mini'app'les

apple computer user group newsletter

VoL IV No 1 JANUARY 1981

OFFICERS:

President	Dan Buchler	890-5051 13516 Grand Burnsville Mn 55337	Vice President	St~ve K Johnson	869-3447
Treasurer	Marilyn Thomas	872-7669 2735 Irving Minneapolis Mn 55408	Secretary	A Michael Young	884-2841 9319 Elliot Bloomington Mn 55420

MEMBERS-OF-THE-BOARD:

Newsletter Editor	Chuck Thiesfeld	831-0009 830-5020 8416 Xerxes Bloomington Mn 55431
Bibliogr´er	Chuck Boody	933-5290
Prg.Editor	Ken Slingsby	507-263-3715
Sales Co´r	Hugh Kurtzman	544-7303

Librarians:		
-Acquisition	Keith Madonna	474-3876
-Organization	Terry Pinotti	786-7118
Consultants:		
-Communications	Nelson Capes	463-0414
-Technical	Jim Henke	869-0361

NEXT MEETING

W~dnesday, January 21st, 1981

3 DIMENSIONAL GRAPHICS

Jim White and Ken Slingsby will demonstrate and comment on the 3 major graphic packages currently available:-
Sub-logic
Apple-world
Bill Budge

We may also be able to see some of the work of Dr Saliterman who is working on a graphics system that creates full color images with shaded planes and hidden lines eliminated!
Anyone interested in the fantastic potential of HIRES and its current state of the art cannot miss this meeting. Come early if you want a seat!

Meeting will take place as usual at
Minnesota Federal Savings and Loan
9th Avenue South
Hopkins
Minnesota.

IN THIS ISSUE

HIGHER TEXT -a review

by Dan Buchler

Tho e of you more familiar with Hires Graphics realize that any picture, shape, graphic form, etc may be represented by a collection of dots. All of the half tone pictures in our daily newspaper are good examples of such images.
Within the standard APPLE TEXT screen, characters are formed within a 5 x 7 dot image area. Within a few months of the availability of the first Apple II in late 1967, character sets had been created in the form of APPLE HIRES SHAPES and not too many months later Rob Wentworth of Mini´app´les, and others elsewhere had created directly formed character sets. Let us stop for a moment and define the difference between SHAPES and Directly Formed Character sets.

Most of you know that an Apple SHAPE is a technique available in HIRES graphics whereby graphic forms are created by user defined vectors. Refer to pages 92-100 in the APPLESOFT manual. The Applesoft or HIRES routines will analyze the Shape table and create the dot patterns that represent that shape on the screen.
A directly formed character, in this definition, is the result of a more direct creation of the above described bit patterns representing the character. In that method, which directly parallels the techniques used by the hardware character generators, a 5x7 character is stored in 8 consecutive bytes of memory.
Example

```
- the letter E          is stored in 8
                        bytes as:
      .....             00011111
      .                 00010000
      .                 00010000
      .....             00011111
      .                 00010000
      .                 00010000
      .....             00011111
                        00000000
```

The all zero byte serves as a one scan line seperator so there is daylight between succesive lines. In most display systems using a 5 dot wide letter, only 7 of the 8 dots (bits) of a byte are used, resulting in 2 dots between each letter.

Well, what has this got to do with a review of something called HIGHER TEXT. I want you beginners, who may be reading this to get an appreciation of what HIGHER TEXT is really about. You see, it was´nt until very recently that any really useful general purpose TEXT packages were available that did anything more than create a standard 5*7 font. 5*7 Text packages, as stated earlier have been available and have gotten more sophisticated. For example, the APPLE DOS TOOL KIT, which costs $65, contains a good 5*7 HIRES character set, which has several different fonts, including various Roman, upside down Roman, Greek, etc. In common with all other directly formed characters, they are fast in creation (much faster than Shapes), are created from they keyboard by simply typing the letters and are very useful for labelling graphs and any other Hires picture.

BUT

These systems produce ordinary letters just like the TEXT screen. They, lack color and above all variation in size. Thats where HIGHER text comes in. The Aldrich brothers, who have already acheived fame in the world of Apple Software, put their very considerable experience to work creating a magnificant tool - HIGHER TEXT.

Basically Higher Text produces 4 sizes of characters:-
5*7
10 wide by 7 high
14 high by 5 wide
10 wide by 14 high

The largest letters are formed in a 14 dot wide by 16 dot high image area, the unused dots simply providing the horizontal and vertical spaces between the letters. The wide letters and tall letters are similar to those used in dot matrix printers. They are formed by doubling dots either horizontally or vertically. For the large letters, 2 basic types are available
EXPANDED - which are a combination of wide and tall where 4 dots replace one of the standard 5*7 formats
and LARGE FONT where the letters are more perfectly formed.

In the case of LARGE FONT, they provide several different Fonts such as

Standard

Roman

Apple

Countdown

Old English

Except for the small, all letters can be displayed in the standard 4 HIRES colors plus White. Because, the Expanded letters are constructed of 4 dot groups-
..
..
those letters can be formed into 4 additional colors since the 4 dot groups can themselves be combinations of colors!
It all works very well. Character sizes are selected with Control characters which are easily implemented in Applesoft.
For example, a Ctl Q, gives large letters. To use that in a program you simply say

```
PRINT CHR$(17);"WORDS TO BE PRINTED"
```

similarly, colors are changed by a Ctl-3 followed by a single digit representing the color.

In fact one can display on the same line in one print statement, letters of different shapes, sizes and colors!!!

Lower case is supported in the large and small fonts. A ctl-S sets lower case shift lock. A ctl-A takes you back to upper case. This is not as convenient as the conventional 'ESC' key approach.

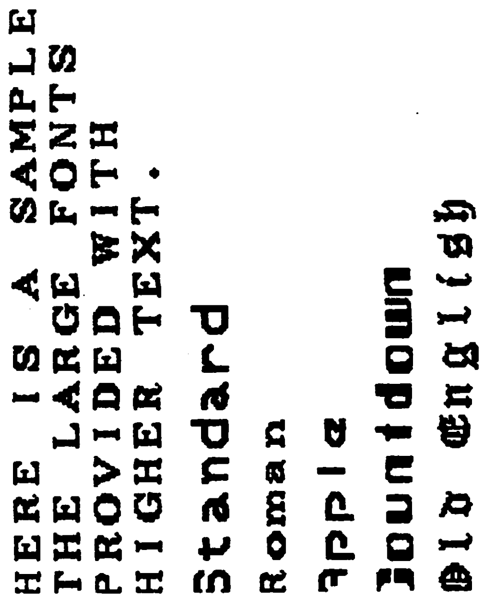
Last but not least is the treatment of backgrounds. To do justice to the Aldrichs, I will not attempt to explain all the options. they are really only understood by demonstration. Let us say, that you can print, black letters on any color background, white letters on a colored background, and print letters on top of letters or other things.

In addition to the HIGHER TEXT package itself, one gets an absolute magnificant editor to create your own FONTS or modify someone elses. I have used the editor and must say it is amazing.
Because Higher Text loads in low memory, a special routine called LOMEM: is provided to move your Applesoft program above the Higher Text program and Graphic Display area: Typically you will start your Applesoft Basic Program at 16384. I admit, that Higher Text uses a fair amountof memory. However if you are pushed, the space from 800 to C00 is also avaiable unless you are using Integer Basic.

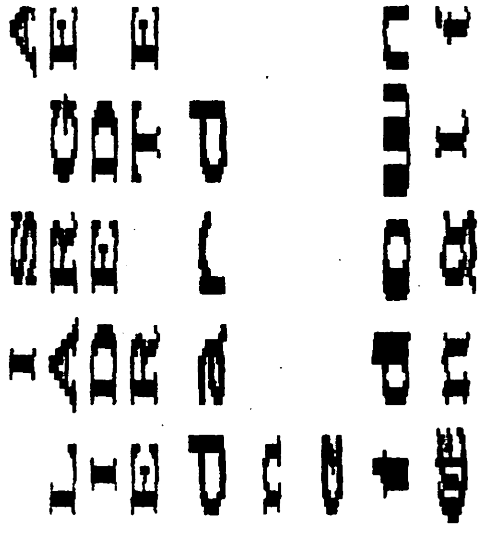
HIGHER TEXT may also be used for non textual types of applications. A good example is shown in the Integer Basic demo provided with the package which includes a 'TANK' font in which a army tank with moving tracks travels across the screen. The Tank is built as a Font of Tanks in different positions relative to the first dot of a large letter. The Editor, which allows one to move the created object within the image area with one keystoke makes it easy to do this sort of thing. And, because the letters are directly formed (see above) the tank can move very fast!

Lastly, the Package comes with a good concise manual which leaves little in doubt except perhaps in exactly how backgrounds work. However a little experimentation makes you an expert.

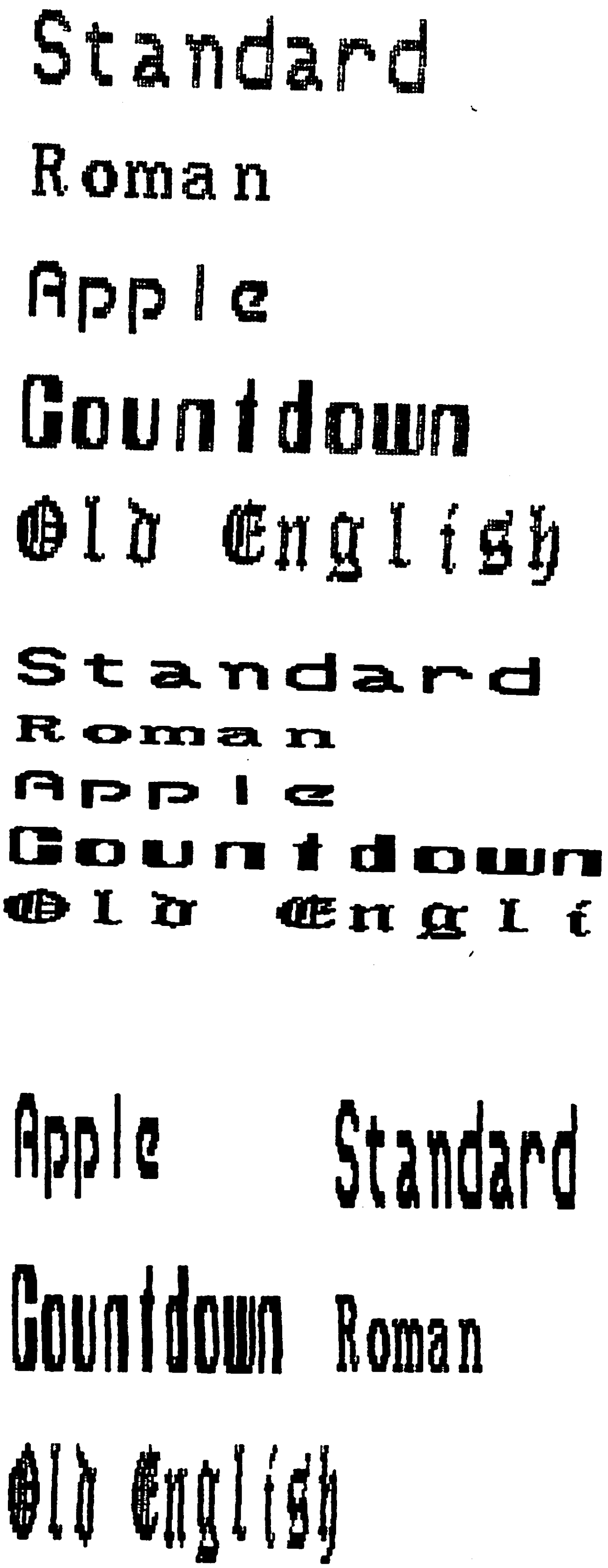
HIGHER TEXT may be purchased through CALL A.P.P.L.E, if you are a member for $20 (Theres a sale on - its usually 22.50) or from Synergistic Software for $35. Theres nothing better currently on the market.



