

52-NOTES

Volume 4 Number 3

48/39/38

March-May 1979

Newsletter of the SR-52 Users Club
published at
9459 Taylorsville Road
Dayton, OH 45424

A KEYBOARD SBR N IN ONE STEP (58/59)

Ed Westenhaver (1095) has discovered another useful end-of-partition function (V3N6p5), noting that a program-executed SBR at the last step presents the user with the ability to specify continued execution at any defined label without having to manually key SBR. For example, with a 59 at turn-on, and the following sequence written in RAM: LA GT0 479... Lbl RCL... R/S... 479: SBR, pressing A causes a halt with flashing display. If RCL is then pressed, the code following Lbl RCL is executed.

Mechanizing this discovery in programs with many entry points can save a significant number of input keystrokes. Ed found an improved typewriter (V2N9p5) to be a good example. The program which follows is based on one Ed wrote, combining his key-character assignments with faster processing. The space, and alphabet characters can be input at a rate of about one per second (compared with about one per 2 seconds for the TI LE and MU CROM routines), and require only a single keystroke per letter. The remaining 37 characters require the shift key (2nd) as a prefix, and can take as long as 2 seconds for input processing, depending upon how far down in RAM their labels are.

I've found that the plastic key-code overlay (factory-supplied with each 58 or 59) makes a good overlay base on which adhesive paper strips can be mounted, and appropriately inscribed as follows: A=start/capital π , C=delete (backspace), D=next, E=space/ Π , 2nd=2nd, INV=A/ Σ , lnx=B/delta, CE=C/ \bar{x} , CLR=D/ x , xXt=E/xXt, $x^2=F/2$, $\sqrt{x}=G/\sqrt{}$, 1/x=H/up arrow, STO=I/, RCL=J/!, SUM=K/*, $y^x=L$, EE=M/%, (=N/(,)=0/), $\div=P/\div$, GT0=Z//, 7=/7, 8=/8, 9=/9, X=Q/X, SBR=Y/!', 4=/4, 5=/5, 6=/6, -=R/-, RST=X/?, 1=/1, 2=/2, 3=/3, +=S/+, R/S=W/e, 0=/0, .=V/., $\pm=U/\pi$, ==T/=, where for $a=b/c$, a is the stamped name of a key, b is the assigned first function and c the assigned second function. The B key is free to be used for either first or second function assignment to one or two frequently used characters, in which case B and/or B' could replace the currently assigned labels for those characters.

Members may wish to modify this program to store assembled print-code for data recording, and are invited to try to speed up execution.

TI-59/PC Program: Fast Typewriter

Ed Westenhaver (1095)/Ed

User Instructions: Press A to initialize; see number of character-positions remaining (20) flashed. Input characters per overlay. Full lines are automatically printed; press D to initiate the printing of less than 20 characters. Press C to backspace, and press LRN LRN to clear the flashing-display error-state.

The SR-52 Users Club is a non-profit loosely organised group of TI PPC owners/users who wish to get more out of their machines by exchanging ideas. Activity centers on a newsletter, 52-NOTES, edited and published by Richard C Vanderburgh in Dayton, Ohio. The SR-52 Users Club is neither sponsored nor officially sanctioned by Texas Instruments, Inc. Membership is open to any interested person: \$6.00 (\$10.00 US abroad) includes 6 issues of 52-NOTES; back issues start June 1976 @ \$1.00 each (\$1.67 abroad).

Program Listing:

```
000: S*0 Op20 Dsz1 477 GTO 323 LE 0 RST L INV 13 RST L lnx 14 RST L CE
027: 15 RST L CLR 16 RST L xXt 17 RST L x2 21 RST L  $\sqrt{x}$  22 RST L 1/x 23
054: RST L STO 24 RST L RCL 25 RST L SUM 26 RST L yx 27 RST L EE 30 RST
080: L ( 31 RST L ) 32 RST L ÷ 33 RST L X 34 RST L - 35 RST L + 36 RST
110: L = 37 RST L ± 41 RST L . 42 RST L R/S 43 RST L RST 44 RST L SBR
137: 45 RST L GTO 46 RST L Dsz 1 RST L Ifflg 2 RST L DMS 3 RST L  $\pi$  4 RST
161: L xGET 5 RST L  $\Sigma$ + 6 RST L x 7 RST L x=t 10 RST L Nop 11 RST L op 12
187: RST L Adv 40 RST L Prt 53 RST L List 64 RST L Stflg 71 RST L Grad
210: 47 RST L Lbl 65 RST L Rad 20 RST L Pause 63 RST L Deg 50 RST L Eng
235: 61 RST L Fix 55 RST L Int 56 RST L Abs 72 RST L CMs 57 RST L Exc 73
262: RST L Prd 51 RST L cos 52 RST L Write 54 RST L sin 70 RST L P/R 62
287: RST L tan 60 RST L CLR' 66 RST L CP 67 RST L log 75 RST L INV' 77
312: RST L E' 74 RST L A' 76 RST L D Op0 R6 + R5 EE 2 INV EE+R4 EE 4 INV
342: EE + R3 EE 6 INV EE + R2 EE 8 = INV EE Op1 R11 + R10 EE 2 INV EE +
370: R9 EE 4 INV EE + R8 EE 6 INV EE + R7 EE 8 = INV EE Op 2 R16 + R15
398: EE 2 INV EE + R14 EE 4 INV EE + R13 EE 6 INV EE + R12 EE 8 = INV EE
424: Op3 r21 + R20 EE 2 INV EE + R19 EE 4 INV EE + R18 EE 6 INV EE + R17
452: EE 8 = INV EE Op4 Op5 LA CMs 3 S00 19 S1 LC Op30 Op21 R1 SBR
```

