

SPACE

NEWSLETTER January, 1996

President's Corner

By Mike Fitzpatrick

As you are all aware the weather forced a cancellation of the Christmas Party. Many thanks to:

Rich Mier.
Ray Wafer.
Greg Leitner.

and others who passed the word along.

Also, we had help from KOOL-108 and WCCO radio in getting the message out over the radio waves.

The plan is to have a party at the January meeting. Please try to bring what you signed up for plus a little bit for yourself. The sheet cake that we picked up was well received by all our neighbors at our impromptu snow removal party the night of the canceled SPACE meeting.

The method <maximum frustration> of contacting folks was exacerbated by the loss of the BBS, with all the member records, two days prior to the meeting. Hopefully that will never happen again. I do think that a calling plan for a member daisy chain of contacting might be a point to consider.

Here's hoping that you had a great Holiday Season and 1996 will be filled with Health, Happiness, and Prosperity to you all.

There is one item that I'll save for Lance to discuss in January about new developments for the classic Atari 8-bit computer.

BBS Report

As you know the BBS went down on the 6th of December. A REALLY horrible event. The problem with the BBS was the hard drive. The older MFM & RLL drives were warranted for 500 hours of service life. Well in six months time 4500 hr's is the norm. This will be an increasingly growing problem as those types of drives are only available as used; true condition unknown. For the technical folks: a post mortem on the drive indicated that the bearing(s) lost the permanent lub. Friction and heat did the rest.

Imbedded SCSI's are available but they tend to be expensive. I would also save that type of drive for the ST where the full potential of the drive can be utilized.

It appears that I was able to save the user dat file so that few if any had to re-log on. There is still a minor problem with the 3rd party f-mail and I plan on working that out with K-Products and Steve Cardin (BBS Express-Pro programmer).

As of this writing the files have not been reloaded but the plan is to do it on Superbowl Sunday when few if any will be bbsing.

Minutes of Space meeting

December 08, 1995

It was the night before the Space annual Christmas party and all through the house, nothing was stirring except me. I was busy preparing the big 13-pound turkey for our Christmas party, Friday, December 8, 1995. I put it in the oven to cook over night. In the morning I pulled a delicious looking turkey out of the oven. I then put it in the ice box. In the evening I was going to add it to all the delicious food at the party. You all know the rest, the club party was canceled by MOTHER NATURE. Luckily we have a chance to celebrate a Christmas party at our January 12, 1996 meeting. We can also include celebrating a new year. For all members that signed up to bring something on the Christmas potluck list, please bring food in for the Christmas party in January. Be prepared to have a fun time too!

The midwinter Madness show is rapidly approaching. It will be held on February 10, 1995 at Blaine Sports Center in Blaine. Be sure to bring plenty of money because there will probably be plenty of bargains.

Last of all, because we didn't have a club meeting in December because of MOTHER NATURE, there aren't any minutes to report. See you at the January Space meeting, and be prepared to PARTY HARDY and FEAST LIKE KINGS!

Michael Weist
Club Secretary

SPACE'S 13TH CHRISTMAS PARTY POTLUCK SIGN-UP LIST

PERSON	POTLUCK ITEM
Terry Streeter	chicken salad
Mike Fitzpatrick	cake
Greg Leinter	meat tray
J. Overbaugh	veggies
D. Wold	Salad
Bill Cotter	Meatballs
Glen Kirschenmann	48 cans pop
Al Noble	Special K Bars
Don Langford	BBQ Smookies
Lance Ringquist	Cookies
Larry Serflatten	????????????
Ray Wafer	Potatoe salad
Mike Weist	????????????
Mike Schmidt	Plates, napkins, utensils

Larry's
ACTION! TUTORIAL

#17 VICTORY

From the last issue:

The 12 different boxes of product need 11 gates to provide 12 conveyors to the 12 automatic shelves. My set of eleven tests began with a scale weighing out any package less than 15 Lbs. If your solution has something different, that is perfectly acceptable, provided it meets the needs of the job. Basically the first test should divide the group of products roughly in half, but it is not mandatory. The important part is that it performs as required, and you can check that by using your flowchart to see where each box would get diverted to. The criteria was that the boxes should end up on separate shelves. You might imagine the supervisor may want to invest more cash into the system and ask you to add in the needed equipment to reject certain boxes which fail in a unique way, and return all rejects on a new conveyor to the plant.

Such is the way it goes in programming, just when you think your done, somebody shows up and wants to change the program. A good flowchart helps in modifying an already complete program. With a flow chart, the programmer can make necessary changes to the algorithm, and can quickly add in the needed code in the proper spot. This is just a reminder to always document your programs, either by the use of a flow chart, or in the source code itself. You never know when you may decide to add more stuff to a finished program.