Sideways dump - Letters are regular upright Higher Text



Exampl^ of Graphics dump with MPI 88-G
These examples are all reproduced full size.



Note this article was printed on a Diablo 1620 at 10 cps, 33 columns of print wide.

The remainder of the newsletter is a mixture of 10 cps and 12 chars per inch using 35 cols and 42 cols respectively.

DAN ON PRINTERS

3 members have bought the Epson MX-80 ($650) so far. I want to say a little more about that printer and the MPI 88-G.

The MX-80 deserves close scrutiny for several reasons:-
1. It has a 9 wire head but does´nt cost a fortune.
2. It provides a feature not seen on too many other printers. This is the emphasized mode. The printer makes two passes or (3 if wide letters are being used), offsetting the image slightly on the 2nd or 3rd pass. The effect is that of a very bold letter. The reason I think this is a great feature is that it tends to make the dots disappear. The overall effect is one of a typeset character. Except for the many wire print heads (11 or greater), its the closest thing to letter quality I have seen on a low cost printer. The MPI also has an emphasized mode, but it doe´nt seem to produce quite as bold an effect as the Epson. The Malibu you all saw at our October meeting uses a similar technique. But the Malibu costs over $2000!
3. The Epson is compact.
4. It uses standard paper (up to 3 part)

Incidentily, this writer has no commercial interests in any particular printer manufacturer. I am just trying to report on the facts.

The main disadvantage of the Epson MX-80 is that it currently does not have dot graphics. Their model TX-80 does, but thats only a 7 wire printer. The MX-80 supports what are called block graphics. These are a set of 64 shapes made up out of 3*3 dot squares. They are not particularly useful to Apple owners. Also, because the block graphics are selected by setting the high order bit in the byte sent to the printer, some fidling with the driver is needed to effectively use same.

Now, the manufacturer does claim that they will make a plug-in PROM, which will create dot-graphics, available within 3 to 6 months. This may well be true. I must tell you at this point that the manufacturers of the MPI made a similar prediction when the MPI-88-T first came out. Unfortunately, it is not possible to upgrade a MPI 88-T to a G. One diffrence here is that the Epson 88-T graphics option is obtained through a PROM purchase, so maybe Epson can follow through.

Both Epson and MPI are sold locally by a distributor/dealer who claim to have trained repair persons on their staff. That is definitely of importance as printers are electro-mechanical devices subject to wear and tear. The MPI is distributed by Bolig Associates who have demonstrated that they do provide good service. Epson is distributed by Computer Warehouse Inc who claim the above, but, since they are new to Apple products have yet to prove this fact. I do beleive that they are sincere. It should be pointed out that many of the printers we have talked about in the past are not available locally. This includes: Base 2, Emako, etc.

L^ts talk about the MPI 88-G. It is a definite improvement over the 88-T. These improvements include:-

1. Graphics capability
2. An 11*7 ´serif style matrix´. This does not match the Epson ´emphasized´ mode in my opinion, although I am sure there are some of you who will like it better.
3. An emphasized mode, similar to the Epson, but as stated above, not quite as effective.
4. Quick Cancel. An advertised capability which any self respecting printer should have. Apparently the 88-T would print blanks right across the line even if the line was only 1 character long!
5. The 88G prints in a 7*7 matrix and thus lacks the descenders of the Epson 9*7. Some people don´t seem to mind. I do!

The graphics is good. The software provided allows two pictures side by side, or pictures to be expanded (and thus distorted deliberately) in either x or y direction.

Both the Epson and MPI offer Bi-Directional printing, which this writer finds useless for anything except rough draft listings. None of the printers from the cheapest up to the most expensive Malibu, Diablo, etc can achieve good left margin alignment in Bi-directional mode. However, the MPI does perform its emphasized mode by overprinting on the return travel. For graphics, forget it!

The Epson and MPI both offer

80 & 132 columns
tractor/pressure feed
cartridge ribbon (not sure if thats an advantage)
(3 million characters/cartridge for Epson- 5 million for MPI)

Examples using these printers are included in this newsletter. Graphics examples are from MPI. Note that the large letters were generated using Higher Text. Examples of MPI and Epson regular and emphasized print are also shown.

MPI regular print at 16.5 chars/inch.

```
     "...here  , you see, it takes all the
running you can do, to keep in the same
place.  If you want to get somewhere else,
you must run at least twice as fast as
that." (The White Queen to Alice, in Through
the Looking-Glass)
```

Regular MPI print

```
     "...here  , you see, it tal
running  you  can do, to keep  :
place.  If you want to get some
you  must  run  at least twice
```

```
:56789:;<=>?@ABCDE
]^_`abcdef
```

```
JKLMNOPQRSTUVWXYZ[\]^_`abcdefghijklmnop
JKLMNOPQRSTUVWXYZ[\]^_`abcdefghijklmnop
```

Example 11*7 ´serif style matrix´

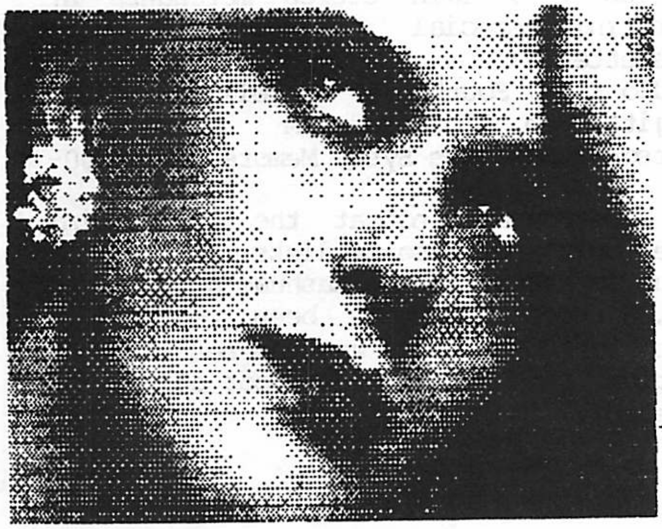
```
     THE NEXT QUESTION ASKED WILL BE
N)?'  WHEN THE PRINTER SEES A WHITE
BLACK DOT.  THEREFORE, MANY PICTURES
THE PICTURES PROVIDED ON THIS DISKET
ONCE A PICTURE IS INVERTED, IT NEED
UNLESS THE PICTURE IS RELOADED.
     YOU WILL NOW BE ASKED WHETHER Y
```

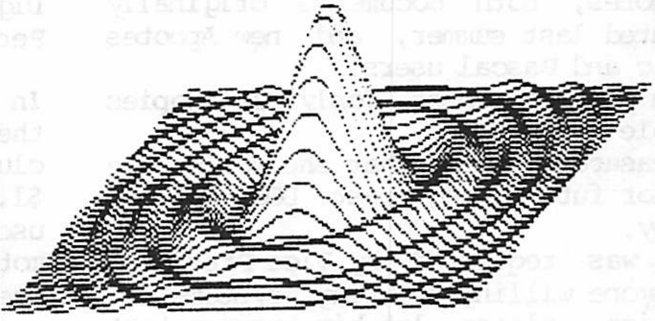
MPI emphasized print.

```
:56789:;<=>?@ABCDEFGH
]^_`abcdef
```

```
'XYZ[\]^_`abcdefghijklmnopqrstuvwxyz{|}~!"
pqrstuvwxyz{|}~!"#$%&'()*+,-./0123456789:;<=>?@ABCDEF
```

MPI 88G screen dump with a HIRES





Epson MX-80 Print

10 CPI Standard

```
ABCDEFGHIJKLMNOPQRSTUVWXYZABCDEFGHIJKLMNOPQRSTUVWXYZ
abcdefghijklmnopqrstuvwxyzabcdefghijklmnopqrstuvwxyz
12345678901234567890!#$%&'()*=+<>?!#$%&'()*=+<>?
THIS IS THE NORMAL PRINTING MODE WHICH
This is the normal printing mode which
```

```
        This is the standard double printing
mode. It is done with double strike.
```

same technique as used by MPI 88-G for emphasized (previous page)

```
RUN
THIS IS AN EXAMPLE OF THE EMPHASIZED MODE
The printing is more dense than the normal 9 x 9 matrix.
ABCDEFGHIJKLMNOPQRSTUVWXYZABCDEFGHIJKLMNOPQRSTUVWXYZ
abcdefghijklmnopqrstuvwxyzabcdefghijklmnopqrstuvwxyz

THIS IS THE DOUBLE STRIKE AND EMPHASIZED MODE PUT TOGETHER.
This is the double strike and emphasized mode put together.
```

5 CPI Double Emphasized

```
THIS IS AN EXAMPLE OF THE DOUBLE WIDE
CHARACTERS EMPASIZED ANDDOUBLE STRUCK
```

8.25 CPI Double Emphasized

```
ABCDEFGHIJKLMNOPQRSTUVWXYZ
abcdefghijklmnopqrstuvwxyz
1234567890:-!#$%&'()*=+<>?
```

16.5 CPI Standard

```
ABCDEFGHIJKLMNOPQRSTUVWXYZABCDEFGHIJKLMNOPQRSTUVWXYZ
abcdefghijklmnopqrstuvwxyzabcdefghijklmnopqrstuvwxyz
12345678901234567890!#$%&'()*=+<>?!#$%&'()*=+<>?
This is an example of the condensed type of printing
```

MINUTES of "EC 1980 MEETING

1. Meeting called to order at 7.45pm with approximately 80 persons present.
2. President described items being sold tonight:

Apnotes, both documents originally distributed last summer, and new Apnotes for Basic and Pascal users.

DOM for November (only 20 copies avaialable tonight).

3. Treasurer stated that she would take orders for future and past DOMs but not any money.
4. It was requested by Vice-President that anyone willing to bring systems to the meeting, please let him know so that we can plan who has to bring a system to the meeting. If enough volunteers can be found, it probably will only be necessary for a specific individual to bring a system once per year.
5. The President mentioned that there was an IAC election forthcoming for the position of Regional Director. A letter from James E. Hassler

AppleNet
129 Park Avenue, Orchard Valley,
Cheyenne, Wyoming, 82001.

was noted. Hassler is campaigning for the position of regional director. His organization is an Apple user group of Ham radio operators. They meet on the air every Sunday night at 0100 Zulu (GMT) on 14.329Mhz. His contention is that, by the nature of his operation he has better contact with users in the district!.

