The dominant theme of this issue is fast mode, but more specifically fast mode using the STP IND sequence at the end of the current partition combined with Patrick Acosta's idea for selecting any desired program location for the jump at fast mode entry. The emphasis on that subject was triggered by Dejan Ristanovic's observation that he still sees many fast mode programs using the antiquated "load-and-go" method which is so sensitive to card reader problems.

The "star" article is not on fast mode however; it is Maurice Swinnen's translation of the text which accompanies the Science et Vie program which calculates 1287 digits of pi. See page 21. Not only is the program superbly crafted, but the text makes pertinent observations on programming style and on differences in calculators.

This issue carries additional information on the CC-40 including two graphics programs by Maurice Swinnen. Users of the CC-40 continue to be crippled by the unavailability of peripherals. For example, when analyzing the accuracy of the sine and cosine functions of the CC-40, I read the errors from the CC-40 display and entered the values into my TI-59 so that I could obtain a plot using Michael Sperber's classic Plot-60 program.

This issue also continues the coverage of precision started earlier this year with important inputs from George Thomson, Laurance Leeds, and Bob Fruit. Bob Fruit's input on the IBM PC was particularly interesting. The device performs very well on the interest problem that he proposed, but quite poorly on the problem of recovering a value for pi (see pages 4 and 14). For the first time I had a chance to look somewhat closely at the HP-75. Although I knew that both EDN and BYTE had reported that the HP-75's calculator mode was A.O.S., somehow I never really believed it. But it really is true! H-P also broke ranks with their traditional ten digits--the HP-75 has twelve. The real advantage of the HP-75 would seem to be the compatibility with the HP-IB interface bus so that several peripherals are already available. In contrast, we wait patiently (?) for the CC-40 related devices. No new information on the TI-66 has become available.

The newsletter is not copyrighted and may be reproduced for personal use. When material is used elsewhere we ask as a matter of courtesy that the source be mentioned. The use of material in this newsletter os entirely at the user's risk. No responsibility as to the accuracy and the consequences due to lack of it will be borne by either the club or the editor.
ERRATUM - 1188 Digits of Pi - Lars Hedlund of Sweden writes that there is an error in Program C on V8N1P22. The command at locations 104/105 should be RCL 08 not STO 08 as indicated in the listing. This error was not present in the original program submitted by Jovan Puzovic but was inadvertently introduced during an optimization by the editor.

CLARIFICATION - 1287 Digits of Pi - George Thomson writes that comment (2) on V8N3P8 is slightly misleading. Program steps 125 and 126 are used to define the number of registers used to store the answer. The user is not limited to the options indicated, but may set locations 125/126 to the number of places desired divided by 13 and rounded up.

HP-67 VS TI-59 WITH EE-INV-EE - Laurance Leeds writes "It has been asserted that using the EE INV EE routine with the TI-59 will give the same result as is given by the HP-67. This is not correct. Consider the problem involving the y^x function where x = 2.5 and y = 2.543210631. Then

| Exact solution                  | 10.314 68158 50719 ...
| Exact 10d rounded              | 10.314 68159
| TI-59 solution                 | 10.314 68158 506 including the guard digits.
| TI-59 rounded                  | 10.314 68159
| HP-67 solution                 | 10.314 68158

In this case and several others as well the TI-59 gives the exact 10 digit rounded answer, whereas the HP-67 is in error."

Editor's Note: The discussion of the HP-67, the TI-59 and other calculators appeared in V8N3P13/14. I had compared the HP-67, the HP-41 and the HP-11 with the TI-59 for the case where five square roots were followed by five squares. Laurance asked whether the HP-41 makes the same mistakes for the y^x function above as the HP-67. I do not have an HP-41 readily available. My HP-11C gets the same correct ten digit answer as the rounded answer from the TI-59 for each of six problems posed by Laurance. Can one of our bi-linguals help us with an answer for the HP-41?

MU-17 CLARIFICATION - Don Graham reports that the documentation for the Romberg Integration program (MU-17) in the Math/Utilities module indicates that data registers R18 and beyond are used for data, the exact number being dependent on input parameters (and therefore accessed by indirect addressing). However, the listing for the program refers explicitly to R19 and R20 (see steps 047-048, 063,064, 071,072, 077-078, 086-087, and 173-174). It's not actually contradictory, but it could be a little misleading in certain cases.
FAST MODE ENTRY - Dejan Ristanovic writes: ...I still see many fast
mode programs using "load-and-go" methods to
initialize fast mode. I think that this method is very user un-
friendly since:

* Memory is completely erased.
* Partitioning is returned to start-up condition.
* One cannot use a normal mode for data entry using labels
and a fast mode for data processing without going through
an intermediate step of writing magnetic cards which can
be re-entered after fast mode initialization.
* Reading magnetic cards require extra steps, and includes
the unpleasant possibility of card reading errors.

Dejan then recommends the use of the method which involves place-
ment of a STF IND at the end of the current partition. This tech-
nique was discussed by Patrick Acosta in V6N8P3/4 and has been
used in several programs in TI PPC Notes; for example, in

* Jovan Puzovic's program for 1186 digits of pi (V8N1P21).
* Laurance Leeds' speedy factor finder program (V8N2P27).
* Peter Messer's fast mode exact factorial program (page 5
of this issue).

All three of those programs share a common characteristic. The
constant used in the fast mode entry sequence is chosen such that
program execution after fast mode entry begins at location 001.
The three programs also all use the 7 EE sequence after the
flash 1 has been obtained in the display after the STF IND
at the end of the partition.

(Editor's Note: There is more versatility built into this method
of fast mode entry. Patrick Acosta's discussion in V6N8P3/4 used
7 INV instead of 7 EE; and experimentation suggests that any
number key can be used in place of the 7, and any command for
which the second digit is a 2 can be used in place of the EE.
Furthermore, V6N8P3/4 describes the use of the ninth through
twelfth digits of the initialization constant to obtain a jump
to a location other than 001 at fast mode entry. That technique
is also illustrated in this issue on pages 15 and 25 of this issue.

EVEN MORE ON NUMERICAL PRECISION - The subject of the comparative
numerical precision of various
hand-held calculators was introduced in V8N2P3, Additional material
was presented in V8N3 on pages 13 through 16. Several readers re-
sponded with their own observations:

George W. Thomson: "...The neophyte will never understand what all
the shouting is about until he has a disaster. He routinely inverts
a matrix or solves a set of simultaneous equations or a differential
equation and he doesn't even get the right SIGN, much less the right
magnitude. The literature on "ill-conditioning" goes back about
150 years. So, a long word length is good insurance, even although
you never seem to need it. So, keep on shouting."
EVEN MORE ON NUMERICAL PRECISION (cont)

Bob Fruit: "...I think you are testing the wrong function to determine the precision of a computer. Not many people have need for high precision trigonometric functions. Maybe astronauts need them, but not many other people. From the people I know the lack of computer precision causes trouble most often when raising a number to a power. This shows up often when calculating amortization tables. Those of us accustomed to using the TI-59 don't even think about the precision of the raise to a power function since it often has more precision than the desired results. On a computer it often turns out that the result is only good to 4 or 5 significant digits. For a table covering 30 years that precision is nowhere near good enough. I would propose a test using the calculation of one dollar contributed to a savings account every month, with interest compounded every month for thirty years. The algebraic form for this test case (it is the solution to a geometric series) is:

$$S_n = \frac{(1 + i)^n - 1}{i}$$

To standardize the calculation an annual interest rate of 10% was used. The effective monthly interest rate is 10%/12. There are 360 monthly compounding periods in 30 years. I have calculated the results for this formula on the IBM PC in two ways. The first way is the straightforward application of the formula using the raise to the power function, and yields the answer

$2260.487924796093$

The second way is to avoid that function by squaring the value until the power is reached. Not only is this method faster than multiplying the number times itself 360 times (which would introduce a lot of rounding error), this is the least number of times a number can be multiplied to achieve the desired results:

$$x^{360} = x^{256} x^{64} x^{32} x^8$$

This method yields the answer

$2260.487924796093$ (identical with that above)

The IBM PC used double precision algebra to get those answers. Using a TRS-80 Model II at work for the same problem yielded:

$2260.5$ in single precision, and

$2260.502243041992$ in double precision, where

the built-in BASIC functions are single precision. FORTRAN for the Model II does have double precision functions--how many ways are we going to test? Straightforward mechanization of the problem on the TI-59 using the $y^x$ function yields an answer of (including the guard digits):

$2260.487924713$

Editor's Note: For comparison the sinking fund option of ML-19 yields the identical answer if one removes the Fix 2 display mode from the output. Solutions on other computers include:

- HP-11C 2260.487641
- CC-40 2260.4879241984
- Model 100 2260.4879247471
EXACT FACTORIALS IN FAST MODE - Peter Messer. V8N3 included two exact factorial programs, one for the TI-59 by Peter Messer and another for the TI-57 by Reginald van Genechten. The TI-57 program would find 34! in 2 minutes 30 seconds, while the TI-59 program required 3 minutes 15 seconds for the same calculations. Peter decided to rearrange his TI-59 program for increased speed, and then added fast mode using the Stflg-Ind-7-EE method. The fast mode version delivers 34! in just 1 minute 21 seconds, and delivers 100! in 10 minutes 30 seconds. The operating instructions are similar to those in V8N3P10, but with additional keystrokes for fast mode entry:

(1) Enter the program.

