52-NOTES

Volume 2 Number 5 48/39 May 1977

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

QUASI-ALPHANUMERICS: A COMPARISON OF SR-52 AND HP-67 CAPABILITIES

A few months ago an exciting HP-67 discovery (consequences of which are still unfolding) opened up the possibility of displaying quasi-alphanumeric messages on that machine... a capability with similarities to SR-52 fractured digits and which got me to wondering about using a mixture of fractured digits and numerals for SR-52 messages. I don't think the value of displayed messages needs to be defended, but as an insight into what matters most with limited capabilities, it is instructive to examine what each machine can do. For purposes of this discussion, display positions for both machines are numbered from left to right as 0-13. An HP-67 user can store and recall special "numbers" to and from data registers which when displayed can show the ten digits, facsimiles of the letters: r, C, o, d, and E, the minus sign, and blank, in accordance with what appear to be the following rules: Position 0 can only be either blank or the minus sign, each of the positions 1 through 10 can be any character except the minus sign, position 11 can only be blank, and positions 12 and 13 can be any character except the minus sign, but their combined value determines the position of the decimal point (which cannot be suppressed). Once a complete display line has been synthesized and successfully stored in a data register, it cannot be merged with changing data; i.e. selected positions cannot be masked. However, there is a way to modify pure data to be flicker-displayed with minus signs, blanks, and decimal points. Messages can be displayed effectively using the two built-in pause functions, a software-generated flicker pause, and the data-entry-sensing capability that flag 3 provides: automatic continued program execution when a number is entered from the keyboard during a pause-displayed prompting message. It is apparently by design that the 5 fractured digits look like letters, since they are all used in the 2 words Error and Crd generated by firmware and displayed as user cues.

So far as I know, in addition to the blank, the SR-52 can produce only 4 fractured digits, only one of which (the square o) is very letter-like. But before we dismiss the SR-52 message-display potential as being too limited, let's see what can be done. I have experimented using the following character assignments: A=°, B=8, D=0, E=3, F=', G=9, H=", I=1, J=', L=1, N=", O=0, P=6, R=7, S=5, T=-, U=", Y=4, Z=2, and find that when these symbols are grouped in the display to form familiar words, intended meaning is generally conveyed. But it's probably best not to hold to fixed character assignments, as specific juxtapositions may suggest variations. For example, by the above assignments, the

The SR-52 Users Club is a non-profit loosely organised group of SR-52/56 owners/users who wish to get more out of their machines by exchanging ideas. Activity centers on a monthly 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 includes six future issues of 52-NOTES; back issues start June 1976 @ $1.00 each.

word GOOD would be spelled 9000. But $9^{00}0$ is probably more recognizable, even though it might, in certain contexts, be interpreted as GOAD.

The message synthesizer program below may be used to see what various trials look like, and it produces the 2 numbers required to produce the message by a running program using the sequence: RCL 98 + STO 60 RCL 99 HLT (with = pressed manually). After a desired message has been synthesized, the contents of Reg 98 and 99 can be transferred to 2 program registers (unused by a user program) for permanent storage. 14 characters are entered as single digits followed by A for the first one, then either E (if to appear normally) or C (if to appear fractured). The fracture codes are per V1N2p5. Positions 0 and 11 may only be either blank or -, position 10 is only zero for the SR-52A, else must be one of the ten digits (0-9), and zeros in positions 12 or 13 are displayed as blanks. For example, try: 6A, 7E, 4E, 3C, 5C, 9E, 5C, 1E, 2C, 3C, 0E, 6C, 1E, 5E; then after you press =, see: TRY AGAIN OTIS. Or (SR-52 only) 3A, 9E, 3E, 6C, 3C, 5C, 2C, 3C, 3C, 5E, 7E, 6C, 5E, 2E; Press = and see GET AN SR-52.

One can, ofcourse, mix messages with changing data. During the synthesis process, substitute zeros for the positions to be dynamically filled, and design the user program to put the desired data into these positions via careful register summing for positions 1-10, and and multiplication for positions 12 and 13. Keep a copy of the original "template" (with the pre-positioned zeros) intact, as a new template will be needed for each data change.

Out of all this, it is evident that the limitations of each machine are quite different. The HP-67 fractured digits are more letter-like, but there are fewer "natural" multiple assignments (such as " standing for H, N, or U for the SR-52); the SR-52 can mask selected positions, but display isn't as automatic or fancy (pausable or flickerable); each HP-67 line takes considerably more effort to produce, but each can be coded into the contents of a single data register, recorded, and produced (displayed) with a single instruction (RCL). Neither the HP-97 nor the SR-52/PC-100 can print fractured digits.

