



mini'app'les

apple computer user group newsletter

VOL IV NO 5

MAY 1981

NEXT MEETING

Wednesday May 20th, 1981; 7:30pm;
Minnesota Federal Savings and Loan,
9th Avenue South,
Hopkins, Minnesota.

This is the first of a new meeting format. The meeting will consist of two or more parts:

- A Demo of hardware or software or a general presentation of interest to most members.
- An educational topic on programming or hardware.

For the May meeting, First Part:

A presentation on the Integration of Opto-Electronics with the Apple by Dave Young of MEI. This will include a demonstration of a low cost optical card reader.

Second part:

Chuck Thiesfeld will give a talk on the "Organization of Basic Programs"

This will probably be the last regular meeting to be held at this bank. See next month's meeting notice. The PASCAL special interest group will continue to meet at the bank.

PASCAL GROUP MEETING

The PASCAL special interest group will meet on May 6th; 7:30pm at:

Minnesota Federal Savings and Loan
9th Avenue South, Hopkins.

(See note elsewhere on ATTACH BIOS)

JUNE MEETING

THURSDAY - JUNE 18TH

HENNIPIN-SOUTHDALE
PUBLIC LIBRARY

This is a change in day of week and a change in meeting place. Only the time remains the same at 7:30pm. The Library is on Xerxes Avenue near Target and the Yorktown Shopping Center. Take 62 Crosstown to Xerxes, then South on Xerxes.

The change in the day of week is only temporary. We could not reserve the building on the 3rd Wednesday of June. Please watch your newsletter for meeting-place announcements as the meeting-place is likely to change from month to month.

FORT SNELLING

Programming Special Interest Group
(Dave Nordvall - coordinator)

The next meeting of this group will be on Monday May 11th at the Lake Nakomis Community Center. It is on Minnehaha Parkway.

C.B.Allen (of last months Apple Valley meeting fame) will make a presentation on Structured Programming.

Other scheduled meetings of this group will be held on:
June 8th and June 22nd.

This is a very active group!

TABLE OF CONTENTS
IS ON
BACK PAGE

MINI'APP'LES INFORMATION

MINI'APP'LES OFFICERS

President	Stephen K. Johnson 6053 Wentworth Ave S. Minneapolis, Minnesota, 55419	869-3447
Past President and Newsletter Editor	Daniel B. Buchler 13516 Grand Avenue S. Burnsville, Minnesota, 55337	890-5051
Vice President	Chuck Thiesfeld 8416 Xerxes, Bloomington, Minnesota, 55431	831-0009 830-5020
Treasurer	Marilyn Thomas 2735 Irving, Minneapolis, Minnesota, 55408	824-9892
Secretary	Ron Androff 1725 Crest Ridge Lane, Eagan, Minnesota, 55122	452-5230

CLUB INFORMATION

This is the Newsletter of Mini'app'les, the Apple II Users' group of the Twin Cities of Minneapolis and St. Paul.

Questions

Please direct questions to appropriate board member or any officer. Technical questions should be directed to one of the Technical Advisers listed on the left.

Membership

Applications for membership should be directed to the Membership Co-ordinator.

Dues are \$10 per year up to July 1st; \$5 per year in July/Aug/Sept. After Oct 1st, \$10 buys membership for current and next year. Members receive a subscription to this newsletter and all club benefits.

DOMs

DOMs (Diskettes of the Month) are available at meetings or contact Software Sales coordinator.

Newsletter Contributions

Please send articles, information, etc., to the Newsletter Editor. Hard copy binary or text files (ASCII coded) are preferred, but any form will be gratefully accepted. Deadline for publication is the Wednesday before the 1st Wednesday of the month in which the item might be included. An article will be printed when space permits if, in the opinion of the Newsletter Editor, it constitutes suitable material for publication in this newsletter.

Advertising rates

Rates are as follows:
Full Page \$30/issue
Half Page \$20/issue
Double column \$4/inch/issue
Single column \$2.75/inch/issue

Circulation 450 (approx)

MEMBERS OF THE BOARD

Membership Co-ordinator	Ann Bell 8325 39th Avenue N. New Hope, Minnesota, 55427	544-4505
Newsletter Editor	Dan Buchler	890-5051
Bibliographer	Chuck Boody	933-5290
Librarian	Terry Pinotti	786-7118
MECC Librarian	Dave Nordvall	724-9174
Program Editor	Ken Slingsby	507/263-3715
Software Distr. - Mail and Software Sales	Hugh Kurtzman	544-7303
Hardware Sales	Al Peterman	721-3295
Disk Sales	Peter Gilles	475-3916
Publicity Co'tor	A. Michael Young	884-2841
Spcl Interest - Pascal	Keith Madonna	474-3876
Spcl Int. - Geneology	Bill Decoursey	574-9062
Spcl Int. - Nibble	John Schoeppner	455-8613
Spcl Int. - Z80/CPM & Meeting Hdw Support	Rick Gates	735-0373
Technical Advisers	Dave Laden Jim White	489-8321 636-4865
Assistant Librarians:	Bill Decoursey	see above
Assistant Prog Editors:	Tom Edwards Rick Gates	
Assistant Editors:	Tom Edwards E. Michael Young	
Assistant Sales Co'tor	Dave Nordvall	

Note: Co'tor = Coordinator.

DOM #4 by Ken Slingsby

The Disk Of the Month number 4 contains the remainder of the programs from the IAC #4 Disk, several original programs and programs from magazines donated by local members, and a repeat from a previous DOM back by popular demand. A new feature is being introduced this month. With the help of Acme Computer the DOS on this DOM and those that follow is bootable (? !) on either 3.2 or 3.3 systems. Try it and see!

**** PULSAR**

This IAC demo is a series of machine language routines that produce high speed pulsating displays on the screen. An Integer program explains how to run the routines.

**** HIRES-TO-LORES DEMO**

This demo is from APNOTES. It converts a random hires screen to a lores equivalent.

**** INPUT DEMO BY TWE**

This demo incorporates routines from fall 1980 issues of Mini'App'Les and especially John Risken. Input does not create the "EXTRA IGNORED" problem. The demo is designed to have the subroutines stripped out for use in program development.

**** INT CATALOG**

A hello program which enables running a program (etc.) with just a keypress or two.

**** JANE'S EGG TIMER**

A large digital count down clock for timing events. REMS in the program explain how to customize the clock for your computer to increase its accuracy.

****POLAR PLOTS**

An educational graphic program that uses the "polar" coordinate method for the construction of geometric shapes.

**** SPIRAL DEMO**

An IAC program that plots spirals in polar coordinates.

**** SURFACE**

An IAC Demo which plots a Three-D mathematical surface.

**** ALPHABETIZE**

A drill for kids in the second grade on the organizing of words alphabetically. Uses "read-data" word entry. Response is by typing in the choice for correct list position.

**** ASTEROYDER**

This game is similar to one found on the last DOM. You control a ship firing at a multitude of targets coming at you. This program requires the paddles and some skill to score. BRUM the program. From the Japan Baked Apple User Group.

**** LIT'L RED BUG**

This game originally appeared in the Creative Computing Magazine. It is a lores paddle game for one player. Try to go as fast as you can and keep your car on the road as it snakes ahead of you. Many of you will recognize the original author's name as one of the first premiere writers of Elppa software.

**** MOONIE**

A great hires game from a recent "SOFTSIDE" magazine (Fall 1980). This is better than the run-of-the-mill lunar lander. The length of the program and stored data clobbers some of the program. Therefore SAVE it BEFORE you RUN it.

**** EQUI-PROBABLE**

A demonstration which plots the equiprobable distribution of a group of random integers. From the IAC.

**** BASE CONVERTER**

A program to convert numbers from integer to hexadecimal to binary. It also has the capability to add or subtract, which becomes useful when calculating addresses in the computer.

