52-NOTES

Volume 2 Number 4 48/39 April 1977

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

EDITORIAL: A Perspective On Machine Comparisons and Loyalties
INVITATION: To Engage In Friendly Competition

During the two years that the HP-65 reigned supreme as the first and only pocket (personal) programmable calculator (PPC), there was no question of loyalty to a specific machine or manufacturer, or of measuring machine capability against a competitor's. If you wanted a PPC, the HP-65 was it. Back in late 1973 those of us who had some inkling of what a PPC might do could hardly wait for the first one to become available, and stood in line for months for the privilege of paying $800 for an HP-65. We were primarily motivated by two expectations: 1) Having exclusive possession of a considerably more powerful computational tool than ever before, and 2) Having the opportunity to win the recognition and acclaim of our peers which would follow discovery and invention.

I submit that these same two expectations continue to dominate our thinking as we attempt to choose from a growing proliferation of new PPCs. Most of us who already had HP-65s and who still decided to buy SR-52s over a year ago, were unaware of user access to as many as 60 storage registers, yet saw the power of, and could imagine the potential for inventiveness offered by the IND function. Then, having both machines made us face up to the matter of machine/manufacturer loyalty. I had a strong bias in favor of HP machines, having worked extensively with the 3 HP PPCs: 65, 55, and 25. And although I still prefer the HP machines for manual use, when the HP-67 appeared, it didn't seem to have enough new programming features to lure me into buying it, especially since it lacks some important programming capabilities which the SR-52 has. Since each manufacturer is motivated to design new machines based on what it thinks the market wants and that it can sell for a profit, it would be surprising if one manufacturer consistently produced machines with the greatest discovery/invention potential. So I see no point in restricting one's patronage to any one company or product line in this game.

Unfortunately, excessive loyalty toward one machine makes it difficult if not impossible to compare competing machines impartially, or even correctly. And I suspect that all else being equal, uni-loyalty becomes the strongest toward the most expensive of competing machines; the rationale goes: it just has to be better since I paid more for it.

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, Incorporated. Membership is open to any interested person: $6.00 includes six future issues of 52-NOTES; back issues start June 1976 @ $1.00 ea.

I bring up this matter of machine loyalty because I think it helps to explain some of the poor comparisons that abound in print, and to suggest that there may be a more productive and stimulating way to compare rival PPCs than by extoling (or quashing) virtues: Write and demonstrate tough programs on one machine and challenge users of a rival machine to match or better them. To be sure, there will always be some disagreement over what is "best", but there are probably enough solid judging criteria such as: required number of keystrokes, precision of results, technical validity, execution time, I/O ease, max number of inputs, handling of worst-case conditions, etc to make the outcome of programming contests conclusive in most cases.

I have discussed the idea of friendly competition with Richard Nelson (Editor and publisher of 65-NOTES), and we have agreed to give it a try. Machines would generally be matched as follows: SR-52 vs HP-65 or HP-67; SR-52/PC-100 vs HP-97; SR-56 (without printer) vs HP-25; although other combinations might be appropriate. Richard and I will do the judging, calling upon tie-breaking expertise as required. Users of TI machines should send candidate challenger programs to me; HP to Richard. We will decide jointly which challenger programs to accept, and announce them in our respective newsletters along with suspense dates for response. Letting better challenger programs supercede original ones may be allowed, but changes would be subject to mutual agreement. For the machines with card readers, candidate programs must fit on one card, although manual code modifications will be allowed and will count in the total of required user keystrokes. Data may be read in via additional cards. For the other machines, there will be no upper limit on program size; overflow will automatically incur the penalty of additional user keystrokes. All programs will be subjected to mutually agreed upon test cases. Additional rules, requirements and constraints will be announced as they suggest themselves and are mutually accepted. To get the ball rolling, to Richard I suggested Barbara Osofsky's SR-52/PC-100 5 X 5 Determinant and Inverse program (V2N2) as an HP-97 challenger, and he has accepted. The particular version of Barbara's program has yet to be decided.

