

```

4      * Adventures in Flatland
5      * FLATLANDP2 v2.0: second part article for supporting
6      * 65C02 instructions disassembly listing on
7      * enhanced and unenhanced Apple //e, and ][(+) even when
8      * the ROM monitor doesn't allow for such.
9      * v2.0 brings a 65C02 mini assembler for 2, 2+, original
10     * //e, original //c (mini-assembler from scratch) and on
11     * enhanced //e (mini-assembler there but only for 6502)
12     * Created under the creative commons
13     * Author: Benoit Gilon
14     * Designed sometime in 2017 (say august/september)
15     * Due to be published in OSB issue 1
16     * OSB (One Step Beyond) is a personal magazine
17     * (editor, publisher and staff all in one person) about
18     * personal computing.
19     * Parts of published code (ROM monitor extract)
20     * were/are copyrighted Apple Computer
21     * S. Wozniak, A. Baum, John A, R. Aurricchio, B. Stearns,
22     * E. Beernink, R. Williams
23
24     OPTBI      =      0              0 for DOS 3.3, 1 for ProDOS BI
25     * b0: not meaningful
26     * b1: mini assembler already there (but not 65C02)
27     * b2: 65C02 disassembly already there.
28     R2P        =      0              Apple 2 or 2plus
29     R2E        =      1              Original ROM //e
30     R2EE       =      2              Enhanced ROM //e
31     R2CO       =      4              Original ROM //c
32
33     ROMMODEL =      R2EE
34
35     * Monitor equates
36             DUMMY $3C
003C: 00      37     A1L      DS      1
003D: 00      38     A1H      DS      1
003E: 00 00   39     A2L      DS      2
0040: 00 00   40     A3L      DS      2
0042: 00      41     A4L      DS      1
0043: 00      42     A4H      DS      1
43             DEND
44     * Used by mini assembler
45     FMT        EQU      $44
46     L          EQU      $35
47     * Misc equates
48     PCL        EQU      $3A
49     LENGTH     EQU      $2F
50     INSDS2     EQU      $F88C
51     PCADJ      EQU      $F953
52     MOVE       EQU      $FE2C
53     GETLNZ     EQU      $FD67
54     ZMODE      EQU      $FFC7
55     GETNUM     EQU      $FFA7
56     TOSUB      EQU      $FFBE
57     MODE       EQU      $31
58     PROMPT     EQU      $33
59     YSAV       EQU      $34
60     BELL       EQU      $FF3A
61     A1PC       EQU      $FE75
62     PRBYTE     EQU      $FDDA

```

```

63 PRBL2 EQU $F94A
64 MNEML EQU $F9C0
65 MNEMR EQU $FA00
66 LMNEM EQU $2C
67 RMNEM EQU $2D
68 FORMAT EQU $2E
69 COUT EQU $FDED
70 PRMN1 EQU $F8F5
71 PRYX2 EQU $FD96
72 PRBLNK EQU $F948
73 FMT2 EQU $F9A6
74 SCRN2 EQU $F879
75 IN EQU $0200
76 CURSUP EQU $FC1A

```

```

77
79 NMCMD = $17
80 CHRTBL EQU $FFCC
81 CHAR1 EQU $F9B4
82 CHAR2 EQU $F9BA
92 NNBL EQU $FF13
93 IDXMINI = 3

```

Mini assembler ROM get next c

```

104 * DOS 3.3 equates
105 DBUFP EQU $9D00
106 EXFLG EQU $AAB3
107 SSAVE EQU $AA59
108 TRUECSW EQU $AA53

```

Where the DOS buffers are
Non zero if EXEC file active
Save area for S upon DOS entry

```

110 * My own equates
111 AUXPTR EQU $06

```

```

112
113 * Some helper macros
116 MSGBE MAC
117 JSR PRMSG
118 <<<

```

```

119
120 MSGP MAC
121 LDX #0
122 MSGBE j1
123 <<<

```

```

124
125 STID MAC
126 LDA #j1
127 STA j2
128 LDA #>j1
129 STA j2+1
130 <<<

```

```

131
133
134 ORG $4000

```

```

135
136 * Ensure that this patch is suitable for the host Apple 2
137 * ROM model..