**** CATALOG MANAGEMENT**

A repeat of a previous DOM. This program enables the running (etc.) of programs from the catalog with just a keypress or two. It also allows a short review of info stored about the programs on the disk.

**** DAY FINDER**

A utility program that finds the day of the week for any date entered (since about 1500 AD).

**** DISK ACCESS UTILITY**

This utility allows display or printing of the disk sector by sector. The program includes some self documentation.

**** DISPLAY MEMORY IN HEX & ASCII**

This routine will allow you to examine the RAM, displaying the addresses in both Hexidecimal and ASCII. The program will fill one screen then stop to allow close examination.

**** NUMBER FORMATTER**

A routine which can be EXECed into a calling program to provide a given number of digits after the decimal point with rounding of the last digit. A demo is included which explains the workings of the routine. The routine occupies lines 59000 - 59080.

**** NUMERICAL KEYPAD**

A utility program which aids in numerical input such as into a data base. BRUN the program; then when the numerical keypad is needed, type a control S. This turns the letters U I O J K L into numbers. The top row of keys provides the 7 8 & 9 with the space bar providing the zero. This may be a little hard to visualize, but it works. Another control S will exit the routine.

**** PAGE 2 UTILITY**

This utility allows the use of the lores page 2 of the Apple.

COMMENTS ON D.O.M.

Documentation for the DOM was part of the job for the Program Editor, me, as I saw it when I volunteered for the position. Up to now with a few exceptions, this has been the case. Thanks to the assistance of two of the people who donated programs for this DOM, it was much easier to assemble this month's disk. Both people provided programs with liberal REM's explaining what was going on. They also provided the file PROG.LIST with explanations of the programs on the disks. We do not necessarily require either of these items for submitting programs for the DOM but I think everyone can appreciate the help their inclusion provides. The program CATALOG MANAGEMENT was not available before now and explains why PROG.LIST was not included before. In most cases a person can follow the flow of a program without all the REMs. In some cases I added comments to programs where I could figure out what was going on. In one or two instances, I have worried that my interpretation was not the same as the programmer's.

Put yourself in my shoes (if they fit). Would you feel right guessing at the intent of a program portion then displaying that guess to the users of varying experience levels?

TURNING THE PAGES

with David E. Laden

This issue signals the first anniversary of "TURNING THE PAGES." One year ago, this column began with entries from three computer magazines. Over the months, it has been expanded to include nine magazines containing material about or for the Apple computer.

(Congratulations Dave -- Your Editor)

Over this past year I have concentrated my efforts on the general computer magazines and on 6502 related periodicals. I have hesitated including entries from those periodicals devoted entirely to the Apple; because it would, in essence, be duplicating their table of contents page. It would not be the most efficient use of my time.

The magazines included regularly are:

Byte
 COMPUTE!
 Creative Computing
 Interface Age
 Kilobaud Microcomputing
 Micro
 onComputing
 Personal Computing
 Recreational Computing

Each entry generally consists of the following items: Article title, Author, Pages, a brief statement pertaining to the content of the article, and type of program (Integer, Applesoft, Pascal, Machine language) included with the article. The last two items are included when they are not obvious from the article title.

Now that you know a little more about this column, I would like to know a little about you (in a sense).

What do you think?

- Which magazines do you regularly read?
- Which magazines (not covered in this column) should be added?
- Which magazines (covered in this column) should be dropped?
- Which should be kept?
- What do you think about the format of the column? Do you have any suggestions for improvement?

I look forward to hearing some of your opinions in the next few months.

Well, enough of the comments...on to this months PAGES...

COMPUTE! -- APRIL 1981

Computers and Society by David D. Thornburg. Pages 10-13. Current and alternative keyboard arrangements are discussed.

The Beginner's Page by Robert Lock. Pages 14-16.

An Applications Commentary: Stimulating Simulations by Gregory R. Glau. Pages 18-22. "...finding the answers is not the problem... The difficulty is finding the problem itself..." Thirteen suggestions for simulation are included.

Resolving Applesoft and Hires Graphics Memory Conflicts by Jeff Schaefer. Pages 76-78.

Fill The Screen With Your Messages Generating Large ~~Multi-Colored~~ Characters Using Apple ~~Low-Resolution~~ Graphics by Francis A. Harvey, Rosann W. Collins, and Theodore C. Hines. Pages 80-86. Applesoft subroutines and program are included.

Decrementing The For...Next & Endless Loops by Derek Kelly. Pages 88-90. Short program examples are included.

CREATIVE COMPUTING -- APRIL 1981

The Centronics 737 by David Mannering. Pages 11-13. This is a review of the 737 printer from Centronics.

MPI 888 by Dale Archibald. Pages 14-17. A review of the MPI 888 printer from Micro Peripherals, Inc.

Project 80: 80 Columns for the Apple by Joe Alinsky and Winston Gayler. Pages 18-24. A comparison of the Videx Videoterm, Double Vision, and M & R Sup'rTerminal 80 column boards for the Apple.

ABM by Dale Archibald. Page 30. A new game from Muse is reviewed.

Wars In Space by Dale Archibald. Pages 37-40. In this article, some games set in outer-space are reviewed.

Apple as Time-Sharing User by James Parr. Pages 60-65. Some tips and programs are given to make communicating with a time-share system easier.

A Manager and His Machine by Robert Heltman. Pages 66-74. A manager writes of his experiences with obtaining and using an Apple computer in his office.

Calling Information: Telecomputing with Personal Computers by Tracy R. Licklider. Pages 78-84. This article describes some of the information, systems, and services available to owners of a computer and modem.

MCCCCBBS by Terrence Ryan. Pages 94-96. The Montgomery County Community College CBBS system for the Apple, its development, and its use are described in this article.

Landing Simulator by Jake Jacobs. Pages 156-166. Includes a hi-res graphics program written in Applesoft BASIC.

Apple-Cart by Chuck Carpenter. Pages 200-208.

Effective Writing: Good Writers Make Good Programmers by Wayne Dickson. Pages 218-219.

INTERFACE AGE -- MAY 1981

Al Baker's Game Corner: What's Your Reading Level? Pages 27-29. This short Applesoft program calculates the Fog reading level index for text you enter.

Learning with Micros by Louis E. Frenzel. Pages 34-36. This author takes a brief look at "real educational software."

Hardware Evaluation: Smart Plotter Eases Program Drudgery Wantanabe's Miplot Plotter by Roger H. Edelson. Pages 58-62.

The Micro As A Writing Tool by Tom Lukers, Ph.D. Pages 70-72 and 144. Although the author's system is not an Apple, this article is nonetheless a good introduction to text/word processing, its features, and characteristics.

Word Processing Apple-ications by Robert Moskowitz. Pages 92-99. The article contains a comparison of six word-processing programs for the Apple.

MICRO -- APRIL 1981

S-C Assembler Modifications by Ned W. Rhodes. Pages 7-10.

Apple Memory Maps, Part 1 by Peter A. Cook. Pages 27-35. A machine language program to create memory maps will be given in part 2.

Oh No - It's Garbage Collect! by Gordon A. Campbell. Page 43. Eventhough the author has a PET, the information still applies to the Apple. Some tips are given to reduce the annoying effects of garbage collect.

Integer BASIC Internals by Glen R. Sogge. Pages 65-66.

PERSONAL COMPUTING -- APRIL 1981

Editorial: Software Piracy: The problem is getting worse by Jules H. Gilder. Page 6.

Disk Memories: What You Should Know Before You Buy Them by Dave Bursky. Pages 20-27. This article is also good for those who just want general information about disk storage. Includes a Disk Glossary and Vendors Guide.

