mini'app'les

apple computer user group newsletter

SEPTEMBER 1979 VOL II No 8

Daniel B.Buchler, President & Activity Coordinator	890-5051
Chuck Thiesfeld, Treasurer & Newsletter Editor	(830-5020 (831-0009
Chuck Boody, Secretary	873-2227
Keith Madonna, Librarian	474-3876
Rob Wentworth, Program Editor	825-9086
James Henke, Technical Advisor	869-0361
Dean Anderson, Bibliographer	466-5562

Contributions Complaints Correspondance	D.Buchler 13516 Grand Ave S Burnsville, Mn 55337
Membership	C.Thiesfeld 8416 Xerxes Ave. S Bloomington Mn 55431

IN THIS ISSUE

	PAGE
Minutes	1
Messing With Printers	2
Guest	2
Easy Loading Tapes	3
Sure Line Input	3
Mail Apple Membership	3
Dan On Printers	4
Basic Course	4
Tape Recorder Counters	4

MINUTES

Minutes of the July Meeting

The meeting was called to order by D. Buchler. The minutes of June newsletter were approved as printed.

OLD BUSINESS:

1. The MIN'APP'LES USER BANK is now in operation. Make sure that you are on the list to use the bank then add what you can of your own.

2. The BASIC teaching program discussed at the previous meeting is still in limbo. Dan is looking into setting a program the club can use.

NEW BUSINESS:

1. An appeal was made for volunteers for 10 minute talks for next meetings program. It will be an application night where members talk on their applications for their APPLE. It's still not to late to call Dan.

2. Dan announced that BLOOMINGTON CABLE TV is interested in involving MINI'APP'LES in public service broadcasting. If anyone is interested in a great project contact Dan.

3. A suggestion was made that a group purchase of disks could be made at some savings, if there is enough interest. Talk to Dan if you are.

4. A question was asked if the APPLE is tax deductable. No conclusion was reached. This would be a good article for the newsletter if someone has any information on the subject.

5. A request was made for ID cards for MINI'APP'LES members. Bill Weldon volunteered to get price and availability information.

J. Henke made a motion to adjourn, and seconded by Rich Rowe at 8:04 P.M.

The program for the night was demonstrations of the APPLE TALKER, HAYES MICRO MODEM, and HEATHKIT printer

NEXT MEETING

The topic for the Sept meeting is "APPLICATIONS". We plan to have about 4 to 6 members talk for about 10 minutes each on their particular application for the Apple. In general these will be serious applications, not games, where the Apple is being used to solve a real life problem. We hope that the speakers will get ideas from the audience, and the audience ideas from the speakers. Please come and support your fellow members and bring a guest.

As usual the meeting will be at the
MINNESOTA FEDERAL SAVINGS AND LOAN
9th Ave South,
HOPKINS,
The date is SEPT 19th at 7:30Pm

The Oct 17th meeting will be a

The Oct 17th meeting will be a program exchange night.

MESSING AROUND WITH PRINTED OR DISPLAYED OUTPUT
(This is sort-of for beginners)

By D.Buchler

As evidenced by several questions recently submitted to me, many Apple Users are unfamiliar with the way in which the system uses the COUT vector locations $36 & $37. These two locations are fundamental to the way in which a printer driver is interfaced to BASIC; an upper/lower case display is tied into the system; the Hayes software ties into the system; or DOS connects/disconnects.

Assuming that DOS or those other goodies are not in use, and assuming also that you are not using some nonstandard I/O scheme like that employed by XPLO, then $36 & $37 will contain the address $FDF0. If you look in your red manual, you will see that $FDF0 is labelled COUT1. COUT1 is the entry point in the monitor to display on the CRT a single character. That character will be displayed at the current cursor position as determined by CH & CV. CH ($24) is the horizontal cursor position in the range 0 to 39. CV ($25) is the vertical cursor position in the range 0 to 23. After COUT1 has received the character and displayed it, CH & CV will be updated accordingly. BASIC, DOS, etc, when they are ready to do an output (this includes, PRINT statements, TRACEs, LISTs, MONITOR outputs such as memory dumps and disassembler printouts, and DOS PRINT D$ when MON C or I or O is in effect) will transfer control to the monitor location COUT1 with a JMP COUT. Again, refering to the red manual on page 86, you will see that COUT contains a JMP (0036) which means make an indirect Jump to $36 or in other words transfer control to the address contained at $36 & $37. When $36 & $37 contain $FDF0 , control passes to the Monitor and the character is output. Note that this happens once for each character being printed. (in the 6502 everything is done one character at a time).

When a print driver is incorporated into the system, we will normally require that the driver output a character to its device, then give control to the Monitor so it can display the character. This is accomplished as follows:

Store in $36 & $37 the starting address of the driver. This can be done by POKEing to 54 ($36) and 55 ($37) from the BASIC program the driver start location.

Code the driver to do its thing, and exit from the driver to $FDF0.