```

```

4000: 20 B2 43
4003: A2 00
4005: C0 02
4007: D0 09

```

```

138 SUITE JSR RECON
139 LDX #M0-MESBASE
150 CPY #2
156 BNE :E

```

```

157 * This patch might have been already installed.. Warn

```

```

158 * the user if this is the case.
4009: A2 6B      159      LDX    #M1-MESBASE
400B: AD 9E 9E   168      LDA    $9E9E
400E: C9 AD      169      CMP    #$AD
4010: F0 03      170      BEQ    *+5
4012: 4C FA 43   172      :E      JMP    PRMSG
173
4015: A9 D3      185      LDA    #$9CD3
4017: 38          186      SEC
4018: ED 00 9D   187      SBC    DBUFP
401B: 8D 5E 40   188      STA    :0+1
401E: A9 9C      189      LDA    #>$9CD3
4020: ED 01 9D   190      SBC    DBUFP+1
4023: AA          191      TAX
192 * If first utility to claim memory this way, then
193 * ask for one additional page for our own purpose..
4024: 0D 5E 40   194      ORA    :0+1
4027: F0 01      195      BEQ    *+3
4029: CA          196      DEX
402A: 8E 66 40   197      STX    :1+1
199 * Relocate code (don't move it yet)
402D: A9 DF      200      LDA    #AROMBA
402F: A0 40      201      LDY    #>AROMBA
4031: 85 3A      202      ]LOOP   STA    PCL
4033: C9 17      203      CMP    #FCODE-FNDVAR2+AROMBA
4035: 98          204      TYA
4036: E9 43      205      SBC    #>FCODE-FNDVAR2+AROMBA
4038: B0 35      206      BCS    :4
403A: 84 3B      207      STY    PCL+1
403C: A2 00      208      LDX    #0
403E: 20 8C F8   209      JSR    INSDS2
4041: A4 2F      210      LDY    LENGTH
4043: C0 02      211      CPY    #2
4045: D0 22      212      BNE    :3
4047: B1 3A      213      LDA    (PCL),Y
4049: AA          214      TAX
404A: 88          215      DEY
404B: B1 3A      216      LDA    (PCL),Y
404D: A8          217      TAY
404E: C9 00      218      CMP    #FIN          Only addresses with range
4050: 8A          219      TXA
4051: E9 9C      220      SBC    #>FIN          Must be < FIN to be relocated
4053: B0 14      221      BCS    :3
4055: C0 2D      222      CPY    #FNDVAR2      And >= FNDVAR2
4057: 8A          223      TXA
4058: E9 99      224      SBC    #>FNDVAR2
405A: 90 0D      225      BCC    :3
405C: 98          226      TYA
405D: E9 00      227      :0      SBC    #0
405F: A0 01      228      LDY    #1
4061: 91 3A      229      STA    (PCL),Y
4063: C8          230      INY
4064: 8A          231      TXA
4065: E9 00      232      :1      SBC    #0
4067: 91 3A      233      STA    (PCL),Y
4069: 20 53 F9   234      :3      JSR    PCADJ          Adjust PCL to length byte
406C: 4C 31 40   235      JMP    ]LOOP

```

```

236
237 * Relocates some non trivial references (i.e.
238 * table of internal addresses)
406F: A2 01 239 :4 LDX #ADT1-ADB1-1
4071: A0 00 240 LDY #0
4073: 38 241 SEC
4074: BD D9 44 245 ]LOOP LDA ADB1,X
4077: 85 06 246 STA AUXPTR
4079: BD DB 44 247 LDA ADT1,X
407C: 85 07 248 STA AUXPTR+1
407E: B1 06 249 LDA (AUXPTR),Y
4080: ED 5E 40 250 SBC :0+1
4083: 91 06 251 STA (AUXPTR),Y
4085: BD DD 44 252 LDA ADB2,X
4088: 85 06 254 STA AUXPTR
408A: BD DF 44 255 LDA ADT2,X
408D: 85 07 256 STA AUXPTR+1
408F: B1 06 257 LDA (AUXPTR),Y
4091: ED 66 40 258 SBC :1+1
4094: 91 06 259 STA (AUXPTR),Y
4096: CA 260 DEX
4097: 10 DB 261 BPL ]LOOP
262
275 * Now move the code
4099: A9 2D 276 LDA #FNDVAR2
409B: 38 277 SEC
409C: ED 5E 40 280 SBC :0+1
409F: 85 42 282 STA A4L
40A1: A9 99 283 LDA #>FNDVAR2
40A3: ED 66 40 284 SBC :1+1
40A6: 85 43 285 STA A4L+1
286 STID AROMBA;A1L
287 STID FIN-1+AROMBA-FNDVAR2;A2L
40B8: A0 00 288 LDY #0
40BA: 20 2C FE 289 JSR MOVE
302 * For DOS 3.3, we reconstruct DOS buffer below..
40BD: AD 00 9D 303 LDA DBUFP
40C0: AE 01 9D 304 LDX DBUFP+1
40C3: C9 D3 305 CMP #$9CD3
40C5: D0 05 306 BNE *+7
40C7: E0 9C 307 CPX #>$9CD3
40C9: D0 01 308 BNE *+3
40CB: CA 309 DEX
40CC: 38 310 SEC
40CD: E9 D3 311 SBC #FIN-FNDVAR2
40CF: A8 312 TAY
40D0: 8A 313 TXA
40D1: E9 02 314 SBC #>FIN-FNDVAR2
40D3: 8C 00 9D 315 STY DBUFP
40D6: 8D 01 9D 316 STA DBUFP+1
40D9: 20 D4 A7 317 JSR $A7D4
40DC: 4C 05 41 318 JMP SETUPD+AROMBA-FNDVAR2
320
329 * DOS 3.3 origin setting
340 AROMBA ORG $992D
347 FNDVAR2
992D: A5 33 352 PX LDA PROMPT