VECTORED PROCESSING WITH Op 10 (V1N3p5, V1N4p3, V2N8p1, V2n11p2, V3N2p3,
and throughout the calendar printer articles: V3N5-V3N9)

Past articles have discussed various methods by which the input datum is used either directly or following a transformation to "point" to one of several processing choices, in addition to its possible function as a processing input. The primary advantage of such mechanizations is fast processing: time is not wasted conducting multiple comparisons. For example, consider a program designed to accept one of ten possible inputs in the 1-10 range, where for an input 1, processing is to begin at step 10, for 2 at step 20,... for 10 at step 100. One way to accomplish this is to compare the input successively with 1,2,... 9 and to branch to the desired address when a match is found. But this approach is slow and code-consuming compared with a method which branches indirectly on ten times the input as a pointer. The first approach might look like: LA xXt 1 x=t 010... 9 x=t 090..., and the second like: LA S0 X 10 = S1 GTO*1... Here, the input is vectored to the desired processing via a simple transformation (ten times the input).

Other more complicated transformations have been discussed in earlier articles, involving carefully contrived arithmetic manipulations to eliminate, or at least minimize "collisions" (V2N8 and Scientific American Apr 77 p63-80). For the special case where there are only 3 processing choices to be made, Jared Weinberger (221) and Bill Skillman (710) have been exploring a new approach using the 58/59 Op 10 function.

From: LA (CE + Op10 S4 Op24 R*4 S04 0) GTO*4 by Jared which handles a 3-way branch on a negative, zero, or positive input through Reg 0, 1, and 2 as pointers, Bill has devised eleven related routines to cover all possible conditional relationships on both a single variable WRT zero, and 2 variables WRT each other: LA (CE + Op10 S4 Op24 R*4 S04 0) GTO*4 LB (CE + Op10 S4 R*4 S04 0) GTO*4 LC (CE + Op10 ± S4 R*4 S04 0) GTO*4 LD (CE + Op10 Abs S4 R*4 S04 0) GTO*4 LE (CE - R3 + Op10 S4 Op24 R*4 S4 R3) GTO*4 LA' (CE - R3 + Op10 S4 R*4 S4 R3) GTO*4 LB' (CE - R3 + Op10 ± S4 R*4 S4 r3) GTO*4 LC' (CE - R3 + Op10 Abs S4 R*4 S4 R3) GTO*4 LD' (R5-r3)

Op10 S4 Op 24 R*4 S4 GTO*4 LE' (R5-R3) Op 10 S4 R*4 S4 GTO*4 L+ (R5-R3)
 OP10 ± S4 R*4 S4 GTO*4 L= (R5-R3) Op10 Abs S4 R*4 S4 GTO*4. In all
 cases prestore 2-way branch addresses in Reg 0, 1, and 3-way branches
 in Reg 0, 1, and 2.

It would appear that practical applications of this approach
 would be in cases where no suitable hash function can be found, and
 the T-register is needed for other purposes. Members are invited to
 make improvements and/or report their own practical applications.

In another application of Op 10, Jared beats the TI 58/59 Manual's
 IV-96 check-service-charge program with: LA S0 fix2 X .1 A' A' LA' - 5
 INV SUM0 R0 X (Op10 + 1) ÷ 200 = rtn which is shorter, and faster for
 inputs greater than 14.

BOOK REVIEW: COUNTDOWN: SKYDIVER, ROCKET AND SATELLITE MOTION ON
 PROGRAMMABLE CALCULATORS by Robert Eisberg and Wendell Hyde (dilithium
 Press Box 92 Forest Grove, OR 97116; 1979, 107 pp, \$6.95)

This new publication appears to be a sequel to Eisberg's earlier
 work: APPLIED MATHEMATICAL PHYSICS WITH POCKET PROGRAMMABLE CALCULATORS
 (V2N1p6), simplifying the sections on freefall, fall with friction, and
 central force motion, targeting the algorithms to the newer low-end
 PPCs: TI-57 and HP-33E.