6. Chuck Boody introduced Howard Keyser, a noted Apple proponent who has recently moved to the Twin Cities.

Varied technical questions from the floor were directed to anyone who thought they could answer.

The meeting adjourned for the program of the evening which consisted of:-

1. Presentation by 3M corporation to Terry Pinotti who won programming contest.
2. Talk by Harry Weingartner of Dain Bosworth about Apple Stock

DISK PURCHASES f-r MEMBERS

Currently 3 Twin Cities merchants are offering special purchase deals for diskettes.-

Audio King, Southdale -	Maxell	at $3.25
Digital Den, Maplewood-	3M	at $3.00
Personal Business Syst-	Memorex	at $3.00-

In case you were not at the Nov meeting, the bulk purchase of disks made by the club for Verbatim and Nashuas at $1.00 and $1.25 repectively has been reserved for use for DOMs. This was the result of a motion and vote on the floor during that meeting.

THANK YOUs

We wish to thank Kent Didrickson and Gail Emerson of the 3M corporation for coming to our meeting to present disks to the Winner Terry Pinotti. Terry showed his program which used Bill Budge's graphic package to make clever animations with the 3m logo. Terry also used the Video Message package to create text on the screen.

We wish also to thank Harry Weingartner of Dain Bosworth for coming to talk to us about Apple Stock. He noted that Apple had gone public with 5,000,000 shares at $22.

Thank you to those who broght systems:
Dan Buchler
Rick Gates
et al.

Thank you to Keith Madonna for demonstrating two new games:
Dog Fight
and
CyberStrike by Sirius Software

Both of these programs employ excellent graphics.

DISKS OF the MONTH (DOM)

We, the Mini´app´les, board again apologize for any inconvenience caused by our inability to deliver sufficient quantity two months in a row. This last time was caused by a technical problem that occured at the last minute just when one of the two sets of reproductions was about to be started. We did manage to make 20 copies (thanks to Peter Giles). Approximately 50 copies of the Nov DOM have now been delivered.

Only one DOM has been distributed to date. That is the November DOM. A listing of programs in same was published in the Nov Newsletter. The next DOM is being prepared, and if it is ready in time we will have it at the next meeting plus copies of the November DOM. We plan to make 80+ copies of the new one. Catalog, etc will be published when available.

ADVERTISEMENT

For sale
APPLE II p us, 48k, 2 Disk Drive , B+W M^nitor, Interface card, IBM Selectric II typewriter, Interface between Apple card and Selectric, Printer Cable.
Also Bunker Ramo 32 col printer and interface card.
Will sell under list - make offer!
To see equipment, call
Gene Altstatt
5219 Wayzata Blvd, Suite #232,
St Louis Park, Mn, 55416. 545-4459

MEMBERSHIP IN MINI´APP´LES

Annual dues are $10

For that you receive newsletter, free copying of user bank at Zim Computer, right to purchase DOMs and other material at low membership prices, right to buy diskettes at discount from participating merchants, etc, etc.

IAC APNOTES

Th^ latest ׀ndex for the IAC Apnotes is included in this newsletter (Dated Dec 6th, 1980). New items included in this index and not previously distributed are being reproduced and will be sold at next meeting. We also try to have more copies of the Pascal notes that we ran out of at the last meeting.

1981 ADVERTISING RATES

Ad rates for 1981 are as follows:

Full page	$30/is^ue
Half Page	$20/issue
Double column	$4/inch/issue
Single column	$2.75/inch/ussue

Classified ads for members are free

THE BEST OF NSAUG

NSAUG = North Suburban Apple Users Group (Chicago, Illinois)

This issue contains a selection made by C. Boody, D. Buchler and S. K. Johnson of the best articles from the last two years of the Newsletter of NSAUG. All articles are in the Public domain and we thank NSAUG for the opportunity to publish them. We hope that you also will find them interesting. Page 15 - Page 30

We plan periodically, to provide our members with the opportunity to see the best articles from the many newsletters that we receive on a regular basis. The purpose of our club is to sisseminate information of interest to Apple owners and this is one way we think we can do that. Some clubs pepper their newsletters with extracts from other NLs. We think our method is a little more organized.

TURNING THE PAGES with DAVE LADEN

BYTE -- JANUARY 1981

Electromagnetic Interference by Steve Ciarcia. Pages 48-68.

Languages Forum: A Bug in BASIC by W. D. Maurer. Pages 188-196. A FOR-NEXT looping bug existing in six of fifteen microcomputers, including Apple's Applesoft BASIC.

The Picture-Perfect Apple by Phil Roybal. Pages 226-235. The author discusses how to dump Apple's high-resolution screen to a Qume Sprint Micro 3 printer. A flow chart is included.

Whose BASIC Does What? by Teri Li. Pages 318-327. This is a series of three tables: system commands, statement types, and functions.

CREATIVE COMPUTING -- DECEMBER 1980

Apple Pascal by Steve North. Pages 43-46.

The Apple II and Apple II Plus by David Lubar. Pages 54-55.

Comparative Evaluations of Basic Systems by M. Firebaugh, T. Fossum, P. Sorensen, and W. Stone. Pages 70-80. This is a detailed look and comparison of the different versions of BASIC.

Do Computers Byte? by Judy Neyhart. Pages 92-93. This is an account of one persons encounter with the Apple.

Christmas Tree by Stephen R. Berggren. Pages 124-125. An Applesoft program to create a hi-res Christmas tree.

Apple Nuclear Power Plant by Stephen R. Berggren. Pages 128-137. An Applesoft program listing is included.

Weather Station by Paul Raymer. Pages 142-143. This is an Applesoft program.

Apple-Cart by Chuck Carpenter. Pages 202-208. Mr. Carpenter reviews a few books and new or updated gadgets for your Apple.

INTERFACE AGE -- JANUARY 1981

File Cabinet and Ampersort II by Dr. R. F. Zant. Pages 94-96. Helps you integrate Ampersort II into File Cabinet.

KILOBAUD MICROCOMPUTING -- JANUARY 1981

Managing the Small System Environment by Steven K. Roberts. Pages 43-46.

Simulation Of Musical Instruments by Hal Chamberlin. Pages 53-58. This is the first of a two part series.

Apple Connections by Rolf A. Deininger and Dom Tujaka. Pages 122-123. Some practical tips if you move your Apple a lot.

Space Race by Robert M. Hirbernik. Pages 126-128. A game written in machine language and Integer BASIC.

MICRO -- DECEMBER 1980

Graphing Rational Functions by Ron Carlson. Pages 7-9. Applesoft program listing included.

An Apple Flavored Lifesaver by Gregory L. Tibbetts. Pages 25-30. The game of life written in machine language and Integer BASIC.

Creating An Applesoft BASIC Subroutine Library by N. R. McBurney. Pages 37-40. This is another use for the DOS EXEC Command.

Multiplying on the 6502 by Brooke W. Boering. Pages 71-74. Machine language subroutine included.

PERSONAL COMPUTING -- JANUARY 1981

Word Processing Software Roundup by Steven Jong. Pages 26-33. Includes background information, descriptions of word processing programs, comparison tables, and a vendor guide.

How to Plan For Your Business Computer by Larry Watkins. Pages 42-44.

A Pseudo-Numeric Key Pad for the Apple II by Robert A. Pritchett. Pages 46-47. "It acts as a keyboard filter and changes the right hand side of the keyboard into a numeric key pad."

Understanding BASIC Language Operations by Tom Swan. Pages 68-77. "This article provides an introductory course in Basic program anatomy." Listings of two machine language utilities for the Apple are also included.

Best wishes and happy reading in 1981.

INTERNATIONAL APPLE CORE TM

NEWS BULLETIN

This is our first issue of the IAC News Bulletin, the purpose of which is to provide to you on a monthly basis everything you ever wanted to know about the IAC but were afraid to ask.

Our format will change in response to feedback from you.

First off, we would like your suggestions for a name - IAC News Bulletin doesn't hack it.

So we hereby announce a name contest to begin as of the date of this issue. The contest entries must be postmarked no later than January 15, 1981. Entries will be judged and the winner selected by the IAC Board. We'll announce the winner and the prize in the February issue.

We have a constitution and bylaws! After considerable time and effort we are officially an organization. Sorry about not involving more of you - too difficult to pull off. However, according to powers provided to you, you may call for changes through special meetings and procedures. Check your copy for further details. Joe Budge, our Secretary, will mail them to you soon.

We also elected Jerry Vitt of Dallas to be Chairman of the Board.

Coming up - a call for nominations for Board Directors. Joe has sent out information describing the procedures. Be sure to advise your entire club membership. By all means, participate. IAC is here to serve you.

The annual IAC general meeting will be held in Chicago May 2 and 3. More information to follow as to time and place.

Now that we are "organized" we can get on to the business of defining more precisely what there is in it for you to be a member of IAC. Under consideration are such goodies as workshops for small businessmen (doctors, lawyers, bowling alleys, etc.), IAC ABBS through low cost WATS type lines, review of serious works and major hardware, ombudsman services, sponsorship of major projects, professional certification of members, and more. More on that later...

Reminder - renewal memberships will be due March 31, 1981. Your club will be billed. Also we prefer individual membership subscriptions to Apple Orchard. Please accept our apologies for the delay in getting Vol. 1, No. 2 of the Orchard to our subscribers. Yes, we goofed! However, from now on, subscribers are to get first preference, meaning three weeks earlier than either through the club or at the computer stores.

"APPLE" is the registered trademark of Apple Computer, Inc.
INTERNATIONAL APPLE CORE is licensed by Apple Computer, Inc. to use certain of the latter's trademarks.

A plea - Val Golding needs your input. It's your journal - remember, your magnum opus gets considerably greater visibility through publication in the Orchard. Try it!

Have you heard...?

...Craig Vaughan of Peripherals Unlimited has moved closer to the SOURCE. He now resides in Virginia. His new firm, Microsoftware Systems, has acquired the rights to all Peripherals Unlimited's software, and he can be reached at (703) 385-2944.
...Programma International has been bought out by Hayden Publications. Our Treasurer, Dave Gordon, is Vice President and General Manager for Programma.
...Videx has announced a new "keyboard enhancer" which provides upper and lower case display and entry with the standard 40-column APPLE keyboard.
...Novation has announced a new modem interface card for the APPLE.
...The "Smarterm" 80-column cards distributed by Apple are on their way to the dealers.
... Apple sent a note to all of their Level One Service Centers stating that old disk drives may need some modification to their analog cards to work properly with 16-sector DOS 3.3 and Pascal.
...The IAC Board has awarded a contract to DiLithium Press to print issues No. 4 through 7 of the Apple Orchard. Grawin Publications will do issue No. 3. We thank Grawin for the excellent job they have done on No. 2.

