

⟨> Code - Revisions 2 ☆ Stars 29 ♀ Forks 10



## Revisions

Split Unified

**beneater revised** this gist on May 31, 2023.

± 1 changed file with 38 additions and 35 deletions.

~	73	wozmon.	s 🖵							
		@@ -12,76 +12,	76 @@ MO	DE = \$2B		; \$00:	=XAM, \$7F=STOR, \$A	AE=BLOCK	XAM	
12						12				
13		IN = \$0200			; Input buffer	13	IN = \$0200			; Input buffer
14 15	_	KBD = \$D010			; PIA.A keyboard	14 15	+ ACIA_DATA =	\$5000		
		input			,					
16	-	KBDCR = \$D011			; PIA.A keyboard	16	+ ACIA_STATUS =	\$ <u>5001</u>		
17		control regist DSP = \$D012	er		; PIA.B display	17	+ ACIA_CMD =	\$5002		
17	_	output registe	r		, TIAID display	1,	T ACIA_CID	\$ <del>300</del> 2		
18	_	DSPCR = \$D013			; PIA.B display	18	+ ACIA_CTRL =	\$5003		
10		control regist	er			10				
19 20		RESET:				19 20	RESET:			
21	_		CLD		; Clear decimal	21	+	LDA	#\$1F	; 8-N-1, 19200
		arithmetic mod					baud.			
22	-		CLI LDY	#\$7F	; Mask for DSP	22	+ +	STA LDA	ACIA_CTRL #\$0B	; No parity, no
25	_	data direction		πφ/ι	, Hask 101 bsi	23	echo, no inter		#400	, No parity, no
24	-		STY	DSP	; Set it up.	24	+	STA	ACIA_CMD	
25	-		LDA	#\$A7	; KBD and DSP	25	+	LDA	#\$1B	; Begin with
26	_	control regist	STA	KBDCR	; Enable		escape.			
		interrupts, se			,					
27	-	-	STA	DSPCR	; positive edge					
28		sense/output m	ode.			26				
29		NOTCR:				27	NOTCR:			
30	-		CMP	#\$DF	; Backspace key?	28	+	CMP	#\$ <u>08</u>	; Backspace key?
31			BEQ	BACKSPACE	; Yes.	29 <b>30</b>		BEQ	BACKSPACE	; Yes.
32 33	-		CMP BEQ	#\$ <mark>9B</mark> ESCAPE	; ESC? ; Yes.	31	+	CMP BEQ	#\$ <u>1B</u> ESCAPE	; ESC? ; Yes.
34			INY		; Advance text	32		INY		; Advance text
2.5		index.	DDI	NEVECHAR		2.2	index.	200	NEVEGUAR	
35		longer than 12	BPL 7.	NEXTCHAR	; Auto ESC if line	33	longer than 12	BPL	NEXTCHAR	; Auto ESC if line
36		tonger than 12				34	tonger than 12	.,.		
37		ESCAPE:				35	ESCAPE:			
<b>38</b> 39	-		LDA JSR	#\$ <u>DC</u> ECH0	; "\". ; Output it.	36 37	+	LDA JSR	#\$ <u>5C</u> ECH0	; "\". ; Output it.
40			JON	LCHO	, output it.	38		JSK	LCIIO	, output it.
41		GETLINE:				39	GETLINE:			
42	-		LDA	#\$ <mark>8D</mark>	; Send CR	40	+	LDA	#\$ <u>0D</u>	; Send CR
43 44			JSR	ECH0		41 42		JSR	ECH0	
45			LDY	#\$01	; Initialize text	43		LDY	#\$01	; Initialize text
		index.					index.			
46		BACKSPACE: index.	DEY		; Back up text	44	BACKSPACE: index.	DEY		; Back up text
47		Index.	BMI	GETLINE	; Beyond start of	45	Ilidex:	BMI	GETLINE	; Beyond start of
		line, reinitia	lize.				line, reinitia	lize.		
48 49		NEXTCHAR:				46 47	NEXTCHAR:			
50	_	NEXTCHAR:	LDA	KBDCR	; Key ready?	48	+	LDA	ACIA_STATUS	; Check status.