Much of the incentive for trying to create messages on either machine is due to the challenge: it is mildly difficult, and often considerably frustrating. But, ofcourse, as soon as a PPC comes along with real alphanumerics, nobody is going to bother, any more than we would spend time now developing efficient manual keystroke sequences for the non-programmable scientifics. But it should be fun for awhile! So send me your best messages and/or applications, and perhaps quasi-alphanumerics will become a lively part of the TI/HP users challenge.

SR-52 Program: SR-52 Message Synthesizer Ed

User Instructions:

1. Key position 0 code (3 or 6), press A, see 1.
2. Key position i code, press C for fracture, E for normal, see i + 1; repeat for i=1,2, ...13.
3. Press =, see message.
4. To transfer message code to another program:
   a. Write other program into memory
   b. In RUN mode: Key RCL 98 STO ab RCL 99 STO cd, where ab is address of program register to hold operator number and cd the operand register.

SR-52 Message Synthesizer Program Listing

```
000:  -6 = *ifzro 009 *stflg 0 CLR 9.00 STO 98 STO 99 .01 STO 69 9 STO 00
030:  11 STO 68 B *ifflg 1 053 *E' SUM 99 *dsz 035 GTO 061 *E' SUM 98
057:  *dsz 035 B *E' SUM 99 B - 6 = *ifzro 076 *stflg 2 B *ifflg 1 096
082:  - 1 = X 10 = SUM 68 GTO 103 EE +/- 12 SUM 98 B *ifflg 1 119 -1 =
112:  SUM 68 GTO 131 X 10 = INV *log *fix 0 EE *PROD 98 RCL 68 INV *log
136:  *fix 0 EE *ifflg 2 146 EE +/- *PROD 99 *ifflg 0 159 1 +/- *PROD 99
159:  RCL 98 + STO 60 RCL 99 HLT *LBL *E' X RCL 69 = *rtn *LBL E INV
181:  *stflg 1 *rtn *LBL C *stflg 1 *rtn *LBL B .1 *PROD 69 1 SUM 67
200:  RCL 67 INV EE HLT *LBL A = *rset
```

- - - - - - - - - - - -

SR-52A/PC-100A

Peter Stark (321) notes that while the SR-52A has been alluded to in 52-NOTES, it has not been properly identified. It is the "newer" SR-52 I refer to on V2N2p6 under "Display Arithmetic Modification" and so far as I know, such machines are not physically marked as such ("SR-52A" is not printed or stamped on the machine). Bob Edelen (100) notes that Alan Charbonneau's Yahtzee program (V2N2p5) will not work with an SR-52A, and suggests the following changes for use with the newer machines: replace the 5 with 6 at step 98; replace the sixes at steps 210-214 with ifflgs. All five dice will show, followed by n irrepressible zero. While the TI trouble shooting manual (V2N4p4) suggests a number of hardware differences between the SR-52 and SR-52A machines, the only difference I am aware of that is noticeable to the user is the 13th digit truncation.

Bob Edelen has made some electric power measurements and reports the following: The PC-100A charges the battery PAC at a constant 120ma even when the power switch is off. His SR-52A with adaptor charges a full PAC at 100 to 110ma with SR-52 off; on: 50ma. A dead PAC draws 120ma with SR-52A off; on: 60ma. SR-52A turned on with 0 displayed draws 160ma, with full display 260ma, which discharges a full PAC 45ma during charge. SR-52A execution (with the 2 minus signs displayed) draws 145ma. Since continuous charging is not good for batteries, Bob has modified his PC-100A to allow the SR-52A to discharge the PAC while connected to the printer. Members interested in details are invited to send Bob a SASE (note his address change elsewhere in this issue).

TI NOTES

Machine Exchange_At TI_Centers: Although individual retailers may be changing their exchange policies, it is still TI policy (see V2N1p2) for its regional centers to continue to exchange good machines (of any model vintage) for bad, that are still in waranty.

Adding_Chips_(52): TI is now naking available to the public the TMC 0599 chip via Mr S Riggs, Box 53, Lubbock, TX 79408 for $17.77 each. Reportedly, 2 of these will provide the missing SR-52 registers (20-59) if properly installed. Michael Rak (502) has obtained details from a TI source, and if you are experienced at assembling modern electronics components and are willing to risk machine damage, send 25¢ and a SASE to Mike at 1823 El Cerrito Pl #F Los Angeles, CA 90068 for a copy of his info. Neither Mike, nor TI nor the SR-52 Users Club assumes any responsibility whatsoever for any consequences which might arise from chip ad-on mods. TI does not now, and does not plan to perform this mod for the general public.

TIPS

Mag Card_Marking: Noting that the first issue of the PPX-52 newsletter mentions use of a "Sharpie" pen, Mack Maloney (246) has found that the Sanford Sharpie #49 is permanent, and that a Vis-a-Vis 69 is smearless with markings that can be removed with a moistened fingertip or cloth, and which seem to work in place of the write-protect mag card tabs.