If you issue a PRINT command from inside a program while running under DOS, DOS itself must determine if there is a Ctl D in the output. Therefore the address in $36 & $37 is a pointer to the software which checks for the Ctl D and performs the disk I/O. This same software will eventually return control to the BASIC program issuing the PRINT command, or if MON C,I,O is in effect, will JMP $FDF0, and the output appears on the screen also ! The particular problem to be overcome is how to couple to a printer driver or other special display routine because we already have a pointer to DOS in $36 & $37. DOS has overcome this problem with some special software as follows. The address of the driver or special display routine is placed in $36 & $37. Then (in DOS 3.2) a CALL 1002 is executed from the BASIC program. 'CALL 1002' causes DOS to pick up the contents of $36 & $37 and save them in its own equivalent vector location. The CALL 1002 also replaces the contents of $36 & $37 with the original pointer back to itself. Then after DOS has done its thing, it returns control to the address saved by the CALL 1002 (This would be $FDF0 if the CALL 1002 was not executed). Thus we now have a situation where 3 different pieces of I/O processing take place namely: DOS looking for Ctl D and doing disk I/O if required, A special driver doing its output, The Monitor displaying the character on the screen.

For those of you who have ROM or EPROM based driver software such as that which comes on the APPLE serial interface card, or with the HAYES modem, you will be doing a PR#n. For example if your board is in slot 4, and you do a PR#4, what BASIC does is to Jump to $C400. The ROM software then places a $C402 into $36 & $37, the 4 of the $C402 being the slot number. $C400 also happens to be the address of the first location of the ROM memory on the board in slot 4. If the board was in slot 3, the address automatically is assigned to $C302, etc. The program in the ROM will do its thing and then transfer control to $FDF0 (in most cases). If you are running with DOS 3.2, you should follow the PR#n with a CALL 1002 before any disk I/O is performed, so that the C#02 is saved by DOS as explained earlier.

QUEST by Dick Meyer, Owatanna.

A new game program has been added to the MINI'APP'LES program library. It is based on Will Crowther's Superprogram 'Adventure'; Roger Chaffee's 'QUEST' copyright (c) 1978, published in Byte magazine July 1979 as a version for the PET.

Quest is simple intent: enter a cavern system, find a treasure, and return to the outside with it. Initial execution of the game will quickly show that it is not a simple task. The cavern is a system of 27 interconnected rooms or places, with movement from place to place selected by one-word directions via the keyboard. Each place is identified by a colorfull text description.

The game is benign in that hazards are not violent or harmful. You may get lost; the treasure may be stolen; you may think that you'll never find the treasure; you may think that it's impossible to exit with the treasure..... but, you will not be hurt or killed. Quest makes limited use of random numbers. This makes it uninteresting after several sessions, but the initial play is quite challenging. Once learned, it is enjoyable to observe someone new to Quest struggling with the cavern topology. Chaffe, in the Byte article, cautions about 'helping' a friend's Quest —It's good advice: help only when your friend is obviously stuck.

A MODIFICATION TO EASE LOADING TAPES

By Ken Slingsby, Welch, Mn.

I have found that a simple modification to my cassette recorder has made saving and loading programs much easier. I feel that others will also benefit, especially those new to computing. The modification allows one to hear the data tones as they are generated by the cassette interface in the Apple. This insures that the recorder's input cable is properly connected. The mod enables one to hear the data tones during a load. Thus one is freed of having to unplug the cassette output cable, search for the beginning of the leader, and reconnect.
The tones heard as a result of the mod are at a lower volume with the cables plugged into the recorder than without the cables. This makes the tones less obtrusive. The recorder's volume control probably will not need adjusting as is often the case without the modification.
To make the modification, disconnect the power to the recorder and the cables to the Apple. The cassette cables should be disconnected from the Apple as well a they will be used later to verify the mod. Turn the recorder upside down on a soft cloth and remove the screws holding the case halves together. On my recorder (Panasonic RQ-2309) there were four screws to remove. Separate the case halves.

Carefully trace the leads to the speaker. One is connected to ground. The other is probably connected to the output jack used for the Apple. There are usually three leads connected to this jack. The shell of the plug connects to ground. The third lead connects to the audio output. This last connection is the one of interest for this mod.
Perform the following two simple steps:

-1 Obtain a 100 and a 300 ohm resistor (1/4 watt or larger). You will only use one of these and some experimentation will be required to determine which one works best for you. Connect one lead of the resistor (you decide which one to start with) to the connection on the jack identified above.
Note that the higher value resistor will give you a lower volume of output.

-2 Connect the other lead from the resistor to the hot side of the speaker. You may have to splice a wire on the resistor to make it reach. Dress the lead away from other components.

To test the mod temporarily connect the power and plug the output cable to the recorder (and NOT into the Apple yet). Insert a recorded tape and press play. The volume should be considerably less with the cable connected than without.You may have to adjust the value of the resistor if needed. More resistance means less volume and vice versa.
When you are satisfied, disconnect the power and output cable and reassemble the recorder.