Your IAC disks and APnotes should be coming to you directly from the mail order firm responsible to Joe Budge. We're sure you will agree that the Apnotes are great and we plan to improve the quality and documentation of our disks. Very shortly you will be getting standards notes.

Check the Help Wanted section of your next issue of the Orchard...

About the DOS 3.3 problem. Apple tells us that it was an unfortunate oversight that caused the problem - how many of you have 32K machines? As soon as they heard, they broke records patching the system and had it completed in one week! They do try hard.

Let us know of news, rumors, whatever you would like to share with your counterparts. Comments and feedback should be sent to Bernie Urban, Editor, at the IAC address. Comments may also be passed along to Chairman of the Board Jerry Vitt, who may be reached on his daytime modem (214) 369-0427. We encourage you to pass on this information to your membership through your newsletter or otherwise. Feel free to reproduce anything herein.

December 1, 1980

INTERNATIONAL
APPLE CORE
TM
APNOTE
P. O. BOX 976, DALY CITY, CALIFORNIA 94017 USA

APNOTE INDEX

December 6, 1980

A. HARDWARE MODIFICATIONS

1. Lower Case Apple
2. User Firmware (2716)
4. Adding Colors to Hires
6. Five Additional characters from the Keyboard

B. BASICS

2. Applesoft Hires Routines
3. Applesoft Array Eraser
4. Applesoft Hires Screen Function
5. Generating Tones in Applesoft
6. Literal Input Routine
8. Print Using Simulator
9. Converting Integer Basic Programs to Applesoft
10. Applesoft Random Numbers
11. Applesoft Out of Memory
12. VTAB and HOME Converter for Sup-R-Terminal

C. MACHINE LANGUAGE

3. Apple II Mini Assembler
8. Adding Features to LISA
9. Putting Programma M/L onto Disk

D. INTERFACING INFORMATION

2. Cassette Interface
4. DEL Character Killer
5. Correction To Sup-R-Terminal Preliminary Manual

E. PRINTER INTERFACING

3. Serial Handshake Modification with Tabs
6. Tabbing with Apple Peripherals

F. DATASHEETS

2. Graphics Tablet
5. Apple Post
6. Silentype Thermal Printer

This application note has been provided by an Apple Computer user. The International Apple Core does not guarantee the accuracy of this information in any way and cautions that modifications may void a manufacturer's warranty. "APPLE" is the registered trademark of Apple Computer, Inc. INTERNATIONAL APPLE CORE is licensed by Apple Computer, Inc. to use certain of the latter's trademarks.

G. PASCAL

1. Lower Case Patch
2. Linefeed
3. Take 280
4. Getrem
5. Transfer & Sum 512
7. Foreign & Gettext
9. Comm Card Baud Rate Changer
10. Interfacing Foreign Hardware
11. Long Integer Fix
12. Hires
13. Known Pascal Bugs
16. Pascal Units
17. Pascal Peeks & Pokes

H. TEXT AND GRAPHICS INFORMATION

1. Text Screen Mapping and Use

I. LISTS

J. DOS

1. DOS Demo Programs
2. 3.2.1 Append Fix

TABLE OF CONTENTS OF NSAUG ARTICLES

Catalog with 181 files by Tim
Dave´s PEKEs by Dave Selig
DOS 3.3 - 1st bug found by Dav Holle
DOS Patches by Mark Pump
DOS Patches (more of) by Mark Pump
File Access by Dave Sheeman
Leon´s Bell by Leon Alexander
LOMEM/HIMEM in Integer BASIC by Mike Reinhart
Print Text File in Pascal by Pascal Pete
Remote Control for Your Apple by Todd Peterson
RF Interference by Dave Alpert
Shift Reset Mod by Blake Hobson
Title Page Program by Rich Lundeen
Vidizzy by Dave Alpert

↓

```
*******************
*                 *
* FEATURE ARTICLE *
*                 *
*******************
```

MARK PUMP'S DOS PATCHES

The following article is presented from notes on the lecture given by MARK PUMP at last months meeting. We cannot say enough about how much we appreciate the work Mark has done on this subject. His talk and slide show were certainly one of the highlights of the year for our group.

The patches listed below are contained in the program that follows the article. Since they are very long and will be tedious to key in, the club will have the program in the club library immediately. It will appear on ALL disks in the library that are not already full. Out of town members can send a disk to the club librarian with return postage and will receive the program ASAP. The daring will key it in themselves. No prizes will be awarded to the first person who keys it in without error on the first try!

All patches except the DOS Performance Patches and the Free Sector patch are to be used with DOS version 3.2 only. The performance patch lists the location to be changed depending on the DOS used. The free sector patch will work with either version of DOS as is.

DOS PERFORMANCE PATCHES

The changes below will allow disk access at speeds faster than normal. While sacrificing only a few seconds to boot DOS. BLOADs, BSAVEs, READs, WRITEs will be considerably faster.

Sets SKEW factor = 2 for fastest file loading.

```
   * BF62:02           DOS 3.2
   * BF66:02           DOS 3.2.1
```

Faster INIT of track 0

```
   * BEAD:30           DOS 3.2
   * BEB1:30           DOS 3.2.1
```

Faster INIT of tracks $1 to $22.

```
   * BFAC:EA EA        DOS 3.2
   * BFAE:EA EA        DOS 3.2.1
```

DOS PATCHES

Show number of free sectors on CATALOG command. Works with either DOS 3.2 or 3.2.1.

```
   * BCBA:A2 0C 20 4A F9 A9 00 85
          40 85 41 A0 C8 18 B9 F2
          B3 F0 0E 0A 90 FB 48 E6
          40 D0 02 E6 41 68 18 90
          F0 88 D0 E9 A6 40 A5 41
```

CONTINUED ON PAGE 16

```
* BCE2:20 1B E5 for integer basic
                only-NOP in FP
* BCE2:20 24 ED for APPLE II+ only !
                do not use with INT
                Basic
* BCE2:20 41 F9 returns HEX for
                both INT and FP

* BCE5:20 2F AE
* BCE8:60
* ADC3:20 BA BC
```

DOS ERROR INTERCEPT
HOOK FOR INTEGER BASIC

Causes "GOTO 30583" when a DOS error occurs, with error code = PEEK (63). If PEEK (63) = end of data error, then GOTO end of data routine elsewhere in your program.

```
* 03A5:86 3F A9 77 85 CE 85 CF
       4C 5E E8
* 9D5A:A5 03
* 9D66:A5 03
```

BINARY FILE PATCH

Shows address and length of binary file immediately after issuing BLOAD or BRUN command-if MON C is in effect. Works with either FP or INT Basic. WHEN THIS PATCH IS USED, THE INIT COMMAND WILL BE DISABLED.

```
* A382:20 00 B6
* B600:20 7A A4 AD 5E AA 29 40
       F0 1C A0 C1 20 27 B6 AD
       73 AA AE 72 AA 20 41 F9
       A0 CC 20 27 B6 AD 61 AA
       AE 60 AA 20 41 F9 60 A9
       AC 20 ED FD 98 20 ED FD
       A9 A4 20 ED FD 60
```

REMOVE INIT COMMAND

his frees up the page at $B600 and prevents SLAVE disks from being created from this patched DOS. Used with above patch.

```
* A884:09 0E 09 EA
```

PARTIAL INIT (DOS 3.2 only)

llows the partial initialization of a disk. This can be used to try to resurrect a blown disk f Just parts of DOS are blown.

```
* BFA3:track + 1 of upper bound
       to be INITed
* BFB8:00
* 3D0G

> INIT program name
> go to monitor with RESET or
  CALL-151
* BFA3:23
* BFB8:4C
* 3D0G

> BRUN UPDATE 3.2
```

CONTINUED ON PAGE 17

Best of NSAUG

BINARY HELLO PROGRAM

To run a binary as your HELLO program when you boot up.

```
* 9E42:34
```

ELIMINATE PAUSE DURING CATALOG

```
* AE39:EA EA EA
```

DOS COMMANDS IN IMMEDIATE MODE

Allows direct use of DOS commands from keyboard that are otherwise only accessable from program command.

```
* A021:EA EA EA
```

SUPPRESS SETTING NOMON IOC AT DOS ENTRY

Leaves MON or NOMON set same as prior to RESET

```
* 9DD7:EA EA EA
```

ALLOW S(lot), D(rive), V(olume) operands with MON and NOMON

```
* A92B: 40 F0 40 F0
```

ALLOW LOWER CASE IN TEXT FILES

Store lower case letters in TEXT files.

```
* A648:29 FF
* A656:09 00
```

****
* USEFUL STUFF *
****

HEX TO DECIMAL CONVERSION

```
* ctrl E
* :HI LO (enter high byte-lo byte)
* E51BG in INTEGER BASIC
    or
* ED24G in APPLESOFT
```

number returned is decimal value of number entered

DECIMAL TO HEX CONVERSION

```
> CLR
> A=decimal number to convert
* 804.805 returns the low byte
          and the high byte
          in that order
```

CONTINUED ON PAGE 18

CONTINUING INTEGER BASIC AFTER RESET WITHOUT AUTOSTART ROM

Displays line number where program stopped when a RESET was hit.

```
  * E3E3G
     or
  * E8C8G
```

FINDING AN INTEGER BASIC PROGRAM LINE IN MEMORY

```
  > LIST line number preceding the
         one desired
  * E2.E3 E6.E7 will return address
               of line number
               desired
```

LIST AN INTEGER BASIC LINE, GIVEN ITS ADDRESS

```
  * E4:LO HI (byte of line number
             address)
  * E063G
```

TO ABORT A LONG CATALOG LISTING DURING THE PAUSE BY HITTING "ESC"

```
  * AE39:20 80 B6
  * B680:20 0C FD C9 9B D0 03 4C
          D0 03 60
```

TO ALLOW LENGTH OPERAND TO BE GREATER THAN $7FFF

Allows saving a binary file that is >32k.

```
   * A964:FF
```

TO SHOW DELETED FILES IN ADDITION TO NORMAL FILES ON THE CATALOG

```
   * ADD9:EA EA
```

Most of these mods are in the program listing on the next page. These POKES will not allow the disk being used to INIT another disk as the program overwrites the page at B600 used for the INIT routines.

Also included in the program is the 'WILD CARD' mod that allows you to enter an abbreviation of the filename followed by a - sign whenever a filename is required on a DOS command. For instance RUN THE- will suffice for RUN THE INFINATE NUMBER OF MONKEYS. This can also be used to rename files that have some funny control characters in them. For example, suppose a file called GLUNK appears in your catalog. However, assume there is a control character in the name and you can't run it without knowing the control character and where it is. The solution is to RENAME G-, GLUNK and now it will really be named GLUNK but without the control characters.

If you ever wanted to know what certain keyboard codes were and didn't have a manual for reference nearby, thy the following in the monitor.

```
   * N C000 34:0 <cr>
```

(note:there is a blank between the 34:0 and the <cr>)
Now you can press any key on the keyboard and the appropriate hex keycode will appear on the screen. To abort, press RESET.