The restricted and general case versions of "Corner The Lady" (see A GAME CHALLENGE below) will likely be good subjects for future challenges, but I prefer to let them get used and improved upon for a while first. In the meantime, send me your candidate challengers: programs you've written for SR-52, SR-52/PC-100, or SR-56 that you think would be difficult to program on the HP-67, HP-97, or HP-25, respectively; and get ready for challenges from the other side!

MORE ON PRINTER GRAPHICS (52)

As several of us noticed, the 17 March 77 issue of ELECTRONICS has an article describing various approaches to and uses of SR-52 printer graphics (see V1N3p2), some of which appear to be worth trying. However, the listed program is not very well optimized, and could easily be improved for faster execution, and input ease. Our thanks go to the auther for the 52-NOTES V1N3 program credit.

MEMBERSHIP RENEWALS

Although I will continue my policy of not issuing individual reminders, I'll note that many memberships become due for renewal in May. If you are in doubt as to your status, send me a SASE.

A GAME CHALLENGE

Martin Gardner appeared to be throwing down the gauntlet to SR-52 users when he stated in his Mathematical Games column in the March 1977 SCIENTIFIC AMERICAN that "...It is easy to program an HP-97 printing calculator or the HP-67 pocket calculator to play a perfect game" without mentioning possible SR-52 mechanization. The game is one related to the NIM family which Gardner calls "Corner The Lady". As it turns out, the HP Games PAC program he refers to (called Queen Board by HP) handles only the restricted case of an 8 X 8 standard chess board for the playing field, with each of the 64 squares numbered in a way that facilitates processing, but which is rather cumbersome for the user and decidedly unconventional for rectangular notation. In any case, the 8 X 8 game should indeed by easy to mechanize on an HP-67, since a modest effort produced an SR-56 program (below) that is more user oriented, and still leaves program steps and data registers to spare. The real programming challenge lies in mechanizing the algorithm that applies to any size board, to produce a program that will fit within 224 SR-52 steps. I had to try several approaches before arriving at the SR-52 program that follows, which seems to work, but is slow. Since finding enough program memory was the biggest problem, and since an integer/fraction routine is required, and also since there are data registers to spare, I suspect that this may be one case where the HP-67 has a decided advantage over the SR-52. However, there may be alternate viable approaches that would make better use of SR-52 features. I have purposely left the code in relocatable form, and invite the membership to improve on it before someone writes a better HP-67 program!

Gardner's article delves into some interesting facets of Fibonacci numbers, and I expect many of you will find the time spent perusing it to be worth while. For my approach, the ten-digit display size limits the maximum playing field to about 100 X 100. This is reduced further to about 60 X 60 to gain an extra program step. I say "about" in each case since there is a position dependency caused by the conversion of base ten numbers into Fibonacci Notation in non-canonical form, as max board size is approached. The game is played on a standard or enlarged chess board (the squares need not be of alternate color) with one queen as the only playing piece. Starting at any square along two adjacent edges of the board, the queen is moved alternately by each of two players toward the corner opposite the corner formed by the starting field edges. Moves are along files or diagonals as in chess, but the queen may not move backwards or "sideways" (such that no progress is made toward the corner goal). The player that moves the queen into the required corner wins. For the two programs that follow, the playing field is addressed by standard x,y notation, such that for the 8 X 8 case, the starting field is the set of squares: k,7 and 7,k where k=0, 1, ...7, and the winning square is 0,0. Advantage is taken of symmetry, and actual x and y values are translated into playable numbers via a and b as defined. The SR-56 game is definitely more "playable", and should be easy to convert to SR-52ese if you are more interested in playing the game than in mechanizing the generalized algorithm.

SR-52 Program: Corner The Lady Game (Fibonacci Notation Verson) Ed

User Instructions:

1. Key a LT b LT 55, press A, see 0 after about 20 seconds
2. Key b, press RUN; if display flashes, see b'; (a'=a)* else, see a'; press RUN, see b'

*if b'=b, machine's move is a, b-1
For x=a, y=b, x'=a', y'=b'; for x=b and y=a, x'=b' and y'=a'
a,b = player's move; a',b' = machine's move as translated for play