Another One-Step_Error_Producer: Jared Weinberger (221) suggests repeating an operator to create an error condition. For example, the sequence: RCL 99 + + RCL 98 = performs the desired addition as well as creating an error condition.

More_on Trig_Function Anomalies: Karl Hoppe (507) notes that attempts to take the sin, cos, or tan of a number whose absolute value is between zero and 3.6 D-97 in degree mode (or 6.283185307180 D-99 in radian mode) result in an error condition. This may be due to the creation of underflowed intermediate results by the trig firmware.

IND Prefix_to a Subroutine_Call (52): Several members have noted that placing an IND before a subroutine call has not yet been mentioned explicitly in 52-NOTES. An IND prefix maintains viability through a subroutine call, just as an INV does (V1N1p2). Further, the sequences: INV *IND or *IND INV are viable together. For example, the routine: *LBL A SUM 99 *rtn prefaced with INV *IND will cause the contents of the register pointed to by Reg 99 to be decremented.

Battery Charging_PAC: Noting that TI does not supply a battery charging PAC for any of its PPCs, Bob Edelen (100) claims that the HP PAC will work with connector mods, and a rubber band to hold things together.

More_on the 0 divide_error_Condition: (52): Howard Cook (556) reports that an underflow resulting from dividing a small number by a number GE 10 produces the 0 divide error condition (see V1N1p2). However, product by a number LE .1 does not do the same thing.

Fast_Relative Addressing_(52): John DeHaven (548) notes that the machine will branch to a label located within the first five or so steps faster than it will with an absolute branch.

ROUTINES

Odd/Even_Determiner (52): Jim Rosenberg (516) has devised a clever way to determine whether a positive integer is odd or even: ... STO 00 50 *PROD 00 *IND RCL 00 *ifzro GTO ..., which branches to GTO on odd, continues on even. Here is another way a non-register (50) can be put to good use (see V2N2p6).

Efficient Temperature Units Conversions (52): John DeHaven (548) has an algorithm for conversions between Centigrade and Fahrenheit which combines the 2-way calculations efficiently and which can be mechanized by: *LBL A *E' *LBL B + 40 = X 1.8 *iferr *LBL *LBL *1/x - 40 = CE HLT. With temp in degrees F in the display, press A for conversion to C; B for C to F. (E' must not be defined).

MEMBERSHIP ADDRESS CHANGES

100: 1632 Yosemite Denver CO 80220; 128: 356 Dart Dr Hanover PA 17331; 252: 2111 Bidwell Rd #C1 Muscatine, IA 52761; 353: 2725 Monte Mar Terrace Los Angeles, CA 90064.

FRIENDLY COMPETITION

The version of Barbara Osofsky's 5 X 5 Matrix program currently the challenger for HP-67 users lists below (see V2N4p2), and needs no further comment (see V2N2p3), except that I take credit only for using Rick Wenger's error-flag approach; credit for eliminating more keystrokes and the potential wipe-out danger (V2N3p3) goes to Barbara.

Another challenger candidate is Karl Hoppe's (507) "Square of 70 Digit Factor" SR-52 program, which follows. While there might possibly be a sufficiency of data registers, ten-digit arithmetic will be a hurdle for HP-67 mechanization. One construct which Karl uses identifies an EE viability of which I was unaware: Decapower display softening by EE carries through a subroutine return into main-program continued execution. For example, the call to D at step 112 returns 1D21 in the display register, which then gets modified to 1D16 by the 6 at step 113.

SR-52/PC-100 Program: 5 X 5 Determinant and Inverse Osofsky/Ed

User Instructions:

1. Initialize: Key 95, press C
2. Key e_i, press E; repeat for i=1,2,...25 with column-wise catenation; see 1 displayed.
3. Get determinant: with 1 in display, press D; first col of adjoint printed; determinant displayed.
4. (optional) Print Determinant: On printer: press ADV, PRINT,ADV
5. Get Inverse: With determinant in display, press D; first column printed, 1 displayed.
6. Get ith column: Key i, press RUN, repeat for i=2,3,4,5; ith column printed, 1 displayed. For new problem, go to step 1.