Mark wants to thank Dav Holle, Jeff Garbers and Scott Glick for their INPUT on this subject.

Best of NSAUG

```
]LIST

10  POKE  - 17222,162: REM BEGIN
20  POKE  - 17221,12:: REM FREES SECTORS
30  POKE  - 17220,32:: REM ROUTINE
40  POKE  - 17219,74
50  POKE  - 17218,249
60  POKE  - 17217,169
70  POKE  - 17216,0
80  POKE  - 17215,133
90  POKE  - 17214,64
100  POKE  - 17213,133
110  POKE  - 17212,65
120  POKE  - 17211,160
130  POKE  - 17210,200
140  POKE  - 17209,24
150  POKE  - 17208,185
160  POKE  - 17207,242
170  POKE  - 17206,179
180  POKE  - 17205,240
190  POKE  - 17204,14
200  POKE  - 17203,10
210  POKE  - 17202,144
220  POKE  - 17201,251
230  POKE  - 17200,72
240  POKE  - 17199,230
250  POKE  - 17198,64
260  POKE  - 17197,208
270  POKE  - 17196,2
280  POKE  - 17195,230
290  POKE  - 17194,65
300  POKE  - 17193,104
310  POKE  - 17192,24
320  POKE  - 17191,144
330  POKE  - 17190,240
340  POKE  - 17189,136
350  POKE  - 17188,208
360  POKE  - 17187,233
370  POKE  - 17186,166
380  POKE  - 17185,64
390  POKE  - 17184,165
400  POKE  - 17183,65
410  POKE  - 17182,32:: REM CHANGE FOR
420  POKE  - 17181,27:: REM IB-FP-OR
430  POKE  - 17180,229: REM HEX OUTPUT
440  POKE  - 17179,32
450  POKE  - 17178,47
460  POKE  - 17177,174
470  POKE  - 17176,96
480  POKE  - 21053,32
490  POKE  - 21052,186
500  POKE  - 21051,188: REM END OF FREE SECT
510  POKE 933,134:::::: REM BEGIN DOS ERROR
520  POKE 934,63::::::: REM INTERCEPT HOOK
530  POKE 935,169:::::: REM FOR IB
540  POKE 936,119
550  POKE 937,133:
560  POKE 938,206
570  POKE 939,133
580  POKE 940,207
590  POKE 941,76
600  POKE 942,94
610  POKE 943,232
620  POKE  - 25254,165
630  POKE  - 25253,3
640  POKE  - 25242,165
650  POKE  - 25241,3::: REM END
660  POKE  - 24543,234: REM DOS CMNDS IN
670  POKE  - 24542,234: REM IMMED MODE
680  POKE  - 24541,234: REM END
690  POKE  - 22396,9::: REM REMOVE INIT
700  POKE  - 22395,14
710  POKE  - 22394,9
720  POKE  - 22393,148: REM END
730  POKE  - 23678,32:: REM START OF
740  POKE  - 23677,0::: REM BINARY FILE
750  POKE  - 23676,182: REM PATCH ROUTINE
760  POKE  - 18944,32
770  POKE  - 18943,122
780  POKE  - 18942,164
790  POKE  - 18941,173
800  POKE  - 18940,94
810  POKE  - 18939,170
820  POKE  - 18938,41
830  POKE  - 18937,64
840  POKE  - 18936,240
850  POKE  - 18935,28
860  POKE  - 18934,160
870  POKE  - 18933,193
880  POKE  - 18932,32
890  POKE  - 18931,39
900  POKE  - 18930,182
910  POKE  - 18929,173
920  POKE  - 18928,115
930  POKE  - 18927,170
940  POKE  - 18926,174
950  POKE  - 18925,114
960  POKE  - 18924,170
970  POKE  - 18923,32
980  POKE  - 18922,65
990  POKE  - 18921,249
1000  POKE  - 18920,160
1010  POKE  - 18919,204
1020  POKE  - 18918,32
1030  POKE  - 18917,39
1040  POKE  - 18916,182
1050  POKE  - 18915,173
1060  POKE  - 18914,97
1070  POKE  - 18913,170
1080  POKE  - 18912,174
1090  POKE  - 18911,96
1100  POKE  - 18910,170
1110  POKE  - 18909,32
1120  POKE  - 18908,65
1130  POKE  - 18907,249
1140  POKE  - 18906,96
1150  POKE  - 18905,169
1160  POKE  - 18904,172
1170  POKE  - 18903,32
1180  POKE  - 18902,237
1190  POKE  - 18901,253
1200  POKE  - 18900,152
1210  POKE  - 18899,32
1220  POKE  - 18898,237
1230  POKE  - 18897,253
1240  POKE  - 18896,169
1250  POKE  - 18895,164
1260  POKE  - 18894,32
1270  POKE  - 18893,237
1280  POKE  - 18892,253
1290  POKE  - 18891,96
1300  POKE  - 18890,32
1310  POKE  - 18889,12
1320  POKE  - 18888,253
1330  POKE  - 18887,201
1340  POKE  - 18886,155
1350  POKE  - 18885,208
1360  POKE  - 18884,6
1370  POKE  - 18883,160
1380  POKE  - 18882,6
1390  POKE  - 18881,104
1400  POKE  - 18880,136
1410  POKE  - 18879,208
1420  POKE  - 18878,252
1430  POKE  - 18877,96
1440  POKE  - 18876,141
1450  POKE  - 18875,216
1460  POKE  - 18874,181
1470  POKE  - 18873,160
1480  POKE  - 18872,30
1490  POKE  - 18871,140
1500  POKE  - 18870,3
1510  POKE  - 18869,178
1520  POKE  - 18868,136
1530  POKE  - 18867,177
1540  POKE  - 18866,66
1550  POKE  - 18865,201
1560  POKE  - 18864,160
1570  POKE  - 18863,208
1580  POKE  - 18862,4
1590  POKE  - 18861,136
1600  POKE  - 18860,208
1610  POKE  - 18859,247
1620  POKE  - 18858,96
1630  POKE  - 18857,201
1640  POKE  - 18856,173
1650  POKE  - 18855,208
1660  POKE  - 18854,3
1670  POKE  - 18853,140
1680  POKE  - 18852,3
1690  POKE  - 18851,178
1700  POKE  - 18850,96: REM END
1710  POKE  - 20935,32:
1720  POKE  - 20934,54
1730  POKE  - 20933,182
1740  POKE  - 20003,32:
1750  POKE  - 20002,68
1760  POKE  - 20001,182
1770  POKE  - 25129,234: REM SUPPRESS
1780  POKE  - 25128,234: REM SETTING
1790  POKE  - 25127,234: REM MONIOC--
1800  POKE  - 22229,64:: REM ALLOW S,D,V
1810  POKE  - 22228,240: REM WITH MON &
1820  POKE  - 22227,64:: REM NOMON CMND
1830  POKE  - 22226,240: REM END
1840  POKE  - 22968,41:: REM ALLOW LC IN
1850  POKE  - 22967,255: REM TEXT FILES
1860  POKE  - 22954,9
1870  POKE  - 22953,0: REM END
1880  POKE  - 22172,255:
1890  END : REM WHEW!!
1900  REM FROM MARK PUMP
1901  REM PUBLISHED IN THE HARVEST
1902  REM NEWSLETTER AUG, 1980
1910  REM NO COPYRIGHT
```

MORE PUMPS PATCHES

As a result of the article appearing on these pages last month, club member Mark Pump will be going to Seattle in Novemeber to speak to a meeting of the APPLE Pugetsound Program Library Exchange users group publishers of CALL APPLE. His article will be the basis for articles in their magazine as well as The Orchard, the International APPLE Core magazine. If this is not an incentive to write for The Harvest, I can't imagine what else there is. (The Editor)

From Mark:

Here are some changes and errata to the DOS patches in the last newsletter.

First of all, all of the patches are applicable to both 3.2 as well as 3.2.1, instead of 3.2 only, as previously indicated.

The partial init procedure is rather cumbersome, so here is a simpler one...

```
$B7F4:4
$BFA5:TRK + 1 of the upper bound
      to be INITED
$3E3SS 3D9G
```

This is much simpler to use, but requires the old monitor ROM with the instruction STEP command.

The patch on page 4 to allow lower case in text files is not necessary and should not be used.

Also, I would like to make the following offer to club members:

I will fix your disk which was renendered unusable by an I/O error in the catalog or some file on the disk, or will replace the DOS, if that is the problem. There is {{no charge}} to NSAUG members for this service.

Either see me at the club meeting or send me the following:

1. The disk with the problem.
2. A blank disk.
3. A brief description of the problem encountered (won't boot, I/O error in CATALOG or I/O error in file, etc.
4. Your name and phone number where I can reach you if I need additional information.

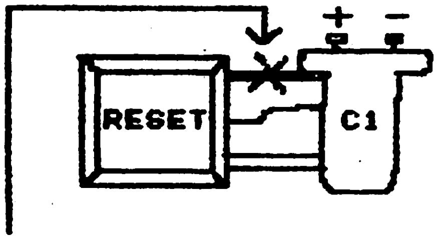
I will return the disk to you at the next meeting unless you provide a S. A. S. E. so I can mail it back to you. If I am unable to fix your disk, I will return both disks to you unchanged. If I succeed in fixing your disk (current success rate 95%), I will return the fixed disk and keep the blank as a backup in case you encounter additional problems.

<<<<NSAUG MEMBERS ONLY, PLEASE>>>>

FROM

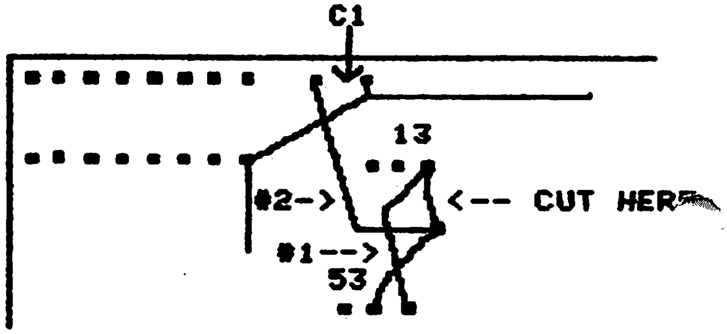
SHIFT RESET MOD

TOPSIDE OF KEYBOARD



CUT TRACE AT X

BOTTOMSIDE OF KEYBOARD
UPPER LEFT CORNER



Best of NSAUG

The short program listed below is copied from Programmers Software Exchange Winter 1980 issue. It is available from them at Box 199, Cabot, Arkansas 72023 for $1.50. Also contained in it are their own review of their own program INSTRUMENT which is an aircraft simulator program and a catalog of their products.

BRUN this program to listen to your cassette player when LOADing a program. This way you don't have to remove the cable from the tape recorder.