```

992F:	C9	AA	354		CMP	# " * "
9931:	D0	1C	355		BNE	GF
9933:	AE	59	AA	361	LDX	SSAVE
9936:	A0	05	363		LDY	#5
9938:	E8		364]LOOP	INX	
9939:	BD	00	01	365	LDA	\$0100,X
993C:	59	65	9B	366	EOR	STKL,Y
993F:	D0	0E	367		BNE	GF
9941:	88		368		DEY	
9942:	10	F4	369		BPL]LOOP
9944:	A9	9B	370	E01	LDA	#>NXTITM-4
9946:	9D	00	01	371	STA	\$0100,X
9949:	CA		372		DEX	
994A:	A9	4A	373		LDA	#NXTITM-4
994C:	9D	00	01	374	STA	\$0100,X
994F:	AD	B3	AA	378	GF	LDA EXFLG
9952:	60		379		RTS	
			381			
9953:	A9	20	384	SETUPD	LDA	#\$20
9955:	8D	9E	9E	385	STA	\$9E9E
9958:	A9	2D	386		LDA	#PX
995A:	8D	9F	9E	387	STA	\$9E9F
995D:	A9	99	388		LDA	#>PX
995F:	8D	A0	9E	389	STA	\$9EA0
9962:	60		390		RTS	
			392			
			394		* Substitute for LDA MNEML,Y	
9963:	C0	F8	395	LBS01	CPY	#\$FAB8-\$F9C0
9965:	B9	C7	9A	396	LDA	NMEMNEML-\$FAB8+\$F9C0,Y
9968:	B0	0D	397		BCS	:0
996A:	B9	C0	F9	398	LDA	MNEML,Y
996D:	D0	08	399		BNE	:0
996F:	A9	1C	400		LDA	#\$1C
9971:	C0	10	401		CPY	#\$D0-\$C0
9973:	F0	02	402		BEQ	:0
9975:	A9	AD	403		LDA	#\$AD
9977:	60		404	:0	RTS	
			405		* Substitute for LDA MNEMR,Y	
9978:	C0	F8	406	LBS02	CPY	#\$FAF8-\$FA00
997A:	B9	CE	9A	407	LDA	NMEMNEMR-\$FAF8+\$FA00,Y
997D:	B0	0D	408		BCS	:0
997F:	B9	00	FA	409	LDA	MNEMR,Y
9982:	D0	08	410		BNE	:0
9984:	A9	C4	411		LDA	#\$C4
9986:	C0	10	412		CPY	#\$10-\$00
9988:	F0	02	413		BEQ	:0
998A:	A9	06	414		LDA	#\$06
998C:	60		415	:0	RTS	
			416			
998D:	98		417	NEWOPS	TYA	
998E:	A2	18	418		LDX	#INDX-OPTBL-1
9990:	DD	CD	9B	419]LOOP	CMP OPTBL,X
9993:	F0	04	420		BEQ	:0
9995:	CA		421		DEX	
9996:	10	F8	422		BPL]LOOP
9998:	60		423		RTS	
9999:	BD	E6	9B	424	:0	LDA INDX,X

999C:	A0 00	425		LDY	#0	
999E:	60	426		RTS		
		427				
999F:	A6 3A	428	MINSDS1	LDX	PCL	
99A1:	A4 3B	429		LDY	PCL+1	
99A3:	20 96 FD	430		JSR	PRYX2	Print address Y,A hexa
99A6:	20 48 F9	431		JSR	PRBLNK	Followed by a blank
99A9:	A1 3A	432	MINSDS2	LDA	(PCL,X)	
99AB:	A8	433		TAY		
99AC:	4A	434		LSR		;Even/odd test
99AD:	90 05	435		BCC	:E	
99AF:	6A	436		ROR		
99B0:	B0 0C	437		BCS	MERR	
99B2:	29 87	438		AND	#\$87	Mask bits
99B4:	4A	439	:E	LSR		
99B5:	AA	440		TAX		
99B6:	BD 6B 9B	441		LDA	NFMT1,X	
99B9:	20 79 F8	442		JSR	SCRN2	
99BC:	D0 04	443		BNE	GETFMT	
99BE:	A0 FC	444	MERR	LDY	#\$FC	
99C0:	A9 00	445		LDA	#0	
99C2:	AA	446	GETFMT	TAX		
99C3:	BD AF 9B	447		LDA	NFMT2,X	
99C6:	85 2E	448		STA	FORMAT	
99C8:	29 03	449		AND	#3	
99CA:	85 2F	450		STA	LENGTH	
99CC:	20 8D 99	451		JSR	NEWOPS	get index for new opcodes
99CF:	F0 05	452		BEQ	GOTONE	
99D1:	A5 2E	453		LDA	FORMAT	
99D3:	4C AF F8	454		JMP	\$F8AF	
99D6:	60	455	GOTONE	RTS		
		456				
99D7:	20 9F 99	457	MINSTDSP	JSR	MINSDS1	Get format and length bytes
99DA:	48	458		PHA		
99DB:	B1 3A	459]LOOP	LDA	(PCL),Y	
99DD:	20 DA FD	460		JSR	PRBYTE	
99E0:	A2 01	461		LDX	#1	
99E2:	20 4A F9	462]JLOOP	JSR	PRBL2	
99E5:	C4 2F	463		CPY	LENGTH	
99E7:	C8	464		INY		
99E8:	90 F1	465		BCC]LOOP	
99EA:	A2 03	466		LDX	#3	
99EC:	C0 04	467		CPY	#4	
99EE:	90 F2	468		BCC]JLOOP	
99F0:	68	469		PLA		
99F1:	A8	470		TAY		
99F2:	20 63 99	471		JSR	LBS01	
99F5:	85 2C	472		STA	LMNEM	
99F7:	20 78 99	473		JSR	LBS02	
99FA:	85 2D	474		STA	RMNEM	
99FC:	4C F5 F8	475		JMP	PRMN1	
		476				
99FF:	20 75 FE	477	MYLIST	JSR	A1PC	Move A1 (2 bytes) to PC if
9A02:	A9 14	478		LDA	#\$14	specified and disassemble 20
9A04:	48	479]LOOP	PHA		
9A05:	20 D7 99	480		JSR	MINSTDSP	
9A08:	20 53 F9	481		JSR	PCADJ	Adjust PC after each instr.

