

BREAKPOINT ROUTINE
FOR 6502s

John Zeigler
8 Seaview Dr., Pittsburg CA 94565
(415) 894-3661

[This routine was distributed at the Homebrew Computer Club meeting, March 17, 1976. It is reprinted with the author's permission.]

This routine is entered via a software breakpoint. It is entered when the processor encounters a 00 op-code. Upon

entering, the program counter is printed, followed by the active flags, accumulator, X index register, & index register, and stack pointer, terminated by a carriage return and line feed. It then waits for the user to type in a new op-code. Upon receiving that op-code, the original 00 code is replaced with the op-code that was input, the stack is returned to pre-interrupt status, and execution of the original program continues from the breakpoint.

To use this routine, it is necessary to load the interrupt vector, FFFE and FFFF, with 64 and 02, respectively, and place the 00 breakpoint op-code in the desired location. The following storage is required: 0000-0007, 0200-02E3, FFFE-FFFF. Note: This routine calls subroutines located in the TIM Monitor.

BUG PROGRAM LISTING

VERSION 1

0200	85 07	NEG	STA 07	;}SAVE MODIFIED P STATUS
0202	A9 4E		LDA #\$4E	;}LOAD A WITH 'N'
0204	20 C6 72		JSR WRT	;}TYPE 'N'
0207	A5 07		LDA 07	;}RESTORE MODIFIED P
0209	4C 7F 02		JMP V	;}RETURN TO PR0G. V
020C	85 07	OVERFL	STA 07	;}SAVE MODIFIED P
020E	A9 56		LDA #\$56	;}LOAD A WITH 'V'
0210	20 C6 72		JSR WRT	;}TYPE 'V'
0213	A5 07		LDA 07	;}RESTORE MODIFIED P
0215	4C 82 02		JMP B	;}RETURN TO PR0G. B
0218	85 07	BRK	STA 07	;}SAVE MODIFIED P
021A	A9 42		LDA #\$42	;}LOAD A WITH 'B'
021C	20 C6 72		JSR WRT	;}TYPE 'B'
021F	A5 07		LDA 07	;}RESTORE MODIFIED P
0221	4C 86 02		JMP D	;}RETURN TO PR0GRAM D
0224	85 07	DEC	STA 07	;}SAVE MODIFIED P
0226	A9 44		LDA #\$44	;}LOAD A WITH 'D'
0228	20 C6 72		JSR WRT	;}TYPE 'D'
022B	A5 07		LDA 07	;}RESTORE MODIFIED P
022D	4C 89 02		JMP I	;}RETURN TO PR0GRAM I
0230	85 07	IR0DIS	STA 07	;}SAVE MODIFIED P
0232	A9 49		LDA #\$49	;}LOAD A WITH 'I'

0234	20	C6	72		JSR WRT	‡TYPE 'I'
0237	A5	07			LDA 07	‡RESTORE MODIFIED P
0239	4C	8C	02		JMP Z	‡RETURN TO PROGRAM Z
023C	85	07		ZERØ	STA 07	‡SAVE MODIFIED P
023E	A9	5A			LDA #55A	‡LOAD A WITH 'Z'
0240	20	C6	72		JSR WRT	‡TYPE 'Z'
0243	A5	07			LDA 07	‡RESTORE MODIFIED P
0245	4C	8F	02		JMP C	‡RETURN TO PROGRAM C
0248	85	07		CARRY	STA 07	‡SAVE MODIFIED P
024A	A9	43			LDA #543	‡LOAD A WITH 'C'
024C	20	C6	72		JSR WRT	‡TYPE 'C'
024F	A5	07			LDA 07	‡RESTORE MODIFIED P
0251	4C	92	02		JMP CØNT	‡RETURN TO PROGRAM CØNT
0254	85	00			STA 00	‡SAVE A IN 00
0256	86	01			STX 01	‡SAVE X IN 01
0258	84	02			STY 02	‡SAVE Y IN 02
025A	68				PLA	‡PULL P ØT A
025B	85	03			STA 03	‡SAVE P IN 03
025D	68				PLA	‡PULL PCL TO A
025E	85	04			STA 04	‡SAVE PCL IN 04
0260	68				PLA	‡PULL PCH TO A
0261	85	05			STA 05	‡SAVE PCH IN 05
0263	BA				TSX	‡MOVE S TO X
0264	86	06			STA 06	‡SAVE S IN 06
0266	D8				CLD	‡NOT DECIMAL MØDE
0267	20	8A	72		JSR CRLF	‡DØ A CRLF
026A	20	CF	02		JSR MØDPC	‡CØRRECT PCL & PCH
026D	A5	05			LDA 05	‡LOAD A WITH PCH
026F	20	B1	72		JSR WRØB	‡TYPE PCH IN HEX
0272	A5	04			LDA 04	‡LOAD A WITH PCL
0274	20	B1	72		JSR WRØB	‡TYPE PCL IN HEX
0277	20	77	73		JSR SPACE	‡SPACE 1 CHARACTER
027A	A5	03			LDA 03	‡LOAD A WITH P
027C	2A				RØL A	‡RØTATE N FLAG TO CARRY
027D	B0	81			BCS NEG	‡BRANCH IF N FLAG SET
027F	2A			V	RØL A	‡RØTATE V FLAG TO CARRY
0280	B0	8A			BCS ØVERFL	‡BRANCH IF V FLAG SET
0282	2A			B	RØL A	‡RØTATE PAST UNUSED BIT
0283	2A				RØL A	‡RØTATE B FLAG TO CARRY
0284	B0	92			BCS BRK	‡BRANCH IF B FLAG SET
0286	2A			D	RØL A	‡RØTATE D FLAG TO CARRY
0287	B0	9B			BCS DEC	‡BRANCH IF D FLAG SET
0289	2A			I	RØL A	‡RØTATE I FLAG TO CARRY