Dave Alpert

```
0800-   20 FD FC    JSR    $FCFD
0803-   AD 30 C0    LDA    $C030
0806-   18          CLC
0807-   90 F7       BCC    $0800
```

FIRST DOS 3.3 BUG FOUND!!!

from Dav (sic) Holle

Whenever DOS 3.3 is booted up, be it with a PR#6 or at the first time the APPLE is turned on, DOS 3.3 clobbers whatever Basic you might have loaded into the Language System already. That means a big pain if you freguently reboot, since you are forced to relaod the Language System just as freguently. I found that I can disable this feature with Dan's Disk Utility (which still works on 3.3) like this!

```
0,9R
CC:10
W
```

...That's all there is to it. If you do this to a master DOS, you will be able to put this mod on other diskettes with the Master Create program. --Dav

RF INTERFERENCE HARDWARE PACKAGE FROM APPLE

I recently called APPLE repair in Cupertino and requested their RF interference repair kit. It seems they will sent out a package containing items to modify the RF to help eliminate interference. I have not received it as yet, but will report next month on its contents. In the meantime, if you are having trouble, call the friendly folks at 408-996-1010 and ask for repair. Tell them your problem and they will send you a kit.

Dave Alpert

SHIFT RESET MOD

*

by Blake Hobson

Do you have an older type keyboard that does not have the control-RESET protection? Do you hit RESET when you don't want to? Would you like to prevent the RESET key from being active unless the SHIFT key is also depressed at the same time?

Read on then!

Power off the APPLE. Tear up your warranty (it is probably outdated anyway). Turn your machine over and remove the bottom. Remove the keyboard from the cover of the APPLE.

If you don't know how to do these things without more explanation, then you should not be making this mod by yourself. Get some help from someone who is experienced at this type of thing.

To the right of the RESET key cut the trace where indicated in diagram below. This may reguire a very small drill as the traces are usually hard to cut with a knife.

Turn the board over placeing it so the upper left corner of the bottom is as shown in the diagram below. Cut the trace below the right point below #13 as shown. Connect the right point below #13 and the right side of #53 where indicated by #1. Connect the left side of the C1 pin with the point just above the right side of the #53 row as indicated by the connection marked #2.

That's it. Put it all back together and you have an old APPLE that acts like a new APPLE. You have to hit SHIFT RESET to actually get a RESET.

Best of NSAUG

May 9, 1980

Dear Dave,
Here's a short program you may want to include in the next newsletter. It allows you to modify one track on a disk to continue the catalog. Normally you are restricted to 84 filenames in a catalog but, after running this program, you can have as many as 181 files on the disk. I ran into this problem while trying to save some spelling lists for a school, the disk was only half full but the directory track (track 17) was full so we kept getting a disk full error.

One must use caution to specify a track that currently is not in use on the disk. Probably it would be best to just INITilalize a new disk, run this program, then save your files. Of course there are some good Disk Map programs available that would do the trick as well.

Keep up the good work on the Newsletter.

Sincerely,
Tim

```
]LIST

1  REM ROUTINE TO SET UP CATALOG
       TO HOLD 181 FILENAMES.
       DISK MUST HAVE ONE TRACK
       THAT IS ENTIRELY EMPTY.
2  REM BY T. HARTLEY
       3268 COACH LANE #2A
       KENTWOOD, MI 49508
       (616) 942-8987
3  REM FEEL FREE TO DISTRIBUTE
       THIS PROGRAM
100  INPUT "WHICH TRACK DO YOU WA
     NT TO USE? ";TR
110  GOSUB 900: REM
:::::REM SETS
      UP SHORT MACHINE LANGUAGE
      PROGRAM
120  CALL 45047:
:::: REM READS
     VTOC
122  POKE 46067 + 4 * TR,0: POKE
     46068 + 4 * TR,0:
:::: REM
     RESERVES SPECIFIED TRACK IN
     VTOC
125  CALL 45051:
:::: REM RESAVES
     VTOC
150  POKE 47092,1: POKE 47084,17:
      POKE 47085,1:
:::: REM  GET
     SET TO READ TRK 17, SECT 1
160  FOR T = 12 TO 1 STEP  - 1:
     :: REM  DO SAME FOR ALL SECT
     ORS IN EXTRA CATALOG TRACK
165  POKE 47084,TR: POKE 47085,T:
      POKE 38402,T - 1
170  CALL 768
180  NEXT T: REM
185  POKE 38401,0: POKE 38402,0: POKE
     47085,0:
:::: REM LEAVE LAST
     SECTOR WITH NO LINKS (0,0)
190  CALL 768
199  END :
900  REM MACHINE LANG. SUBROUTINE
905  FOR K = 768 TO 788: READ A: POKE
     K,A: NEXT : RETURN
920  DATA  169,0,141,235,183,141,
     240,183,169,150,141,241,183,
     169,183,160,232,32,181,183,9
     6
```

DAVE'S PEEKS or

things I learnt while looking up other things

this little ditty allows you to exit your program or to continue

```
10 PRINT "PRESS 'ESC' TO END"
   or to return to menu
20 PRINT "PRESS SPACE BAR TO
          CONTINUE"
30 IF PEEK (-16384) = 155 THEN END
40 IF PEEK (-16384) <> 160 THEN
    GOTO XXX (where XXX is
   place where program continues):
   POKE -16368, 0
```

remember that this routine is inserted into your program

when using ONERR routines in APPLESOFT, the following will print the error code and the line number the error occurred in.

```
IF PEEK (222) <> 5 THEN PRINT D$:
PRINT "ERROR # "; PEEK (222); " IN
LINE "; PEEK (218) + PEEK (219) * 256
```

when using an INPUT statement that calls for a single character response, the following line reads the keyboard INPUT immediately without waiting for a carriage return

```
10 PRINT "XXX (Y/N)? ";: GET A$:
   PRINT A$
20 IF A$ <> "Y" THEN GOTO YYY
```

where XXX is your question and YYY is the line # you want to GOTO

VOL. 2, NO. 1 SEPTEMBER 1980

THE HARVEST

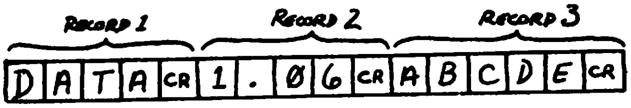
```
*******************
*                 *
* FEATURE ARTICLE *
*                 *
*******************
```

FILE ACCESS

Dave Sherman

One of the better reasons for having a disk system is the use of text files. They let you store and recall data for a program with greater speed and efficiency than any other previously available system, mainly a cassette recorder. The problem is how to use them in the most efficient manner.

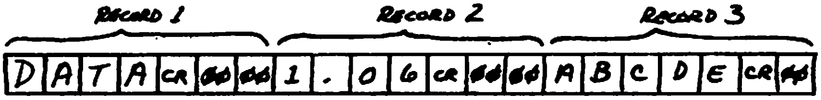
The most basic type of text file is the sequential text file. A program saved on disk, for example, is a sequential file; all of the bytes which represent the program are sequentially saved so that they may be recalled at a later time and thus reload the program. A sequential text file, then, is sequential pieces of data written out as the file one after another. A sequential text file consisting of strings and numbers would look like this:



D	A	T	A	CR	1	.	Ø	6	CR	A	B	C	D	E	CR

where each box is a character position in the text file. Remember, a carriage return (cr) is a character in it's own right; pressing 'return' from the keyboard does not just signal the computer, it sends a character which the computer interprets as a control code that it has to do something about. Carriage returns are said to 'delimit' data in a text file, just as the quotes in a PRINT statement delimits what's suppossed to be printed. Text file data, whether a string or a number, is stored as a sequence of "ASCII" characters (pp. 138-139 in the Applesoft manual). For instance, the string "ABCDE" takes up five characters while the number 11.265 takes up six (the period takes up a character, just as on the screen when printed). In a sequential text file, the data between the carriage returns (including the trailing carriage return) is called a 'record'.

The other type of text file is a random access text file. This is almost exactly the same as a sequential text file with one important difference: all records are the same length, carriage return or NO carriage return. What this means is that DOS can easily calculate the exact location in the file of a particular entry given its number. This ability to reach in anywhere in the file and pull out any entry without the need to read any of the preceding entries is why it's called random access. Every time a random access is opened you must specify a length parameter (,Lxx), because DOS doesn't store this information anywhere. As far as DOS is concerned, all text files look alike. The previous example in a random access file of length seven would look like this:



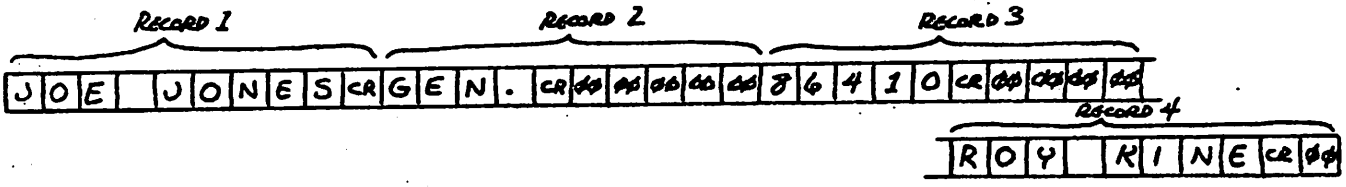
D	A	T	A	CR	ØØ	ØØ	1	.	0	6	CR	ØØ	ØØ	A	B	C	D	E	CR	ØØ

In order to make all records the samee length, DOS pads them all with ASCII 00's, which means you can not use a random access text file as a sequential text file (ASCII 00 is the DOS end-of-file character, so you get an OUT OF DATA error if you try). A random access text file is therefore not as 'efficiency' as a sequential text file because it wastes characters.

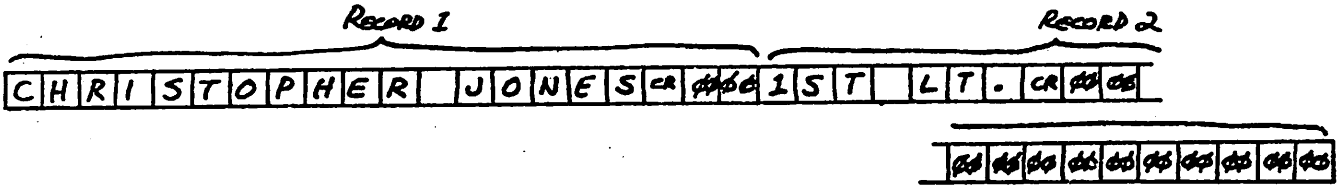
Both types of text files are equally important because each one has different uses. A sequential text file should be used when the data is a list of some sort that does not need to be accessed randomly and could use the greatest possible storage density, such as a list of answers to test questions. Also, if the file is small enough, it can be loaded entirely into memory as, say, a string array and do all random access within this array; this has the advantage of speed at the expense of size. Random access text files, on the other hand, should be used where random access of records is of prime importance; i.e. in an interactive file such as for sorting or searching. Using a random access file is the exact opposite of loading an entire file into memory; the advantage of size at the expense of speed.