Moving from words in a flowchart, to code on the computer can be made easy to do. The more details included in a flowchart, the easier it is to translate into code. Flowcharts also help in optimization of algorithms. Careful inspection of a flow chart can reveal redundant loading of a variable, or testing of a value, that may be consolidated or reduced. Looking at the whole process, might give insight to a different algorithm, or method, that will do the same thing even quicker. Some times a better algorithm will be enough, and should be attempted first, but there are occasions when assembly is the best choice. Remember you need only to rework the code that causes a delay, or that will make a perceptable difference. There is no big advantage in optimizing the part of a program that is already performing well.

As if to fly in the face of good advice, look back at issue #12's procedures. Here we had a few good routines that were made even better by translating part of them to machine code (AscToInt). EchoS has a loop in it, one of the most basic programming structures, and the most time consuming. Translating this routine to machine code will provide an example of the loop structure as it appears in machine language.

First the flowchart:

1. Save STRING address
2. Calculate DESTINATION address
3. Initialize counter
4. Load byte from STRING
5. Save byte to DESTINATION
6. Decrement counter
7. Test for end condition
8. Loop to "4" until done
9. RETURN

Now the code:

```
PROC EchoS=(BYTE ARRAY S)
;1. Save STRING address
[$85 $A2      ;STA $A2 Save S LO
$86 $A3      ;STX $A3 Save S HI
;2. DEST=(Y*40)+SAVMSC+X
$A5 $54      ;LDA $54 Y pos
$A2 40       ;LDX #40
$20 MultiplyB ;JSR MultiplyB
$18          ;CLC
$A5 $58      ;LDA $58 Screen LO
$65 $55      ;ADC $55 X pos
$65 $A0      ;ADC $A0 (Y*40) LO
$85 $A4      ;STA $A3 DEST LO
$A5 $59      ;LDA $59 Screen HI
$65 $A1      ;ADC $A1 (Y*40) HI
$85 $A5      ;STA $A4 DEST HI
;3. Init counter
$A0 $00      ;LDY 0
$B1 $A2      ;LDA ($A2),Y
$A8          ;TAY
$88          ;DEY
;START=START+1 S(0) is string Length!
$18          ;CLC
$E6 $A2      ;INC $A2
$D0 $02      ;BNE 2
$E6 $A3      ;INC $A3
;4. LOOP start, load byte
$B1 $A2      ;LDA ($A2),Y
$20 AscToInt ;JSR AscToInt
;5. Store byte
$91 $A4      ;STA ($A4),Y
;6. Decrement counter
$88          ;DEY
;7 & 8. Test and Loop until done
$10 $F6      ;BPL (LOOP)
;9. RETURN from whence we came
$60]         ;RETURN
```

As you can see, most of this procedure is spent setting up the parameters for the loop. The loop itself is quite small, only 10 bytes (Two bytes are used to indicate where AscToInt is located). This loop can still be optimized further, if desired, by moving all the code from the AscToInt routine to its proper place in the loop. This results in eliminating the need for jumping (using time to store values on the processor stack and in the program counter) which is time consuming when done repeatedly. The code from the AscToInt routine is needed for this loop to function, so moving it 'in line' will make the loop larger, but will actually reduce the execution time.

Were you wondering how to call a procedure using machine language? You can also call ACTION! routines that you have written. ACTION! passes variables using the A and X registers, and returns function values in \$A0 and \$A1.

Larry's
ACTION! TUTORIAL

#18 BREAKING THE RECORD!

BYTES, CARDS, POINTERS, and ARRAYS can be used for most of your programming. There comes a time when you must work with a large amount of data, and a mixed set of BYTES, CARDS, and ARRAYS. At times it makes sense to organize your data so that it may be easily processed, other times it makes just as much sense to organize your data for optimum use with files on a disk. Records are used to help in both of these areas.