(2) Press A to initialize. See a zero in the display.

(3) To find n!:
   (a) Enter the first factor and press B. You may use either a 1 or a 2.
   (b) Enter the second factor and press C. You may use a 2 if you used a 1 in step 2.a., or a 3 if you used a 2 in step 2.a.
   (c) Enter n and press D. See a flashing 1 in the display. Ignore the flashing and press 7 and then press EE. The calculator will be in fast mode. You will be unable to interrupt operation with either RST or R/S. A "-0" in the display signals the end of the calculations.
   (d) Press E to display the highest order block. Press R/S to see each remaining block in order. The number of trailing zeroes is displayed as a negative integer.

(4) To find n!/m!, first initialize as in step 2 above.
   (a) Enter m+1 and press B.
   (b) Enter m+2 and press C.
   (c) Enter n and press D. See a flashing 1 in the display. Proceed as in steps 3.c. and 3.d. above.

As with the program on V8N3P10:
* All blocks are ten digits.
* The highest factorial is 461!.

If you tire of all this speed and want to see how fast the program will run in normal mode simply proceed as above, but in response to the flashing 1, press CLR and see a steady zero. Press RST followed by two R/S's. The normal mode run time for 34! will be about 2 minutes 46 seconds. Tests of other factorials in normal and fast mode will show that the fast mode runs slightly better than two times faster than normal mode.

You might also want to experiment with other responses to the flashing 1 which signals readiness to enter fast mode. My tests show that you may use any number key (0 through 9) in place of the 7, and may use any function key which has a code which has a 2 in the second digit (STO, x t, INV or B) in place of the EE.

The program appears on the next page.
EXACT FACTORIALS IN FAST MODE (cont)

Program Listing:

000 91 R/S  027 65  x  054 45 45  080 76 LBL  106 61 GTO  133 00 00
001 01 1  028 43 RCL  055 76 LBL  081 43 C  107 00 00  134 01 01
002 44 SUM  029 99 99  056 41 8  082 42 RCL  108 87 87  135 30 30
003 98 98  030 95  =  057 47 CMS  083 98 99  109 69 DP  136 01 1
004 42 STD  031 63 EX +  058 01 1  084 91 R/S  110 30 30  137 44 SUM
005 00 00  032 00 0  059 00 0  085 76 LBL  111 42 RCL  138 96 96
006 73 RC  033 49 DP  060 49 DP  086 15 E  112 00 00  139 61 GTO
007 00 00  034 20 20  061 17 17  087 43 RCL  113 67 EQ  140 00 00
008 65 x  035 61 GTO  062 01 1  088 97 97  114 01 01  141 04 04
009 43 RCL  036 00 0  063 52 EE  089 42 STD  115 25 26  142 00 0
010 98 98  037 06 0  064 01 1  090 00 0  116 42 STD  143 00 0
011 95 +  038 68 NDIV  065 00 0  091 73 RC  117 97 97  144 00 0
012 43 RCL  039 76 LBL  066 42 STD  092 00 0  118 97 DSZ  145 02 0
013 94 94  040 14 D  067 95 95  093 91 R/S  119 95 95  146 94 99
014 99 INT  041 25 75  -  068 25 CLR  094 97 DSZ  120 00 0  147 01 1
015 99 -  042 43 RCL  069 91 R/S  095 00 0  121 01 01  148 02 2
016 67 EQ  043 98 98  070 76 LBL  096 00 0  122 25 CLR  149 94 +/–
017 01 01  044 85  +  071 15 B  097 91 91  123 94 +/–  150 85 +
018 09 09  045 01 1  072 65 DP  098 43 RCL  124 81 RST  151 01 1
019 95 -  046 22 INV  073 20 20  099 96 96  125 42 RCL  152 95 =
020 43 RCL  047 44 SUM  074 72 ST +  100 65 x  126 97 97  153 22 INV
021 99 99  048 98 98  075 00 0  101 01 1  127 42 STD  154 52 EE
022 95 95  049 95  =  076 91 R/S  102 00 0  128 00 0  155 58 FIX
023 42 STD  050 42 STD  077 61 GTO  103 95 =  129 00 0  156 00 0
024 94 94  051 95 95  078 00 0  104 94 +/–  130 63 EX +  157 60 ENG
025 22 INV  052 61 GTO  079 72 72  105 91 R/S  131 00 0  158 86 STF
026 99 INT  053 01 01

ZERO AS A CALLABLE LABEL - Myer Boland. Myer suggests adding the availability of zero as an additional label to the list of quirks in V8N3P3. He notes that you can address LBL Ø by hand or from a program by using IND A (or any other user defined key A through E'). For example with the following routines in user memory:

LBL C 100 RTN
LBL Ø 200 RTN
LBL A x IND A = RTN
LBL B x C = RTN

If you put a 5 in the display and press A you will get 1000. With a 5 in the display, pressing B will return 500. If you have a soft 5 (no decimal point) in the display, pressing C will yield 5100; but with a hard 5. (with a decimal point) pressing C will yield 100. With a soft 5 in the display, pressing IND A will yield 5200; but with a hard 5. in the display, pressing IND A (or IND E' or IND followed by any other user defined key) will yield 200.

The effect is as if there is one more user defined label.

Editor's Note: Myer didn't recall where he had seen this technique. My research shows that the effect was reported in V4N3P4 of 52 Notes by Maurice Swinnen, who stated that he had found the technique in the newsletter DISPLAY. Maurice reported that zero was the only numeral that worked this way, and that somehow data register 00 gets into the act. For example, if you had the number 123.55 in data register 00 and press the keys 5 2nd IND A with the routines above in the calculator and in TRACE mode then the integer portion of the contents of R00, or 123 will be printed at the right hand side on the line before 5200. RTN is printed. If you try the technique with no data registers (Ø Op 17) you get a flashing display.
EDUCALC MAIL STORE - Educalc sells software, accessories, books and supplies for portable computing. Devices covered range from the TI-59 and HP-41 to the CC-40 and the HP-75. The shipping and handling charge is only $1.00 no matter how large the order. Order from EduCALC Mail Store 27953 Cabot Road Laguna Niguel CA 92677

Free Catalog

That is a slightly different address than has been listed previously. If you order, mention TI PPC Notes. Items of interest to TI-59 users:

Stock No. A-514 - Leather Calculator Case for the TI-58/59. These hard leather cases are made of high quality, 7 to 8 ounce cowhide. They are perfect for outdoor applications such as surveying, construction or a machine shop environment. A belt loop is available on the back of each case. Light tan in color. $29.95.

Stock No. P-150 - Designed to fill a very real gap in TI's manual documentation. Also, appendices give useful information on Registers versus Program Memory and Pseudo-Instruction Codes. Gives procedures for interfacing ROM programs in user memory. $12.00. (Editor's Note: This is listed as the User Survival Guide for the TI-58/59 Master Library, but old timer's will recognize it as Fred Fish's book which has been previously mentioned in V6N6/7P1. Maurice is no longer providing these books for members.)

Stock No. I-159 - Problem Solving with the Programmable Calculator by Dunlop and Sigmund. This unique approach to problem-solving will develop your programming skills. Here are dozens of puzzles, games and simulations for a wide variety of applications in math and science. Each activity comes complete with detailed user instructions, flowcharts and/or keystroke listings, and solutions. 227 pages, soft bound. $10.95. (Editor's Note: This book was reviewed by Maurice Swinnen on V8N1P16).

Stock No. M-135 - Curve Fitting for Programmable Calculators (Second Edition) by William Kolb. Formulas, graphs, and sample problems for a huge assortment--38 in all--of different curves to be fitted to your data. Most of these are for one independent variable, but multiple linear regression is also covered, along with exotics like Hoerl Functions and Logistic Curves. The equations are designed for any programmable calculator; however, you may need to change register assignments if you have fewer than 100 data registers. HP-41 programs and barcodes are included for half of these curves. In addition some program listings are given for the HP-75 and the TI-59. 148 pages. Spiral bound. $13.95. (Editor's Note: I reviewed this book on V8N2P20).

Stock No. B-81 - Programmable Calculators: Business Applications by Aronofsky, Frame and Greyneolds. For the manager, analyst, or student who wants a more business-like and understandable treatment than the TI-58/59 manuals offer, this comprehensive introduction is a spin-off of an MBA course at SMU. 203 pages, softbound. $11.95.