51	_		BPL	NEXTCHAR	; Loop until	49	+	AND	#\$ <b>0</b> 8	; Key ready?
		ready.		1/00				550	NEVEGUAR	
52	_	B7 should be '	LDA 1'.	KBD	; Load character.	50	+ ready.	BEQ	NEXTCHAR	; Loop until
						51	+	LDA	ACIA_DATA	; Load character.
							B7 will be '0'			
53		buffer.	STA	IN,Y	; Add to text	52	buffer.	STA	IN,Y	; Add to text
54		buller:	JSR	ECH0	; Display	53	burrer:	JSR	ECH0	; Display
		character.					character.			
<b>55</b> 56	-		CMP	#\$ <u>8D</u> NOTCR	; CR?	<b>54</b> 55	+	CMP BNE	#\$ <u>0D</u>	; CR?
56			BNE	NUICK	; No.	56		DIVE	NOTCR	; No.
58			LDY	#\$FF	; Reset text	57		LDY	#\$FF	; Reset text
F.0		index.	104	##00	L Eam VAM made	FO	index.	104	##00	L Eon VAM and
59 60			LDA TAX	#\$00	; For XAM mode. ; X=0.	58 59		LDA TAX	#\$00	; For XAM mode. ; X=0.
33					, 🕶	60	+ SETBLOCK:			, , .
						61	+	ASL		
61 62		SETSTOR:	ASL		; Leaves \$7B if	62 63	SETSTOR:	ASL		; Leaves \$7B if
02		setting STOR m			, Leaves \$75 II	03	setting STOR m			, Leaves \$75 II
63	_	SETMODE:				64	+	STA	MODE	; \$00 = XAM, \$74 =
64			STA	MODE	; \$00 = XAM, \$7B =		STOR, \$B8 = BL	OK XAM.		
04	_	STOR, \$AE = BL		IIUUL	, \$00 - AMI, \$/D =					
65		BLSKIP:				65	BLSKIP:			
66		to to	INY		; Advance text	66		INY		; Advance text
67		<pre>index. NEXTITEM:</pre>				67	index. NEXTITEM:			
68		ACATTICIT.	LDA	IN,Y	; Get character.	68	NEXTTEN.	LDA	IN,Y	; Get character.
69	_		CMP	#\$ <mark>8D</mark>	; CR?	69	+	CMP	#\$ <u>0D</u>	; CR?
70		line	BEQ	GETLINE	; Yes, done this	70	lino	BEQ	GETLINE	; Yes, done this
71	_	line.	CMP	#\$AE	; "."?	71	line. +	CMP	#\$2E	; "."?
72			BCC	BLSKIP	; Skip delimiter.	72		BCC	BLSKIP	; Skip delimiter.
73	-	mode	BEQ	SETMODE	; Set BLOCK XAM	73	+ mode	BEQ	<u>SETBLOCK</u>	; Set BLOCK XAM
74	_	mode.	CMP	#\$BA	; ":"?	74	mode.	CMP	#\$3A	; ":"?
				· <del>-</del>	-	1	ı		· <u> </u>	-

					Revision	ns · wozmon.s			
75		BEQ	SETSTOR	; Yes, set STOR	75		BEQ	SETSTOR	; Yes, set STOR
	mode.					mode.			
76	_	CMP	#\$D2	; "R"?	76	+	CMP	#\$52	; "R"?
77		BEQ	RUN	; Yes, run user	77		BEQ	RUN	; Yes, run user
	program.					program.			
78		STX	L	; \$00 -> L.	78		STX	L	; \$00 -> L.
79		STX	Н	; and H.	79		STX	Н	; and H.
80		STY	YSAV	; Save Y for	80		STY	YSAV	; Save Y for
	comparison					comparison			
81					81				
82	NEXTHEX:				82	NEXTHEX:			
83		LDA	IN,Y	; Get character	83		LDA	IN,Y	; Get character
	for hex test.					for hex test.			