Simulation: How Can It Help? by Charles J. Wilson. Pages 29-31, 63-64, and 87-88. This last part in a series of three illustrates the theory/steps presented previously with a real-world problem. Unfortunately (for Apple users) a TRS-80 was chosen for the example program.

Estimate Trip Costs in BASIC or Pascal by Sam Gaylord. Pages 33-35. Both an Applesoft BASIC and an Apple Pascal program are listed for use and comparison.

Choosing An Assembler For Machine Language Programming by Tod Zipnick. Pages 40-45. The author discusses the many "characteristics" of assemblers. Included is a list of "27 Questions To Ask About Assemblers."

An Introduction To Printer Interfaces by Jules H. Gilder. Pages 46-48, 71, and 97. This article provides general information on the centronics parallel, IEEE-488 parallel, and serial RS-232C interfaces.

It's Not All Drill and Practice by Carol Klitzner. Pages 51-53. Three approaches to software and their relationship to software selection are covered: drill and practice, programmed instruction (tutorials), and using the computer as a tool.

How To Boot Binary Programs by Jules H. Gilder. Pages 56-57. The Integer BASIC program included modifies DOS to allow it to boot a binary "greeting" program.

I think that this should be enough PAGES for this month. It sure was enough for me, I just about kept reading right on through the newsletter article deadline. If you are reading this now, then I guess I did make it in enough time...

(Just -- your editor. But it was worth waiting for!)

PONDERING PASCAL

A Very Short Program
by Daryl Hammond

A couple of weeks ago I bought a joystick for my Apple. When I got home I decided to write a short Pascal program that would allow me to use the joystick with Turtlegraphics to create an Etch-a-Sketch effect. I quickly laid out the overall program structure in my head and then sat down to type it in. I hate to admit it, but I spent most of that Saturday afternoon getting the program to work. The problems I ran into included:

- Applestuff and Turtlegraphics procedures which give integer results and expect integer parameters,
- Internal computations that need to be done with reals if the results are to be useful,
- Finding an appropriate hardware timing delay.

In the Apple Pascal Language Reference Manual, on page 103, it is indicated that a hardware delay is necessary when you attempt to read one paddle and then the other. To do this the manual recommends the following code:

```
FOR I := 0 TO 3 DO;
```

I found that I needed a value of 4 here to avoid interference between paddles. In the program I have some computations between successive paddle reads and can get away with a smaller delay. You can

tell if you have too small a delay by trying to draw a border around the screen. I ended up with a ragged border in the lower left hand corner when my delay was too small. Happy doodling.

PROGRAM JOYSTICK;

```
(*****)
(* *)
(* Program : JOYSTICK *)
(* Written : 04-11-81 *)
(* By : Daryl Hammond *)
(* Phone : (612) 487-1002 *)
(* *)
(* This program takes input from the *)
(* paddles and converts it to *)
(* display on the screen using *)
(* Turtlegraphics. *)
(* *)
(* Note: The line that appears on *)
(* the screen when this program is *)
(* first run is caused by the *)
(* position of the paddles relative *)
(* to the center of the screen. *)
(* *)
(* Pressing button(0) clears the *)
(* screen. Pressing both buttons *)
(* exits the program. *)
(* *)
(*****)
```

USES TURTLEGRAPHICS, APPLESTUFF;

VAR

```
X,
Y,
I : INTEGER;
A,
B : REAL;
```

BEGIN

```
INITTURTLE;
PENCOLOR(WHITE);
REPEAT;
  REPEAT
    (* Read horizontal value and
       translate to turtlegraphics
       coordinate. *)
    X := PADDLE(0);
    A := X;
    X := ROUND(A * 279 / 255);

    (* Hardware delay *)
    FOR I := 0 TO 1 DO;

    (* Read vertical value and
       translate to turtlegraphics
       coordinate. Subtracting from
       191 causes the Y coordinate to
       match the orientation on my
       joystick. *)
    Y := PADDLE(1);
    B := Y;
```

PEEKs - POKES & CALLS

THANKS TO CHRIS ESPINOSA, PAUL WYMAN, & APPLE COMPUTER

CALLS

Most of these functions can be executed by a CALL to the proper address from an Interger BASIC program. Some CALLs may not be executed in the middle of a line in an Interger BASIC program: The CALL must be alone or at the end of a program line.

HEX ADD	CALL	EXPLANATION
\$E000	CALL -8192	Reset Integer BASIC Integer Basic Kill -Clear workspace
\$E048	CALL -8117	List Integer program
\$E836	CALL -6090	Run Integer
\$F07B	CALL -3973	Load Integer Program from tape
\$F140	CALL -3776	Save Integer Program to tape
\$F666	CALL -2458	Enter mini-assembler
\$F800	CALL -2048	Plot a point
\$F819	CALL -2023	Draw a horizontal line
\$F828	CALL -2008	Draw a vertical line
\$F832	CALL -1998	Clear Graphic Screen
\$F847	CALL -1977	Calc. graphics baseaddress
\$F85F	CALL -1953	Change Color +3
\$F864	CALL -1948	Adjust color byte for both halves equal
\$F871	CALL -1935	Read a point of the low res graphics screen 1
\$FA43	CALL -1469	Perform a single step
\$FA86	CALL -1402	IRQ Handler
\$FA92	CALL -1390	Break handler
\$FAD7	CALL -1321	Display user reg
\$FB1E	CALL -1250	Read paddle controls
\$FB2F	CALL -1233	Screen initialization Reset TEXT mode
\$FB39	CALL -1223	Set text screen
\$FB40	CALL -1216	Set graphics screen
\$FB48	CALL -1205	Set normal window
\$FB63	CALL -1181	Multiply routine
\$FB84	CALL -1148	Divide routine
\$FBC1	CALL -1087	Calculate text base address
\$FBE4	CALL -1052	Produce a bell
\$FBF4	CALL -1036	Advance Cursor
\$FBFD	CALL -1027	Output a reg as ASCII on text screen 1
\$FC10	CALL -1008	Backspace Cursor
\$FC1A	CALL -998	Move cursor up
\$FC22	CALL -990	Perform vertical tab
\$FC2C	CALL -980	Escape functions
\$FC9C	CALL -868	Clear to end of line
\$FC42	CALL -958	Clear to end of screen
\$FC58	CALL -936	Home and Clear
\$FC62	CALL -926	Carriage return
\$FC66	CALL -922	Cursor down-line feed
\$FC70	CALL -912	Scroll screen
\$FC9C	CALL -868	Clear to end of line
\$FCA8	CALL -856	Wait loop
\$FDC0	CALL -756	Wait for Keypress
\$FD1B	CALL -741	Monitor keyin routine
\$FD35	CALL -715	Read key & per esc fun if necessary
\$FD62	CALL -670	Perform a line cancel(/)
\$FD67	CALL -665	Perform Line Feed and wait for input
\$FD6A	CALL -662	Prompt and wait for input
\$FD6F	CALL -657	Get a Line
\$FDDA	CALL -550	Print a byte as 2 hex digits
\$FDE3	CALL -541	Print hex digit
\$FDE0	CALL -531	Output char. via user output
\$FDF0	CALL -528	Monitor char. output
\$FE2C	CALL -468	Perform memory move
\$FE36	CALL -458	Perform memory verify
\$FE5E	CALL -418	Disassemble 20 instr.
\$FE80	CALL -384	Set Inverse Mde

\$FE84	CALL -380	Set Normal Mde
\$FE80	CALL -336	Jump to basic
\$FE83	CALL -333	Continue basic
\$FE8F	CALL -321	Display registers A,X,Y,P,S Current Values
\$FEC2	CALL -318	Perform monitor trace
\$FEC0	CALL -307	Write to tape
\$FEFD	CALL -259	Read from tape
\$FF3A	CALL -198	Ring bell
\$FF4A	CALL -182	Save registers
\$FF59	CALL -167	Enter System Monitor, reset text mode
\$FF65	CALL -155	Enter System Monitor & Ring Bell
\$FF69	CALL -151	Enter System Monitor No beep
\$FF70	CALL -144	Scan input buffer
\$FF70	CALL -144	Scan the input buffer

The following CALLS require various address to be POKED into locations in the Apple's memory. These locations are organized into three "registers", called A1, A2, and A4. The actual address for A1, A2, and A4 can be found in the section on PEEKs.