Program Listing:

```
000:  *LBL D 1 + 5 *rtx = div 2 = y^x RCL 00 div 5 *rtx = *fix 0 *D.MS *rtn
022:  *LBL E + 1 = STO 19 CLR 9 STO 00 *LBL *2' D *dsz *1' RCL 69 *rtn
044:  *LBL *1' - RCL 19 = *ifpos *2' +/- STO 19 RCL 00 INV *log *D.MS
063:  SUM 69 GTO *2' *LBL C RCL 18 div 10 - *fix 0 *D.MS STO 18 = *rtn
085:  *LBL B STO 18 CLR 1 STO 00 *LBL *5' C *ifzro *6' D SUM 69 *LBL *6'
106:  1 SUM 00 RCL 00 - 10 = INV *ifzro *5' RCL 69 *rtn *LBL A STO 98 E
130:  STO 17 STO 18 0 HLT STO 99 E STO 16 1 STO 15 *LBL *3' 1 +/-
153:  *PROD 15 C *ifzro *3' RCL 15 INV *log *fix 1 EE *PROD 17 RCL 15
173:  INV *ifpos *9' RCL 16 - RCL 17 = *ifpos *9' RCL 99 - RCL 98 -1= E
197:  + 1 = X 10 = STO 99 B HLT RCL 99 X 10 = B HLT *LBL *9' RCL 17 B
```

- - - - - - - -

SR-56 Program: Corner The Lady Game (Standard Chessboard Version) Ed

User Instructions:

1. Initialize: press RST, R/S
2. Key a.b 8 GT a GT b, press R/S see new position: a'.b' after machine's move. Flashed 0.0 indicates machine's win. For new player's move, go to step 2. For new game, go to step 1

Program Listing:

```
00:  R/S *Subr 72 *x=t 63 STO 1 - *Int STO 0 *CP = STO 2 *x=t 67 X 10 =
24:  xEt RCL 0 *x=t 67 INV *dsz 48 RCL 0 + RCL 2 = *Subr 72 INV *x=t 30
47:  RST RCL 1 - 1.1 = *Subr 72 INV *x=t 50 RST - 1 = RST 0 *fix 1 + =
72:  xEt 1.2 *x=t 95 2.1 *x=t 95 5.3 *x=t 95 7.4 xEt *rtn
```

- - - - - - - - - -

BOOK REVIEW: SR-52 and SR-52A TROUBLESHOOTING GUIDE by Jim Johnson, Kevin Hines, Donna Davis, Don Young; published by TI (see V2N3p6); 32 pages plus two fold-out schematics.

This booklet is of mimeograph-quality print on 8½ x 11 paper, one side only. Most diagrams are handsketched; some detail is difficult to read. The stated purpose "... is to describe the operational theory of the SR-52 and SR-52A and to give guidance in its repair". Theory of operation covers: power supply, clock, the MOS chip set, keyboard, display interface, motor control, card sensor input, mag read/write mechanism, and a reference circuit. Illustrations include the keyboard matrix, display wiring, motor drive circuit, card sensor circuit, reference circuit, SR-52A Block Diagram, 3 pages of waveforms, an unlabeled page of step functions associated with the display digit select signals, four pages of component layouts, 7 pages of parts lists, and 2 fold-out electronic schematic diagrams. Since I am not a EE, I will defer technical review to whomever feels qualified and would like to volunteer. Members able to put any or all of the info inthis manual to good use are invited to share their successes via 52-NOTES.

CORRECTION

Step 193 of the SR-52 program on the previous page should be - not =.

PSEUDO 83: Part I: Accessing the Instruction Address Register (IAR)

Of all the pseudos, 83 stands out as the most fascinating and enigmatic. Orla Damkjer (393) has examined p83 in considerable detail and reports a wide variety of results from its execution, with interrelated dependencies upon such things as the number of digits in the display following the decimal point, display notation/format, angular mode, soft or hard display, whether or not p83 execution has been prefixed by CLR or rset, and even at what step the p83 has been executed. This last dependency was also discovered by Jim Rosenberg (516) who notes that it provides a means of access (albeit somewhat gross) to the IAR. This will be the first topic covered in a series on p83. Members are invited to explore this and other facets of p83 behavior that might lead to new programming capabilities, and to share their significant results. A word of caution: p83 behavior is so dependent upon so many pending conditions or machine states that it is easy to make false assumptions about experimental control conditions.