9A0B:	85	3A	482		STA	PCL	
9A0D:	84	3B	483		STY	PCL+1	
9A0F:	68		484		PLA		
9A10:	38		485		SEC		
9A11:	E9	01	486		SBC	#1	Next of 20 set of instructions
9A13:	D0	EF	487		BNE]LOOP	
9A15:	60		488]RET	RTS		
			493				
9A16:	85	33	494	LBS20	STA	PROMPT	
9A18:	4C	67	495		JMP	GETLNZ	
			496				
			497	MTOSUB	DO	ROMMODEL-R2CO	
9A1B:	C0	0F	498		CPY	#\$0F	Index for L command
9A1D:	F0	07	499		BEQ	:0	
9A1F:	C0	03	501		CPY	#IDXMINI	!" idx on all a2 platforms
9A21:	F0	09	502		BEQ	:1	
9A23:	4C	BE	503		JMP	TOSUB	
9A26:	20	C7	505	:0	JSR	ZMODE	
9A29:	4C	FF	506		JMP	MYLIST	
9A2C:	20	C7	508	:1	JSR	ZMODE	
9A2F:	4C	A6	509		JMP	MYMINI	
			510				
9A32:	E9	81	511	REL	SBC	#\$81	Is FMT compatible
9A34:	4A		512		LSR		;with relative mode?
9A35:	D0	14	513		BNE	ERR3	No
9A37:	A4	3F	514		LDY	A2L+1	
9A39:	A6	3E	515		LDX	A2L	Double decrement
9A3B:	D0	01	516		BNE	REL2	
9A3D:	88		517		DEY		
9A3E:	CA		518	REL2	DEX		
9A3F:	8A		519		TXA		
9A40:	18		520		CLC		
9A41:	E5	3A	521		SBC	PCL	;Form Addr - PC - 2
9A43:	85	3E	522		STA	A2L	
9A45:	10	01	523		BPL	*+3	
9A47:	C8		524		INY		
9A48:	98		525		TYA		
9A49:	E5	3B	526		SBC	PCL+1	
9A4B:	D0	4A	527	ERR3	BNE	ERR	
			528				
9A4D:	A4	2F	529	FINDOP	LDY	LENGTH	
9A4F:	B9	3D	530]LOOP	LDA	A1H,Y	;Move instruction to PC
9A52:	91	3A	531		STA	(PCL),Y	
9A54:	88		532		DEY		
9A55:	10	F8	533		BPL]LOOP	
9A57:	20	48	536		JSR	PRBLNK	For enhanced //e
9A5A:	20	1A	538		JSR	CURSUP	
9A5D:	20	1A	539		JSR	CURSUP	
9A60:	20	D7	540		JSR	MINSTDSP	
9A63:	20	53	541		JSR	PCADJ	Update PC
9A66:	84	3B	542		STY	PCL+1	
9A68:	85	3A	543		STA	PCL	
9A6A:	4C	A6	544		JMP	NEXTLINE	
9A6D:	A5	3D	545	TRYNEXT	LDA	A1H	Get trial OpCode
9A6F:	20	AB	546		JSR	MINSDS2+2	
9A72:	A8		547		TAY		
9A73:	20	78	549		JSR	LBS02	

