1951
 1952
 1953
 1954
 1955
 1956
 1957
 1958
 1959
 1960
 1961
 1962
 1963
 1964
 1965
 1966
 1967
 1968
 1969
 1970
 1971
 1972
 1973
 1974
 1975
 1976
 1977
 1978
 1979
 1980
 1981
 1982
 1983
 1984
 1985
 1986
 1987
 1988
 1989
 1990
 1991
 1992
 1993
 1994
 1995
 1996
 1997
 1998
 1999
 2000

MODEM.M65 Cross reference table

VM.ORG=	A000	3#	262*																	
=	9218	26#	299																	
31#																				
32#																				
33#																				
34#																				
35#																				
36#																				
37#																				
38#																				
39#																				
40#																				
41#																				
42#																				
43#																				
44#																				
45#																				
46#																				
47#																				
48#																				
49#																				
50#																				
51#																				
52#																				
53#																				
54#																				
55#																				
56#																				
57#																				
58#																				
59#																				
60#																				
61#																				
62#																				
63#																				
64#																				
65#																				
66#																				
67#																				
68#																				
69#																				
70#																				
71#																				
72#																				
73#																				
74#																				
75#																				
76#																				
77#																				
78#																				
79#																				
80#																				
81#																				
82#																				
83#																				
84#																				
85#																				
86#																				
87#																				
88#																				
89#																				
90#																				
91#																				
92#																				
93#																				
94#																				
95#																				
96#																				
97#																				
98#																				
99#																				
100#																				

CHECK 227# 228

MODEM COMMUNICATIONS PROGRAM
 CROSS REFERENCE TABLE
 WITH A MODEM

PROGRAM CROSS REFERENCE

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
100	101	102	103	104	105	106	107	108	109	110	111	112	113	114
115	116	117	118	119	120	121	122	123	124	125	126	127	128	129
130	131	132	133	134	135	136	137	138	139	140	141	142	143	144
145	146	147	148	149	150	151	152	153	154	155	156	157	158	159
160	161	162	163	164	165	166	167	168	169	170	171	172	173	174
175	176	177	178	179	180	181	182	183	184	185	186	187	188	189
190	191	192	193	194	195	196	197	198	199	200	201	202	203	204
205	206	207	208	209	210	211	212	213	214	215	216	217	218	219
220	221	222	223	224	225	226	227	228	229	230	231	232	233	234
235	236	237	238	239	240	241	242	243	244	245	246	247	248	249
250	251	252	253	254	255	256	257	258	259	260	261	262	263	264
265	266	267	268	269	270	271	272	273	274	275	276	277	278	279
280	281	282	283	284	285	286	287	288	289	290	291	292	293	294
295	296	297	298	299	300	301	302	303	304	305	306	307	308	309
310	311	312	313	314	315	316	317	318	319	320	321	322	323	324
325	326	327	328	329	330	331	332	333	334	335	336	337	338	339
340	341	342	343	344	345	346	347	348	349	350	351	352	353	354
355	356	357	358	359	360	361	362	363	364	365	366	367	368	369
370	371	372	373	374	375	376	377	378	379	380	381	382	383	384
385	386	387	388	389	390	391	392	393	394	395	396	397	398	399
400	401	402	403	404	405	406	407	408	409	410	411	412	413	414
415	416	417	418	419	420	421	422	423	424	425	426	427	428	429
430	431	432	433	434	435	436	437	438	439	440	441	442	443	444
445	446	447	448	449	450	451	452	453	454	455	456	457	458	459
460	461	462	463	464	465	466	467	468	469	470	471	472	473	474
475	476	477	478	479	480	481	482	483	484	485	486	487	488	489
490	491	492	493	494	495	496	497	498	499	500	501	502	503	504
505	506	507	508	509	510	511	512	513	514	515	516	517	518	519
520	521	522	523	524	525	526	527	528	529	530	531	532	533	534
535	536	537	538	539	540	541	542	543	544	545	546	547	548	549
550	551	552	553	554	555	556	557	558	559	560	561	562	563	564
565	566	567	568	569	570	571	572	573	574	575	576	577	578	579
580	581	582	583	584	585	586	587	588	589	590	591	592	593	594
595	596	597	598	599	600	601	602	603	604	605	606	607	608	609
610	611	612	613	614	615	616	617	618	619	620	621	622	623	624
625	626	627	628	629	630	631	632	633	634	635	636	637	638	639
640	641	642	643	644	645	646	647	648	649	650	651	652	653	654
655	656	657	658	659	660	661	662	663	664	665	666	667	668	669
670	671	672	673	674	675	676	677	678	679	680	681	682	683	684
685	686	687	688	689	690	691	692	693	694	695	696	697	698	699
700	701	702	703	704	705	706	707	708	709	710	711	712	713	714
715	716	717	718	719	720	721	722	723	724	725	726	727	728	729
730	731	732	733	734	735	736	737	738	739	740	741	742	743	744
745	746	747	748	749	750	751	752	753	754	755	756	757	758	759
760	761	762	763	764	765	766	767	768	769	770	771	772	773	774
775	776	777	778	779	780	781	782	783	784	785	786	787	788	789
790	791	792	793	794	795	796	797	798	799	800	801	802	803	804
805	806	807	808	809	810	811	812	813	814	815	816	817	818	819
820	821	822	823	824	825	826	827	828	829	830	831	832	833	834
835	836	837	838	839	840	841	842	843	844	845	846	847	848	849
850	851	852	853	854	855	856	857	858	859	860	861	862	863	864
865	866	867	868	869	870	871	872	873	874	875	876	877	878	879
880	881	882	883	884	885	886	887	888	889	890	891	892	893	894
895	896	897	898	899	900	901	902	903	904	905	906	907	908	909
910	911	912	913	914	915	916	917	918	919	920	921	922	923	924
925	926	927	928	929	930	931	932	933	934	935	936	937	938	939
940	941	942	943	944	945	946	947	948	949	950	951	952	953	954
955	956	957	958	959	960	961	962	963	964	965	966	967	968	969
970	971	972	973	974	975	976	977	978	979	980	981	982	983	984
985	986	987	988	989	990	991	992	993	994	995	996	997	998	999
1000	1001	1002	1003	1004	1005	1006	1007	1008	1009	1010	1011	1012	1013	1014