\$FE2C	CALL -468	Move a Range of Memory
\$FE36	CALL -458	Verify Two Ranges of Memory
\$FEC0	CALL -307	Write a Range to Tape
\$FEFD	CALL -259	Read a Range from tape
\$FD92	CALL -622	Print A1 in Hexadecimal

The following subroutines require you to POKE several values into some memory locations. These locations represent the microprocessor's A,X,Y and Status registers, and the Stack Pointer (These are the same registers you can examine with the Monitor's CTRL E command). For the address of these locations see section on PEEKs.

\$FEB6	CALL -622	Call a Subroutine
--------	-----------	-------------------

(This loads the internal registers with the values you have POKED into locations 69 through 73, and jumps to the subroutine whose address has been POKED into locations 58 and 59 (PCL & PCH). The subroutine acts just as if it had been CALLED from BASIC directly, and returns accordingly.

\$FEBF	CALL -321	Display Registers (69 thru 73).
\$FDDA	CALL -550	Display the A-register in Hex (location 69).

POKES

HEX ADD	POKE DEC ADD	EXPLANATION VALUE LIMITS
\$0020	POKE 32, X	Window Left 0-39
\$0021	POKE 33, X	Window Width 0-40
\$0022	POKE 34, X	Window Top 0-24
\$0023	POKE 35, X	Window Bottom 0-24
\$0024	POKE 36, X	Cursor Horizontal
\$0025	POKE 37, X	Cursor Vertical
\$0026	POKE 38, X	GBASL Graphics baseaddress
\$0027	POKE 39, X	GBASH
\$0028	POKE 40, X	BASL Text screen base address
\$0029	POKE 41, X	BASH
\$002A	POKE 42, X	BAS2L Temp. base addr.
\$002B	POKE 43, X	BAS2H
\$0030	POKE 48, X	Color (0-15)
\$0031	POKE 49, X	Mode

POKES (CONT.)

\$0032 POKE 50, X COUT text modes
 255-Normal mode
 63-Inverse mode
 127-Flashing mode
 \$0033 POKE 51, X Prompt Character
 \$0035 POKE 54, L CSHL
 \$0037 POKE 55, H CSMH:CHARACTER OUTPUT
 \$0038 POKE 56, L KSWL
 \$0039 POKE 57, H KSWH:CHARACTER INPUT

The POKES below may be used before a CALL -327 to CALL a machine language program with parameters.

\$003A POKE 58, L PCL
 \$003B POKE 59, H PCH
 \$0045 POKE 69, X A-register
 \$0046 POKE 70, X X-register
 \$0047 POKE 71, X Y-register
 \$0048 POKE 72, X Status register
 \$0049 POKE 73, X Stack Pointer

POKES below (60-67) can be used in conjunction with various CALLS to perform many System Monitor commands from BASIC (see CALL list).

\$003C POKE 60, L A1L
 \$003D POKE 61, H A1H
 \$003E POKE 62, L A2L
 \$003F POKE 63, H A2H
 \$0042 POKE 66, L A4L
 \$0043 POKE 67, H A4H

C\$010 POKE-16368, X Clear Keyboard Strobe
 \$C020 POKE-16532, X Toggle Cassette output
 \$C030 POKE-16336, X Toggle Speaker
 \$C040 POKE-16320, X Strobe Game I/O
 \$C050 POKE-16305, X Set Graphics Mode
 \$C051 POKE-16303, X Set Text Modes
 \$C052 POKE-16302, X Full-Screen Graphics
 \$C053 POKE-16301, X Mixed Text and Graphics
 \$C054 POKE-16300, X Display Primary Page
 \$C055 POKE-16299, X Display Secondary Page
 \$C056 POKE-16298, X Set Block Graphics
 \$C057 POKE-16297, X Set Dot Graphics
 \$C000 POKE-16384, X Read Key to see which key.

A POKE to one of the locations below specifies the state of the output on the pin of the Game I/O Connector associated with that Annunciator. Set is +5 volts; clear is 0 volts.

\$C058 POKE-16296, X Clear,
 \$C059 POKE-16295, X Set Annunciator 0
 \$C05A POKE-16294, X Clear,
 \$C05B POKE-16293, X Set Annunciator 1
 \$C05C POKE-16292, X Clear,
 \$C05D POKE-16291, X Set Annunciator 2
 \$C05E POKE-16290, X Clear,
 \$C05F POKE-16289, X Set Annunciator 3

\$C054 X=PEEK(-16300) Primary page 1.
 \$C055 X=PEEK(-16299) Secondary page 2.
 \$C056 X=PEEK(-16298) LO-RES Graphics
 \$C057 X=PEEK(-16297) HI-RES Graphics
 \$C060 X=PEEK(-16288) Cassette Input
 \$C061 X=PEEK(-16287) Read Pushbutton controller 0
 \$C062 X=PEEK(-16286) Read Pushbutton controller 1
 \$C063 X=PEEK(-16285) Read Game I/O Pin 4

X = PEEK (218) + PEEK (219) x 256 sets X equal to the line number of the statement where an error occurred if an ONERRGOTO statement has been executed.
 IF PEEK (216)>127 THEN GOTO ----. If bit 7 at memory location 222 (ERRFLG) has been set true, then an ONERRGOTO statement has been encountered.
 POKE 216,0 Clears ERRFLG so that normal error messages will occur.
 Y = PEEK (222) Sets variable Y to a code that described type of error that caused an ONERRGOTO jump to occur.

The locations below may be PEEKed to determine what type of Apple Intelligent Interface card is installed in each slot. If X is 44 - Communications Card, 162 - Disk Controller, 72 - Heuristics Speechlab. Others not known at this time.

\$C100 X=PEEK(-16128) Slot 1
 \$C200 X=PEEK(-15872) Slot 2
 \$C300 X=PEEK(-15616) Slot 3
 \$C400 X=PEEK(-15630) Slot 4
 \$C500 X=PEEK(-15104) Slot 5
 \$C600 X=PEEK(-14848) Slot 6
 \$C700 X=PEEK(-14592) Slot 7

A PEEK at this location determines which Monitor ROM is installed in the Apple. If X is 0, then the Auto-Boot Monitor is in; if X is 1, then the original Apple Monitor ROM is installed.

\$FAFF X=PEEK(-1281) Monitor

The PEEKs below all require you to convert the values in two separate memory locations into one decimal number which BASIC can handle. This conversion involves two steps: first, obtaining the values in the locations; and second, amalgamating the two values into one decimal number. The following pairs of PEEKs give two values, stored in the two variables L and H.