9A76:	C5	42	553	CMP	A4L	
9A78:	D0	13	554	BNE	NEXTOP	
9A7A:	20	63	99 556	JSR	LBS01	
9A7D:	C5	43	560	CMP	A4H	
9A7F:	D0	0C	561	BNE	NEXTOP	
9A81:	A5	44	562	LDA	FMT	
9A83:	A4	2E	563	LDY	FORMAT	
9A85:	C0	9D	564	CPY	#\$9D	Trial format relative?
9A87:	F0	A9	565	BEQ	REL	Yes
9A89:	C5	2E	566	CMP	FORMAT	Same format
9A8B:	F0	C0	567	BEQ	FINDOP	Yes.
9A8D:	C6	3D	568	NEXTOP DEC	A1H	;Try next trial OpCode
9A8F:	D0	DC	569	BNE	TRYNEXT	
9A91:	E6	44	570	INC	FMT	;No more, try with LEN=2
9A93:	C6	35	571	DEC	L	Was L=2 already
9A95:	F0	D6	572	BEQ	TRYNEXT	No
			573			
			574	* Mini assembler unrecognized instruction		
9A97:	A4	34	575	ERR LDY	YSAV	
9A99:	98		576	ERR2 TYA		
9A9A:	AA		577	TAX		
9A9B:	20	4A	F9 578	JSR	PRBL2	
9A9E:	A9	DE	579	LDA	#" ^"	
9AA0:	20	ED	FD 580	JSR	COUT	
9AA3:	20	3A	FF 581	RESETZ JSR	BELL	
			582			
			583	MYMINI		
9AA6:	A9	A1	584	NEXTLINE LDA	#"! "	
9AA8:	20	16	9A 585	JSR	LBS20	
9AAB:	20	C7	FF 586	JSR	ZMODE	
9AAE:	AD	00	02 587	LDA	IN	
9AB1:	C9	A0	588	CMP	#" "	
9AB3:	F0	12	589	BEQ	SPACE	
9AB5:	C9	8D	590	CMP	#\$8D	
9AB7:	D0	01	591	BNE	*+3	
9AB9:	60		592]RET RTS		
9ABA:	20	A7	FF 593	JSR	GETNUM	;Get a number
9ABD:	C9	93	594	CMP	#\$93	":" terminator
9ABF:	D0	D8	595	ERR4 BNE	ERR2	No: error
9AC1:	8A		596	TXA		;No address preceding ":"
9AC2:	F0	D5	597	BEQ	ERR2	
9AC4:	20	75	FE 598	JSR	A1PC	Move Address to PCL,PCH
9AC7:	A9	03	599	SPACE LDA	#3	Count of characters in mnem.
9AC9:	85	3D	600	STA	A1H	
9ACB:	20	13	FF 601	NXTMN JSR	GETNSP	
9ACE:	0A		602	NXTM ASL		
9ACF:	E9	BE	603	SBC	#\$BE	Substract offset
9AD1:	C9	C2	604	CMP	#\$C2	Legal character?
9AD3:	90	C4	605	BCC	ERR2	No
9AD5:	0A		606	ASL		;Compress left justify
9AD6:	0A		607	ASL		
9AD7:	A2	04	608	LDX	#4	
9AD9:	0A		609]LOOP ASL		;Do 5 triple words shifts
9ADA:	26	42	610	ROL	A4L	
9ADC:	26	43	611	ROL	A4H	
9ADE:	CA		612	DEX		
9ADF:	10	F8	613	BPL]LOOP	

9AE1:	C6 3D	614	DEC	A1H	Done with 3 chars?
9AE3:	F0 F4	615	BEQ]LOOP	Yes, but do 1 more shift
9AE5:	10 E4	616	BPL	NXTMN	No
9AE7:	A2 05	617	FORM1	LDX #5	5 chars in addr mode
9AE9:	20 13 FF	618	FORM2	JSR GETNSP	
9AEC:	84 34	619	STY	YSAV	
9AEE:	DD B4 F9	620	CMP	CHAR1,X	1st char match pattern?
9AF1:	D0 13	621	BNE	FORM3	No
9AF3:	20 13 FF	622	JSR	GETNSP	Yes: get 2nd character
9AF6:	DD BA F9	623	CMP	CHAR2,X	Match?
9AF9:	F0 0D	624	BEQ	FORM5	Yes
9AFB:	BD BA F9	625	LDA	CHAR2,X	No: is second half zero
9AFE:	F0 07	626	BEQ	FORM4	Yes
9B00:	C9 A4	627	CMP	#\$A4	
9B02:	F0 03	628	BEQ	FORM4	
9B04:	A4 34	629	LDY	YSAV	
9B06:	18	630	FORM3	CLC	;Clear bit means no match
9B07:	88	631	FORM4	DEY	;Back 1 character
9B08:	26 44	632	FORM5	ROL FMT	Form format byte
9B0A:	E0 03	633	CPX	#3	Time to check for Address
9B0C:	D0 0D	634	BNE	FORM7	No
9B0E:	20 A7 FF	635	JSR	GETNUM	Yes
9B11:	A5 3F	636	LDA	A2L+1	
9B13:	F0 01	637	BEQ	FORM6	High order byte zero
9B15:	E8	638	INX		;No, incr for 2-byte
9B16:	86 35	639	FORM6	STX L	
9B18:	A2 03	640	LDX	#3	Reload format index
9B1A:	88	641	DEY		;Back 1 character
9B1B:	86 3D	642	FORM7	STX A1H	Save index
9B1D:	CA	643	DEX	Done	with format check?
9B1E:	10 C9	644	BPL	FORM2	
		645	* Important note: at this time A1H value is \$FF		
		646	* Will serve as Trial OpCode for next assembly step.		
9B20:	A5 44	647	LDA	FMT	Put length into the 2
9B22:	0A	648	ASL		;lsb of FMT
9B23:	0A	649	ASL		
9B24:	05 35	650	ORA	L	
9B26:	C9 20	651	CMP	#\$20	Add "\$" if non zero length
9B28:	B0 06	652	BCS	FORM8	and don't already have it
9B2A:	A6 35	653	LDX	L	
9B2C:	F0 02	654	BEQ	FORM8	
9B2E:	09 80	655	ORA	#\$80	
9B30:	85 44	656	FORM8	STA FMT	
9B32:	84 34	657	STY	YSAV	
9B34:	B9 00 02	658	LDA	IN,Y	
9B37:	C9 BB	659	CMP	#";"	Start of comment?
9B39:	F0 04	660	BEQ	FORM9	
9B3B:	C9 8D	661	CMP	#\$8D	
9B3D:	D0 80	662	BNE	ERR4	
9B3F:	4C 6D 9A	663	FORM9	JMP TRYNEXT	
		664			
		681	GETNSP	EQU NNBL	
		683			
9B42:	D8	684	MON	CLD	
9B43:	20 3A FF	685	JSR	BELL	
9B46:	A9 AA	686	MONZ	LDA #" *"	
9B48:	20 16 9A	687	JSR	LBS20	

