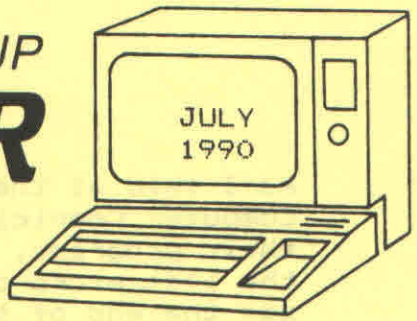
CEDAR VALLEY 99'ER USER GROUP
NEWSLETTER



CEDAR RAPIDS/MARION, IOWA

****

OFFICERS

PRESIDENT:
Gary Bishop
3270 28th Ave.
Marion, Iowa 52302
(319) 377-9574

VICE PRESIDENT:
John Johnson
398 Forest Dr. SE
Cedar Rapids, Iowa 52403
(319) 366-4541

SECRETARY:
Bill Paeth
923 Owen St. NW
Cedar Rapids, Iowa 52405
(319) 396- 6470

TREASURER:
Bruce Winter
702 Fernwood Dr. NE
Cedar Rapids, IA 52402
(319) 393-0610

COMMITTEES

PROGRAM:
Ed Edwards
102 N. Davis St.
Anamosa, Iowa 52205
462-2329

PUBLICITY:
Paul Mortensen
301 Pebble Lane
Hiawatha, Iowa 52233
393-6022

EDUCATION
Jerry Canad y
6616 Kent D r. NE
Cedar Rapids, Iowa 52402
377-9382

EDITOR:
Jim Green
377 Cambridge Dr. NE
Cedar Rapids, Iowa 52402
377-4073 (Home) or
395-1898 (Office)

****NEWSLETTER TOPICS****

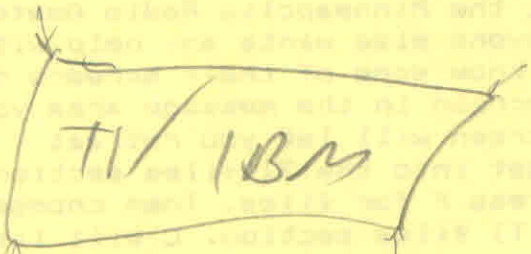
1. Future Meeting Dates
2. Next Meeting Notes
3. The Library Blurb
4. Of Clocks, Time, and Other Stuff
5. Minutes from the June Meeting

****FUTURE MEETING DATES****

Please mark the following dates on your calendar for future meetings: JULY 10, AUGUST 14, SEPTEMBER 12

********NEXT MEETING********

The regular monthly meeting will be TUESDAY, JULY 10, at West Music, Cedar Rapids, with open discussion starting at 6:30 PM. Come join the fun in air-conditioned comfort. Please remember to check your calendar to sign up for the Hamfest table coming in August.



COMPUTER TRIVIA OF MONTH

look up mic for VCR

TIPS FROM THE TIGERCUB

#54

TIGERCUB SOFTWARE
156 Collingwood Ave.
Columbus, OH 43213

Distributed by Tigercub Software to TI-99/4A Users Groups for promotional purposes and in exchange for their newsletters. May be reprinted by non-profit users groups, with credit to Tigercub Software.

Over 120 original programs in Basic and Extended Basic, available on cassette or disk, NOW REDUCED TO JUST $1.00 EACH!, plus $1.50 per order for cassette or disk and PP&M. Minimum order of $10.00. Cassette programs will not be available after my present stock of blanks is exhausted. The Handy Dandy series, and Color Programming Tutor, are no longer available on cassette.
Descriptive catalogs, while they last, $1.00 which is deductable from your first order.