\$0028 L=PEEK(40) BASL
 \$0029 H=PEEK(41) BASH-Cursor Base Address
 \$0036 L=PEEK(54) CSHL
 \$0037 H=PEEK(55) CSMH-Output Subroutine
 \$0038 L=PEEK(56) KSWL
 \$0039 H=PEEK(57) KSWL-Input Subroutine
 \$004E L=PEEK(78) RNDL
 \$004F H=PEEK(79) RNDL-Random Number Seed
 \$004A L=PEEK(74) LOMHML
 \$004B H=PEEK(75) LOMEMH
 \$004C L=PEEK(76) HIMHML
 \$004D H=PEEK(77) HIMEMH
 \$00CA L=PEEK(202) PPL
 \$00CB H=PEEK(203) PPH-Program Pointer
 \$00CC L=PEEK(204) PVL
 \$00CD H=PEEK(205) PVH-Variable Pointer
 \$00DC L=PEEK(220) LINEL
 \$00DD H=PEEK(221) LINEH-Line Number
 \$00EE L=PEEK(224) STL
 \$00E1 H=PEEK(225) STL-Beginning of Statement

P E E K S

HEX	PEEK	EXPLANATION
\$0020	X=PEEK(32)	Window Left
\$0021	X=PEEK(33)	Window Width
\$0022	X=PEEK(34)	Window Top
\$0023	X=PEEK(35)	Window Bottom
\$0024	X=PEEK(36)	Cursor Horizontal
\$0025	X=PEEK(37)	Cursor Vertical
\$0030	X=PEEK(48)	Color
\$0032	X=PEEK(50)	Video mode
\$0033	X=PEEK(51)	Prompt Character
\$C000	X=PEEK(-16384)	Keyboard
\$C010	X=PEEK(-16368)	Clear key. strobe
\$C050	X=PEEK(-16304)	Graphics mode
\$C051	X=PEEK(-16303)	TEXT mode
\$C052	X=PEEK(-16302)	all TEXT or GRAPHICS
\$C053	X=PEEK(-16301)	mix TEXT and a Graphics mode



LOCATING I/O PERIPHERAL CARDS
PRACTICAL COMPUTING, INC.
NEWSLETTER A80-001-01

C.E. Collins of CEC Systems,
Minneapolis (546-5694), a
local Apple consultant and
club member, provided this
article.

COPYRIGHT 1981 BY EARL ALLEN

THIS PUBLICATION MAY BE REPRODUCED PROVIDED THIS COPYRIGHT NOTICE IS INCLUDED.

In order to reference any peripheral device such as a printer, a disk drive, a modem, etc, a program must determine the I/O slot in which the peripheral card resides. Obviously, the program could ask the user; however, this is not always preferable in the case of a user not familiar with the Apple's I/O connections.

This newsletter describes a method whereby an Apple program can easily locate an I/O card by peeking at the Peripheral Card Prom locations documented on pages 80 and 81 of the Apple II Reference Manual (A2L0001A).

The Peripheral Card Prom area, beginning at 49152 (-16384), maps the Programmable ROM code actually residing on the I/O card ROM chips. Each I/O slot maps 256 bytes of a card's ROM code into this area. This ROM code can be tested via Peek instructions for determining the type of I/O card residing in each slot.

The example below demonstrates searching the programmable ROM space for either a standard Apple Serial card or a standard Apple Parallel card. If another vendor's card were inserted in the Apple, that card's code would not be detected by these instructions.

The example on page 12 searches all seven I/O peripheral slots and matches their ROM code against several commonly available cards. The program's data statements obviously do not reflect all possible cards available for the Apple. If any reader of this article is kind enough to provide ROM code for any additional cards, I will provide him with an update of this article.

The code to be verified is up to the programmer; however, multiple bytes should be tested since several I/O cards use similar code. Note that the Apple Graphics Tablet's code changes after twelve bytes depending on the slot in which it resides. It looks as if its coding occupies only twelve bytes in the 256 byte ROM area. Its additional code is setting in the 2K Expansion ROM space at \$C800.

```
10 FOR SLOT=1 TO 7:PK=49152+SLOT*256
20 IF (PEEK(PK)=24) AND (PEEK(PK+1)=176) AND (PEEK(PK+2)=56) AND
(PEEK(PK+8)=8) THEN PO=2:GOTO 100:REM FOUND PARALLEL
30 IF (PEEK(PK)=44) AND (PEEK(PK+1)=88) AND (PEEK(PK+2)=255) AND
(PEEK(PK+6)=144) THEN PO=1:GOTO 100:REM FOUND SERIAL
40 NEXT SLOT: REM DID NOT FIND APPLE SERIAL OR PARALLEL
```

The following table displays several common I/O cards with the first 16 bytes of their ROM code:

Vendor/card	ROM Code (in Hexidecimal)
Apple Parallel	18 B0 38 48 8A 48 98 48 08 78 20 58 FF BA 68 68
Apple Disk II (13 sector ROM)	A2 20 A0 00 A9 03 85 3C 18 88 98 24 3C F0 F5 26
Apple Disk II (16 sector ROM)	A2 20 A0 00 A2 03 86 3C 8A 0A 24 3C F0 10 05 3C
Apple Serial (W/O handshaking)	2C 58 FF 70 04 38 90 FE 18 88 08 78 86 35 48 8A
Apple Silentye	2C 58 FF 70 04 38 90 18 B8 48 8A 48 98 48 08 78
C. C. S. Serial (Asynchronous-7710A)	2C CB FF 70 04 18 B0 38 B8 48 8A 48 98 48 08 78
D.C. Hayes Modem	18 B0 38 B8 50 18 B0 38 2C 58 FF B0 03 8D 78 07
Mtn Hardware ROM +	2C 58 FF 38 70 04 38 90 18 B8 48 8A 48 98 48 08
Apple Graphics Tablet	18 B0 38 B8 08 78 2C FF CF 20 00 C8 ** ** ** ** **

]

RUN

SLOT 1 - CCS ASYNCH SERIAL
 SLOT 2 - EMPTY OR UNKNOWN
 SLOT 3 - D.C. HAYES MODEM II
 SLOT 4 - EMPTY OR UNKNOWN
 SLOT 5 - APPLE GRAPHICS TABLET
 SLOT 6 - APPLE DISK II - 16 SECTORS
 SLOT 7 - MTN HARDWARE ROM +

Mr. Allen has been programming for 17 years in languages such as ALGOL, FORTRAN, COBOL, Autocoder, PL/1, APL, BASIC, Assembler Language for the IBM 360/370/303X computer systems, and Assembler Language on the Apple II microcomputer. Currently he provides customized business programs and professional consultation for the Apple II microcomputer. In addition, he advises on subjects such as Data Processing contracts and acquisition of D.P. equipment/services. He may be consulted at 312-837-9259.

```

10 TEXT : HOME : REM SET MAX WINDOW CLEAR SCREEN
20 READ CARDCOUNT: DIM DESCR$(CARDCOUNT),CODE(CARDCOUNT,11): REM DETERMINE NUMBER OF CARDS AND DEFINE CARD DESCRIPTION AND
    RAYS
25 REM LINES 30 THRU 80 WILL BUILD CARD DESCRIPTION AND CODE ARRAYS FROM DATA STATEMENTS AT END OF PROGRAM
30 FOR LOOP = 1 TO CARDCOUNT
40 : READ DESCR$(LOOP)
50 : FOR INDEX = 0 TO 11
60 :: READ CODE(LOOP,INDEX)
70 :: NEXT INDEX
80 : NEXT LOOP
90 :
92 :
94 :
96 REM LINES 100 THRU 180 WILL PERFORM LOOP OF EACH I/O SLOT AND MATCH CODE AGAINST CODE ARRAY ENTRIES
100 FOR SLOT = 1 TO 7: REM DO ALL SEVEN SLOTS
110 : FOR LOOP = 1 TO CARDCOUNT: REM CHECK CODE AGAINST ALL CODE ARRAY ENTRIES
120 :: FOR INDEX = 0 TO 11: REM CHECK ON TWELVE BYTES
130 :: IF CODE(LOOP,INDEX) < > PEEK (49152 + INDEX + SLOT * 256) THEN 160: REM IF A BYTE DOESN'T MATCH, GO TRY NEXT CODE ARRAY
    TRY
140 :: NEXT INDEX: REM IF ALL TWELVE BYTES MATCH, THEN CARD TYPE KNOWN
150 : PRINT "SLOT ";SLOT;" - ";DESCR$(LOOP): GOTO 180: REM FOUND CARD MATCH - SAY SO AND GO ON TO NEXT SLOT
160 : NEXT LOOP: REM GO ON TO NEXT CODE ARRAY ENTRY
170 : PRINT "SLOT ";SLOT;" - ":"EMPTY OR UNKNOWN": REM CARD NOT FOUND - SAY SO AND GO ON TO NEXT SLOT
180 : NEXT SLOT: REM GO ON TO NEXT I/O SLOT
190 END
195 DATA 9
200 DATA "APPLE PARALLEL",24,176,56,72,138,72,152,72,8,120,32,88
210 DATA "APPLE DISK II - 13 SECTORS",162,32,160,0,169,3,133,60,24,136,152,36
220 DATA "APPLE SERIAL",44,88,255,112,4,56,144,254,24,184,8,120
230 DATA "D.C. HAYES MODEM II",24,176,56,184,80,24,176,56,44,88,255,176
240 DATA "MTN HARDWARE ROM +",44,88,255,56,112,4,56,144,24,184,72,138
250 DATA "APPLE GRAPHICS TABLET",24,176,56,184,8,120,44,255,207,32,0,200
260 DATA "APPLE SILENTYPE",44,88,255,112,4,56,144,24,184,72,138,72
270 DATA "CCS ASYNCH SERIAL",44,203,255,112,4,24,176,56,184,72,138,72
280 DATA "APPLE DISK II - 16 SECTORS",162,32,160,0,162,3,134,60,138,10,36,60