Stock No. P-141 - Calculator Programming for Chemistry and the Life Sciences by Frank Clarke. This book gives students and lab scientists new approaches to experimental design and data interpretation with specific detailed examples. The four main topics are Molecular Formulas, Coordinate Transformations (for X-ray crystallography), Potentiometric Titrations, and Correlation Analysis (up to 5 variables). Programs are given for the TI-59 with printer. 226 pages. $26.50.
MODULE SELECTOR - V5N8P3 and V7N1/2P25 described module selectors which permit selection, either from the keyboard or under program control, of up to four modules. The devices are available from American Microproducts Inc., 705 North Bowser, Richardson TX 75080 or call them at (214)-238-1815. Joseph Thomas who has one of the devices writes:

A word of caution to those fellow programmers who use the automatic module selector. One sleepless night I decided to put my ams through its paces and found some interesting results. The instructions for accessing a module are

\[
\begin{align*}
X & \hspace{1em} 0p \hspace{1em} 04 \hspace{1em} 0p \hspace{1em} 06
\end{align*}
\]

where \( X \) is the module port and ranges from zero through three. This is not the only way to access the modules. It seems that if the alphanumeric code for \( \Sigma \) (77 from the print code table) occurs anywhere in the program the ams senses it and transfers (switches) to the appropriate (or depending on the situation, inappropriate) module. That is, a legal command is

\[
\begin{align*}
X & \hspace{1em} 0p \hspace{1em} Y \hspace{1em} 0p \hspace{1em} Z
\end{align*}
\]

where \( X \) is the module port, \( Y \) is 1 through 4, and \( Z \) is 5. This might enable the user to print prompt statements such as

\[
\text{ACCESSING MODULE } X\Sigma
\]

with an \( 0p \hspace{1em} 05 \) command. But at the same time the user must be careful not to inadvertently transfer module connections during a critical point in his program. Consider some examples:

\[
\begin{align*}
13 & \hspace{1em} 77 \hspace{1em} 0p \hspace{1em} 04 \hspace{1em} 0p \hspace{1em} 06
\end{align*}
\]

prints an \( A\Sigma \) and selects port 3, while

\[
\begin{align*}
19 & \hspace{1em} 77 \hspace{1em} 0p \hspace{1em} 01 \hspace{1em} 0 \hspace{1em} 77 \hspace{1em} 0p \hspace{1em} 03 \hspace{1em} 0p \hspace{1em} 05
\end{align*}
\]

prints \( 8\Sigma \) and selects port 1 as controlled by the \( 0p \hspace{1em} 01 \) command. Furthermore, suppose that some part of a user program contains a \( \Sigma+ \) command. While listing a program the module selector will switch to port 1 no matter where it is originally located, it is not the key code 78 which initiates the switching process, but seems to be the \( \Sigma \) character which has a print code 77.

Editor's Note: I do not have a module selector, so I am unable to verify these effects. Readers who use the ams are invited to comment.

OTHER TI-59 PERIPHERALS - The following page describes two other peripherals which are available for use with the TI-59. I do not own, nor have I used, either of these devices. Contact the manufacturers for more information. Readers with experience with either device are invited to comment.

SNYDER's CHALLENGE - Ralph Snyder's Challenge (V8N1P18) must have been too hard. Ralph agrees that it may have been a little cryptic, and he should have provided a definition of the symbol \( v \) in his formula (5). It's a standard symbol for \((1+i)^{-1}\). The single response chose to tackle the problem from an entirely different aspect, not consistent with Ralph's guidelines. The bottom line is that all bets are off, and no prize money was paid. Case closed.
HUGE STORAGE CAPACITY - PROGRAM OR DATA

*****SAVE UP TO $100.00*****

This coupon is good until May 30, 1983 for a five percent discount on Hand Held Products, Inc. EPROM Memory Storage Units ordered through Educade and is limited to a maximum value of $1,000.00 on a single purchase of hardware. Limit one coupon per company or person per purchase during the discount period.

*****MAIL BEFORE MAY 30, 1983*****

FOR HP-41C/CV USERS AND NOW AVAILABLE FOR TI 58C/59 USERS

Storage capacities of 4K, 8K, 16K, and even 32K bytes are possible for the HP-41 and up to 10K bytes for the TI 58/59. This added memory is accomplished through the use of EPROM (Erasable Programmable Read Only Memory) technology and emulates the ROM (Read Only Memory) capacity of your hand held computer. The advantages are the following:

1. Significantly larger memory capacity than is available through a custom ROM module
2. The speed of delivery and programming time, typically 2 weeks delivery versus months for custom ROM
3. EPROM's are erasable and producible by the use of the EPROM programmer, and can be programmed with absolute precision and accuracy in the event that errors are detected.
4. Complete system support for in-house design
5. Low power circuit, no battery required

The expanded storage space can be used to maintain large reference tables or program libraries or both. It's ideal for field sales and service, in-house engineering or professional software development organizations. Relatively low volume projects become cost effective. EPROM modules plug directly into the HP-41 (any port) or the TI 58/59 (using an adapter) with no modification. All RAM space remains available for your use.

TI 58/59 APPLICATION

The most recent development by Hand Held Products has been an EPROM package for the TI 58/59. It consists of an interface adapter (HHP-IA) which plugs into the back of the TI 58/59 and a plug in-program module (HHP-PM). It offers twice as much storage (10K bytes) as a custom ROM and is small enough to fit into the PC-100 printer storage compartment (2.5" x 2.25"). No calculus or printer modification is required. Private or Non-private mode may be used.

Support for the TI 58/59 includes the following:

1. 10K RAM/ROM emulator board for the IBM-PC to support program testing of application libraries without the necessity of burning EPROMs.
2. Source programs can be created by using the User Language Translator (ULT) software system. ULT accepts source code created by the programmer using a standard word processor and compiles it into native TI 58/59 binary code while producing a detailed source listing with user comments for evaluation. The binary program modules are then loaded to RAM/ROM for testing and for EPROM storage using the ROM Build System (RBS). Several other methods are available to create EPROMS for the HP-41. For example, programs stored on magnetic cards or HP Development system disc can be converted to EPROM storage through Hand Held Products, OEM application support services include EPROM production, custom labeling, and application development assistance.

For additional information, Call Hand Held Products. (704)377-3841

HP-41 APPLICATION

Two memory units are available, 16K and 32K bytes. Physically they are the same size (3.6" x 5.8") about the size of your HP-41. Since its introduction in November 1981 by Hand Held Products, Inc., the system support has included the following:

1. A User Code Compiler (UCC) which is a full function cross-compiler for the HP-41. Working with a text editor, the programmer develops source code which is then compiled with all necessary comments/notes/definitions embedded as simply as drafting a letter on a word processor. When ready to compile, the programmer directs UCC to cross-compile the program and produce (A) a fully documented source code listing; (B) complete cross reference index of register, flag and label utilization and optionally; (C) a printer code image file.
2. A ROM Development System (RDS) handles all of the functions necessary to build, interlink, catalog and burn the EPROM chips with minimum technical input. All software tools are designed for the CP/M operating system and are fully operational on the Osborne I and IBM-PC with baby blue.

PRICE LIST

<table>
<thead>
<tr>
<th>Model</th>
<th>Price</th>
</tr>
</thead>
<tbody>
<tr>
<td>HHP-16K</td>
<td>$241.00</td>
</tr>
<tr>
<td>HHP-32K</td>
<td>$495.00</td>
</tr>
<tr>
<td>HHP-IA (TIA)</td>
<td>$199.00</td>
</tr>
<tr>
<td>HHP-PM (TIPM)</td>
<td>$213.00</td>
</tr>
<tr>
<td>HHP-ER (T1 Emulator BD)</td>
<td>$149.00</td>
</tr>
</tbody>
</table>

One Time Charge

<table>
<thead>
<tr>
<th>Specification</th>
<th>Price</th>
</tr>
</thead>
<tbody>
<tr>
<td>HHP-ULT (T1 User Language Trans)</td>
<td>$995.00 (each)</td>
</tr>
<tr>
<td>HHP-RBS (TI ROM Build System)</td>
<td>$395.00 (each)</td>
</tr>
<tr>
<td>HHP-UCC (HP User Code Comp)</td>
<td>$395.00 (each)</td>
</tr>
<tr>
<td>HHP-RDS (HP ROM Develop Syst)</td>
<td>$295.00 (each)</td>
</tr>
</tbody>
</table>

SAC DC-59 free-standing Model 59 or 52 interfaces consist of two printed circuit boards and a power supply housed in a small, attractive cabinet, for use with a Texas Instruments' SR-59 or SR-52 hand-held programmable calculators. One cable is supplied which connects the interface to the calculator. The user's calculator is modified by SAC to accept this cable.

Input from the user's device to the interface is via a 50 pin connector. A 50 pin cable transition card is supplied for the user to construct a cable between the device to be monitored and the DC-59. This innovative approach expands the use of the calculator by providing a means of monitoring keyboard closures on the calculator. The interface is designed to connect any standard output devices such as digital voltmeters, A/D converters, event counters, frequency meters, and strain gages to the programmable calculators.

The combination of TI's vast library of programs and the speed and convenience of direct automatic data input to the calculator can significantly increase the utility of these electronic tools. SAC's free standing interfaces represent the first non-keyboard entry devices for use with handheld programmable calculators. The DC-59 allows the user to create his own solutions to problems by programming the TI calculator.
CC-40 GRAPHICS - Maurice Swinnen. These whimsical little programs illustrate the use of the CHAR command (page 5-15 of the CC-40 User's Guide to generate user defined characters. The characters are then placed in sequence to provide an illusion of motion. The first program moves a character across the screen while performing the old "jumping jack" exercise. The second program uses seven characters (all that are allowed) to generate a "soccer" figure which moves the ball back and forth across the screen.