Tigercub Full Disk Collections, reduced to $5 postpaid. Each of these contains either 5 or 6 of my regular catalog programs, and the remaining disk space has been filled with some of the best public domain programs of the same category. I am NOT selling public domain programs - they are a free bonus!
TIGERCUB'S BEST, PROGRAMMING TUTOR, PROGRAMMER'S UTILITIES, BRAIN GAMES, BRAIN TEASERS, BRAIN BUSTERS!, MANEUVERING GAMES, ACTION GAMES, REFLEX AND CONCENTRATION, TWO-PLAYER GAMES, KID GAMES, MORE GAMES, WORD GAMES, ELEMENTARY MATH, MIDDLE/HIGH SCHOOL MATH, VOCABULARY AND READING, MUSICAL EDUCATION, KALEIDOSCOPES AND DISPLAYS

NUTS & BOLTS DISKS

These are full disks of 100 or more utility subprograms in MERGE format, which you can merge into your own programs and use, almost like having another hundred CALLs available in Extended Basic. Each is accompanied by printed documentation giving an example of the use of each. NUTS & BOLTS (No. 1) has 100 subprograms, a tutorial on using them, and 5 pp. documentation. NUTS & BOLTS No. 2 has 108 subprograms, 10 pp of documentation. NUTS & BOLTS #3 has 140 subprograms and 11 pp. of documentation. NOW JUST $15 EACH, POSTPAID.

TIPS FROM THE TIGERCUB

These are full disks which contain the programs and routines from the Tips from the Tigercub newsletters, in ready-to-run program format, plus text files of tips and instructions.
TIPS (Vol. 1) contains 50 original programs and files from Tips newsletters No. 1 through No. 14. TIPS VOL. 2 contains over 60 programs and files from Nos. 15 thru 24. TIPS VOL. 3 has another 62 from Nos. 25 through 32. TIPS VOL. 4 has 48 more from issues No. 33 through 41.
NOW JUST $10 EACH, POSTPAID.

```
**************************
*       NOW READY        *
* TIPS FROM TIGERCUB VOL.5 *
* Another 49 programs and  *
* files from issues No. 42 *
* through 50. Also $10 ppd *
**************************
```

TIGERCUB CARE DISKS #1,#2,#3 and #4. Full disks of text files (printer required).
No. 1 contains the Tips news letters #42 thru #45, etc. Nos. 2 and 3 have articles mostly on Extended Basic programming. No. 4 contains Tips newsletters Nos. 46-52. These were prepared for user group newsletter editors but are available to anyone else for $5 each postpaid.

This program uses the program that writes a program technique to create a program that can be used over and over to create quiz programs.

When you key in the these routines, DON'T change any line numbers!

First key in this routine and run it to create a D/V 163 file named ASCII on the disk in drive 1.

```
100 OPEN #1:"DSK1.ASCII",VAR
IABLE 163,OUTPUT
110 FOR J=1 TO 125 :: X$=X$&
CHR$(J):: X2$=X2$&CHR$(J+125
):: NEXT J
120 PRINT #1:CHR$(0)&CHR$(23
0)&"X$"&CHR$(190)&CHR$(199)&
CHR$(125)&X$&CHR$(0)
130 PRINT #1:CHR$(0)&CHR$(24
0)&"X2$"&CHR$(190)&CHR$(199)
&CHR$(125)&X2$&CHR$(130)&"J$
"&CHR$(190)&"X$"&CHR$(184)&"
X2$"&CHR$(0)
140 PRINT #1:CHR$(255)&CHR$(
255)
```