MODEM.M65 Cross reference table

ADC	120	228													
AND	46	49	53	275											
ASL	254	255	256	257											
BCC	44	69													
BEQ	50	55	57	59	61	111	131	143	154	174	194	235			
BGE	271														
BLT	52	267	269	274											
BNE	47	62	105	137	140	240	243	299							
CLC	118	228													
CLD	90														
CMP	51	54	56	58	60	104	130	139	142	234	239	242	266	268	
	270	273													
DEX	136														
INC	132	133	147	298	300										
JMP	28	29	30	31	32	33	34	35	36	37	73	75	77	158	
	224	295													
JSR	48	65	71	84	86	87	92	101	110	113	114	116	123	125	
	127	134	135	138	141	155	157	165	166	168	170	171	172	177	
	179	180	181	183	184	186	188	190	195	197	199	201	203	205	
	207	209	211	212	213	214	216	218	220	222	223	237	241	248	
	249	252	253	259	260	279	280	282	284	286	287	289	291	293	
	294														
LDA	40	42	45	67	70	85	91	93	106	119	145	148	150	153	
	156	164	182	185	187	189	193	196	198	200	204	206	208	210	
	215	217	219	221	228	278	281	283	285	288	290	292			
LDY	128														
LSR	43	68													
ORA	261														
PHA	94	146	176	236											
PLA	144	173	178	238											
RTS	79	228	244	250	263	276	301								
SBC	272														
STA	41	88	95	96	97	98	99	107	108	109	115	121	124	129	
	149	151	167	169	228	258	262								
TAX	112														
.BYTE	38														
.END	303														
.MCALL	227														
.PAGE	24	82	162	230	246										
.TITLE	1														

Errors detected: 0

*.MODEM=MODEM

Run-time: 0 0 0 Seconds

Core used: 8K

END Job MODEM Req #912 for KSPROUL Date 11-Sep-82 20:43:33 Monitor: Rutgers/LCSR DEC-20 (Red), TOPS-20 Monit **END**

END Job MODEM Req #912 for KSPROUL Date 11-Sep-82 20:43:33 Monitor: Rutgers/LCSR DEC-20 (Red), TOPS-20 Monit **END**

* * * L P T S P L R u n L o g * * *

20:42:46 LPDAT LPTSPL version 104(16650 Rutgers/LCSR DEC-20 (Red), TOPS-20 Monit
20:42:46 LPDAT Job MODEM sequence #1427 on Printer 0 [LOCAL] at 11-Sep-82 20:42:46
20:42:50 LPMSG Starting File PS:<KSPROUL>MODEM.LST.1
20:43:33 LPMSG Finished File PS:<KSPROUL>MODEM.LST.1
20:43:33 LPEND Summary: 14 Pages of Output
20:43:33 LPEND 4 Disk Pages Read
20:43:33 LPEND 5.895 Seconds CPU Time Used

END Job MODEM Req #912 for KSPROUL Date 11-Sep-82 20:43:33 Monitor: Rutgers/LCSR DEC-20 (Red), TOPS-20 Monit **END**

END Job MODEM Req #912 for KSPROUL Date 11-Sep-82 20:43:33 Monitor: Rutgers/LCSR DEC-20 (Red), TOPS-20 Monit **END**

END Job MODEM Req #912 for KSPROUL Date 11-Sep-82 20:43:33 Monitor: Rutgers/LCSR DEC-20 (Red), TOPS-20 Monit **END**

END Job MODEM Req #912 for KSPROUL Date 11-Sep-82 20:43:33 Monitor: Rutgers/LCSR DEC-20 (Red), TOPS-20 Monit **END**

END Job MODEM Req #912 for KSPROUL Date 11-Sep-82 20:43:33 Monitor: Rutgers/LCSR DEC-20 (Red), TOPS-20 Monit **END**

END Job MODEM Req #912 for KSPROUL Date 11-Sep-82 20:43:33 Monitor: Rutgers/LCSR DEC-20 (Red), TOPS-20 Monit **END**

END Job MODEM Req #912 for KSPROUL Date 11-Sep-82 20:43:33 Monitor: Rutgers/LCSR DEC-20 (Red), TOPS-20 Monit **END**

END Job MODEM Req #912 for KSPROUL Date 11-Sep-82 20:43:33 Monitor: Rutgers/LCSR DEC-20 (Red), TOPS-20 Monit **END**

END Job MODEM Req #912 for KSPROUL Date 11-Sep-82 20:43:33 Monitor: Rutgers/LCSR DEC-20 (Red), TOPS-20 Monit **END**

END Job MODEM Req #912 for KSPROUL Date 11-Sep-82 20:43:33 Monitor: Rutgers/LCSR DEC-20 (Red), TOPS-20 Monit **END**