JUMPING JACK

100 CALL CHR(0,"0E0E150E04040A11"):CALL CHR(1,"0E0E040E150E0404")
110 FOR I=1 TO 31:FOR J=0 TO 1:DISPLAY AT(I),CHR(J):PAUSE .3
120 NEXT J:NEXT I
130 FOR K=31 TO 1 STEP -1:FOR L=0 TO 1:DISPLAY AT(K),CHR(L)
140 NEXT L:NEXT K
150 GOTO 110

SOCCER

100 CALL CHR(0,"0E0E150E04040A11"):CALL CHR(1,"001A1A180028509")
110 CALL CHR(2,"000100030051313191") CALL CHR(3,"0001000301312901")
120 CALL CHR(4,"150E04040A110E0E"):CALL CHR(5,"001A141814131313")
130 CALL CHR(6,"00000081000891412")
140 FOR A=10 TO 21:FOR B=0 TO 6:DISPLAY AT(A),CHR(B):PAUSE .1
150 NEXT B:NEXT A:PAUSE .5
160 FOR A=21 TO 10 STEP -1 FOR B=6 TO 0 STEP -1
170 DISPLAY AT(A),CHR(B):PAUSE .1
180 NEXT B:NEXT A:PAUSE .5 GOTO 140

TI-99/4A SOFTWARE DIRECTORY - This directory listing over 1200 software packages for the TI-99/4A contains both programs available from Texas Instruments and those developed by and available from third party authors. The directory can be purchased directly from TI at a cost of $5.95 plus $2.00 for shipping and handling. The directory is currently being updated for release in the fourth quarter of 1983 and will then include listings for more than 2000 software packaged.

Former PPX Exchange members will recognize the format of the index which includes cross-references by category, author and keyword with individual listings including a short abstract.

To obtain a copy send a check or money order to: Texas Instruments Consumer Relations P.O. Box 53 Lubbock, Texas 79408

Copies may also be obtained by calling 1-800-858-4565.
ACCURACY OF THE CC-40 SINE AND COSINE FUNCTIONS - Palmer Hanson

V8N3P18/19 presented George Thomson's analysis of the accuracy of the sine and cosine functions of the TI-58/59. The CC-40 calculates the trigonometric functions to fourteen places and might be expected to yield more accurate results than the TI-59. Examination of the CC-40 sine function for one degree increments from 0 through 90 degrees shows the following errors:

**CC-40 Sine Errors**

Mean Error = 8.2E-14
RMS Error = 18.3E-14
Peak Error = 59E-14

The peak error of 59E-14 occurs at 79 degrees. For a graphic comparison with the TI-59 results the following plots show the TI-59 errors without compensation (same as the top plot on V8N3P19) and the CC-40 errors using the same scale for both plots:

**TI-59 Errors without any compensation**

Mean Error = 1.6E-13
RMS Error = 6.8E-13
Peak Error = 17E-13

**CC-40 Errors**

Mean Error = 0.8E-13
RMS Error = 1.8E-13
Peak Error = 5.9E-13

Over the examined range the CC-40 results are nearly four times more accurate than the TI-59. As with the TI-59 the cosine function is less accurate over the same range. The mean cosine error is 5.8E-14, but the RMS cosine error is 37.1E-14, nearly twice that of the sine.
PROMPTING ON THE CC-40 - In V7N7/8P24 Maurice Swinnen described a multi-language capability built into the TI-88 such that prompting could be in English, German or French. The CC-40 provides an extended multi-language prompting capability through the use of the CALL SETLANG(n) command. The assigned language codes are:

- 0  English
- 1  German
- 2  French
- 3  Italian
- 4  Dutch
- 5  Swedish
- 6  Spanish

For n = 1 the system messages and error messages are in German. For example, the response to the incorrect entry sequence ATN( ENTER is "ungleiche Klammern". For any other value of n the system messages and error messages are in English. In response to the incorrect sequence ATN( ENTER the English response is "Unmatched parenthesis". This output of error messages in text is one of the attractive features of the CC-40. The user need not memorize error codes or translation tables to avoid frequent reference to the manual. The manual does provide extended discussion of each error message.

For programs from a Solid State Software™ module the prompts and messages from the module may be in any of the languages if supported by the particular module. My Mathematics module supports English, German and French. For the Prime Factors program the various messages are:

<table>
<thead>
<tr>
<th>English</th>
<th>German</th>
<th>French</th>
</tr>
</thead>
<tbody>
<tr>
<td>PRIME FACTORS</td>
<td>PRIMZAHLN</td>
<td>FACTEURS PREMIERS</td>
</tr>
<tr>
<td>Use Printer?</td>
<td>Drucker benutzen?</td>
<td>Utilisation d'une Imprimante?</td>
</tr>
<tr>
<td>Enter # To Be Factored:</td>
<td>-&gt; Zahl:</td>
<td>-&gt; Nb a Decomposer:</td>
</tr>
<tr>
<td>Exit Program?</td>
<td>Programm verlassen?</td>
<td>Fin du Programme?</td>
</tr>
</tbody>
</table>

The responses to the questions asking for yes/no answers are Y or N in English, J or N in German, and 0 or N in French. I have not found any information in the manual for the Mathematics module which would tell me which languages are supported. Language codes 3 through 6 result in English messages for that module.

PRIME FACTORS WITH THE CC-40 MATHEMATICS MODULE - The speed of the prime factors program in the CC-40 Mathematics module is disappointing, about ten to forty percent faster than the fastest program for the TI-59, but substantially slower than some programs for the HP-41. Representative speeds for some of the standard problems are:

<table>
<thead>
<tr>
<th>Program/machine</th>
<th>111111111111</th>
<th>103569859</th>
<th>987654321</th>
<th>9999999967</th>
</tr>
</thead>
<tbody>
<tr>
<td>CC-40</td>
<td>11 sec</td>
<td>32 sec</td>
<td>41 sec</td>
<td>1 hr 55 min</td>
</tr>
<tr>
<td>Fast Mode Modulo 210</td>
<td>45 sec</td>
<td>58 sec</td>
<td>2 hr 8 min</td>
<td></td>
</tr>
<tr>
<td>Leeds FM (V8N2P26)</td>
<td>17 sec</td>
<td>46 sec</td>
<td>61 sec</td>
<td>2 hr 31 min</td>
</tr>
<tr>
<td>Acosta FM 58C</td>
<td>27 sec</td>
<td>61 sec</td>
<td>79 sec</td>
<td>3 hr 6 min</td>
</tr>
<tr>
<td>M/U Module - 59</td>
<td>43 sec</td>
<td>163 sec</td>
<td>215 sec</td>
<td></td>
</tr>
</tbody>
</table>
PRIME FACTORS ON THE CC-40 (cont)

For large primes such as 9999999967 the execution speed of the CC-40 Mathematics module program is about 0.069/N. Page 19 of the July 1981 issue of the PPC Calculator Journal reported a speed of 0.035/N for the HP-41C; but the HP-41C cannot maintain that speed for input integers of more than ten digits.

The CC-40 Mathematics module program has other deficiencies:

* The program stops as each factor is found. A better technique is to store the factors as they are found and continue the search until all factors are found. This minimizes operator attention. A simple additional routine provides for recall of the factors. The technique was illustrated in Laurence Leeds' Speedy Factor Finder in V8N2P26.

* Multiplicity of factors is not indicated. Indication of the multiplicity using a technique such as that devised by George Vogel in his prime factor program in the article "It Pays to Analyze Your Problem" in the January/February 1981 issue of PPEX Exchange would be preferred. As George said in that article "Peeemeal presentation of results is slow and inconvenient (try factoring 7, 247, 757, 312). Yet it is not difficult to make the program count the number of times each prime factor occurs, and output the count." George used a decimal point notation where the number after the decimal point indicated the multiplicity. For the number mentioned the output would be 2.28 and 3.03 meaning $2^{28} \times 3^3$.

* Although the CC-40 program can factor input integers of up to twelve digits, it does not provide an ability to recall the input integer correctly for more than ten digits. For example, factor the number 111,111,111,111. You will obtain the correct solution on the first pass, and an "N" in response to the prompt "Exit Program?" will bring the input value back to the display but in exponential notation 1.1111111E+11. If you run with that value you will get the factors for 111,111,100,000!

We will have to wait until someone finds out how to download the programs in the modules before we can know if there will be ways to use segments of the module programs, say in the manner in which we can enter the library modules of the TI-59 with the sequence Pgm-XX-SBR-nnn.

CC-40 PERIPHERALS - Peripherals for the CC-40 include a Printer/Plotter, a Wafertape™ Digital Tape Drive, and an RS-232 interface. Currently, none of these are available in the Tampa Bay area. The devices are listed in the Sears Fall/Winter 1983 catalog (page 869), in the Educalc Mail Store catalog issue 16 (page 34), and in the Elek-Tek catalog Volume VI (page 17). Inquiries indicate the peripherals will be available in early fall.