Next, key in this part -

```
220 CALL CLEAR :: CALL SCREE
N(5):: FOR SET=1 TO 12 :: CA
LL COLOR(SET,2,8):: NEXT SET
 :: DIM L$(250,4)
230 !skip to line 280!
280 READ M$ :: DISPLAY AT(2,
14-LEN(M$)/2):M$ :: FOR J=1
TO C :: READ M$ :: DISPLAY A
T(6+J,4):J;M$ :: NEXT J
290 DISPLAY AT(12,1):"Catego
ry to match? (1-"&STR$(C)&")
" :: ACCEPT AT(12,26)SIZE(1)
VALIDATE("1234"):M :: IF M>C
 THEN 290
300 IF C=2 AND M=1 THEN A=2
:: GOTO 320 ELSE IF C=2 AND
M=2 THEN A=1 :: GOTO 320
310 DISPLAY AT(14,1):"Match
against (1-"&STR$(C)&")" ::
ACCEPT AT(14,21)SIZE(1)VALID
ATE("1234"):A :: IF A>C OR A
=M THEN 310
320 DISPLAY AT(16,1):"How ma
ny choices? (2-5)" :: ACCEPT
 AT(16,25)SIZE(1)VALIDATE("2
345"):CH :: IF CH>N-1 THEN 3
20
330 FOR J=1 TO N :: FOR L=1
TO C :: READ L$(J,L):: NEXT
L :: NEXT J
340 X$=SEG$(J$,1,N):: FOR J=
1 TO CH :: RANDOMIZE :: X=IN
T(LEN(X$)*RND+1):: Y(J)=ASC(
SEG$(X$,X,1)):: X$=SEG$(X$,1
,X-1)&SEG$(X$,X+1,255):: NEX
T J
350 Z=INT(CH*RND+1):: IF L$(
Y(Z),1)=Y$ THEN 350 ELSE Y$=
L$(Y(Z),1)
360 DISPLAY AT(8,1)ERASE ALL
:L$(Y(Z),M):: FOR J=1 TO CH
:: DISPLAY AT(10+J,4):J;L$(Y
(J),A)
370 NEXT J :: DISPLAY AT(23,
1):""
380 DISPLAY AT(20,1):"(1-";S
TR$(CH);")?" :: ACCEPT AT(20
,8)SIZE(1)VALIDATE(DIGIT):Q
:: IF Q=0 OR Q>CH THEN 380
390 IF L$(Q,M)<>L$(Z,M)THEN
410 :: DISPLAY AT(23,1):"COR
RECT!"
400 CALL SOUND(100,659,5)::
CALL SOUND(100,784,5):: CALL
 SOUND(400,1047,5):: GOTO 34
0
410 DISPLAY AT(23,1):"WRONG!
" :: CALL SOUND(300,110,0,-4
,5):: GOTO 380
```

Enter MERGE DSK1.ASCII and then SAVE DSK1.QUIZ,MERGE

Then key in -

```
100 OPEN #1:"DSK1.QUIZ",VARI
ABLE 163,INPUT :: OPEN #2:"D
SK1.QUIZ/2",VARIABLE 163,OUT
PUT
110 FOR J=220 TO 410 STEP 10
 :: LINPUT #1:M$ :: CALL LIN
E(J,LN$)
120 PRINT #2:LN$&CHR$(156)&C
HR$(253)&CHR$(200)&CHR$(1)&"
1"&CHR$(181)&CHR$(199)&CHR$(
LEN(M$))&M$&CHR$(0):: NEXT J
130 PRINT #2:CHR$(255)&CHR$(
255):: CLOSE #1 :: CLOSE #2
140 SUB LINE(LN,LN$):: LN$=C
HR$(INT(LN/256))&CHR$(LN-256
*INT(LN/256)):: SUBEND
```

Run that to convert the merge file QUIZ into another merge file QUIZ/2. Then key this in -

```
100 CALL CLEAR :: CALL SCREE
```

OF CLOCKS, TIME, AND OTHER ESOTERIC STUFF . . .

This is the first of what I hope will be several articles about the real time clock available for the TI. It has appeared in several hardware forms, such as kits, add-ons to expansion boards, inside speech synthesizers, etc. The real time clock is such a useful device, that it is difficult for me to imagine operating the TI without it. My plan of attack is: to describe the hardware and its uses, provide some software to utilize the clock, and then a uniform examination of how to treat the usage of time in programs. Along the way, I have several "wish list" items I will discuss. Maybe these discussions will trigger even more ideas, with the result that dreams will become reality.