```



P. O. BOX 976, DALY CITY, CALIFORNIA 94017 USA

BULLETIN

Coming soon...

The next IAC Disk will include ATTACH BIOS 1.1 for your Pascal system. It also includes FORTFIX which swats the bugs in Apple's Fortran. This time the disk will be mailed to all clubs, associates and sponsors.

Hear ye, hear ye...

Apple is holding a traveling road show called Apple Expo 81. The first of four has already been held in Dallas on March 31. Hopefully you will see this in time for the next three. Next will be held at the Statler in New York on April 7, followed by the Hyatt Regency in Chicago on April 14, and then at the Biltmore in Los Angeles on April 28. General admission hours are from 1 - 9 PM. Presentations will include hands-on seminars covering applications, hardware, etc. for both the II and //+. There will be approximately forty major vendors of Apple associated hardware and software also exhibiting at the Expos. N.B. admission tickets are available at no cost through any Apple dealer. However, you will be charged \$10.00 admission if you show up without a ticket. There will be drawings for prizes and special offers by the vendors. See your dealer soon!

Bernie Urban, Ed.

LOGO

The following document was received through a timesharing network by your Editor and Jim White, a club member, and was written by Patrick G. Sobalvarro. It is in response to the questions:-

What do you know about Apple LOGO?

I've got some experience with it; I wrote parts of it.

I'm not quite sure what you want to know, so I'll tell you a few things about the software.

First, you need an Apple II or an Apple II Plus with 48K and a language card; a language card being one of those 16K memory extensions that comes with the Apple Pascal system or Apple Fortran system. I understand that ComputerCity markets something called an Andromeda board, which is a language card without the language; we haven't tested it yet, so I don't know how identical it is.

Apple Logo is an interpretive, procedure-oriented, list processing language with turtle graphics. The interpreter is quite fast; someone here ran some benchmarks and Logo came out ahead of Applesoft Basic, although I have no idea what the benchmark was.

Apple Logo is procedure oriented, like any Logo. Variables are dynamically scoped, as in Lisp. Internally, the language is very Lisp-like; essentially, we could make it a vanilla lisp by modifying the reader.

As you may know, the high resolution color graphics on the Apple are a complete kludge. Some colors cannot be made to appear on some screen locations, certain pairs of colors cannot be made to appear next to each other. So if you use Applesoft or Apple Integer Basic, or even Apple UCSD Pascal, you'll notice that your colors lose. With Basic they don't appear sometimes; with Pascal they appear in the wrong colors. We've taken the trouble to work out an algorithm that does the smartest thing possible in every situation. So color works in Apple Logo.

There is a screen-oriented procedure editor that one is automatically put into when defining or modifying a procedure; the editor is actually a subset of Emacs. Lots of fun.

Garbage collection is mark and sweep, and pretty quick (a garbage collect of nodespace takes less than half a second, I think).

There is a version that includes primitives for using the ALF Music card (6 voices of square waves); it's currently being used by Jeanne Bamberger's group at the Division for Study and Research in Education.

There is a company being formed to market Apple Logo; they're called LCSi (Logo Computer Systems, Incorporated).

EPSON BLOCK GRAPHICS

Another Way - by Lou Adornato

Like many other Apple owners, I was dissappointed to find that the Epson MX-80 could not use block graphics with the interface card supplied (by EPSON). I was further perplexed after looking over the schematic for the interface. It appears that the designer intentionally disabled this ability! After talking this oddity over with some other Epson owners, I still don't know why this was done, but I have come up with a "fix" that works without interfering with the normal operation of the card. This fix can be installed in about five minutes by anyone, and requires only a soldering iron, a small vice, a 16-pin wire-wrap socket, and a wire wrap tool.

First, locate IC 2A on the interface card. This is one of the 16-pin IC's just below the connector. You can find it by referencing figure 6 on page 10 of the booklet that came with the interface card. Now find pin 15 of this chip. This is just below the uppermost pin on the right side of the chip. Turn the board over, locate this pin again. (Remember, everything is reversed now).

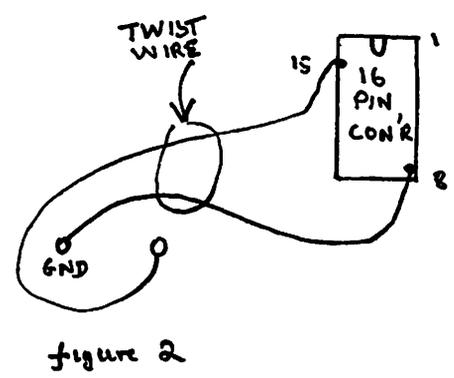
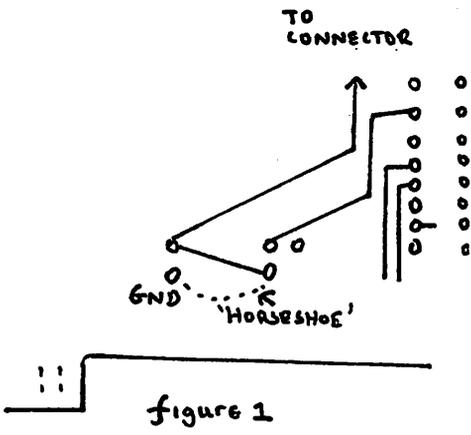
Follow the track coming from this pin with your finger until it stops. There you will find three tracks coming to an end in a triangular pattern. Two of these are connected by a small wire "horseshoe" on the topside. This is the object of our work.

One of these tracks leads to the connector, the other is the ground. See figure 1. The wide track on the IC side of the board is the ground.

Clamp the card (gently!) on the edge, or have someone hold it, while you grab the horseshoe with a small pair of needle-nose pliers. Carefully touch the soldering iron to the backside of the board at one end of the horseshoe, and pull with the pliers (wiggling them helps) till the heated end of the wire comes out of the hole. Now remove the other end.

Next, take 2 pieces of wire-wrap wire about 18" long, and mark the end of one with a piece of tape. Strip a half inch of insulation off of the other end and insert the bare wire into the GROUND hole from which the horseshoe came. The hole on the left is the ground as you view the board with the Apple 50 pin mother board edge connector downwards. You'll have to heat the solder in the hole as you insert the wire (leaving about 1/4" of bare wire), then hold the wire in place for a few minutes after you remove the iron. Now go through the same procedure (except for marking the wire) for the other hole.