Although the Manufacturer's Suggested Retail Price for the CC-40 is $249.95, catalog prices range from $199.99 (Sears) through $189.95 (Educalc) to $189.00 (Elek-Tek). I have seen the CC-40 at local discount houses for as low as $179.95. The CC-40 packs a lot of "bang for the buck" at those prices.
NEWCOMER'S SPECIALS - These are additional copies of old issues that Maurice Swinnen forwarded for disposition. As with V6N9/10 which was described in V8N3P23, these are single issues which are not parts of complete one year sets.

V7N4/5, 1982 - This is another of Maurice's double issues which includes thirty pages. There are eleven complete programs; two, Bill Carpenter's Loan Repayment and Ralph Snyder's Rate of Return are of interest to those who work in business and finance. Bob Fruit's pi program runs in fast mode and finds 460 digits in six hours eighteen minutes--the best for the TI-59 until the programs from Yugoslavia and France were published this year. My article on Powers of -1 examined that exercise from nearly every aspect. Charlie Williamson's 13 Digit Printer and Sidney Hack's 100 Random Integer programs were responses to earlier challenges. A copy of the index is at the right. Send two dollars (no checks, please) for a copy.

V7N6, 1982 - This single issue contains eighteen pages, with four programs and thirteen (count them) solutions to a puzzle proposed in V7N4/5. Dejan Ristanovic examines the responses of the TI-59 to twelve different hexadecimal codes and describes a way to place a TI-57 in a mode that minimizes energy use, effectively converting it to a TI-57C. George Thomson makes the observation that the correlation coefficient is a poor measure of the quality of a fit, and proposes a concise method for finding the standard error for a straight line fit using values already accumulated with the linear regression functions. A copy of the index is at the right. Send one dollar for a copy.

MORE PI FROM BASIC - V8N3P25/26 discussed finding pi in BASIC and gave several examples. More examples include:

<table>
<thead>
<tr>
<th>From 4*ATN(1)</th>
<th>From 2*ATN(N)</th>
</tr>
</thead>
<tbody>
<tr>
<td>AMS-55 Reference</td>
<td>3.1415 92653 58979</td>
</tr>
<tr>
<td>IBM PC Single Precision</td>
<td>3.1415 93</td>
</tr>
<tr>
<td>Double Precision</td>
<td>3.1415 92979 43115 2</td>
</tr>
<tr>
<td>HP-75</td>
<td>3.1415 92653 59</td>
</tr>
</tbody>
</table>
13 DIGIT MODULO 30 SPEEDY FACTOR FINDER - Palmer Hanson. The original version of this program used the h-12 method for fast mode entry, with all of the attendant button pushing (PPX 398278). This revised version uses the STF IND at the end of the current partition technique (see page 3 of this issue). Previously published programs using the technique have all had a jump to location 001 at fast mode entry, a necessary feature associated with the initialization constant used. Use of the same technique with this programming would have caused substantial reprogramming effort. Reprogramming was minimized by using a jump to another location through use of another initialization constant. The selection of the appropriate constant was defined by Patrick Acosta in V6N8P3; when accessing fast mode the calculator will jump to address 8*(WXY) + Z + 1 where W, X, Y, and Z are the ninth through the twelfth digits. To use large segments of the existing h-12 type program as is needed to jump to program location 230. Therefore, the ninth through twelfth digits needed to be 0235 yielding an address of (28*8 + 5 + 1 = 230). A 2 is needed in the thirteenth digit for fast mode entry, and a one in the first digit avoids any flag setting problems. Therefore, the initialization constant selected was 1.000000002852 which is generated at locations 462 through 474 of the program.

The typical algorithm used for factor finding has been of the form

\[ \text{RCL 01 DIV RCL 02 = INV INT} \]

The integer to be factored is in R01. The factor to be tested is in R02. If the result is zero then the value in R02 is a factor of the value in R01. This algorithm permits factoring input integers of up to 12 digits. The truncation characteristics of the guard digits can be used to obtain a thirteen digit capability. While the ten digit display is obtained by rounding from the three guard digits as described on pages C-1 and V-5 of Personal Programming, the thirteen digits in the display register seem to be obtained in a truncated, not a rounded format. If we fill the display register with a thirteen digit integer and divide by a small integer such that the integer part of the quotient is still a thirteen digit number then a "built-in" integer function has been performed. We can use this characteristic to do modulo arithmetic to find prime factors with the program sequence

\[ \text{RCL 01 - (CE DIV RCL 02)} \times \text{RCL 02 =} \]

Again, if the result is zero the value in R02 is a factor of the value in R01. To cover the cases where the division yields a quotient with less than thirteen digits to the left of the decimal point we add an integer function to the sequence, yielding

\[ \text{RCL 01 - (CE DIV RCL 02)INT x RCL 02 =} \]

where the double integer function which occurs when the quotient has thirteen digits to the left of the decimal point is an idiosyncrasy of the algorithm which does no harm.

The PRT command prints what is in the display, not what is in the display register. Thus, if the integer 111,111,111,111 had been synthesized in the display register, say with a key sequence such as 1111111111 x 100 + 11 = then the display will show 1,1111111111, and that is what would be printed with a PRT command. The actual number could be recovered by the reverse of the entry process. An example appears in the Leeds Factor Finder program on V6N2P26/27.
13 DIGIT FACTOR FINDER - (cont)

This program prints all the digits of the input integer on one line by using a conversion algorithm similar to that originally developed by Robert Snow (VSN1P2). Integers of ten digits or less are printed using the PRT command. Integers of 11, 12, or 13 digits are printed using the routine from locations 250 through 315. That routine is used both for the input integer and for a final factor which is greater than ten digits. The use of indirect addresses is required to permit the converter to work in fast mode where subroutines are not allowed. The return addresses are set up before the converter routine is called.

The particular sequence of test integers in locations 001 through 167 was selected to provide an easy calculation of the return addresses from the multiplicity routine in locations 340 through 403.

Other features of the program include:

* Multiplicity of factors is indicated using the Vogel technique which has been discussed on page 13 of this issue.

* Factors are stored for later recall. Thus, when operating without a printer the program need not stop each time a factor is found.

Operating Instructions:

(1) Enter the program.

(2) Place the integer to be factored in the display register and press A. After a short time a flashing "1" will appear in the display. Do not clear it, but press 7 and then EE. The input integer will be printed followed by the factors with their multiplicity as illustrated at the right. A flashing "1" indicates that the solution is complete.

<table>
<thead>
<tr>
<th>111111111111</th>
<th>111111111111</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.01</td>
<td>7.01</td>
</tr>
<tr>
<td>11.01</td>
<td>13.01</td>
</tr>
<tr>
<td>37.01</td>
<td>101.01</td>
</tr>
<tr>
<td>9901.01</td>
<td>7247757312.</td>
</tr>
<tr>
<td></td>
<td>2.28</td>
</tr>
<tr>
<td></td>
<td>3.03</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>8074195354368</th>
<th>8074195354368</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.08</td>
<td>3.06</td>
</tr>
<tr>
<td>11.01</td>
<td>13.02</td>
</tr>
<tr>
<td>17.01</td>
<td>37.02</td>
</tr>
</tbody>
</table>

(3) To recall the input integer and the factors with their multiplicities, press B. The input integer will appear in the display. Press R/S as many times as needed to see the factors. A flashing "1" indicates that there are no more factors. This mode does not provide a printout since it is provided primarily for use without a printer.

(4) To test additional input integers, return to step (2) above.

Execution Times:

Execution times are somewhat longer than for the twelve digit factor finders due to the more complex test algorithm. The factors of 987654321 are found in about one minute thirty-three seconds. For large input integers the execution time is about 0.137√N. After somewhat over 120 hours the program will declare 9,999,999,999,971 to be prime. Until the advent of the Science et Vie pi programs that was a long period.
IMPEDEACE MATCHING, ATTENUATING T-PAD CIRCUIT.- Maurice Swinnen. This program allows you to compute the three resistors necessary to form the t-pad circuit, with as inputs the input and output impedances and the attenuation in dB. As an added bonus, the program also computes the minimum attenuation in dB necessary to assure positive values for all three resistors and thus provide a realizable t-pad attenuator. To my knowledge, this is the first program to do that. Other programs, although computing the three values correctly, require you arrive at realistic values by a time-consuming "trial and error" method.

\[ \alpha_{\text{min}} \text{ (the minimum attenuation in dB) is determined as follows:} \]
\[ R_1 = Z_1 \left( \frac{N+1}{N-1} \right) - R_3 \text{ and } R_2 = Z_2 \left( \frac{N+1}{N-1} \right) - R_3 \text{ for certain values of } R_3 = \frac{2V_{\text{in}}Z_1}{N-1} \]

\( N = 10^\alpha \), \( R_1 \) and \( R_2 \) can be negative. That is, for a given \( Z_{\text{in}} \) and \( Z_{\text{out}} \) there is a minimum \( \alpha \), i.e. \( \alpha_{\text{min}} \), below which \( R_1 \) and/or \( R_2 \) are negative.