The "clock" I mention is not a wall hanging pendulum type, nor is it a wind up or water powered one. The "clock" is a circuit consisting of a National Semiconductor MM58167 integrated circuit. This chip has many interesting features, such as extremely low power for battery back up operation, alarm capability, snooze features, and a host of other interesting capabilities. The complete circuit for the clock includes the MM58167, a couple of garden variety decoder chips, a crystal, a couple of diodes, and a small battery of some sort. The MM58167 is the same clock chip used in the IBM PC's, so it must be a solid and worthwhile device. I will talk later on some IBM style uses that can be incorporated into our TI. I have not included the schematic for the clock, so if you are a hardware builder and want to try your hand at construction, contact me for the details. The Zeno board will hold the circuit easily, but it can be installed almost anywhere. Most of the components are available at Radio Shack, and their blank PC board can also be used. The Triple Tech card from Rave also has clock capability. I know there are probably several other sources for the clock, but I can't recall all of them at this writing. The operation of all of them is the same, as far as I know.

The MM58167 uses a battery to keep time when the power is turned off. Although I use a small rechargeable nicad battery, coin sized watch batteries will also work just fine. Expect to get at least one year's use from a nonrechargeable battery. The coin batteries are also available at good ole Radio Shack, and are quite reasonable. For the operation of the clock, it doesn't matter which type of battery is used.

The clock is connected so that it can be read from and written to, just like a RAM memory location in the computer. The clock doesn't use any existing RAM, however. It shares addresses with the sound generator chip. Our computer reserves a large block of addresses just to make the sound chip work. The sound chip will respond to any address in the block. All software is written to make sounds by sending data to only the first address in the block. The sound chip will still respond to any address in the block, however. Well, the clock is connected to an address in the middle of the block reserved for the sound chip. What this means is that software using the sound chip will not affect the clock, but clock users will affect the sound chip. The sound chip is not a highly critical component, so all that has to be done after using the clock is to reset the sound chip. Easy! But remember, it must be done after every access or groups of access to the sound chip. If not, the sound chip will simply continue to howl and growl until you reset it. No big deal. So remember, the clock "sneaks" in on some of the addresses for the sound chip.

The clock can be accessed from Extended Basic with Call Peek/Call Load statements directly. Of course, assembly language, FORTH, console Basic with either Editor Assembler or Minimemory modules, and almost any other language can make use of the clock. Two of the most basic uses for the clock are to set the correct time, and to read the correct time. I haven't mentioned

Max	Units
5.5	V
5.5	V
10	μA
20	μA
5	mA
0.8	V
V_{DD}	V
1	μA
0.4	V
	V
	V
1	μA
0.4	V
10	μA

Tens D5	D6	D7	Max BCD Code
D5	D6	D7	9
D5	D6	D7	9
D5	D6	-	5
D5	D6	-	5
D5	-	-	2
-	-	-	0
D5	-	-	3
-	-	-	1

Functional Description (Continued)

TABLE II. ADDRESS CODES AND FUNCTIONS

A4	A3	A2	A1	A0	DECIMAL	HEX	Function
0	0	0	0	0	-31168	8600	Counter—Ten Thousandths of Seconds
0	0	0	0	1	-31166	8602	Counter—Hundredths and Tenths of Seconds
0	0	0	1	0	-31164	8604	Counter—Seconds
0	0	0	1	1	-31162	8606	Counter—Minutes
0	0	1	0	0	-31160	8608	Counter—Hours
0	0	1	0	1	-31158	860A	Counter—Day of Week
0	0	1	1	0	-31156	860C	Counter—Day of Month
0	0	1	1	1	-31154	860E	Counter—Month
0	1	0	0	0	-31152	8610	RAM—Ten Thousandths of Seconds
0	1	0	0	1	-31150	8612	RAM—Hundredths and Tenths of Seconds
0	1	0	1	0	-31148	8614	RAM—Seconds
0	1	0	1	1	-31146	8616	RAM—Minutes
0	1	1	0	0	-31144	8618	RAM—Hours
0	1	1	0	1	-31142	861A	RAM—Day of Week
0	1	1	1	0	-31140	861C	RAM—Day of Month
0	1	1	1	1	-31138	861E	RAM—Months
1	0	0	0	0			Interrupt Status Register
1	0	0	0	1			Interrupt Control Register
1	0	0	1	0			Counters Reset
1	0	0	1	1			RAM Reset
1	0	1	0	0			Status Bit
1	0	1	0	1			GO Command
1	0	1	1	0			$\overline{\text{STANDBY INTERRUPT}}$
1	1	1	1	1			Test Mode