9B4B:	20	C7	FF	688		JSR	ZMODE
9B4E:	20	A7	FF	689	NXTITM	JSR	GETNUM
9B51:	84	34		690		STY	YSAV
9B53:	A0	17		691		LDY	#NMCMD5
9B55:	88			696]LOOP	DEY	
9B56:	30	EA		697		BMI	MON
9B58:	D9	CC	FF	698		CMP	CHRTBL,Y
9B5B:	D0	F8		699		BNE]LOOP
9B5D:	20	1B	9A	700	:0	JSR	MTOSUB
9B60:	A4	34		701		LDY	YSAV
9B62:	4C	4E	9B	702		JMP	NXTITM
				703			
				704	FCODE	EQU	*
9B65:	FF	6F	FD	705	STKL	HEX	FF6FFD77
9B69:	FD	37		707		HEX	FD37
9B6B:	0F	22	FF	712	NFMT1	HEX	0F22FF33CB62FF73
9B73:	03	22	FF	713		HEX	0322FF33CB66FF77
9B7B:	0F	20	FF	714		HEX	0F20FF33CB60FF70
9B83:	0F	22	FF	715		HEX	0F22FF39CB66FF7D
9B8B:	0B	22	FF	716		HEX	0B22FF33CBA6FF73
9B93:	11	22	FF	717		HEX	1122FF33CBA6FF87
9B9B:	01	22	FF	718		HEX	0122FF33CB60FF70
9BA3:	01	22	FF	719		HEX	0122FF33CB60FF70
9BAB:	24	31	65	720		HEX	24316578
9BAF:	00	21	81	721	NFMT2	HEX	00218182594D9192
9BB7:	86	4A	85	722		HEX	864A859D495A
9BBD:	00			723		DS	1
9BBE:	00			724		HEX	00
							Byte index \$0F of NFMT2
9BBF:	8A	8B	A5	725	NMEMNEML	HEX	8A8BA5AC00
9BC4:	8A	8B		726		HEX	8A8B
9BC6:	74	74	76	727	NMEMNEMR	HEX	747476C600
9BCB:	72	72		728		HEX	7272
9BCD:	12	14	1A	729	OPTBL	HEX	12141A1C32343A3C
9BD5:	52	5A	64	730		HEX	525A6472747A7C89
9BDD:	92	9C	9E	731		HEX	929C9EB2D2F2FC
9BE4:	DA	FA		732		HEX	DAFA
9BE6:	38	FB	37	733	INDX	HEX	38FB37FB39213621
9BEE:	3A	F8	FA	734		HEX	3AF8FA3BFAF92221
9BF6:	3C	FA	FA	735		HEX	3CFAFA3D3E3FFC
9BFD:	FD	FE		736		HEX	FD FE
9BFF:	00			737		HEX	00
				739	FIN	EQU	*
				740			
				745		ERR	*-\$9C00
				747		ORG	
				748			
43B2:	A0	07		749	RECON	LDY	#8-1
43B4:	AD	B3	FB	750		LDA	\$FBB3
43B7:	4D	C0	FB	751		EOR	\$FBC0
43BA:	4D	BF	FB	752		EOR	\$FBBF
43BD:	D9	15	44	753]LOOP	CMP	MACMAT,Y
43C0:	F0	04		754		BEQ	:1
43C2:	88			755		DEY	
43C3:	10	F8		756		BPL]LOOP
43C5:	C8			757		INY	
43C6:	C0	02		758	:1	CPY	#2
43C8:	D0	23		759		BNE	:2