\( \alpha_{\text{min}} \) is determined by solving the above equation for \( R_1 = 0 \), \( R_2 = 0 \), using the equation for \( R_3 \) which yields:
\[ \alpha_{\text{min}} = 20 \log_{10} \left( 2\pi + \sqrt{4\pi(Z-1)} \right) \text{ dB for } Z > 1, \]
where \( Z = Z_1/Z_2 \) or \( Z_2/Z_1 \).

USER INSTRUCTIONS: 1. To draw the circuit of a t-pad attenuator, press 2nd E'. For calculator use only, LBL E' may be omitted from the listing altogether without affecting the computation part of the program.
2. Enter attenuation required in dB and press A.
3. Enter the input impedance required, \( Z_{\text{in}} \), and press B.
4. Enter the output impedance required, \( Z_{\text{out}} \), and press C.
5. Compute the three resistance values in ohms: press D. When the printer is attached, values are printed with suitable descriptors. When the calculator only is used, the first value appearing in the display is \( R_3 \). Press R/S to obtain \( R_1 \). Press R/S again to obtain \( R_2 \).
6. To obtain the minimum attenuation possible in dB, press E.
7. It is now possible to obtain new values for the three resistors, without re-entering \( Z_{\text{in}} \) and \( Z_{\text{out}} \), by simply entering a new attenuation value, using either the minimum value computed in 6, or any other value you enter.

NOTE: If you are curious about the derivations of the above equation of \( \alpha_{\text{min}} \), just send me SASE and I will send you a copy of that two-page hand-written "opus". Typing it all out would have constituted an enormous chore and I am not even sure the editor would have wanted it in this article.


EXAMPLE: Design a t-pad circuit with 10 Kohm input impedance and 250,000 ohm output impedance. Is an attenuation of 10 dB possible? If not, what is the lowest attenuation possible. Design for that attenuation.

Press 2nd E'. See drawing.

---
R1
R2
ZIN
ZOUT
---

Enter attenuation 10 and press A.
Enter input impedance 10000 and press B.
Enter output impedance 250000 and press C.
To compute \( R_1 \), \( R_2 \) and \( R_3 \), press D.
Note negative resistance for \( R_3 \).
Obtain minimum attenuation, press E.
Enter new attenuation 20 and press A.
Compute \( R_1 \), \( R_2 \) and \( R_3 \) again, press D.
**TI PPC NOTES**

**IMPEEDANCE MATCHING - ATTENUATING T-PAD CIRCUIT**  
*(cont)*

**Program Listing:**

|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|-----

A number of subscribers have indicated an interest in surveying. This book may be of interest. The abstract in Government Reports Announcements (GRA) states:

This study was done to provide the US Army with the basis for a standardized program package for use with hand-held programmable calculators. The programs are written and published in a format which is immediately useable with the Texas Instruments Programmable 59 Calculator, but which can easily be adapted for use with other programmable calculators. Other Department of Defense agencies may also find them valuable in training and operational efforts. The study includes a general overview of the history of the Topographical Engineers.


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

COMPLEX NUMBER ARITHMETIC - Maurice Swinnen called my attention to this article which appeared in the June 1983 issue of Microwaves & RF. The article "TI-59 Program Simplifies Complex-Number Arithmetic" is by Professor Darko Kajfez of the University of Mississippi. The article states:

The program described in this article is an alternative to the EE module. The routine enables users of the TI-59 calculator to perform the four basic arithmetic operations on complex numbers without the need for any additional modules. The program has 114 steps and requires only four registers (26 through 29) and four flags (1 through 4).

One of the benefits of the new program is that the instructions are easy to memorize, since the standard keys are used. For example, the "+" key is used for addition and the "1/X" key is used for inversion. Memory storage and recall are accomplished by means of the standard keys for these operations ("STO" and "RCL"). Thus, rules for operating the program are fairly simple, eliminating the "search-the-manual" syndrome that can occur with the EE module.

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

LIST PROCESSOR --- George Thomson's response to the list processor problem in V8N3P16 appears at the right. The top of the list is at RO2. Enter the number of items in the list and press A. As you enter each item for the list press E. The program requires about 38 seconds to process a 60 item list.

```
000 76 LBL  015 42 STD  030 69 DP
001 15 E    016 00 00  031 31 31
002 82 HIR  017 73 RG+ 032 82 HIR
003 08 08   018 01 01  033 19 18
004 03 3    019 59 DP  034 72 ST*
005 42 STD  020 31 31  035 01 01
006 01 01   021 72 ST* 036 32 XIT
007 43 RCL  022 01 01  037 92 RTN
008 02 02   023 02 2   038 76 LBL
009 32 XIT  024 44 SUM 039 11 A
010 82 HIR  025 01 01  040 47 CMS
011 17 17   026 37 DESZ 041 82 HIR
012 75 -    027 00 00  042 07 07
013 01 1    028 00 00  043 92 RTN
014 95 =    029 17 17
```
1287 DIGITS OF PI - translation by Maurice Swinnen. V8N3P8/9 presented a program from the French scientific journal "Science et Vie" which would calculate up to 1287 digits of pi. Pierre Flener had obtained permission for us to reprint the program. Maurice Swinnen has provided a translation of portions of the text, which has been reviewed and edited with the help of George Thomson and Bob Fruit. Palmer Hanson performed the final typing and editing, and hence must bear responsibility for any residual inaccuracy:

Large Numbers and Calculators: The Limits
by
Renaud de la Taille

(From "Science et Vie", 12/80, pages 54-57)

With the entry of the Japanese into the programmable calculator market, it has become difficult for amateurs to make a choice among the available calculators. Since we have used several models to handle the difficult art of calculating large numbers, we have formed more accurate opinions about the performance and capacity of these astonishing machines.

Presenting the calculation of large numbers on small machines in our past issue, we thought to have reached the limits of these machines. We had also showed their possibilities to be well superior to what one had believed up to then. In response to that article, our readers have proven to us that we were rather far off the mark, and that the mathematical limits of these programmable calculators were, on the average, at least two times greater than we had thought.

This leads us to discuss this subject again in order to complete the study started in the last issue, and to answer the numerous questions posed to us. It behooves us to note that calculators have seemingly three uses, often rather distinct from each other:

* The resolution of scientific or technical problems such as the computation of the profile of the teeth in gears, or the determination of the positions of the planets for a given date.

* Mathematical games: the calculator acts as a game partner which may ask questions, give responses, and win the game.

* The science of numbers: the machine handles a multitude of arithmetic problems such as the determination of prime factors of integers, the determination of the limit of a series, the evaluation term by term of a series, etc.

Our former article was primarily concerned with this last field, the computation of large numbers, that is, obtaining numeric values with the largest possible number of digits (that is, significant figures). This kind of calculation can be useful for all sorts of values: mathematical constants, logarithms, exponentials, roots, quotients, sums of series, and so on.

The entire art consists of finding the right formula in order to design a short and simple program, and to use the largest possible number of memory registers. As an example, we have previously discussed the calculation of the two most important mathematical constants, e and pi. We had arrived at more than 600 digits with a TI-59 and thought it to be rather difficult, even impossible, to go any further.
1287 DIGITS OF PI (cont)

Well, we were mistaken by a factor of two: on a TI-59 one can evaluate pi to 1287 digits, which exceeds by far the record announced at the round table conference at the Palais de la Decouverte (The Palace of Discovery) at Paris. But we must admit that we would have never arrived at this result without the help of our readers.

Especially, we have to mention M. Labat of Paris, M. Brombeck of Marckolsheim, M. Colmont of Brive-la-Gaillarde and M. Molinaro of Nantes. Thanks to their contributions we have been able to redesign the programs we presented in our last article and go a large step farther in the calculation of pi. The records are:

* 250 digits on the HP-67 by M. Molinaro.
* 576 digits on a TI-58 by M. Brombeck.
* 1287 digits on a TI-59 by M. Colmont.

On the other hand, the programs we have reworked for the HP-41C have allowed us to obtain 3600 digits; but the calculation requires the enormous time of four months.

In order to reach these large numbers of digits one has to start with the series

\[
\pi/2 = 1 + 1/3 + 1*2/3*5 + 1*2*3/3*5*7 + 1*2*3*4/3*5*7*9 + \ldots
\]

Thus \(\pi/2\) is equal to the sum of the series \(k!/1*3\ldots(2k + 1)\) from \(k = 0\) to \(k = x\). This series can easily be factored by Horner’s method, which gives

\[
\pi = ((...((2n/(2n+1) + 2)(n-1)/(2n-1) + 2)(n-2)/(2n-3) + 2)(n-3)/...)
\]

1/3 + 2.

This method avoids the addition of terms of the series to the preceding sums. The resulting program contains only a multiplication and a division.

One will easily recognize the multiplication and division loops in the programs presented. In the other parts, which were much shorter, \(n\) went to 2n+1, then to \(n-1\) for the next cycle after having added the 2.

Some of our correspondents redesigned the programs based on the same principle, but in a more subtle way, which permits reduction of execution time. The chief prize for it goes to M. Molinaro who succeeded in halving the execution time on the HP-67 by replacing the returns with labels with returns with GSB, and in this way increasing the number of available memory registers.