All others unused

yet that the clock also handles the date! Years, too. I haven't seen any discussions on keeping the year, so I'll include that a little later on.

Now, suppose you have built this little jewel, but want to verify its operation before trusting it. I have included a short test program to do just that. Type it in, ground pin 10 on the clock chip, then run this program. The results should look like the printout.

```
1 CALL INIT
80 OPEN #1:"PIO",DISPLAY,VARIABLE 80, OUTPUT
81 REM INSERT YOUR PRINTER SPECIFICATION HERE
85 PRINT #1:"ADDRESS";TAB(12);"128";TAB(20);"64";TAB(28);"32";TAB(36);"16";TAB
(44);"8";TAB(52);"4";TAB(60);"2";TAB(68);"1" ::
86 REM SETS UP HEADER
90 FOR A=-31168 TO -31138 STEP 2 ! RANGE OF ADDRESSES FOR CLOCK
91 PRINT
92 PRINT #1:A;
93 FOR D=7 TO 0 STEP -1
94 REM TEST ONE BIT AT A TIME AT EACH ADDRESS
95 DB=2^D
96 REM CALCULATE THE CORRECT WEIGHT FOR THE BIT POSITION
100 PRINT DB:
110 CALL LOAD(-31130,255,0,255,0,"",A,DB,0)
111 REM WRITE DATA OUT TO THE CLOCK
140 CALL PEEK(A,DA)
141 REM GO GET THE DATA BACK FROM THE CLOCK
143 CALL SOUND(1,20000,30)
144 REM RESET THE SOUND GENERATOR
145 PRINT DA
150 PRINT #1:TAB((7-D)*8+12);DA;
155 NEXT D
160 NEXT A
170 CLOSE #1
```

Now, don't rag on me about my line numbering. Anyway, a correctly operating and wired clock chip will produce the following results on your printer:

ADDRESS	128	64	32	16	8	4	2	1
-31168	128	64	32	16	0	0	0	0
-31166	128	64	32	16	8	4	2	1
-31164	0	64	32	16	8	4	2	1
-31162	0	64	32	16	8	4	2	1
-31160	0	0	32	16	8	4	2	1
-31158	0	0	0	0	0	4	2	1
-31156	0	0	32	16	8	4	2	1
-31154	0	0	0	16	8	4	2	1
-31152	128	64	32	16	0	0	0	0
-31150	128	64	32	16	8	4	2	1
-31148	128	64	32	16	8	4	2	1
-31146	128	64	32	16	8	4	2	1
-31144	128	64	32	16	8	4	2	1
-31142	0	0	0	0	8	4	2	1
-31140	128	64	32	16	8	4	2	1
-31138	128	64	32	16	8	4	2	1

Yeah, I know my columns aren't straight. So fix it! In any event, the program reads and writes every bit available in every location of the clock. Notice

that some addresses don't return all the bits that were written. This is because some locations need fewer than 8 total bits to accomplish the job. An example of this is for the day of the week: the count to represent the day of the week only needs to go to seven, so there is no need to have any bits that are 8 or higher. By the way, the day of the week location is address -31158.

I have included a copy of the data sheet for the MM58167, with the decimal addresses used in the Call Peeks/Call Loads. I have also included the hex address. I may be just an old hacker at heart, but I work in hex much better than I do in decimal, at least when it comes to addresses inside the TI. Addresses beyond -31136 are for special purposes, and I will not discuss them here.