It must be noted that this modification will void the recorder warranty. If this bothers you wait for it to expire. After reassembly you will find that the easiest method of loading programs or data is to type LOAD (or aaaa.bbbbR), start the recorder, wait for the leader to start, and press RETURN. Simple isn't it?

A CURE FOR LINE INPUT by Ken Slingsby, Welch, Mn.

APPLESOFT II lacks a feature found in several other basics commonly called a "line input" capability. This patch will cure that problem. The following lines can be added to the program to prompt for input. Commas and other characters will not cause "EXTRA IGNORRED" or error messages to occur.Just replace INPUT statements with a GOSUB to this routine. After the GOSUB set the variable equal to A$

```
 10 GOTO 110
 20 POKE 511,191
 21  REM  Use  question  mark  as  prompt  char.
 25 CALL 64874
 26  REM  Get  a  line  with  prompt
 30 A$ = ""
 35 FOR I = 512 TO 767
 36  REM  Put  text  buffer  into  A$  array
 40  A$  =  A$  +  CHR$(  PEEK(I))
 45  IF  PEEK(I)  <>  141  THEN  35
 50 GOTO 60
 55 NEXT I
 60  A$  =  LEFT$(A$,  LEN(A$)  -  1)
 65 RETURN
110 GOSUB 20
120  PRINT  A$:    REM  User  program  goes  here.
130 END
```

The value of 191 in line 20 governs the character used as the prompt character for line input. One application of this routine is the ADDR LIST program currently on the MINI'APP'LES user bank. Somehow addresses do not look right without a comma between the city and state.

CREDIS go to :

1 Softsell Associates, 2022-79th Street, New York
2 Cideer Press for their list of Peeks, Pokes, and Calls

CALL APPLE needs members now so they are offering a membership deal this month $25.00 duringthe month of September good for next year also. From October 1st to December 31st the new membership is $27.50. Starting January the fee is $40.00 so get your order in today.

DAN ON PRINTERS by Dan

In last month's column Dan mentioned that he was experimenting with hooking up a $50 printer to his Apple. Well Dan is pleased to report that the interface and driver have been completed successfully. In fact, by the time you read this there may be as many as 10 units of the same type operational on Apples in the Twin Cities area. If anybody else is interested, please contact Dan. There are not too many units left. Arrangements can be made for the construction of an interface board at a resonable price.

For those of you who are interested, the unit will print 32 columns, at a print speed of 100 chars per second. Thruput is about 50 characters per second. The units started life as Red Owl checkout/cash register printers. They print from right to left and contain a one line buffer which is loaded in the opposite direction. They utilize a dot matrix printer mechanism made by Victor and interface electronics by Bunker Ramo. The listing in the article entitled Tape Recorder Counters was printed on Dan's unit. 32 column printers are fine for listings, mailing lists, memory dumps, and draft text output.

Metropoletan Community College offers BASIC programing.
1501 Hennepen Ave, Minneapolis, Minn.
Registration is from Aug. 17-Sept.17 341-7061.
The course includes BASIC programing techniques and runs for 11 monday evenings from Sept 17to Dec 10 for 3 credits.
cost is $38.25 The course uses MECC timesharing

TAPE RECORDER COUNTERS by Dan Buchler

During the course of distribution of the user bank tapes, it became evident that there were two type of counters used on recorders. My counter counts turns of the take-up reel, others count turns of the give-up reel. There seems to be a 50-50 distribution between the two techniques. The reason that the descrepancy surfaced was that the C-60 tapes contain alot of programs and the identification of position of the program on the tape is needed to make life bearable when using those tapes.However a mathematical conversion from readings on the take-up counter to readings to be expected on the give-up counter is easily accomplished using the following program.

```
LIST

10  REM PROGRAM TO COMPUTE TURNS
 OF TAKE UP V TURNS OF GIVE UP
20 L = 5
30 I = 0: REM  TURNS OF TAKE UP
40 T = 493.: REM   TURNS TO FILL
                C-60 REEL
50 R1 = .9375: REM   RADIUS WHEN
                FULL
55 R2 = .375: REM   RADIUS WHEN
                EMPTY
60 D = (R1 - R2) / T: REM   TAPE
                THICKNESS
70 S = 0: REM  TURNS OF GIVE UP
80  FOR R = R2 TO R1 STEP D
90  REM  R IS CURRENT RADIUS
100 I = I + 1
110 X = R / R1: REM   TURNS OF
                TAKEUP/T. GIVEUP
120 R1 = R1 - (X * D): REM   AVG
                RADIUS OF TAKEUP
130 S = S + X: REM   TOTAL TURNS
                OG GIVEUP
140  IF I # L THEN 170
150 L = I + 5: REM   SET PRINT
                INTERVAL
160  PRINT I, INT (S): REM  TAKE
 UP      TURNS V GIVE UP TURNS
170  NEXT R: END
```

```
RUN
 5   2
10   4
15   6
20   8
25  10
30  12
35  14
40  17
45  19
50  21
55  24
60  26
65  29
70  31
75  34
80  36
85  39
90  41
95  44
```

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