028A	B0 A4		BCS IRQDIS	‡BRANCH IF I FLAG SET
028C	2A	Z	R0L A	‡ROTATE Z FLAG TO CARRY
028D	B0 AD		BCS ZERO	‡BRANCH IF Z FLAG SET
028F	2A	C	R0L A	‡ROTATE C FLAG TO CARRY
0290	B0 B6		BCS CARRY	‡BRANCH IF C FLAG SET
0292	20 77 73	C0NT	JSR SPACE	‡SPACE 1 CHARACTER
0295	A5 00		LDA 00	‡GET A
0297	20 B1 72		JSR WR0B	‡TYPE A
029A	20 77 73		JSR SPACE	‡SPACE 1 CHARACTER
029D	A5 01		LDA 01	‡GET X
029F	20 B1 72		JSR WR0B	‡TYPE X
02A2	20 77 73		JSR SPACE	‡SPACE 1 CHARACTER
02A5	A5 02		LDA 02	‡GET Y
02A7	20 B1 72		JSR WR0B	‡TYPE Y
02AA	20 77 73		JSR SPACE	‡TYPE SPACE
02AD	A5 06		LDA 06	‡GET S
02AF	20 B1 72		JSR WR0B	‡TYPE S
02B2	20 8A 72		JSR CRLF	‡DO A CRLF
02B5	20 B3 73		JSR RDHEX	‡READ VALID 0PC0DE
02B8	A2 00		LDX #S00	‡PREPARE TO LOAD 0PC0DE
02BA	81 04		STA (04,X)	‡STORE CORRECT 0PC0DE
02BC	A6 06		LDX 06	‡GET S
02BE	9A		TXS	‡RESTORE STACK POINTER
02BF	A5 05		LDA 05	‡GET PCH
02C1	48		PHA	‡RESTORE PCH TO STACK
02C2	A5 04		LDA 04	‡GET PCL
02C4	48		PHA	‡RESTORE PCL TO STACK
02C5	A5 03		LDA 03	‡GET P
02C7	48		PHA	‡RESTORE P TO STACK
02C8	A4 02		LDY 02	‡RESTORE Y
02CA	A6 01		LDX 01	‡RESTORE X
02CC	A5 00		LDA 00	‡RESTORE A
02CE	40		RTI	‡RETURN TO PROGRAM
02CF	A5 04	M0DPC	LDA 04	‡LOAD PCL IN A
02D1	F0 07		BEQ ALTER1	‡BRANCH IF PCL = 0
02D3	C6 04	ALT1	DEC 04	‡SET PCL = PCL-1
02D5	F0 08		BEQ ALTER2	‡BRANCH IF PCL = 0
02D7	C6 04	ALT2	DEC 04	‡SET PCL = PCL-2
02D9	60		RTS	‡RETURN FROM SUBROUTINE
02DA	C6 05	ALTER1	DEC 05	‡SET PCH = PCH-1
02DC	4C D3 02		JMP ALT1	‡JUMP TO ALT1
02DF	C6 05	ALTER2	DEC 05	‡SET PCH = PCH-1
02E1	4C D7 02		JMP ALT2	‡JUMP TO ALT2
			END	