84	_	EOR	#\$B0	; Map digits to	84	+	EOR	#\$30	; Map digits to
	\$0-9 <sub>•</sub>		_			\$0-9.		_	
85		CMP	#\$0A	; Digit?	85		CMP	#\$0A	; Digit?
86		BCC	DIG	; Yes.	86		BCC	DIG	; Yes.
87		ADC	#\$88	; Map letter "A"-	87		ADC	#\$88	; Map letter "A"
	"F" to \$FA-FF			,		"F" to \$FA-FF.			,
			CETADD.	LDA L 1 V					
• •	@@ -132,17 +1	.32,17 @@	SETADK:	LDA L-1,X		py hex data to			
L32					132				
L33	NXTPRNT:				133	NXTPRNT:			
134		BNE	PRDATA	; NE means no	134		BNE	PRDATA	; NE means no
	address to pr					address to pri	nt.		
L35	-	LDA	#\$ <mark>8D</mark>	; CR.		+	LDA	#\$ <u>0D</u>	; CR.
L36		JSR	ECH0	; Output it.	136		JSR	ECH0	; Output it.
L37		LDA	XAMH	; 'Examine index'	137		LDA	XAMH	; 'Examine index
	high-order by	te.				high-order byte	2.		
138		JSR	PRBYTE	; Output it in hex	138		JSR	PRBYTE	; Output it in h
	format.					format.			
139		LDA	XAML	; Low-order	139		LDA	XAML	; Low-order
	'examine inde	ex' byte.				'examine index	byte.		
140		JSR	PRBYTE	; Output it in hex	140		JSR	PRBYTE	; Output it in he
	format.					format.			
141	_	LDA	#\$BA	; ":".	141	+	LDA	#\$3A	; ":".
142		JSR	ECH0	; Output it.	142		JSR	ECH0	; Output it.
143					143				
144	PRDATA:				144	PRDATA:			
145	_	LDA	#\$A0	; Blank.	145	+	LDA	#\$20	; Blank.
146		JSR	ECH0	; Output it.	146		JSR	ECH0	; Output it.
147		LDA	(XAML,X)	; Get data byte at	147		LDA	(XAML,X)	; Get data byte a
,	'examine inde		(70 ) //	, oct data byte at		'examine index		(70.0.12)717	, 551 4414 5,15
148	CXUMILITO INGC	JSR	PRBYTE	; Output it in hex	148	CXGIII THE THUCK	JSR	PRBYTE	; Output it in he
1-10	format.	3310	TROTTE	, odepat it in nex	140	format.	3311	TROTTE	, odepac ic in in
		72 40 00	DDD)/TE			TOTILIACE			
• • •	@@ -173,15 +1	./3,18 @@	PKBYIE:						
173					173				
174	PRHEX:				174	PRHEX:			
175		AND	#\$0F	; Mask LSD for hex	175		AND	#\$0F	; Mask LSD for he
	print.					print.			
176	-	0RA	#\$ <mark>B0</mark>	; Add "0".	176	+	0RA	#\$ <u>30</u>	; Add "0".
177	_	CMP	#\$BA	; Digit?	177	+	CMP	#\$ <mark>3A</mark>	; Digit?
178		BCC	ECH0	; Yes, output it.	178		BCC	ECH0	; Yes, output it
179		ADC	#\$06	; Add offset for	179		ADC	#\$06	; Add offset for
	letter.					letter.			
					180				
180	ECHO:				181	ECHO:			
	LCIIO:	BIT	DSP	; DA bit (B7)	182		РНА		; Save A.
181				(/			*		
181	-		ECH0	; No, Wait for	183	+	STA	ACIA_DATA	; Output
181 182	- cleared yet?	RMT	EVITO	, NO, WALL TO	103	character.	317	VCTV_DVIV	, output
181 182	- cleared yet?	BMI			184		I DA	##==	. Initializa dal
181 182 183	- cleared yet?			. 0++		+	LDA	#\$FF	; Initialize dela
181 182 183	- cleared yet? - display.	STA	DSP	; Output	104	1			
181 182 183	- cleared yet?	STA		; Output		loop.			_
181 182 183	- cleared yet? - display.	STA		; Output	185	loop. + TXDELAY:	DEC		; Decrement A.