OK. Your clock is built or bought, and passes the above test. Now what? The following two programs are by John Willforth. Remember to remove the ground from pin 10 of the MM58167 if you installed it for the above test.

```
10 REM ********************
20 REM **                  **
30 REM **   CLOCKSET       **
40 REM **                  **
50 REM ********************
100 REM THIS PROGRAM IS USED TO SET THE CLOCK
110 REM THIS PROGRAM NEED ONLY BE RUN IF THE TIME DRIFTS OR WHEN BATTERIES ARE
 REPLACED.
120 CALL INIT
130 DEF SET=X+6*INT(X/10) ! CONVERTS DECIMAL TO BCD FOR OUTPUT TO CLOCK
140 INPUT "MONTH 1-12 ":X
150 MO=SET
160 INPUT "DAY OF MONTH 1-31 ":X
170 D=SET
180 INPUT "DAY OF WEEK 1-7 SUN=1 ":DW
190 INPUT "HOUR 0-23 ":X
200 H=SET
210 INPUT "MIN 0-59 ":X
220 M=SET
230 INPUT "SEC 0-59 ":X
240 S=SET
250 CALL LOAD(-31164,S,0,M,0,H,0,DW,0,D,0,MO) ! SET CLOCK
260 CALL SOUND(1,20000,30) ! RESET THE SOUND GENERATOR
270 CALL CLEAR
280 STOP
```

```
10 REM ********************
20 REM **                  **
30 REM **      TIME        **
40 REM **                  **
50 REM ********************
100 REM THIS PROGRAM DEMONSTRATES HOW THE CLOCK IS ACCESSED BY A PROGRAM.
110 CALL CLEAR
120 DIM WK$(7),MO$(12)
130 DEF TIME=X-6*INT(X/16) ! CONVERTS BCD OUTPUT OF CLOCK TO DECIMAL
140 FOR DW=1 TO 7
150 READ WK$(DW)
160 NEXT DW
170 FOR DM=1 TO 12
180 READ MO$(DM)
190 NEXT DM
200 DATA Sun,Mon,Tues,Wednes,Thurs,Fri,Satur
210 DATA January,February,March,April,May,June,July
220 DATA August,September,October,November,December
230 GOTO 380
240 CALL PEEK(-31164,X1,X2,X3,X4,X5) ! READ SEC. MIN. &&HOUR X2&&X4 NOT USED
250 X=X1 :: SEC$=STR$(TIME)
260 IF X1<10 THEN SEC$="0"&&SEC$
270 X=X3 :: MIN$=STR$(TIME)
280 IF X3<10 THEN MIN$="0"&&MIN$
290 X=X5 :: HR=TIME :: M$=" AM"
300 IF HR>11 THEN M$=" PM"
310 IF HR=0 THEN HR=12
320 IF HR>12 THEN  HR=HR-12
330 HR$=STR$(HR)
340 TI$=HR$&&":"&&MIN$&&":"&&SEC$&&M$
350 DISPLAY AT(6,11):TI$
360 IF X1+X3+X5=213 THEN 380
370 GOTO 240
380 CALL PEEK(-31158,X1,X2,D,X4,X5) ! READ DAY, DATE, && MONTH
390 X=D :: D$=STR$(TIME)
400 X=X5 :: X5=TIME
410 L1$="Today is "&&WK$(X1)&&"day"
420 L1=INT((32-LEN(L1$))/2)
430 DISPLAY AT(2,L1):L1$
440 L2$=MO$(X5)$" "&&D$&&",, 1988"
450 L2=INT((32-LEN(L2$))/2)
460 DISPLAY AT(4,L2):L2$
470 GOTO 240
```

Up until now, I haven't seen anyone attempt to use the year in a clock program. There are certainly registers available to do this, but how is it handled? To find out, I looked into how the year is stored and updated on the IBM. To accommodate rolling over the date and year, address -31150 is used to store the last month in which the clock was accessed, and address -31148 is used to store the last year the clock was read. This means that every time the clock is used, these two addresses should be checked to see if the year has rolled over or not. More on this later.