There are others who made the multiplication and division within the same loop further reducing execution time. Finally, one gains a lot of execution time using the HIR’s on the TI calculators. Let us remind you that these codes give access to internal registers reserved for the levels of parentheses, but are not accessible from the keyboard. A little trick is all that is needed: in order to code, say HIR 46, of which the machine code is 82 46, one simply keys in SUM 82 SUM 46, after which by means of BST and DEL one removes the two 44 codes from the SUM commands, and what remains is 82 46.
1287 DIGITS OF PI (cont)

The first digit of the second part of the HIR command indicates the operation to be performed. The second digit indicates which of 8 registers will be used. As an example, for register 6 HIR 06 = STO 6, HIR 16 = RCL 6, HIR 46 = PRD 6, HIR 56 = INV SUM 6, and HIR 66 = INV PRD 6.

While the program does have one level of parenthesis, we still have registers 3, 4, 5, 6, 7, and 8 available. It is through the use of these registers that M. Brombeck was able to arrive at 576 digits with a TI-58.

It should be well understood that the same procedures could be used in any arithmetic computation; the number pi, the most important of all mathematical constants, has simply served as an example. But one could as well calculate e which is still easier (there is no division whatsoever), a fifth root, the prime factors of a large number, etc.

What one looks for in such a problem is a short program and a large number of memory registers. Let us note that the ideal consists of writing a complete program, which has a data entry system without any superfluous manipulation, and an output of the results which is simple and direct. In this sense, in the programs which we publish, the only complete one is that for the HP-67; one has only to enter the number of digits one requires, from 10 to 250 in increments of 10. Key D provides an output of all the decimals without the possibility of forgetting any of them.

One could have made similar programs for the HP-41C, for the TI-58/59, and for the Sharp, which have selectable partitioning between program memory and data memory, but at the expense of some of the digits in the solution. We should also add that the large numbers obtained are not always easy to manipulate, and that one shouldn’t forget the leading or trailing zeros.

Only the HP-67 program collects the decimals of pi in the form 0.abcdef where one retains only the first five digits following the decimal point, even for numbers such as 0.00357 or 23040. ... On the TI-58/59 the numbers are in the form uvwxyz.abcdefg and one reads 6 digits in front of and 7 digits beyond the decimal point. If necessary, one should add leading zeros or trailing zeros as needed to make up the proper number of digits.

Editor’s Note: The remainder of the article has not been completely translated to date. It is primarily concerned with more detailed discussion of each calculator in terms of capability, price, etc. Only the portions that discuss the TI-58/59 are printed below.

The TI-58: For its price it is the one that offers the greatest possibilities. It has a very large number of registers, the memory can be partitioned, it has all the useful mathematical functions, and it has superb precision thanks to the 13 digits in each register. It also is very complete with respect to programming having all the tests needed, the control of loops, subroutines, symbolic addressing and direct addressing, user-defined keys, etc. It is also a subtle machine in which the capabilities surpass the instructions in the manual: it is thus possible to create loops (Dsk) with any register, not only registers 0 through 9, and one may use the registers reserved for parenthesis by using the HIR commands.
1287 DIGITS OF PI
(cont)

TI-58/59 Program Description: We made a program as short as possible in order to keep the largest number of data registers available for the digits. One can obtain 507 digits in 87 hours with the TI-58 and 1287 digits in 24 days with the TI-59. 1000 digits are reached in 15 days. (Editor’s Note: One must reduce the number of registers which will be used to support only the number of digits desired if one wishes to minimize execution time. The number of registers to use must be entered at program locations 125/126 in the LRN mode prior to running the program.) And, one must calculate the number of terms to use as a function of the number of digits desired, and enter that value at the start of the program. The number of terms must be greater than the number of digits divided by log 2.

The TI calculators are more delicate to program than the HP calculators, and one should pay careful attention to the order of the operations.

Editor’s Note: In another section you will see that I have been able to optimize the TI-59 program in two areas:

* Fast mode has been added using the STF IND technique such that execution time is reduced by about a factor of 0.56.

* A data entry system has been added using the portion of bank 1 from program locations 160 through 239. The number of digits desired is entered (from 1 to 1287 in increments of 1) and user defined key A is pressed. The program calculates and stores both the number of memory registers and the number of terms.

---

1287 DIGITS OF PI IN FAST MODE - Program Listing

```
000 31 R/S  040 01 1  080 55  +  120 00 00
001 81 RST  041 52 EE  081 82 HIR  121 43 RCL  161 00 00
002 69 DP  042 06 6  082 16 16  122 00 00  162 00 00
003 20 20  043 75  083 54  >  123 32 XLT  163 00 00
004 25 CLR  044 59 INT  084 59 INT  124 85  +  164 00 00
005 29 CP  045 65  >  085 65  >  125 82 HIR  165 00 00
006 82 HIR  046 32 XLT  086 32 XLT  126 10 18  166 00 00
007 16 16  047 01 1  087 82 HIR  127 67 EQ  167 00 00
008 66 PHU  048 95  =  088 16 16  128 60 01
009 67 EQ  049 65  >  089 65  >  129 36 36  169 00 00
010 00 00  050 01 1  090 32 XLT  130 65  >  170 00 00
011 73 RC+  051 63 EE  091 63 EX+  131 00 00  171 00 00
012 73 RC+  052 06 6  092 00 00  132 95  =  172 00 00
013 00 00  053 95  =  093 22 INV  133 61 GTO  173 00 00
014 32 XLT  054 74 SM*  094 59 INT  134 00 00  174 00 00
015 55 +  055 00 00  095 95  =  135 67 67  175 00 00
016 01 1  056 97 DZ2  096 65  >  136 03 3
017 52 EE  057 00 00  097 01 1  137 82 HIR  177 00 00
018 07 7  058 00 00  098 52 EE  138 56 56  178 00 00
019 95 +  059 12 12  099 07 7  139 02 2  179 00 00
020 73 RC+  060 72 HIR  100 72 HIR  139 02 2  179 00 00
021 22 INV  061 82 HIR  101 53 <  141 66 66  181 00 00
022 59 INT  062 46 46  102 24 CE  142 52 EE  182 00 00
023 64 XLT  064 82 HIR  104 82 HIR  144 44 SUM  184 00 00
024 85 22 INV  065 36 36  105 16 16  145 01 01  185 00 00
025 59 INT  066 32 XLT  106 54  >  146 61 GTO  186 00 00
026 75 22 INV  067 69 DP  107 59 INT  147 00 00  187 00 00
027 69 INT  069 65 »  108 65  »  148 04 04  188 00 00
028 59 INT  070 01 1  109 32 XLT  149 68 NDP  189 11 R
029 72 STF  070 01 1  110 82 HIR  150 68 NDP  190 42 STD  230 95 +
030 00 00  071 52 EE  111 16 16  151 68 NDP  191 00 00  231 01 1
031 85 +  072 06 6  112 95  +  152 01 1  192 00 00  232 95 +
032 32 XLT  073 85 +  113 32 XLT  153 00 0  193 69 DP  233 22 INV
033 59 INT  074 73 RC+  114 55  +  154 69 DP  194 17 17  234 32 EE
034 65 +  075 00 00  115 01 1  155 17 17  195 95 <  235 36 FIX
035 85 +  076 59 INT  116 32 EE  156 47 CMS  196 93  
036 16 16  077 75  =  117 07 7  157 61 GTO  197 43 RCL  237 60 DEC
037 95 =  078 53 <  118 95  +  158 00 00  198 00 00  238 60 STF
038 55  =  079 24 CE  119 74 SM*  159 02 02  199 55 +
```

---
1287 DIGITS OF PI IN FAST MODE - Palmer Hanson. I started the conversion of the Science et Vie program to fast mode as a demonstration of the ease of conversion when using the STF IND technique combined with Patrick Acosta's method for jumping to program locations other than 001. I soon found that there was adequate space to provide a data entry system which only requires that the number of digits be entered, and which also automatically selects the minimum number of registers as suggested by George Thomson on page 2 of this issue. Two penalties result:

* The output routine at locations 149 through 159 of the original program is eliminated. It is necessary to key in an output routine at the completion of the calculations.

* It is necessary to return to the start-up partitioning and re-enter the program for each new solution.

The program revision concept is to begin execution in that part of bank 1 which includes program steps 160 through 239, set partitioning 9-0p-17 for use by the STF IND fast mode entry at locations 238/239, calculate the number of terms (iterations) and the minimum number of data registers from the number of digits desired and store the values in the appropriate locations, and set up for fast mode entry. At fast mode entry the program jumps to program location 152, changes the partitioning to 10-0p-17, clears the data registers with a Cms, and jumps to location 002 to begin the normal Science et Vie calculations, but in fast mode. The changes required to the original program as listed on V8N3P9 are:

1. Place a R/S RST sequence at program locations 001/002. These commands provide the transparent fast mode exit as defined in V7N1/2P23.

2. Change the address for the EQ command at program locations 010/011 from 149 to 002. The transfer at the completion of the calculations is changed from going to the start of the output routine at location 149 (which will no longer exist) to going to the transparent fast mode exit sequence.

3. Change the 39 at locations 125/126 to HIR 18. The minimum number of registers needed will have been previously stored in hierarchy register 8 by the input routine.