181 182 183	- cleared yet? - display.	STA		; Output		+ TXDELAY: +	DEC BNE	TXDELAY	•
181 182 183	- cleared yet? - display.	STA		; Output	185	+ TXDELAY:		TXDELAY	•
180 181 182 183 184	- cleared yet? - display.	STA		; Output	185	+ TXDELAY: +		TXDELAY	•
181 182 183	- cleared yet? - display.	STA		; Output ; Return.	185 186	+ TXDELAY: + 0.	BNE	TXDELAY	; Until A gets to
181 182 183 184	- cleared yet? - display.	STA ets DA.			185 186 187	+ TXDELAY: + 0.	BNE PLA	TXDELAY	; Until A gets to ; Restore A.

## **beneater created** this gist on May 31, 2023.

```
√ 191 ••••• wozmon.s 
□
     @@ -0,0 +1,191 @@
                                                              1
                                                                    .org $8000
                                                              2
                                                                     .org $ff00
                                                              4
                                                                 + XAML = $24
                                                                                                      ; Last "opened"
                                                                   location Low
                                                              5
                                                                 + XAMH = $25
                                                                                                      ; Last "opened"
                                                                   location High
                                                              6
                                                                 + STL = $26
                                                                                                      ; Store address
                                                                   Low
                                                              7
                                                                 + STH = $27
                                                                                                      ; Store address
                                                                   High
                                                              8
                                                                 + L = $28
                                                                                                      ; Hex value
                                                                   parsing Low
                                                              9
                                                                 + H = $29
                                                                                                      ; Hex value
                                                                   parsing High
                                                                 + YSAV = $2A
                                                             10
                                                                                                      ; Used to see if
                                                                   hex value is given
                                                                 + MODE = $2B
                                                             11
                                                                                                      ; $00=XAM,
                                                                   $7F=STOR, $AE=BLOCK XAM
                                                             12
                                                                 + IN = $0200
                                                             13
                                                                                                      ; Input buffer
                                                             14
                                                             15
                                                                 + KBD = $D010
                                                                                                      ; PIA.A keyboard
                                                                   input
                                                                 + KBDCR = $D011
                                                             16
                                                                                                      ; PIA.A keyboard
                                                                   control register
                                                             17 + DSP = $D012
                                                                                                      ; PIA.B display
                                                                   output register
                                                                 + DSPCR = $D013
                                                             18
                                                                                                      ; PIA.B display
                                                                   control register
```

```
Revisions · wozmon.s
19
20
     + RESET:
21
                       CLD
                                              ; Clear decimal
       arithmetic mode.
22
                       CLI
23
                       LDY
                               #$7F
                                              ; Mask for DSP
       data direction reg.
24
                       STY
                                             ; Set it up.
                               DSP
25
                       LDA
                               #$A7
                                              ; KBD and DSP
       control register mask.
26
                       STA
                               KBDCR
                                              ; Enable
       interrupts, set CA1, CB1 for
27
                       STA
                               DSPCR
                                              ; positive edge
       sense/output mode.
28
29
     + NOTCR:
30
                       CMP
                               #$DF
                                              ; Backspace key?
31
                               BACKSPACE
                       BEQ
                                              ; Yes.
32
                                              ; ESC?
                       CMP
                               #$9B
33
                       BEQ
                               ESCAPE
                                              ; Yes.
34
                       INY
                                              ; Advance text
       index.
                       BPL
                               NEXTCHAR
35
                                              ; Auto ESC if line
       longer than 127.
36
37
     + ESCAPE:
38
                                              ; "\".
                       LDA
                               #$DC
39
                       JSR
                               ECH0
                                              ; Output it.
40
41
     + GETLINE:
42
                       LDA
                               #$8D
                                              ; Send CR
43
                       JSR
                               ECH0
44
45
                       LDY
                               #$01
                                              ; Initialize text
       index.
46
     + BACKSPACE:
                       DEY
                                              ; Back up text
       index.