43CA:	AD	5C	FC	760		LDA	\$FC5C	
43CD:	C9	EB		761		CMP	#\$EB	
43CF:	D0	1C		762		BNE	:2	
43D1:	A0	08		763		LDY	#8	
43D3:	18			764		CLC		
43D4:	FB			765		HEX	FB	65802 XCE
43D5:	08			766		PHP		
43D6:	C2	30		767		HEX	C230	65802 REP \$30
43D8:	20	1F	FE	768		JSR	\$FE1F	
43DB:	8C	E1	44	769		STY	NEWY	
43DE:	28			770		PLP		
43DF:	FB			771		HEX	FB	
43E0:	A0	0C		772		LDY	#12	
43E2:	AD	E2	44	773		LDA	NEWY+1	
43E5:	D0	06		774		BNE	:2	
43E7:	AD	E1	44	775		LDA	NEWY	
43EA:	09	08		776		ORA	#8	
43EC:	A8			777		TAY		
43ED:	B9	1D	44	778	:2	LDA	MCODE,Y	
43F0:	8D	E3	44	779		STA	MACHINE	
43F3:	C0	01		780		CPY	#1	
43F5:	6A			781		ROR		
43F6:	8D	E4	44	782		STA	MUL	
43F9:	60			783]RET	RTS		
				784				
43FA:	BD	2B	44	785	PRMSG	LDA	MESBASE,X	
43FD:	F0	FA		786		BEQ]RET	
43FF:	20	07	44	787		JSR	LBS10+2	
4402:	E8			788		INX		
4403:	D0	F5		789		BNE	PRMSG	Always
				790				
4405:	09	80		791	LBS10	ORA	#\$80	
4407:	2C	E4	44	792		BIT	MUL	
440A:	30	06		793		BMI	LBS11	
440C:	C9	E0		794		CMP	#\$E0	
440E:	B0	02		795		BCS	LBS11	
4410:	29	DF		796		AND	#\$DF	
4412:	6C	53	AA	797	LBS11	JMP	(TRUECSW)	
				798				
4415:	EA	2D	E6	799	MACMAT	HEX	EA2DE6E7F9060502	
441D:	00	40	41	800	MCODE	HEX	0040414280818283C0C1C2C3C4C5	
				801	MESBASE			
442B:	8D			802	M0	HEX	8D	
442C:	D4	E8	E5	803		ASC	"The target and host ROM models do not",8D	
4452:	ED	E1	F4	804		ASC	"match. There is no need to install",8D	
4475:	AA	F4	E8	805		ASC	"*this* patch on *this* comuter!",8D00	
4496:	8D			806	M1	HEX	8D	
4497:	C9	F4	A0	807		ASC	"It seems that this patch or another",8D	
44BB:	F3	E9	ED	808		ASC	"similar is already installed",8D00	
				814		ERR	*-MESBASE/256	
				815				
44D9:	FD			821	ADB1	DFB	E01+7+AROMBA-FNDVAR2	
44DA:	0B			824		DFB	SETUPD+6+AROMBA-FNDVAR2	
44DB:	40			826	ADT1	DFB	>E01+7+AROMBA-FNDVAR2	
44DC:	41			829		DFB	>SETUPD+6+AROMBA-FNDVAR2	
44DD:	F7			831	ADB2	DFB	E01+1+AROMBA-FNDVAR2	
44DE:	10			834		DFB	SETUPD+11+AROMBA-FNDVAR2	

```

44DF: 40      836  ADT2      DFB    >E01+1+AROMBA-FNDVAR2
44E0: 41      839      DFB    >SETUPD+11+AROMBA-FNDVAR2
      841
44E1: 00 00    842  NEWY      DS      2
44E3: 00      843  MACHINE  DS      1
44E4: 00      844  MUL        DS      1          b7 set iif no upper. case
      845
      874      SAV      FLATLANDP3EED33

```

Object saved as FLATLANDP3EED33,A\$4000,L\$04E5

--End assembly, 1253 bytes, Errors: 0

Symbol table - alphabetical order:

A1H	=\$3D	A1L	=\$3C	A1PC	=\$FE75	A2L	=\$3E
? A3L	=\$40	A4H	=\$43	A4L	=\$42	ADB1	=\$44D9
ADB2	=\$44DD	ADT1	=\$44DB	ADT2	=\$44DF	AROMBA	=\$40DF
AUXPTR	=\$06	BELL	=\$FF3A	CHAR1	=\$F9B4	CHAR2	=\$F9BA
CHRTBL	=\$FFCC	COUT	=\$FDED	CURSUP	=\$FC1A	DBUFP	=\$9D00
E01	=\$9944	ERR	=\$9A97	ERR2	=\$9A99	ERR3	=\$9A4B
ERR4	=\$9ABF	EXFLG	=\$AAB3	FCODE	=\$9B65	FIN	=\$9C00
FINDOP	=\$9A4D	FMT	=\$44	? FMT2	=\$F9A6	FNDVAR2	=\$992D
? FORM1	=\$9AE7	FORM2	=\$9AE9	FORM3	=\$9B06	FORM4	=\$9B07
FORM5	=\$9B08	FORM6	=\$9B16	FORM7	=\$9B1B	FORM8	=\$9B30
FORM9	=\$9B3F	FORMAT	=\$2E	GETFMT	=\$99C2	GETLNZ	=\$FD67
GETNSP	=\$FF13	GETNUM	=\$FFA7	GF	=\$994F	GOTONE	=\$99D6
IDXMINI	=\$03	IN	=\$0200	INDX	=\$9BE6	INSDS2	=\$F88C
L	=\$35	LBS01	=\$9963	LBS02	=\$9978	LBS10	=\$4405
LBS11	=\$4412	LBS20	=\$9A16	LENGTH	=\$2F	LMNEM	=\$2C
M0	=\$442B	M1	=\$4496	MACHINE	=\$44E3	MACMAT	=\$4415
MCODE	=\$441D	MERR	=\$99BE	MESBASE	=\$442B	MINSDS1	=\$999F
MINSDS2	=\$99A9	MINSTDSP	=\$99D7	MNEML	=\$F9C0	MNEMR	=\$FA00
? MODE	=\$31	MON	=\$9B42	? MONZ	=\$9B46	MOVE	=\$FE2C
MD?MSGBE	=\$8000	MD?MSGP	=\$8000	MTOSUB	=\$9A1B	MUL	=\$44E4
MYLIST	=\$99FF	MYMINI	=\$9AA6	NEWOPS	=\$998D	NEWY	=\$44E1
NEXTLINE	=\$9AA6	NEXTOP	=\$9A8D	NFMT1	=\$9B6B	NFMT2	=\$9BAF
NMCMD5	=\$17	NMEMNEML	=\$9BBF	NMEMNEMR	=\$9BC6	NNBL	=\$FF13
NXTITM	=\$9B4E	? NXTM	=\$9ACE	NXTMN	=\$9ACB	OPTBI	=\$00
OPTBL	=\$9BCD	PCADJ	=\$F953	PCL	=\$3A	PRBL2	=\$F94A
PRBLNK	=\$F948	PRBYTE	=\$FDDA	PRMN1	=\$F8F5	PRMSG	=\$43FA
PROMPT	=\$33	PRYX2	=\$FD96	PX	=\$992D	R2CO	=\$04
R2E	=\$01	R2EE	=\$02	R2P	=\$00	RECON	=\$43B2
REL	=\$9A32	REL2	=\$9A3E	? RESETZ	=\$9AA3	RMNEM	=\$2D
ROMMODEL	=\$02	SCRN2	=\$F879	SETUPD	=\$9953	SPACE	=\$9AC7
SSAVE	=\$AA59	MD STID	=\$8000	STKL	=\$9B65	? SUITE	=\$4000
TOSUB	=\$FFBE	TRUECSW	=\$AA53	TRYNEXT	=\$9A6D	YSAV	=\$34
ZMODE	=\$FFC7	V JLOOP	=\$99E2	V JLOOP	=\$43BD	V JRET	=\$43F9