My suggested improvements to the above two programs mainly incorporate the year storage and updating. In the CLOCKSET program, substitute or include the following lines:

```
120 CALL PEEK(8198,A) :: IF A<>170 THEN CALL INIT
131 INPUT "YEAR 00-99 ":Y
250 CALL LOAD(-31164,S,0,M,0,H,0,DW,0,D,0,MO,0,0,0,MO,0,Y) ! SET CLOCK
```

So as not to confuse this with John's original version, I'll call this enhancement CLOCKSET1.

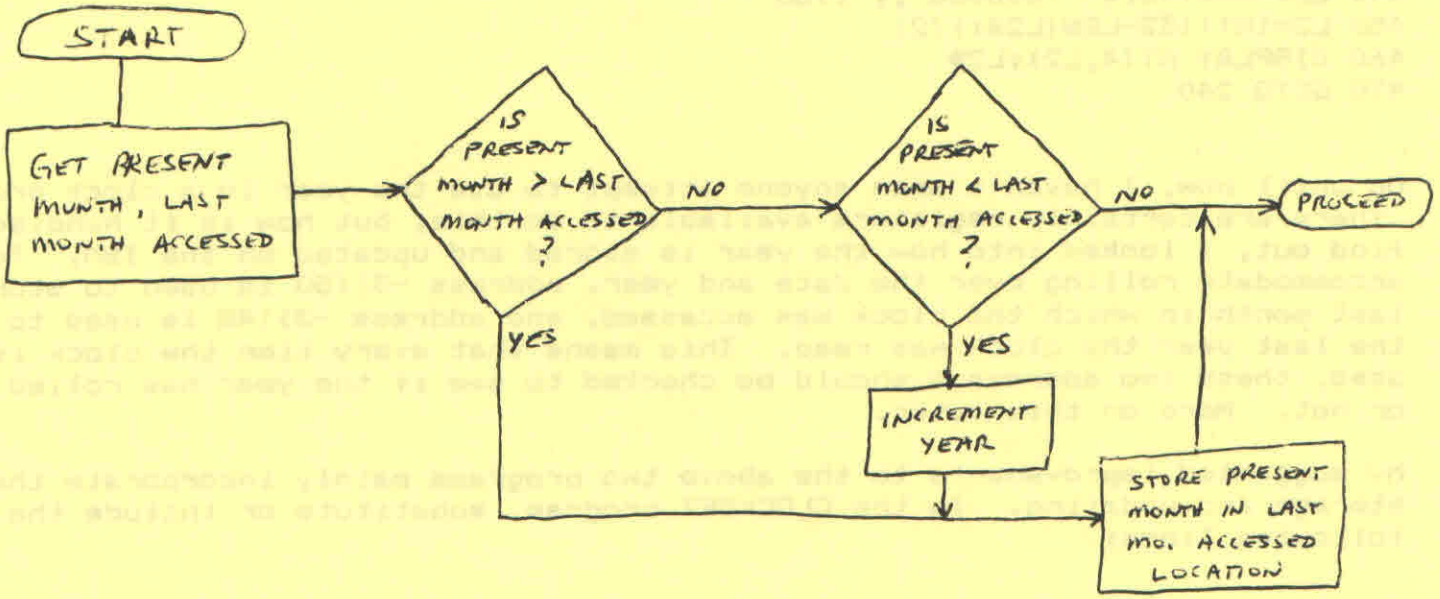
Substitute the following lines in the TIME program to include years:

```
380 CALL PEEK(-31158,X1,X2,D,X4,X5,X2,X2,X2,X2,X2,Y) ! READ DAY, DATE, MONTH,YR
440 L2$=MO$(X5)&&" "&&D$&&", 19"&&STR$(Y)
```

Again, to avoid confusion, I'll call this version TIME1.