47
                       BMI
                               GETLINE
                                              ; Beyond start of
       line, reinitialize.
48
49
     + NEXTCHAR:
50
                       LDA
                               KBDCR
                                              ; Key ready?
51
                       BPL
                               NEXTCHAR
                                              ; Loop until
       ready.
52
                       LDA
                               KBD
                                              ; Load character.
       B7 should be '1'.
53
                       STA
                               IN,Y
                                              ; Add to text
       buffer.
54
                       JSR
                               ECH0
                                              ; Display
       character.
55
                       CMP
                                              ; CR?
                               #$8D
56
                       BNE
                               NOTCR
                                             ; No.
57
58
                       LDY
                               #$FF
                                              ; Reset text
       index.
59
                       LDA
                               #$00
                                             ; For XAM mode.
60
                       TAX
                                              ; X=0.
61
     + SETSTOR:
                                             ; Leaves $7B if
       setting STOR mode.
63
     + SETMODE:
64
                       STA
                               MODE
                                             ; $00 = XAM, $7B =
       STOR, $AE = BLOK XAM.
65
     + BLSKIP:
66
                       INY
                                              ; Advance text
       index.
67
     + NEXTITEM:
68
                       LDA
                               IN,Y
                                             ; Get character.
69
                       CMP
                               #$8D
                                              ; CR?
70
                               GETLINE
                       BEQ
                                             ; Yes, done this
       line.
                                             ; "."?
71
                       CMP
                               #$AE
72
                                             ; Skip delimiter.
                       BCC
                               BLSKIP
73
                                             ; Set BLOCK XAM
                       BEQ
                               SETMODE
       mode.
                                             ; ":"?
                       CMP
74
                               #$BA
75
                                             ; Yes, set STOR
                       BEQ
                               SETST0R
                       CMP
                                              ; "R"?
76
                               #$D2
77
                       BEQ
                               RUN
                                             ; Yes, run user
       program.
78
                       STX
                                             ; $00 -> L.
79
                                             ; and H.
                       STX
                       STY
                               YSAV
80
                                             ; Save Y for
       comparison
81
82
     + NEXTHEX:
83
                       LDA
                               IN,Y
                                             ; Get character
       for hex test.
84
                       E0R
                               #$B0
                                             ; Map digits to
       $0-9.
85
                       CMP
                               #$0A
                                             ; Digit?
86
                       BCC
                               DIG
                                             ; Yes.
87
                                             ; Map letter "A"-
                       ADC
                               #$88
       "F" to $FA-FF.
88
                       CMP
                               #$FA
                                             ; Hex letter?
89
                               NOTHEX
                       BCC
                                             ; No, character
       not hex.
90
     + DIG:
91
                       ASL
92
                       ASL
                                             ; Hex digit to MSD
       of A.
                       ASL
93
                       ASL
94
95
96
                       LDX
                               #$04
                                             ; Shift count.
97
    + HEXSHIFT:
```

```
Revisions · wozmon.s
                                               ; Hex digit left,
98
                       ASL
       MSB to carry.
                       R<sub>0</sub>L
99
                                               ; Rotate into LSD.
                       R<sub>0</sub>L
100
                                               ; Rotate into
       MSD's.
101
                       DEX
                                               ; Done 4 shifts?
102
                                HEXSHIFT
                       BNE
                                               ; No, loop.
103
                       INY
                                               ; Advance text
       index.
104
                               NEXTHEX
                       BNE
                                               ; Always taken.
       Check next character for hex.
105
106
     + NOTHEX:
107
                       CPY
                                YSAV
                                               ; Check if L, H
       empty (no hex digits).
108
                       BEQ
                                ESCAPE
                                               ; Yes, generate
       ESC sequence.
109
                       BIT
110
                                MODE
                                               ; Test MODE byte.
111
                       BVC
                                NOTSTOR
                                               ; B6=0 is STOR, 1
       is XAM and BLOCK XAM.
112
                                               ; LSD's of hex
113
                       LDA
       data.
                       STA
                                (STL,X)
114
                                               ; Store current
       'store index'.
115
                       INC
                                STL
                                               ; Increment store
       index.