Twist these two wires around each other loosely. About two turns an inch should do nicely. Strip an inch off of the free end of the marked wire and wire wrap it onto pin 8 of the wire wrap socket. Remember, as you hold the socket upside down, with the notch pointing away from you, pin 8 is pin closest to you on the right side. This corresponds to ground on the game I/O port. Repeat this procedure with the other wire, connecting it to pin 15 (pin 16 is furthest away, on the left, pin 15 is just below it). This is annunciator 0 in the game port. Again, marked wire (ground) to pin 8, unmarked wire (bit 8 to printer) to pin 15. See figure 2.



As the last step, plug the socket into the game I/O port, making sure the notch faces the front of the computer. Then plug the game paddles into the socket.

In order to run the printer normally, you have to POKE -16295,0. This forces the printer to see only the lower 256 ASCII values. To print the graphics, POKE -16296,1. This forces the upper 256 ASCII values.

One discouraging word, some disk access procedures leave the annunciator bit scrambled, so you'll have to be careful. Otherwise, your nice invoices turn into nice Op-art. (Maybe you could put a POKE -16295,0 in your HELLO program. -- Ed.)

As a last note, remember that this type of work will void the warranty on the interface, and possibly on the printer. So BE CAREFUL! Make sure you double-check everything.

1000 MICRO USERS

by Dan Buchler

On Wednesday April 22nd I attended a meeting of the Presidents of most of the Twin Cities' Micro User groups. 17 persons representing about 1000 local micro users including: TRS-80, PET, S-100, Heath specific computer clubs; 3M, Univac, and Honeywell micro user clubs; Minnesota Computer Society and some others. We, Mini'App'Les, were the largest club represented (by a factor of 2). Leroy Erickson, of the Univac club, did an excellent job organizing the meeting.

Topics discussed included:

- Bulk buying and purchasing power of 1000 users.
- A Joint Fair at the Science Museum
- Interchangeability of software including techniques for that interchange such as via Modem.
- Exchange of Newsletters.
- Monthly or bi-monthly meeting of the consortium
- Combined club meetings in order to attract key speakers.

Anyone wishing to make suggestions, please call your President or Newsletter Editor.

SPECIAL INTEREST ACTIVITY

By your editor

There was quite a bit of activity during the past month among the several Special Interest groups. This indicates a maturing of the club in terms of fulfilling needs for education, etc.

The Pascal group met as usual on the 1st Wednesday of the month.

Dave Nordval held a programming meeting on April 20th for residents in the Fort Snelling area.

C.B. Allen arranged a Disk Programming seminar at his residence (Apple Valley area) on 22 April. The seminar was conducted by Chuck Thiesfeld and was reported to be 'superb'.

Lets try and do more of this sort of things. Let the arrangers please note that, if you can plan ahead by as much as 3 weeks in front of a meeting, we can announce the special interest meetings in this newsletter.

Nano-BOOK REVIEW

by R. Hreka & D. Buchler

Some of you may have recently purchased books on Programming the Apple, BASIC Programming, Assembly Language Programming, 6502 Programming, etc. Please let us have your comments about such literature so that we can all benefit and know what is a good buy and what is not.

Prentice Hall has just published a new book called "Apple Machine Language" by Don and Kurt Inman. It is the first 6502 machine language programming book actually written around the Apple. It contains lots of examples oriented towards the Apple 'Mini-assembler'. The author makes heavy use of Graphics to improve the readability of the book and understanding of the technical matter. Price is around \$10.

DAN ON PRINTERS

Dan is still preoccupied with his Epson MX-80, so it is difficult to be unbiased on this subject.

One thing Dan learnt from the Epson purchase, that applies to any printer purchase, is that there are differences between the various interface cards available on the market. Whatever printer you buy, carefully review the specifications of the interface cards available. The cards made by Apple all have the advantage that they contain 'firmware', but there are other manufacturers who make boards with firmware and these are often cheaper than the Apple card, and sometimes offer additional features. 'Firmware' allows you to activate the printer with a simple

PR# n

where n is the slot number. If you don't have on board firmware, you must BLOAD a driver into memory before printing can commence.

With respect to interface cards for the Epson, you might consider the TYMAC which lists for \$139. It should be available at discount for about the same price as the Epson card and does offer one additional feature. That is, it has the ability to send the high order bit of the data word to the printer without any hardware changes such as described elsewhere in this newsletter. This allows the Epson Block Graphics to work with only a POKE being needed.

We have been deluged with flyers from DIP, inc who are promoting the sale of their DIP-84 printer. I think the Epson is a better buy, but here's a comparison anyway.

DIP-84	EPSON MX-80
-----	-----
\$595 Spring special	\$650 List - bulk prices thru club.
40,48,66,80,96 or 132 chars per line	40,80 or 132 chars per line
Dot graphics incl.	Dot graphics extra
7 wire head	9 wire head
Single sheet cap.	Single paper add as future upgrade
?	Enhanced/double strike mode

One printer which can not be dismissed when one is considering purchase, is the Centronics 737. It costs a bit more than the Epson, but does provide an excellent quality printout. It includes a serif mode which works in conjunction with proportional spacing. An example of this used in conjunction with the Muse Super-Text II word processing package is shown below. (With thanks to John Hansen who provided the example).

DEMONSTRATION OF SUPER-TEXT II

ELONGATION

UPPER CASE, and lower case, and Mixed Faces can be elongated to form headings, subheadings, etc.

UNDERLINING

Underlining can be done with all type faces, but I find it to work well with the Proportional Type style, which is used here.

SUPERSCRIPTS AND SUBSCRIPTS

One reason I chose Super-Text II was because I knew it would do superscripts and subscripts with the Centronics 737 printer, while I knew that Apple Writer would not and was not sure from the literature whether Easy Writer would do both. These are necessary features for me in my work as a chemist.

One should not drink too much $\text{CH}_3\text{CH}_2\text{OH}$ when driving. The penalty in some states can be more than 30² dollars.

As you can see, the characters are very well formed. This is achieved in part by a denser use of dots in the horizontal direction than that used by the Epson. Both the Centronics and the Epson MX-80 use a 9 wire head and thus 9 dots in the vertical direction. If you want a printer for word processing, and want to spend around \$700 to \$800, do consider the 737.

Please note though: none, I repeat NOT ONE, of the word processing systems on the market support proportional spacing with respect to column justification. The "Executive Secretary", from Personal Business Systems here in the Twin Cities, will be offering that capability later this year. (Word processing will be discussed in an upcoming edition of the newsletter)

Other new printer products:

MICROTEK are advertising for sale direct from factory only, the "Byte Writer" for \$299. Its a 7 wire printer that only takes roll paper. But its cheap! (See page 260/261 of May Byte magazine)

Radio Shack are offering their Line Printer VII for \$399. This is a 7 wire head printer with dot graphics capability, tractor feed, serial and parallel interface, wide or normal caharcters, 80 columns normal. However, note that the Epson, through our bulk deal, is not much more expensive!

EPSON NEWS

by Dan Buchler

See, elsewhere in this newsletter, an article by Lou Adornato on a simple hardware fix to make the block graphics available with the Epson interface. If it bothers you to do this yourself or with the help of a club member,

Computer Corner
439 Route 23
Pompton Plains, NJ, 07444

will do the modification for you - at a price of course!

THIS
SPACE
RESERVED
FOR
YOUR
ADVERTISEMENT

OVER
400
POTENTIAL
APPLE
CUSTOMERS
AT
YOUR
BIDDING!

This same feature will allow screen dumps with a minor modification to the program published last month by Chuck Thiesfeld. Again, one can buy commercially what is available through the club. Mirco-Ware, of same address as above, offers a screen dump program called "Super Pix for \$39.95 list.