In deciding which type of file to use for your particular application, sometimes it is best to write your program and base all disk access on a set of low-level subroutines and dedicate certain variables for passing data for disk accesses. For instance, you could use a string variable for all data (numeric or string) to be written, a variable for record number, etc., and use them consistently throughout the program. Then, when the progranm is written, look at all of the file access and handling and THEN decide which type of file to use. This approach has the advantage that you can keep a library of disk file handling routines and save yourself the effort of re-writing the routines each time. (Subroutine Sam wrote on a similar theme a few issues ago. Ed.)

Another thing to think about when working with files is the actual structure of the file itself. For instance, say you wanted to keep a file of people and file them with their name, rank and serial number. So you open a random access file of length 10 and say that every fourth record is name; the record right after that is rank and so on.


Record 1
Record 2
Record 3
J O E J O N E S CR G E N . CR 00 00 00 00 00 8 6 4 1 0 CR 00 00 00 00
Record 4
R O Y K I N E CR 00

But wait one minute. What if the person's name is CHRISTOPHER JONES? His name is going to get truncated and there would be no way to tell him from CHRISTOPHER MCELROY, either (actually, CHRISTOPHER JONES would get saved, but the next record would over write R JONES(cr) with his rank, and his name would end up CHRISTOPHER LT. with a rank of 1ST LT.) The solution to that is simple. Just increase the record length to 20 and get;

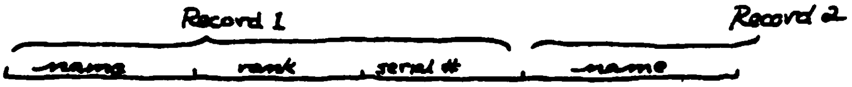

Record 1
Record 2
C H R I S T O P H E R J O N E S CR 00 00 00 1 S T L T . CR 00 00
00 00 00 00 00 00 00 00 00 00

But now there are 10-20 bytes (characters) being wasted with each entry, which is going to be costly later. Besides, using more than one record per entry is really defeating the purpose of the random access file; that is, an easy way to reference each entry by number. What we really want is to have three 'fields' within each record-sort of a mini text file within a text file. Think of it as a two dimensional array;

	name	rank	serial #
Record 1	—	—	—
" 2	—	—	—
" 3	—	—	—

CONTINUED ON PAGE 25

stretched out into a one dimensional array;



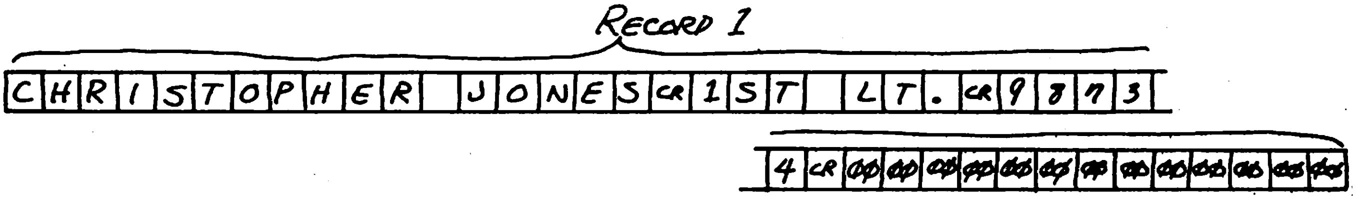
To top it off, we separate the individual fields in each record by carriage returns (we could use any delimiting character that was not valid data, but using carriage returns allows us to get or write each field with a simple INPUT or PRINT statement, i.e.

100 PRINT NAME: PRINT RANK: PRINT SERNUM

and

200 INPUT NAME: INPUT RANK: INPUT SERNUM
where SERNUM is serial number.

As you can see, we need to decide a record length that will hold the longest name plus the longest rank plus the longest serial number, say 45; and thus saving 15 or so bytes per entry over using random access files for each individual entry one at a time.


RECORD 1
C H R I S T O P H E R J O N E S CR 1 S T L T . CR 9 8 7 3
4 CR ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ

Text files are not really as magical or complicated as they might seem at first glance. All the commands and new ideas that go with them are merely intimadating if you've never seen them before. However, the best way to learn about text files is by trying and testing what you have learned, because personal style and experience is just as important as knowing all of the commands and what they do. Remember, it is better to be able to do a great job with imperfect technique because it is the former that gets the job done.

VIDIZZY

Greg Robbins (spelled with one B and not related to our Mike Robins) of Piedmont, Ca. sent this as a letter to Personal Computing magazine and it appeared in their February 1980 issue. After running this program, just trace over the POKE statement and hit RETURN. How about some one to tell us why it does this!!

Dave Alper

```
]LIST

100  TEXT : HOME
110  PRINT "-";
120  FLASH
130  PRINT "U";
140  NORMAL
150  PRINT "@-";
160  FLASH
170  PRINT "T";
180  NORMAL
190  PRINT "@";
200  FLASH
210  PRINT "L";
220  INVERSE
230  PRINT "@O"
235  NORMAL
240  VTAB 5: PRINT " POKE 1032,4:CALL 1024"
245  INVERSE
250  VTAB 4
```

Best of NSAUG

LEON'S BELL

Leon Alexander
as told by
The Editor

Leon gave me a program to demonstrate a form of PRINT USING he had applied to a specific use. When I ran the program, each time I hit a key, the bell sounded. Upon examination I found this nifty little machine program that toggles the bell upon an INPUT or GET statement.

The program starts at $302(770) and uses $301(769) to store a number used for the duration of the bell when a tone is activated. The binary program is loaded either from the prompt line or from BASIC and then a CALL 770 after either the GET or INPUT statement will cause the tone. In using the GET statement, the tone is activated with each keystroke. While using an INPUT statement, the tone is sounded after hitting the c/r.

Thanks Leon. Now where are the other goodies like this the rest of you are hiding? It's stuff like this that make life easier and more interesting for all of us.

```
]LIST                    ]LIST

5  INPUT A$              5  GET A$
10  POKE 769,8           10  POKE 769,8
20  CALL 770             20  CALL 770
30  GOTO 5               30  GOTO 5
```

```
*302.316

0302- AD 30 C0 88 D0 05
0308- CE 01 03 F0 09 CA D0 F5
0310- AE 00 03 4C 02 03 60

*302L

0302-   AD 30 C0    LDA   $C030
0305-   88          DEY
0306-   D0 05       BNE   $030D
0308-   CE 01 03    DEC   $0301
030B-   F0 09       BEQ   $0316
030D-   CA          DEX
030E-   D0 F5       BNE   $0305
0310-   AE 00 03    LDX   $0300
0313-   4C 02 03    JMP   $0302
0316-   60          RTS
```

ADDING LOMEM OR HIMEM TO AN INTEGER BASIC PROGRAM

by Mike Reinhart
M & R Engineering Co.

Step A: enter into your program
0 REM --- (--- is 3 blanks)

Step B: enter monitor(via reset or call -151)
type cb ca(cr)

Step C: type the two numbers returned, a period and add eight (hex) to the first number and type that and a c/r
EXAMPLE:

```
00CB-95
00CA-F8

type 95F8.95FF
```

Step D: 95F8-08 00 00 5D A0 A0 A0 01
is shown on the screen

Step E: type in the following
95F8:08 00 00 11 A0 00 10 01

What happens is this. The hex number 5D in step D is the REM token. The A0s that follow are the spaces. The 01 identifies the end of the program line. When we change the 5D to an 11, we change the REM to a LOMEM (10 would change it to a HIMEM). The 00 and 01 that follow sets LOMEM to 4096 which is HEX 1000. Note that the 00 preceeds the 10 since we enter numbers low order bit first followed by the high order bit. This will protect HIRES Graphics subroutines entered as machine language programs (usually stored around 800 hex) from being clobbered by Integer Basic variables.

THE FORTRANS ARE COMING!
THE FORTRANS ARE COMING!

Heard recently from usually reliable sources that APPLE will begin selli Fortran for the Language System sometime during the third quarter of this year. The cost will be around $200.

A FREE REMOTE CONTROL FOR YOUR APPLE
(IF YOU'RE LUCKY)

Todd C. Peterson
937 Timber Lane
Lake Forest, Il 60045

If you have only a few household electronic items, you will be able to "put together" a hardware "limited" remote control device for your Apple. Perhaps a few things need explaiing: I said "put together" because no actual construction is required for this little device, and "limited" because it offers toggle ON-OFF capabilities (almost) exclusively.

To construct this hardware interface to the Apple, no circuitry need be designed or built -- in fact, you don't even have to open the cover. This is because the cassette input port of the Apple is utilized.

If you're lucky, as I was, you'll have the neccessary items for this interface. They are simply:

A CB Radio (Reciever Only Needed)
Jumper Cable (CB <--> Apple)
One Cheap (Radio Shack) Walkie-Talkie

The CB is anything that recieves the same frequency you walkie-talkie is transmitting on (usually channel 14 -- look on the back). The jumper cable connects the external speaker plug on the CB with the CASSETTE IN plug on the Apple.

Through experimentation it was discovered that with the CB squelch set almost to maximum, the recorded input on the Cassette port would register a constant input level (1 or 0) regardless of input to the CB. Whenever the transmit button on the walkie-talkie was depressed, the iput level would switch states. From this observation, the control program (Listing 1) was developed. The user-interface to this routine comes at FUNC (line 1490). In the listing, the user program simply rang a bell, but a JSR to another machine language program would work.

The interface to BASIC is very simple also. It will only work with Integer BASIC and an example is included in Listing 2. Put a RTS at FUNC and then follow the example. Note that the program can have a counter to do different things each time the program is triggered.

The demo machine language as written will continue to monitor the input until a key on the Apple keyboard is pushed, followed by a toggle from the CB. You can change these termination conditions by altering the last few lines of the routine.