Program Listing:

```
000:  *LBL *B' STO 68 (*IND RCL 66 X *IND RCL 68 - 1 SUM 66 +/- SUM 68
024:  *IND RCL 66 X *IND RCL 68) X*LBL *E' 1 SUM 66 *rtn *LBL C STO 66
047:  *rtn *LBL *C' (8 C 4 *B' *E' 17 *B' +/- + 19 *B' C 7 *B' - 14 *B'
071:  *E' 17 *B' + 19 *B' *E' 2 *B' + 9 *B' *E' 2 *B' - 4 *B' 6 C 12
095:  *B') div RCL 67 +/- STO 67 X *prt RCL 95 + *rtn *LBL *D' 10 +/- A
118:  *LBL B 95 C *A' *A' *A' *A' *LBL *A' *IND RCL 66 *IND EXC 69
137:  *LBL E *IND STO 66 5 *iferr 149 1 SUM 66 *LBL A SUM 69 *rtn
158:  *LBL D +/- STO 67 CE 0 STO 69 *C' B *C' B *C' B *C' B *C' 5 A
180:  *D' *D' *D' *D' *pap 0 = HLT STO 69 94 A *x! B GTO 164
```

- - - - - - - - - - -

SR-52 Program: 70 Digit Square Karl Hoppe (507)

User Instructions:

1. Initialize: Key 99, press B
2. Input up to a 70-digit positive integer (no decapower): Key blocks of ten digits, each followed by RUN; begin with MSD; see prior block of ten digits (from earlier problem).
3. Get square: Press A; after about 200 seconds, see the first ten MSDs of the square.
4. Get remaining digits: Key 6 B; then key RUN for successive blocks of ten digits. For new number, go to step 1.

70 Digit Square   Program Listing:

```
000:  *IND *LBL E RCL 98 *rtn *IND *E' 1 +/- C *rset *LBL D - (STO +
019:  *LBL *D' 1 EE 21 *rtn *LBL C +/- SUM 98 *LBL *C' SUM 68 *rtn
038:  *LBL *B' RCL 68 - 6) *rtn *LBL *A' 18 STO 68 19 *LBL *E' STO 98
061:  *rtn *LBL A 105 STO 69 *A' 0 *IND *E' *IND STO 68 2 C *B' *ifpos
082:  071 9 *C' 6 *C' 105 *E' *IND RCL 69 D 6 - *D' 6) STO 67 = STO 66
110:  *IND E D 6 - *D' 6) STO 64 STO 65 = STO 63 X RCL 67 *PROD 64 +
136:  RCL 65 X RCL 66 *PROD 63) D - *D') SUM 64 + RCL 63) *IND *C' 1 C
163:  RCL 64 div *D' 0) *IND *C' E - 99) *ifpos 110 SUM 69 *B' INV *ifzro
187:  087 *A' *IND E + *D' - *D') *IND INV SUM 98 div *D' 0) *IND *C' 1
210:  C *B' *ifpos 191 CLR 6 *LBL B *E' *rset
```

- - - - - - - - - - -

PPX-52 GEMS

#300032A: 4 X 4 Matrix Multiplication, by Barbara Osofsky (420), facilitates chain multiplications by replacing each successive multiplicand by the next product. The PPX program analyst's note (2) is in error, likely to be misleading, and should read: "Insertions and deletions in the program will change the first 10 entries of matrix 1". As Barbara notes, warnings of this type should be made for any program that uses program registers for data, since data stored in program registers are equivalent to unrelocatable instruction code.

#910050A: Two to Nine Pile NIM, by R L Floyd (82) effectively mechanizes the modulo two addition algorithm that handles up to nine piles, each of which can be up to 1D10 in size. Richard was the only one to respond to my invitation (V1N3p3), but noted at the time that he was submitting his program to the PPX-52 library.

BOOK REVIEW: COMPUTERS AT LARGE, by C J Sippl (239) and R Bullen, Bobbs-Merrill, 1976, 222 pages.

Here is a readable discussion for the layman that covers a broad spectrum of man-computer relationships. Emphasis is on the sociological, psychological and philosophical impact computers have made and will make on man and society. The text is liberally sprinkled with quotes which the reader can pursue in greater depth via a substantial bibliography. Discussion of the varied uses to which computers have been put, and the broad range of consequences to humans provides good food for thought.

MEETING THE SUM-OF-THE-DIGITS CHALLENGE (V2N3p2)

Howard Cook (556) beats Jared's shortest by one with: *LBL C EE STO EE 00 - 1 SUM 69 = INV *ifzro C *EXC 69 *rtn, and notes that Jared's routine A and his routine C effectively sum the rounded 10-digit display, and that Jared's routine B sums up to 12 of the most significant digits in the display register. Howard goes on with a new challenge: beat his 56 step routine D which sums all 13 digits: *LBL A (STO - .5) EE INV EE *rtn *LBL D STO 01 *fix 0 *log A INV *log A *1/x *PROD 01 *LBL RCL RCL 01 *ifzro *EXC A SUM 69 INV SUM 01 10 *PROD 01 GTO RCL *LBL *EXC *EXC 69 HLT, which he says is clumsy, but which has some clever constructs.

SUBJECT/AUTHOR CROSS REFERENCE

Bob Edelen (100) has kindly volunteered to work up an index covering the first 12 issues of 52-NOTES, and will cut mimeo stencils under computer control, in much the same way Mike Brown did with the alphabetized membership list. I'll be asking Mike for a follow-on soon, that will cover only currently active members.