An IAR is a computer register that tells the machine what instruction to execute next by containing the address of the next instruction. It is incremented during sequential execution, and fed new data for branches. The PPCs all have IARs, but their contents are not accessible to executing programs. When computers are operated in a real-time environment, requirements for program execution to respond immediately to unpredictable external events make it necessary to be able to determine where in a main program, execution stops when interruption occurs. General purpose computers are designed with one or more so-called interrupts which cause program execution to branch to one or more predetermined step locations following completion of the instruction being executed at the time an interrupt occurs, Special interrupt processing routines are written and positioned starting at this (these) predetermined locations to save important information that is vital to continued main-program execution. Included in this saved information is the contents of the IAR, so that after the requirements posed by the interrupt have been met, proper transfer back to the main program can be made. While it may be hard to find practical uses for SR-52 IAR access, mechanization is fairly challenging, and illustrates an important general purpose computer function.

When p83 is executed with 4 digits following the decimal point in the display, a zero divide error state without flashing display appears to result (see V1N1p2) and information is stored in Reg 60 that identifies which program storage register contains the just-executed p83 step. Further, if a program-executed subroutine call precedes the p83 execution, with no intervening branches or rsets, Reg 60 also contains information identifying which program storage register holds the first executable instruction of the subroutine. If we examine the contents of Reg 60 as it appears in the display following a subroutine call - p83 sequence, the following characteristics are observed: The number is either zero, or it is in scientific notation with negative mantissa and negative decapower, in the form: -m. -dd or -m.m -dd . The mantissa identifies the adress (ss) of the program storage register containing the first executable instruction of the called subroutine by the relationships: ss=m+70 or ss=10(m.m)+70. The decapower identifies the address (pp) of the program storage register containing the just-executed p83 by the relationships: pp=dd+48 if the mantissa is in the form -m., and pp=dd+49 if in the form -m.m . However, if the first executable instruction of the called subroutine is in Reg 70, the mantissa is zero which causes the display to be zero, and there is no information concerning the location of the just-executed p83. Also note that when m is either 1 or 10, it is displayed as 1. As an example, program the sequence: *LBL A E p83 HLT starting at step 072 (which puts the p83 in Reg 79), and the sequence: *LBL E 1.2345 *rtn starting at step 215 (which puts the 1 in Reg 97). Press A, RCL 60, and see -2.7D-30, which by the above relationships confirms the locations of the 1 and the p83. To make use of such Reg 60 information during program execution, use the Weinberger exponent extractor (V1N4p6) with rounding, and extract the mantissa with: EE 00. Because of possible ambiguities, a Reg 60 processor routine needs to be written such that the pre-p83 called subroutine is known to be either in the Reg 70-79 block or the Reg 80-97 block.

While all this appears to work under the stated conditions, there may be subtle machine states that go unnoticed which could cause trouble. I've found that if you make the subroutine call manually, or eliminate it, the Reg 60 information will be in a different form, and there are likely to be many more special pre-conditions that will affect results. It might be interesting to write a program that is D/R switch interruptable at many points such that an interrupt processor routine determines within a block of 8 steps where an interrupt occurs, and where the last called subroutine is located.

SR-56 Program Exchange

Dave Johnston's 1 April 1977 catalog lists 21 math, 4 statistics, 23 physics, 7 games, and 6 misc programs for a total of 61. Members who have found particular SR-56 programs to be especially helpful, well-written, etc are invited to share this information via 52-NOTES. Send me the name, Dave's catalog number, and a few lines telling what you like about them.

TI INVITATION TO CLUB MEMBERS

A letter from Robb Wilmot, Manager of TI's Professional Calculators Division, suggests that the mid-June National Computer Conference to be held in Dallas this year might be a good occasion for us users to hold "... informal discussions with some of our (TI's) design and applications people.

"You are aware that, due to our designers work pressures, we do not normally encourage their dialogue with users groups, but the NCC Show being in Dallas, I feel, offers a unique opportunity for a 'once in a while' dialogue."

I have responded that our interest in participating in such dialogue would be determined in large part by the extent to which TI would expect to discuss topics of substance. I will pass along further TI response via future 52-NOTES.