The reader is assumed to be new to both physics and programming,
 and is led step by step in both areas to mechanize a few simple numer-
 ical solutions to some of the fundamental motion problems. The chosen
 applications, and variations on them should be both entertaining and
 educational to the physics/programming novice, and without burdening
 him with the formal definitions, the text announces at the end that
 the reader/user has solved nonlinear second order, and "... coupled,
 second order, differential equations. Congratulations."

MORE ON THE MU CROM (V3N12p3-5)

Bill Skillman (710) has examined and used the routines in this
 CROM, and offers the following comments:

MU-04: Columns of figures can be printed without using MU-03.
 (I note also that with or without MU-03, restarting using old line num-
 bers requires CMs to "erase" previous characters).

MU-06: If data are prestored in Reg 1 to n by non-MU-06 means,
 Reg 0 needs to be initialized with n.

MU-08: The max number of pseudos per real register may be exten-
 ded to 11 by creating the format statement incrementally (register
 arithmetic, or display summing (see V3N7p4)).

MU-11: There is noticeable accuracy degradation for inputs less
 than $\frac{1}{2}$.

MU-14: For call by a user program, follow manual initialization;
 the suggested shortcut is incomplete.

MU-15: A factor of x was omitted in the denominator of the f'(x)
 expression, $x \neq 0$.

MU-17: Contrary to instructions, you can use = and CLR in SBR A',
 but you must preserve t (MU-17 uses T in an xGET comparison at step
 236).

Two observations which Bill makes generalize to all CROMs (as
 noted by Fred Fish (V3N3p1)): 1) Check CROM listings for routines
 which only store inputs, and for user program calls, bypass these with

appropriate user sequences of the form: STO nn, and 2) Check CROM code before and after calls to user routines (via Pgm 00 N) to see if it is safe to use CLR and/or =, since it is not always necessary to heed related caveats in the CROM manuals.

The MU CROM is slowly becoming available (not yet showing up in Dayton), and as more members acquire it, I hope to get more discussion of this unusually good module.

*** DAVID W JOHNSTON (5) -1979 ***

I was saddened to learn via a note from Jim Doman (473) of Dave's death 2 April 1979. In his last letter to me in January, Dave reported that he was undergoing chemotherapy for cancer, and that he was "holding my own".

Dave will be missed, especially by those members who have been active in the 56/57/58 Program Exchange which he ran so efficiently and modestly. I have been in touch with Dave's closest survivor, a cousin living in Massachusetts, and hope to obtain Dave's Program Exchange records. At such time as they become available, I'll pass them along to a volunteer, or lacking the latter will attempt to carry on the Program Exchange service myself.

Another PPC notable died in January: Jim Davidson, a major contributor of input to 65-NOTES/PPC JOURNAL, and a participant in our Friendly Competition (V2N12p4).

- - - - -

ZERO AS A CALLABLE LABEL (58/59)

Maurice Swinnen (779) passes along a discovery published in DISPLAY (V4N5+6S76), which reveals that zero can be used as an addressable label (V3N7p3). It turns out that a call to any of the ten user-defined keys (A-E') preceded by Ind, executes as a call to Label 0 (if defined). The user-defined label may (but need not) be defined. For example: LA seq1 rtn L0 seq2 rtn called by pressing A, executes seq1, while Ind A (or Ind B,C,...E') does seq2.

A call to LA seq1 Ind C seq3 rtn L0 seq 2 rtn with A does seq1 seq2 seq3. Zero appears to be the only numeral which will work this way, and somehow Reg 0 gets into the act, since a print trace reveals its contents following execution of Ind n (n=A,B,...E') as if Reg 0 were being addressed indirectly.

ROM CONSTANTS UPDATE (V4N1p6)

John has explored further and reports: "I have determined the accuracy of all 16 of them. Nine turned out to be good to all 16 places; the other seven are good to the number of places shown (): $\tan^{-1}.1$ (14), $\tan^{-1}.01$ (15), $\ln 10$ (14), π (13), $\pi/2$ (13), $\pi/4$ (15), and $18/\pi$ (13).

"The ROM constants for $\tan^{-1}.1$, $\tan^{-1}.01$, and $\pi/4$ end in zero, and appear to be simply a rounding to the number of digits given. The other four all end either 012 ($\ln 10$, π), or 022 ($\pi/2$, $18/\pi$), so these numbers could be some sort of code. (That the 14th digit of $\ln 10$ (0) is good is probably a coincidence)."

CORRECTIONS: V4N2p5 step 234 is Adv; V4N2p6 step 254 is ÷ (divide) and step 263 is 51.

MEMBERSHIP ADDRESS CHANGES: 395: 569 Broadway Ave Victoria BC V8Z 2G3 Canada; 559: 4427 Rosada St Long Beach CA 90815

52-NOTES V4N3p4 (end)