Symbol table - numerical order:

OPTBI	=\$00	R2P	=\$00	R2E	=\$01	R2EE	=\$02
-------	-------	-----	-------	-----	-------	------	-------

ROMMODEL=\$02	IDXMINI =\$03	R2CO =\$04	AUXPTR =\$06
NMCMDS =\$17	LMNEM =\$2C	RMNEM =\$2D	FORMAT =\$2E
LENGTH =\$2F	? MODE =\$31	PROMPT =\$33	YSAV =\$34
L =\$35	PCL =\$3A	A1L =\$3C	A1H =\$3D
A2L =\$3E	? A3L =\$40	A4L =\$42	A4H =\$43
FMT =\$44	IN =\$0200	MD?MSGBE =\$8000	MD?MSGP =\$8000
MD STID =\$8000	? SUITE =\$4000	AROMBA =\$40DF	RECON =\$43B2
V JLOOP =\$43BD	V JRET =\$43F9	PRMSG =\$43FA	LBS10 =\$4405
LBS11 =\$4412	MACMAT =\$4415	MCODE =\$441D	MESBASE =\$442B
M0 =\$442B	M1 =\$4496	ADB1 =\$44D9	ADT1 =\$44DB
ADB2 =\$44DD	ADT2 =\$44DF	NEWY =\$44E1	MACHINE =\$44E3
MUL =\$44E4	FNDVAR2 =\$992D	PX =\$992D	E01 =\$9944
GF =\$994F	SETUPD =\$9953	LBS01 =\$9963	LBS02 =\$9978
NEWOPS =\$998D	MINSDDS1 =\$999F	MINSDDS2 =\$99A9	MERR =\$99BE
GETFMT =\$99C2	GOTONE =\$99D6	MINSTDSP=\$99D7	V JJLOOP =\$99E2
MYLIST =\$99FF	LBS20 =\$9A16	MTOSUB =\$9A1B	REL =\$9A32
REL2 =\$9A3E	ERR3 =\$9A4B	FINDOP =\$9A4D	TRYNEXT =\$9A6D
NEXTOP =\$9A8D	ERR =\$9A97	ERR2 =\$9A99	? RESETZ =\$9AA3
MYMINI =\$9AA6	NEXTLINE=\$9AA6	ERR4 =\$9ABF	SPACE =\$9AC7
NXTMN =\$9ACB	? NXTM =\$9ACE	? FORM1 =\$9AE7	FORM2 =\$9AE9
FORM3 =\$9B06	FORM4 =\$9B07	FORM5 =\$9B08	FORM6 =\$9B16
FORM7 =\$9B1B	FORM8 =\$9B30	FORM9 =\$9B3F	MON =\$9B42
? MONZ =\$9B46	NXTITM =\$9B4E	FCODE =\$9B65	STKL =\$9B65
NFMT1 =\$9B6B	NFMT2 =\$9BAF	NMEMNEML=\$9BBF	NMEMNEMR=\$9BC6
OPTBL =\$9BCD	INDX =\$9BE6	FIN =\$9C00	DBUFP =\$9D00
TRUECSW =\$AA53	SSAVE =\$AA59	EXFLG =\$AAB3	SCRN2 =\$F879
INSDS2 =\$F88C	PRMN1 =\$F8F5	PRBLNK =\$F948	PRBL2 =\$F94A
PCADJ =\$F953	? FMT2 =\$F9A6	CHAR1 =\$F9B4	CHAR2 =\$F9BA
MNEML =\$F9C0	MNEMR =\$FA00	CURSUP =\$FC1A	GETLNZ =\$FD67
PRYX2 =\$FD96	PRBYTE =\$FDDA	COUT =\$FDED	MOVE =\$FE2C
A1PC =\$FE75	NNBL =\$FF13	GETNSP =\$FF13	BELL =\$FF3A
GETNUM =\$FFA7	TOSUB =\$FFBE	ZMODE =\$FFC7	CHRTBL =\$FFCC