                                NEXTITEM
116
                       BNE
                                               ; Get next item
       (no carry).
117
                       INC
                                STH
                                               ; Add carry to
       'store index' high order.
                       JMP
118
     + TONEXTITEM:
                               NEXTITEM
                                               ; Get next command
       item.
119
120
     + RUN:
121
                       JMP
                                (XAML)
                                               ; Run at current
       XAM index.
122
123
     + NOTSTOR:
124
                       BMI
                                XAMNEXT
                                               ; B7 = 0 for XAM,
       1 for BLOCK XAM.
125
126
                       LDX
                                #$02
                                               ; Byte count.
127
                       LDA
     + SETADR:
                                L-1,X
                                               ; Copy hex data to
128
                               STL-1,X
                       STA
                                               ; 'store index'.
129
                       STA
                                XAML-1, X
                                               ; And to 'XAM
       index'.
130
                       DEX
                                               ; Next of 2 bytes.
131
                       BNE
                                SETADR
                                               ; Loop unless X =
132
133
     + NXTPRNT:
134
                       BNE
                                PRDATA
                                               ; NE means no
       address to print.
135
                       LDA
                                #$8D
                                               ; CR.
136
                                ECH0
                       JSR
                                               ; Output it.
137
                       LDA
                                XAMH
                                               ; 'Examine index'
       high-order byte.
138
                       JSR
                                PRBYTE
                                               ; Output it in hex
       format.
139
                                XAML
                       LDA
                                               ; Low-order
        'examine index' byte.
                       JSR
140
                                PRBYTE
                                               ; Output it in hex
       format.
                                               ; ":".
                                #$BA
141
                       LDA
142
                       JSR
                                ECH0
                                               ; Output it.
143
144
     + PRDATA:
145
                       LDA
                                #$A0
                                               ; Blank.
146
                       JSR
                                ECH0
                                               ; Output it.
147
                       LDA
                                (XAML,X)
                                               ; Get data byte at
        'examine index'.
                       JSR
                                               ; Output it in hex
148
                                PRBYTE
       format.
     + XAMNEXT:
149
                       STX
                                MODE
                                               ; 0 -> MODE (XAM
       mode).
150
                       LDA
                                XAML
151
                       CMP
                                               ; Compare 'examine
       index' to hex data.
152
                                XAMH
                       LDA
153
                       SBC
154
                       BCS
                                TONEXTITEM
                                               ; Not less, so no
       more data to output.
155
156
                       INC
                                XAML
157
                                MOD8CHK
                       BNE
                                               ; Increment
       'examine index'.
158
                                XAMH
                       INC
159
160
     + MOD8CHK:
161
                       LDA
                                XAML
                                               ; Check low-order
        'examine index' byte
162
                       AND
                                #$07
                                               ; For MOD 8 = 0
163
                       BPL
                                               ; Always taken.
                                NXTPRNT
164
165
     + PRBYTE:
                       PHA
166
                                               ; Save A for LSD.
                       LSR
167
168
                       LSR
169
                       LSR
                                               ; MSD to LSD
       position.
                       LSR
170
171
                       JSR
                                PRHEX
                                               ; Output hex
       digit.
172
                       PLA
                                               ; Restore A.
173
174 + PRHEX:
```

	ns · wozmon.s			
175	+	AND	#\$0F	; Mask LSD for hex
	print.			
176	+	0RA	#\$B0	; Add "0".
177	+	CMP	#\$BA	; Digit?
178	+	BCC	ECH0	; Yes, output it.
179	+	ADC	#\$06	; Add offset for
	letter.			
180	+			
181	+ ECHO:			
182	+	BIT	DSP	; DA bit (B7)
	cleared yet?			
183	+	BMI	ECH0	; No, Wait for
	display.			
184	+	STA	DSP	; Output
	character. Sets	DA.		
185	+	RTS		; Return.
186	+			
187	+ .org \$FFFA			
188	+			
189	+	.word	\$0F00	; NMI vector
190	+	.word	RESET	; RESET vector
191	+	.word	\$0000	; IRQ vector