My specialized software development team is currently working on some specific applications for this remote control. If any READERS can beat them to it, I'd like to hear about it.

```
                  1000 CASSIN  .EQ $C060    ;CASSETTE INPUT PORT
                  1010 COUT    .EQ $FDED    ;MONITOR CHAR OUTPUT
                  1020 BELL    .EQ $FF3A    ;MONITOR BELL
                  1030 WAIT    .EQ $FCA8    ;MONITOR DELAY ROUTINE
                  1040 KBD     .EQ $C000    ;KEYBOARD INPUT TRUE IF >$7F
                  1050 STRB    .EQ $C010    ;KBD STROBE MUST HIT AFTER KBD READ
                  1060            .OR $0800
                  1070 *  PROGRAM REMOTE (INPUT,OUTPUT);
0800- 20 31 08    1080 ORG     JSR READ     ;GET DATA IN CARRY BIT
0803- 90 16       1090         BCC SET
0805- A2 00       1100 CLR     LDX #$00
0807- 20 31 08    1110         JSR READ
                  1120 CHECK1
080A- 90 F4       1130         BCC ORG
080C- E8          1140         INX
```

```
080D- 20 31 08 1150        JSR READ
0810- 8A       1160        TXA
0811- D0 F7    1170        BNE CHECK1
               1180 CLOOP
0813- 20 31 08 1190        JSR READ
0816- B0 FB    1200        BCS CLOOP
               1210 *
               1220 *
0818- 4C 41 08 1230        JMP FUNC
081B- A2 00    1240 SET    LDX #$00
081D- 20 31 08 1250        JSR READ
               1260 CHECK2
0820- B0 DE    1270        BCS ORG
0822- E8       1280        INX
0823- 20 31 08 1290        JSR READ
0826- 8A       1300        TXA
0827- D0 F7    1310        BNE CHECK2
               1320 SLOOP
0829- 20 31 08 1330        JSR READ
082C- 90 FB    1340        BCC SLOOP
               1350 *
               1360 *
082E- 4C 41 08 1370        JMP FUNC
               1380 *   FUNCTION READ : BOOLEAN;
0831- 20 3B 08 1390 READ JSR STALL
0834- AD 60 C0 1400        LDA CASSIN
0837- 29 80    1410        AND #$80
0839- 2A       1420        ROL
083A- 60       1430        RTS
               1440 STALL
               1450 *   PROCEDURE STALL;
083B- A9 1C    1460        LDA #$1C
083D- 20 A8 FC 1470        JSR WAIT
0840- 60       1480        RTS
               1490 FUNC
               1500 *  PROCEDURE USER;
               1510 *** PUT ANYTHING HERE TO BE EXECUTED EVERY TIME THE BUTTON
               1520 * IS DEPRESSED.
               1530 **
0841- 20 3A FF 1540         JSR BELL
               1550 *******
0844- AD 00 C0 1560         LDA KBD
0847- 8D 10 C0 1570         STA STRB
084A- 10 B4    1580         BPL ORG
084C- 60       1590         RTS
               1600         .EN
```

```
:$800.84C

0800- 20 31 08 90 16 A2 00 20
0808- 31 08 90 F4 E8 20 31 08
0810- 8A D0 F7 20 31 08 B0 FB
0818- 4C 41 08 A2 00 20 31 08
0820- B0 DE E8 20 31 08 8A D0
0828- F7 20 31 08 90 FB 4C 41
0830- 08 20 3B 08 AD 60 C0 29
0838- 80 2A 60 A9 1C 20 A8 FC
0840- 60 20 3A FF AD 00 C0 8D
0848- 10 C0 10 B4 60
```

SYMBOL TABLE

```
CASSIN C060   COUT  FDED   BELL   FF3A
WAIT   FCA8   KBD   C000   STRB   C010
ORG    0800   CLR   0805   CHECK1 080A
CLOOP  0813   SET   081B   CHECK2 0820
SLOOP  0829   READ  0831   STALL  083B
FUNC   0841
```

```
>or ] LIST

 10 D$="" REM CTRL D HERE
 20 PRINT D$; "BLOAD REMOTE CONTROL"
 30 REM MAKE SURE LINE 1490 IS FUNC RTS
 40 REM IN MACHINE LANGUAGE PROGRAM
 50 REM
 60 REM
 70 CALL 2048
 80  !
 90  !
100  !
     !
     !          YOUR PROGRAM GOES HERE
     !
     !
200  ! REM
300  ! END
```

Best of NSAUG

TITLE PAGE PROGRAM

Rich Lundeen submits this nice program to start off your programs with a title page that will get their attention!

```
]LIST

10  REM  *******************
20  REM  TITLE PAGE ROUTINE
30  REM  WITH ASTERISK BELLS
40  REM  *******************
50  TEXT : HOME
60 A$ = "*": REM   CTRL G HERE
70 B$ = "                        "
75  REM  24 BLANKS HERE
80 T$ = "YOUR TITLE GOES HERE"
90  REM  LIMIT TITLE TO 24 CHARACTERS
130  FOR X1 = 33 TO 8 STEP  - 1
140  VTAB 7: PRINT  TAB( X1 )A$
150  VTAB 14: PRINT  TAB( X1 )A$
160  NEXT X1
170  FOR Y = 8 TO 13
180  VTAB Y: PRINT  TAB( 33 )A$
190  VTAB Y: PRINT  TAB( 8 )A$
200  NEXT Y
210  INVERSE
220  POKE 32,8: POKE 33,25
230  POKE 34,7: POKE 35,14
240  VTAB 8
250  FOR X = 1 TO 6: PRINT B$: NEXT X
270  VTAB 10
280  HTAB  INT (12 - ( LEN (T$) / 2))
290  PRINT T$
300  REM  ROUTINE TO HOLD SCREEN
320  NORMAL : POKE 32,0: POKE 33,40: POKE 34,20: POKE 35,23
330  VTAB 23: PRINT  TAB( 22)"PRESS SPACE BAR";: GET Q$
340  POKE 34,0: REM  NORMAL SCREEN
350  REM
360  REM  START YOUR PROGRAM
370  REM  HERE
380  END
390  REM
400  REM  BY R. A. LUNDEEN (3-1980)
```

SINGING FINGERS

This program appeared on the pages of the Cider Press and on their 'Best of Cider Press diskette'. It starts at 768 ($300) and is 70 (46) bytes long. It can be loaded and CALLed from basic or simply BRUN.

```
*300.346

0300- AD 47 03 85 38 AD 48 03
0308- 85 39 60 E6 4E D0 02 E6
0310- 4F 2C 00 C0 10 F5 91 28
0318- 8A 48 98 48 08 AD 00 C0
0320- 0A 0A 85 06 A9 10 85 07
0328- AD 30 C0 88 D0 04 C6 07
0330- F0 09 CA D0 F6 A6 06 F0
0338- EF D0 ED 28 68 A8 68 AA
0340- AD 00 C0 2C 10 C0 60
```

Best of NSAUG

NEED TO GET A TEXT FILE TO A PRINTER IN PASCAL?

Here is a starter program that is easily modifoed to print as many copies as you need. As listed, the program is set up for a 66 line page. With a little change here and there, you can easily justify your right margins, break a long file for margins at the top and bottom of each page and number the pages.

```
PROGRAM PRINTAPAGE;

(* TAKE A TEXT FILE AND PUT IT TO THE
   PRINTER IN LETTER FORM FOR A SPECIFIED
   NUMBER OF PRINTOUTS...*)

VAR I,J,K,L     : INTEGER;  (* COUNTERS *)
    ANS,QUES    : STRING[80];
    FID1,FID2   : TEXT;      (* FILE VAR *)

BEGIN
  WRITELN(CHR(12));
  GOTOXY(0,12);
  WRITELN('TAKE A FILE TO PRINTER');
  WRITELN('');
  WRITELN('NAME THE FILE: ');
  WRITELN('(I.E. DISKNAME:FILENAME)');
  READLN(ANS);
  ANS := CONCAT (ANS,'.TEXT'); (* -MUST- BE A TEXT FILE *)
  WRITELN(CHR(12));
  GOTOXY(0,12);
  WRITE('HOW MANY COPIES? ');READLN(L);
  GOTOXY(0,12);
  WRITE('ARE LINES SET TO 80 COLUMNS? (YES/NO) '); (* PAGE WIDTH CHECK *)READLN(QUES);
  IF QUES <> 'YES' THEN EXIT (PROGRAM);
  WRITELN(CHR(12));
  GOTOXY(0,12);
  WRITELN('TURN ON THE PRINTER--->');READLN(QUES);
  WRITELN(CHR(12));
  REWRITE(FID1,'PRINTER:');
  FOR K := 1 TO L DO
    BEGIN
      GOTOXY(0,12);
      WRITELN('THE COUNT IS: ',L-K+1:3);
      RESET(FID2,ANS);
      I:= 1;
      WHILE NOT EOF(FID2) DO
        BEGIN
          READLN(FID2,QUES);
          I:= I + 1;
          WRITELN(FID1,QUES);
          IF I>= 66 THEN I:= 1;
        END;
      FOR I := 66-I DOWNTO 0 DO
        BEGIN
          WRITELN(FID1,'');
        END;
      CLOSE (FID2,LOCK);
    END;
  WRITELN(CHR(12));
END.
```

THIS IS A "QUICK" PROGRAM THAT CAN BE EASILY MODIFIED---

PASCAL PETE

MINI'APP'LES SURVEY

NAME ____________________ ____________________ _- DATE / /
last first m

PHONE ( ) -

1-2 Do you own or have regular access to an APPLE II?
(A2) YES (**) NO

3. Computer type?
(-) STANDARD APPLE II (+) APPLE PLUS

4-5. Memory size?
(16) 16K (32) 32K (48) 48K

6-7. Storage media?
(*T) TAPE (*D) DISK (DD) DUAL DISK

8. Monitor type?
(C) COLOR TV (B) B&W TV (V) VIDEO MONITOR

9. Language cards?
(L) LANGUAGE CARD (A) APPLESOFT ROM (I) INTEGER ROM

ADDITIONAL EQUIPMENT (circle all that apply)

10. (M) MODEM 11. (P) PRINTER 12. (G) GRAPHICS TABLET
13. (E) 80 COLUMN BOARD 14. (K) CLOCK CARD
15. (S) MUSIC OR SOUND BOARD 16. (X) 16K EXPANSION CARD TO 64K
17. (Z) Z-80 CARD 18. (R) ROM+ 19. (Q) PA#1

INTERESTS (circle all that apply)

20. (H) HAM 21. (B) BUSINESS 22. (E) EDUCATION 23. (F) HOME FINANCE
24. (G) GAMES 25. (W) WORD PROCESSING

PLEASE RATE YOURSELF IN PROGRAMING ABILITY IN THE FOLLOWING LANGUAGES.
(0=none to 9=expert)

26. Assembly (0) (1) (2) (3) (4) (5) (6) (7) (8) (9)
27. Integer Basic (0) (1) (2) (3) (4) (5) (6) (7) (8) (9)
28. Applesoft Basic (0) (1) (2) (3) (4) (5) (6) (7) (8) (9)
29. Pascal (0) (1) (2) (3) (4) (5) (6) (7) (8) (9)
30. Fortran (0) (1) (2) (3) (4) (5) (6) (7) (8) (9)

THANK YOU FOR YOUR HELP

PASCAL SPECIAL INTEREST GROUP

Regular meetings will be held on the 1st Wednesday of ecah month, also at the Minnesota Federal Building in Hopkins. Next Meeting of that group will be
Wednesday, Feb 4th, 1981

Repeat, this is the PASCAL SPECIAL INTEREST GROUP. However all are wellcome, and in fact are encouraged to attend if you are at all interested in Pascal.

FEBRUARY MEETING

WEDNESDAY , Feb 20th, 1981
Minnesota Federal & Savings
Hopkins

APPLE]I[DEMO.

MINI´APP´LES SURVEY

If you did´nt make it to the Dec meeting, we request that you fill in the survey in this issue and return to Steve Johnson.

6053 Wentworth,
Minneapolis,
Minnesota, 55419

. The survey will allow us to make more intelligent decisions in selecting programs for meetings, material for newsletters, etc.

MEMBERSHIP CARDS

Members should be aware that 1981 Mini´app´les Membership cards will be mailed out this month to paid up members.

MINI´APP´LES
13516 Grand Avenue South,
Burnsville
Minnesota, 55337.

Bulk Rate
U.S. Postage
PAID
Hopkins, MN
Permit 631