The clock chip does not have provisions to keep track of the year. How is it performed on the IBM? Well, if the alarm or timer functions are not being used, addresses -31152 thru -31138 are not needed. That means they can be used to store the year, or anything else, for that matter. These 8 locations can hold any important data you want to survive a reboot or power removal. A "mailbox" for the last filename used in TI Writer, some important status information, or ? In any event, I suspect that most people familiar with the clock didn't realize that these locations were available. There are a total of six full 8-bit locations for storage, addresses -31150, -31148, -31146, -31144, -31140, annd -31138. There is only a high nybble at address -31152, and only a low nybble at -31142. If you want to know, ask me at the next meeting what a nybble is.

Back to years. IBM uses address -31150 to store the month the clock was last accessed, and -31148 for the last year accessed. Now what happens is when ever the clock is read for any reason, the active and battery powered month, which is kept current, is compared to the month the clock was last accessed. If the months are the same, do nothing. The assumption here is that the clock will be used at least once a year, not an unreasonable assumption. If the present month is later than the last month accessed, store the present month as the last month accessed. If the last month accessed is before the present month, the assumption should be that we have rolled past the end of a year, and the year needs to be incremented, along with storing the present month in the last month accessed location. This is harder to explain than to draw, so I have included a flow chart:



Crafty, huh? Let it not be said that we can't borrow the best ideas of the IBM world to keep our orphan alive.

One wish list item I have been planning is to have REMIND ME! on a single console, read the present date and time, and provide the proper display for today. The program may have to be stored in a Minimemory, or some sort of pared down ram disk. My goal would be to have the program autoload, just like John Johnson's MENU program for a ramdisk. I also have a relay installed in the modulator on the back of the TV, eliminating the need to reach back there to activate the computer. All I want to do is turn the TI on, and it automatically switched onto the TV, runs REMIND ME! for today's date, and enter any future events. When I'm done, I just want to turn the computer off, and get back to watching the morning news, or whatever. Talk to me out there, and let's see what can be done.

Gary Bishop

* MINUTES FROM THE JUNE MEETING *

President Gary Bishop called the June meeting to order on Tuesday the twelfth. There were 10 present at the meeting.

The chair, hearing no complaints about the May minutes as printed in the NEWSLETTER, declared them accepted.

The treasurer then announced that he had received the bank statement and had prepared a formal report. The report was accepted as read.

OLD BUSINESS: 1. Gary announced that he still has some items for sale from Bill Gonda. Included is a console. It was moved, seconded and passed that the UG offer Bill $15.00 for the console for UG use as a backup. 2. Bob started a short discussion on the Darkside BBS. Several have tried it. Now is your chance. 3. Sr. Pat sent a letter to Gary which was shared with the UG. Included was a copy of her picture which someone had digitized. She also was donating several items to the Oakland UG. 4. The DOS (Disk of the Summer) is here! Get your copy while they are in stock. By voice vote, it was decided that the UG will furnish the disks and the cost for the DOS would be $1.

NEW BUSINESS: 1. Gary reported the Ham Fest is just about two months away. The UG voted to support a TI table at the fest. We now need people to man the table on the first weekend in August. Sign up at the next UG meeting. 2. John reported he has contacted the Oakland BBS and given them "Hi from the Hawkeye state". 3. Gary is looking for support for a HAM SIG from the UG. There are several in the UG. He will be contacting you soon. Maybe we can make it the "TI Ham Capitol of the Universe".

DISCUSSION: 1. A suggestion for a future meeting was discussed. Trivia at the UG. This might be handled similar to the TV program "Computer Cronicles". Don't be surprised if you see it announced as the program at a future meeting.

The meeting was adjourned.

DEMO: Contrary to the announcement, Ed not Gary gave a good demo on making a Father Day Card using JIFFY-CARD.

Submitted by Bill Paeth, Secretary

NEXT MEETING

TUESDAY, JULY 10

6:30 PM --- WEST MUSIC COMPANY

Cedar Valley 99'er Users Group
377 Cambridge Dr. NE
Cedar Rapids, Iowa 52402

Send To:

GARY BISHOP
124-222
3270 28TH AVE
MARION IA 52302