According to reliable sources, the Epson dot graphics chip set is on the market. It has been seen in California. The chip set will retail (list) for about \$100 and will have an option for either 480 or 960 dots across the page. The latter is a higher density than available on most other printers that have a dot graphics feature.

Incidentally, Byte magazine, has a detailed review of the MX-80 and MX-70.

NEWS FROM AFAR

By your editor

1. According to recent rumours, substantiated by the April 15th edition of the Wall Street Journal, the following problems have been occurring with Apple IIIs:
 - Chip sockets were too 'loose' resulting in chips falling out or breaking contact. 20% of Apple IIIs were "dead on arrival".
 - The Clock Calendar Chip manufactured by National Semiconductor did not meet specification. Apple has given customers a \$50 rebate and has stopped using that chip.
 - The keyboard cable was too short.
 - Connectors had a variety of mechanical problems.

In attempting to fix the socket problem, Apple tightened the sockets so much, that assembly workers broke pins in trying to install integrated circuits. Apple fired the manager of computer operations and now claims that the reliability of the 'III' exceeds that of the 'II'.

Apple still has problems with software on the III. 'Word Painter', the new Word Processor for the 'III' is not expected to be available until December or January 1982!
2. Other unsubstantiated rumours. The 3 key officers of Apple Inc., Wozniak, Jobs and Markkula have switched around their job functions.
3. Anyone interested in Real Estate oriented software? Jerry L. Cooper 231 South Terrace, Wichita, Kansas 67218 tel: 316/685-3582 wants to get in touch with anyone developing Appraisal or Investment software for Real Estate on the Apple II.
4. International Apple Corps have released their "ATTACH-BIOS" diskette and documentation. The later is an 8.5" by 5.5" book of 45 pages. The Software provides Pascal users a means to insert printer drivers or other 6502 machine language routines into a Pascal 1.1 environment. Another module allows Apple Fortran users to modify their Fortran files to work with Pascal 1.1. Contact Keith Madonna, Pascal Special Interest group co-ordinator for futher information. Additional copies of the documentation may be purchased from IAC for \$3.00 each. Contact our club secretary, Ron Androff.
5. According to Byte Magazine, there's a company called Axlon Inc. of Sunnyvale, California that is building a 256k byte memory box. This device, when added to the Apple II, together with appropriate software, will allow it, an Apple II, to emulate an Apple III !
6. Data Ed is offering a 15Mhz Green Monitor at an introductory price of \$129. (See page 17 of May Byte magazine)

For Sale:

Apple Silentype Printer,
 New! Only \$550.
 Earl Keyser 429-5570 or
 633-9110 ext 155

Integer Basic ROM Card - \$150
 Contact Dag Johansen 447-5495

MINUTES APR MEETING

The meeting was called to order at 7:52 p.m., on wednesday March 15th 1981, by the incumbant President, Dan Buchler. The minutes of the past month's board and club meeting were approved.

-DOM #4 was sold at this meeting. Ken Slingsby is trying something new with the DOM's. There is a new DOS on the DOM's. This new DOS has 13 sectors per track, but can be booted on either 3.2 and 3.3 systems. Unfortunately, not all copy programs will copy this DOS. If you have a problem, this new DOS can be replaced with the old DOS using UPDATE 3.2 or MASTER CREATE. Chuck Theisfeld, copied the programs onto 5" disks from his hard disk using his own customized operating system.

-Dan Buchler announced that he was taking orders again for Printer Paper to be purchased in bulk.

The Minneapolis Star and Tribune is publishing a series of weekly (Tuesdays) articles on computing. Last week our 1980/1981 President was interviewed.

-Chuck Theisfeld distributed a VISICALC NEWSLETTER called "VISINEWS". Our thanks for making the club aware of this publication.

-WARREN OSTLUND, is selling a limited supply of protective disk boxes for \$9.00,per box. WARREN, can be reached at 926-3122.

-Max Coe is giving three very good courses at ZIM COMPUTERS, soon. (See elsewhere in newsletter).

-Chuck Boody has another new Music synthezizer board.

-Dan Buchler reported on the Committee to find another Meeting place. Al Peterson, who is on that committee with Terry Pinotti, brought up three suggestions, of which WILDER ELEMENTARY SCHOOL, seemed the most promissing. The school is centrally located at CHICAGO Avenue, and 34TH street in South Minneapolis. It is easy to get

to by street, or freeway. The second offering was the SOUTHDALe LIBRARY. Unfortunately it must be reserved about 3 months in advance. A motion from the floor was made and passed that the June meeting be held at a new location.

-Club Incorporation.. MIKE YOUNG, made a plea for professional lawyers, accountants, and insurrance agents to please contact him. It was mentioned that the club might consider a "NOT-FOR-PROFIT" INCORPORATION STATUS. The suggestions are being considered.

-It was mentioned that Chase Allen was holding a Programming Sub-Group Seminar at his home on April 22nd.

-Anybody using Compuserve, please contact the club about problems.

THE ELECTION RESULTS:::

The President presented the 1981/1982 roster of nominees for office. No further nominations wre received from the floor. A motion was made and passed unanimously to elect the nominees by acclamation.

- President ----- STEPHEN K. JOHNSON
- Past President -- DANIEL B. BUCHLER
- Vice President -- CHUCK THEIESFELD
- Treasurer ----- MARILYN THOMAS
- Secretary ----- RON ANDROFF

The newly elected President then formally nominated the Board members for the jobs listed in the last newsletter. See 'mast-head' for names. Job functions were listed in last newsletter.

A.MICHAEL YOUNG.

MEMBERSHIP APPLICATION FORM

Fill in and return to Membership Co'rd

Name: _____

Address: _____

Zip: _____

Home Telephone: _____

Office Telephone: _____

Apple II configuration: _____

Special interests: _____

Signature.....

ASSEMBLY LANGUAGE CLASS

Max Coe is again conducting an 8 week course on Assembly Language Programming. The primary goal is to teach the student how to follow through an Assembly Language program on the Apple and understand what is the function of the code. The course assumes that the student already has the ability to program in a high level language such as BASIC, APPLESOFT or PASCAL.

The course will be held at Zim Computer, 5717 Xerxes North
 Contact Zim Computer at 560-0336
 or Max Coe at 377-2107(home)
 631-5446(work)

APPLENET

Did you know that there is a group of Amateur Radio (Ham) Operators who own Apples and who have established a regular transmission of Apple Information. This takes place at 8pm every Sunday night at 14.329Mhz. However, don't try to listen-in unless you have high quality equipment. Some of the broadcasts originate in the Rockies, on the East coast, etc. High selectivity in your receiver is essential (This information is courtesy of Richard Hreka)

AMATEUR FAIR

May 30th, State Fair Grounds.
 Swapfest for Ham Radio and Computer Enthusiasts.

MX-80 BULK PURCHASE

Contact Al Peterman 721-3295

TABLE OF CONTENTS

DOM#4 by Ken Slingsby.....	3-4	Special Interest Activity.....	15
Turning the Pages		Nano Book Review	
with David Laden.....	5-7	by Richard Hreka.....	15
Pondering PASCAL		Dan on Printers.....	16,17
by Daryl Hammond.....	7	Epson News by Dan Buchler.....	17
PEEKs, POKES and CALLS.....	8,9	News from Afar.....	18
Locating I/O Peripheral Cards		For Sale.....	18
by Earl Allen.....	10-12	Minutes by A.Michael Young.....	19
LOGO by Patrick Sobalvarro.....	13	Announcements.....	1,20
Epson Block Graphics			
by Lou Adornato....	14,15		
1000 Micro Users			
by Dan Buchler.....	15		

MINI'APP'LES

Dan Buchler - Editor
 13516 Grand Avenue South
 Burnsville, Mn., 55337

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