4. Replace the output routine which had been at locations 149 to 159 with a routine which sets partitioning to 10-0p-17, clears the data memories, and transfers the program to location 002.

5. Add the input routine and fast mode initialization sequence at locations 186 through 239. Locations 195 through 208 calculate the number of terms using the formula defined in V8N3P8, and store the result in hierarchy register 6. Locations 209 through 221 calculate the minimum number of data registers required, and store the result in hierarchy register 8. Locations 222 through 239 set up for fast mode entry. The 187 at locations 222/223/224 provide the transfer address of 18*8 + 7 + 1 = 152 as defined by Patrick Acosta in V6N8P3.

The revised program listing, which should be recorded with partitioning 6-0p-17 (the turn-on partitioning), appears on the facing page at the end of the Science et Vie article translation (page 24).
1287 DIGITS OF PI IN FAST MODE - (cont)

Operating Instructions:

1. Enter the program.

2. Enter the number of digits desired and press A. After a short period a flashing "1" will appear in the display. Do not reset it, but press 7 and then EE at the end of the calculations the calculator will stop with a zero in the display.

3. The solution appears in data registers R01 through R99 as required. You may simply read these values out with sequential RCL statements from the keyboard, thirteen digits at a time, in the format with six digits to the left of the decimal point and seven to the left. You must add leading or trailing zeros to make up that many digits.

As one alternate method of readout you may key in the short seventeen step program at the right. The sample print-out is for an input asking for thirty digits.

A second alternative is to record banks 1 through 4 as required and use one of the thirteen digit print routines to print the results.

4. To run an additional solution, return to the startup partition, re-enter the program card, and go to step (2) above.

Sample Execution Times:

<table>
<thead>
<tr>
<th>Number of Places Wanted</th>
<th>Normal Mode from VSN3P9</th>
<th>Fast Mode</th>
</tr>
</thead>
<tbody>
<tr>
<td>20</td>
<td>52.5 minutes</td>
<td>7.3 minutes</td>
</tr>
<tr>
<td>100</td>
<td>4 h 33 m</td>
<td>2 h 10 m</td>
</tr>
<tr>
<td>500</td>
<td>3d 18h 12m</td>
<td>2d 1h 16m</td>
</tr>
<tr>
<td>1287</td>
<td>24.55 days</td>
<td>13.39 days</td>
</tr>
</tbody>
</table>

The great improvement in speed for the 20 digit case is due to the use of only two data registers with the automatic calculation, where in the test with the original program I had used a Nop at location 125 yielding the use of nine data register. The execution time listed for 1287 digits for the fast mode program is an extrapolation based on the number of iterations completed after five days. One of my TI-59's is still running as of this writing, and will not finish the 1287 digit problem until after VSN4 is published.
STEREO GRAPHICS WITH BALL-STICK OPTION - Don Graham.

This is a highly comprehensive modular system of programs designed to place at the disposal of TI-59/PC-100 users a relatively simple means of generating accurate professional-quality drawings of three-dimensional objects. Although other programs are available that will do this (see, for example, the 3D Graphics Pakette, or Lester Tibbetts' program in V7N3P9), this system has numerous special features that make it, so far as I know, unique. The principal unique feature, but by no means the only one, is that it contains special routines to make it easy to draw ball-stick molecular-type models.

The system consists of 21 semi-independent programs in four groups: data entry, data manipulation, special ball-stick programs, and output. Since all the programs operate on a common database, additional special-purpose routines could be added by those who need them.

The data entry programs permit entry of up to 216 coordinates at a time, either point-by-point, or automatically from user-written subroutines in any of a variety of coordinate systems.

The data manipulation programs permit an almost unlimited range of transformations of the coordinates in memory. Besides provision for all the usual transformations such as translations, rotations, scaling, etc. It is also possible to perform almost any other transformation imaginable by means of user-written subroutines.

The output programs can print lists of coordinates in memory, or generate coordinates for perspective drawings. All output is fully labeled, as is most input.

This is NOT a graphics mode program. All it does is provide coordinates for hand-plotting. It is a "straight" program that doesn't use any unpublished quirks, but does use HIR and two-digit Dsz. It is not a demonstration program. It is intended as a workhorse system for persons who wish to make accurate professional-quality drawings of 3D objects without the expense of a computer graphics system. The Math/Utilities module is required.

Some examples of stereo pairs of drawings, generated with the aid of the system, are shown on the next page. The drawings are placed for crossed, rather than parallel, viewing. That is, the image intended for the left eye is on the right, and vice versa. They are easily viewed by crossing the eyes, then uncrossing them until the images fuse. Most persons seem to find that this gives a better stereo effect than the more conventional parallel method.

The total system contains over 7300 lines of code, requiring 39 card sides. There are 239 pages of documentation, including algorithm derivations, listings, instructions and examples. You can order a copy of the program by sending $20.00 Canadian, plus postage and packing. Postage and packing charges are as follows:

<table>
<thead>
<tr>
<th>Destination</th>
<th>Surface Mail</th>
<th>Air Mail</th>
</tr>
</thead>
<tbody>
<tr>
<td>Canada</td>
<td>$2.50</td>
<td>$3.50</td>
</tr>
<tr>
<td>US</td>
<td>$5.00</td>
<td>$7.00</td>
</tr>
<tr>
<td>Other</td>
<td>$6.00</td>
<td>$10.50</td>
</tr>
</tbody>
</table>

A complete set of 24 pre-recorded cards, 15 of which are recorded on both sides, is available for an additional $20.00 (although I can't guarantee compatibility between the card-reading mechanism on your TI-59 and mine. International money orders only, please - cash is too risky, and checks take forever to clear.
STEREo GRAPHICS WITH BALL-STICK OPTION - (cont)

Editor's Note: This is a monumental effort which I simply haven't had time to review. The documentation seems thorough and the printing is legible. Two sample outputs follow:

---

TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>ERRATA</td>
<td>2</td>
</tr>
<tr>
<td>FAST MODE ENTRY, D. Hishanovic</td>
<td>3</td>
</tr>
<tr>
<td>NUMERICAL PRECISION, G. Thomson, P. Fruit</td>
<td>3</td>
</tr>
<tr>
<td>EXACT FACTORIALS IN FAST MODE, P. Messer</td>
<td>5</td>
</tr>
<tr>
<td>ZERO AS A CALLABLE LABEL, M. Boland</td>
<td>6</td>
</tr>
<tr>
<td>EDUCALC MAIL STORE</td>
<td>7</td>
</tr>
<tr>
<td>MODULE SELECTOR, J. Thomas</td>
<td>8</td>
</tr>
<tr>
<td>SNYDER'S CHALLENGE, Ralph Snyder</td>
<td>8</td>
</tr>
<tr>
<td>OTHER TI-59 PERIPHERALS</td>
<td>9</td>
</tr>
<tr>
<td>CC-40 GRAPHICS, Maurice Swinnen</td>
<td>10</td>
</tr>
<tr>
<td>CC-40 SINE/COSINE FUNCTIONS, P. Hanson</td>
<td>11</td>
</tr>
<tr>
<td>PROMPTING ON THE CC-40, P. Hansen</td>
<td>12</td>
</tr>
<tr>
<td>PRIME FACTORS WITH CC-40 MATH MODULE</td>
<td>12</td>
</tr>
<tr>
<td>MORE PI FROM BASIC</td>
<td>14</td>
</tr>
<tr>
<td>13 DIGIT MOD 30 SPEEDY FACTOR PINDER</td>
<td>15</td>
</tr>
<tr>
<td>IMPEDANCE MATCHING, Maurice Swinnen</td>
<td>18</td>
</tr>
<tr>
<td>BOOK REVIEWS, P. Hanson, M. Swinnen</td>
<td>20</td>
</tr>
<tr>
<td>LIST PROCESSOR, G. Thomson</td>
<td>20</td>
</tr>
<tr>
<td>1287 DIGITS OF PI TRANSLATION, M. Swinnen</td>
<td>21</td>
</tr>
<tr>
<td>1287 DIGITS OF PI IN FAST MODE, P. Hanson</td>
<td>25</td>
</tr>
<tr>
<td>STEREO GRAPHICS/BALL-STICK OPTION, D. Graham</td>
<td>27</td>
</tr>
</tbody>
</table>

MAGNETIC CARD SERVICE

See V8N1P32 for details. One dollar per card plus a stamped and self-addressed envelope.
No checks please. For programs in this issue:

<table>
<thead>
<tr>
<th>Program</th>
<th>Price</th>
</tr>
</thead>
<tbody>
<tr>
<td>FM Exact Factorials</td>
<td>$1.00</td>
</tr>
<tr>
<td>13 Digit SFP</td>
<td>$1.00</td>
</tr>
<tr>
<td>Impedance Matching</td>
<td>$1.00</td>
</tr>
<tr>
<td>FM 128? Digits of Pi</td>
<td>$1.00</td>
</tr>
<tr>
<td>Others:</td>
<td></td>
</tr>
<tr>
<td>Superplot II (V5N8P9)</td>
<td>$1.00</td>
</tr>
</tbody>
</table>