STRINGOUT MODS

Dear Dr. Warren, February 2, 1977

In *DDJ* Vol. 1, No. 10 (p. 50), Marcel Meier presented a revision of the Espinosa 6502 program STRINGOUT, cutting its length from 64 to 44 bytes and making it relocatable in RAM. The following revision cuts it to 40 bytes, allows location in ROM (or RAM), is compatible with 6502 systems other than the OSI-400 (e.g., KIK-1), and allows selection of a variety of subroutines to work on the character string.

```
            ORG $O200
YKEEP   EQU $OOE8
AKEEP   EQU $OOE9
LO      EQU $OOEA
HI      EQU $00EB

0200 84 E8   BEGIN STY YKEEP     (save Y index)
0202 85 E9          STA AKEEP    (save accumulator)
0204 68             PLA          (stock pull return LO)
0205 85 EA          STA LO       (store zero-page LO)
0207 68             PLA          (stock pull return HI)
0208 85 EB          STA HI       (store zero-page HI)
020A AO OO          LDY #00      (clear Y index)
020C C8      ALOOP INY           (increment Y)
020D B1 EA          LDA (LO, Y   (get character)
020F FO 06          BEQ EXIT     (if null, exit)
0211 20 00 01       JSR ANYSUB   (selector subroutine)
0214 18             CLC          (clear carry)
0215 90 F5          BCC ALOOPE   (always loops back)
0217 98       EXIT TYA           (start Y offset add)
0218 38             SEC          (offset is Y + 1)
0219 65 EA          ADC LO       (add to return LO)
02lB 85 EA          STA LO       (LO = L) + offset)
021D 90 02          BCC NOCAR    (carry clear, no page cross)
021F E6 EB          INC HI       (increment page number)
0221 A5 E9   NOCAR LDA AKEEP     (restore accumulator)
0223 A4 E8          JMP (LO)     (return to program)
```

The selector subroutine ANYSUB is located in RAM, at the bottom of the stack where it is relatively secure.

```
0100 20 XX XX  ANYSUB JSR XXXXXX
0103 60               RTS
```

It allows subroutines in ROM (or RAM) to call any of several ROM (or RAM) subroutines, by prior setting of the address XX XX. Although STRINGOUT was intended for teletype output of a character string, ANYSUB allows it to feed characters to other devices. One possibility is a sequence of digital commands to a control system. For this purpose, the above version is restrictive since it excludes 00. This restriction can be removed by making exit conditional on a sequence of two 00 bytes, by modifying (with 8 more bytes) the ALOOP section as follows:

```
ALOOP INY
      LDA (LO), Y
      BNE OUTPUT
      INY            (test next character)
      LDA (LO), Y
      BEQ EXIT       (exit if also = 00)
      DAY            (if not, output the
                        single-null character)
      LDA (LO), Y
OUTPUT JSR ANYSUB
      CLC
      BCC ALOOP
 EXIT TYA            (same as earlier version)
```

One possible use of the unrestricted STRINGOUT would be to relocate blocks of instructions (up to 256 bytes) in RAM. In fact, this is how I tested the program, since I do not have a teletype. Since the Y index (as well as the character) is available to ANYSUB, STA instructions using the Absolute, Y addressing mode can be used.

Sincerely,
H. T. Gordon
University of California, Berkeley

STRINGOUT MODS – ADDENDA

Dear Mr. Warren: February 4, 1977

(This is an addendum to my letter of February 2.)

While playing with the double-null-terminator version of STRINGOUT, it dawned on me that the restriction that its transfer subroutine pre-exist at 0100 is unnecessary, since it can create it from the first 4 bytes of the string. The following alteration replaces the LDY instruction at 020A in the original version by a block of 12 bytes. It causes the first 4 bytes of the string to be copied into 0100-0103. These can be JSR XX XX RTS to create a subroutine, but if the next 2 bytes are 00 00, then STROUT (the routine is now so different that a new name seems justified!) can be used to put *any* 4-byte sequence into 0100-0103 using 9 program bytes.

```
020A A0 04        LDY #04          (set to store 4 bytes)
020C B1 EA SLOOP LDA (LO), Y       (gets 1 of first 4 bytes)
020E 99 FF 00     STA STBASE, Y    (stores in low stack)
0211 88           DEY              (decrement Y index)
0212 DO F8        BNE SLOOP        (loop until zero)
0214 AO 04        LDY #04          (reset Y index)
0216      ALOOP INY                (same as double-null
                                      version)
```

The full program is now up to 58 bytes, but much more versatile.

Sincerely,
H. T. Gordon

P. S. Enclosed is a check for a 1-year subscription (in no way conditional on publishing my stuff – it has to stand on its own merits!). Your journal is really the best on programming

SECRET CODES, ANYONE?

NOTICE:

CRYPTOLOGIA, a journal devoted to all aspects of cryptology, will be published four times a year. There is a need for a forum for the exchange of ideas related to cryptology in the public sector. This journal will meet that need with research papers, survey articles, personal accounts, reviews, educational notes, and problems. Some of the areas which will be discussed are mathematical, computational, literary, historical, political, military, mechanical and archeological aspects of cryptology. Further information may be obtained from CRYPTOLOGIA , Albion College, Albion, Michigan 49224. The Editors: Cipher A. Deavours, Department of Mathematics, Kean College; David Kahn, Department of Journalism, New York University; and Brian J. Winkel, Department of Mathematics, Albion College.