Records allow a programmer to group the fundamental data types together (BYTES, CARDS, and INTEGERS). The grouping must be declared before it is used. The records become real useful when you set up a block of memory to hold an array of records. This is no more than a BYTE array, which is divided into individual records, one record right after the other. Each record can be accessed using a record POINTER, which steps through the array, in steps the size of one record.

```
TYPE DAY=[BYTE hi,lo   ;Record FORMAT
          INT  change]
```

```
BYTE ARRAY memory(400) ;Records storage
```

```
DAY POINTER temp       ;Record ID
```

```
DEFINE RECORD_SIZE="4" ;2 BYTES + 1 INT
DEFINE MAX_DAYS="9"    ;99 max!
```

```
PROC Make_Record()
BYTE i,r=53770,oldtemp
INT chg
```

```
    temp=memory                   ;# 1a
```

```
    oldtemp=75
    FOR i=0 TO MAX_DAYS
    DO
        chg=r&7
        chg== -4
        temp.change=chg
        temp.hi=oldtemp+chg
        temp.lo=temp.hi-5-(r&3)
        oldtemp=temp.hi
        temp==+RECORD_SIZE       ;# 1b
    OD
```

```
RETURN
```

```
PROC SHOW()
BYTE i
```

```
Make_Record()
```

```
Graphics(0)
PrintE(" ")
```

```
printE("DAY   High   Low   Chg")
```

```
    FOR i=0 TO MAX_DAYS
    DO
        temp=memory+(RECORD_SIZE*i) ;# 2
        PrintF(" %U       %U       %U       %I",
                i,temp.hi,temp.lo,temp.change)
        PrintE(" ")
    OD
RETURN
```

You can read your manual to get further information on declaring records. This program is nothing special, it creates a list of each days high and low temperature and indicates the change of the high temp from one day to the next. Take note, there are two methods used to access the records. Method #1 sets 'temp' to the start of the storage area (1a) and steps through each record by adding RECORD_SIZE to 'temp' (1b) each pass through the loop. Method #2 calculates the proper location of each record (#2). This random access method of locating each record is simple to use and works well for records that reside completely in memory.

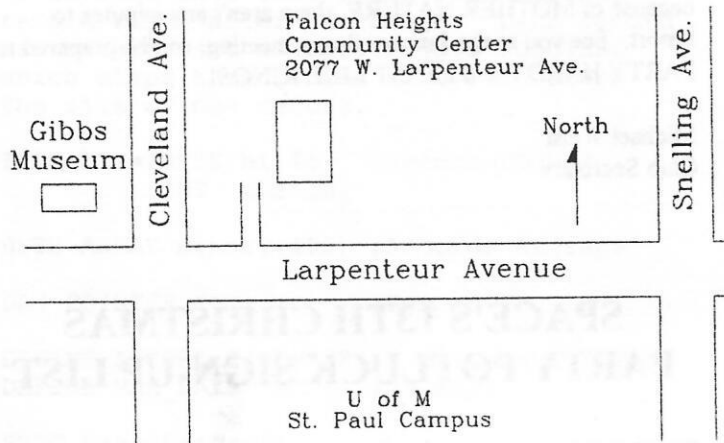
Because each record can be accessed in a uniform manner, the need for special data handling code is reduced. To give you a little practice in using records, try to write a routine that would allow you to enter the data for this program. You can have the user input the high and low temperatures, and the computer calculating the amount of change in high temperature from the previous day. When you have that routine working, write a program to keep track of the time of the sun rising and setting each day. Also keep track of how many hours of daylight there are in each day and the change in daylight hours from one day to the next. The more practice you get, the better you become.

Records are a very useful tool. As will be shown next month, they can be used to provide an easy means of working with data objects (the manual calls them 'virtual records'), we're going to call them worms!

CLUB OFFICIALS

President:	Mike Fitzpatrick	XXX-XXXX
Vice President:	Terry Streeter	XXX-XXXX
Secretary:	Mike Wiest (Temp)	XXX-XXXX
DOM Librarian:	Terry Streeter	XXX-XXXX
Paper Librarian:	Vacant	XXX-XXXX
Software Librarian:	Earline Fitzpatrick	XXX-XXXX
Membership Chairman:	Glen Kirschenmann	786-4790
SPACE Treasurer	Greg Leitner	455-6550
SPACE Newsletter Editor:	Michael Schmidt	757-4192

Saint Paul Atari Computer Enthusiasts (SPACE) meets on the second Friday of each month at 7:30 PM in the Falcon Heights Community Center at 2077 West Larpenteur Ave. Doors open at 7:00 PM.



DISCLAIMER

Published by the Saint Paul Atari Computer Enthusiasts (SPACE), an independent organization with no business affiliation with ATARI Corporation. Permission is granted to any similar organization with which SPACE exchanges newsletters to reprint material from this newsletter. We do however ask that credit be given to the authors and to SPACE. Opinions expressed are those of the authors and do not necessarily reflect the views of SPACE, the club officers, club members or ATARI Corporation.

SPACE BBS

BBS Phone: (612)-427-4317

Sysop: Mike Fitzpatrick

Co-sysop: Open Position

Supporting:

ATARI 8-Bit Computers

ATARI ST Computers

S.P.A.C.E.
P.O. Box 120016
New Brighton, MN 55112

