october 1987

TS·2068 UP·DATE

the user's NEWS



TS-2068 UP-DATE
1317 STRATFORD AVE.
PANAMA CITY, FL 32404

TS-2068 UP-DATE is a quarterly publication devoted to the support of users of the Timex TS-2068. Each of the Disk Drive Systems for the TS-2068 are discussed in detail in special feature sections. Annual subscription is $12,00 per year of issues.

**** BUILDING BLOCK SECTIONS ****

Software and Hardware Articles and Reviews

Technical: Schematics, Data Sheets, Interfacing

The Catalog: Software and Hardware Availability

Electronics Data: Formulas, Symbols, Tables

Subscribers Free Ads: Swap, Sell, Wanted

TS-2068 UP-DATE
1317 Stratford Ave, Panama City, FL 32404
(904 871 4513)

Common Electronics Symbols and Abbreviations

The following are Electronic Symbols and abbreviations, having to do with Voltge, Current, Power, Frequency, Inductance, Reactance, Capacitance, Resistance. For electroncs formulas, refer to the technical (green) section.

English Letter Symbols

A =Ampere (Current Measurement)
C =Coulomb (capacitance Charge)
E =Electromotive Force (Volts)
G =giga (billion or 10 to 9th)
I =Current (Unit=Ampere)
K =Kilo (one thousand)
kV=Kilovolt (thousand volts)
kW=Kilowatt (1000 watts) (power)
M =Mega (1 million)
mA=Milliampere (1/1000th ampere)
mV=Millivolt (1/1000th volt)
n =Nano (1 billionth or 10 to -9)
nF=Nanofarad (1 billionth Farad)
P =Pico (10 to the minus 12)
PW=Picowatt
VA=Voltampere (One Volt at 1 Amp)
Wh=Watthour
uA=Microampere
uH=Microhenry

Bd=Baud (speed of communications)
dB=Decibel (sound measurement)
F =Farad (unit of Capacitance)
GHz=Gigahertz- One Billion Hertz
J =Joule (Inductive energy unit)
kHz=Kilohertz (frequency)
kVA=Kilovoltampere. (1000V*1A)
kWh=Kilowatthour (1000W for 1 hr)
MHz=Megahertz (1 million hertz)
ms=millisecond (1/1000th second)
mW=milliwatt (1/1000th watt)
nA=Nanoampere (1 billionth amp)
ns=Nanosecond (1 billionth second)
pF=Picofarad
V =Volt (Electromotive Force)
W =Watt (unit of electrical power)
u =Micro (1 millionth)
uF=Microfarad
us=Microsecond

Abbreviations:

ac=Alternating Current
am=Amplitude modulation
amp hr=Ampere hour
ant=Antenna
assy=Assembly
avc=Automatic Volume control
bfo=Beat frequency oscillator
B&S=Brown & Sharpe Wire Gage
cb=Common base
cf=Cathode follower
cw=Continuous wave
dB=Decibel (Sound measurement)
demod=Demodulator
dpbc=Double pole, back connected
dpfc=Double pole, front connected
dpsw=Double pole switch
D =Drain; duty factor
emf=Electromotive force
erp=Effective radiated power

af=Audio Frequency
amp=Ampere
ampl=Amplifier
apc=Automatic phase control
atten=Attenuator
AWG=American wire gauge
bp=bandpass
cap=Capacitor
ce=Common emitter
ct=Center tap
C =Capacitance, Capacitor, Coulomb
dc=direct current
dp=Double pole (switch)
dpdt=Double pole, double throw
dpst=Double pole, single throw
dt =double throw
ef=Emitter follower
env=Envelope
E =Voltage; Emitter

Continued inside back (blue section).

About TS-2068 SAFE DISK UP-DATE

TS-2068 SAFE DISK UP-DATE is a general publications to support TS-2068 Users. SDU is orientated to those serious users who have already, or intend to upgrade to Disk Drive system. SDU is written for all levels of user programming. The expert programmer will likely think that the content is too basic, and the novice will feel that it is too advanced. To the latter, knowledge is gained by the study of details that are foreign, as no one learns from re-hashing what one already knows. There are many TIMEX CLUBS, one within your commuting range. These clubs provide a great opportunity to benefit from the exchange of programming information. Some produce club newsletters that are well worth subscribing to. I subscribe to no less than four of these club publications. Current listings of active clubs can be found in Time Designs Magazine, SyncWare News, and Computer Trader Magazine, all of which are "MUST SUBSCRIBE TO" publications that support the TS-2068. TO THE EXPERT PROGRAMMER, this issue is a bit weak on Machine Language programming, but I'll bet that you can find some new kinks in BASIC that will support your programming efforts. AND, the program "SDOS MAIL MERGE" is a natural for assemblying into MC. We will do that in the next issue. Why dont one of you MC artists beat me to the punch and send in a article on this? Do it and your works will be published.

SYNTAX: While SDU is "Disk Drive Orientated", and is now slanted to the Safe Disk Operating System (SDOS), all programming presented can easily adapt to any other Disk Controller syntax, and to Cassette. SDU will begin to support other disk systems as writers begin to support SDU. I received a nice letter from Mr. Larry Kenny of LARKIN ELECTRONICS inquiring if SDU is a "General Publication for the TS-2068". YES, it is intended to be just that. BUT, to cover all systems, I need WRITER CONTRIBUTIONS from those who use the systems. I would love to add a "LARKEN DISK SECTION". Larry Kenny deserves much credit for developing the LARKIN DISK and for his perseverance. He even has a "Universal DOS DISKETTE" that is compatable with the Oliger SAFE DISK System. I haven't tried that, but would like to.

AERCO FD-68 DISK SYSTEM: David Hill started a nice newsletter for Aerco FD-68 Disk Users. I subscribed and received two issues of FD-68 User and I must admit that my money's worth was received in the two issues. BUT, it appears that FD-68 USER is gone. The current issue is well past its time and correspondance to Dave is being returned "no forwarding address". Maybe this is hasty, and I'll be sure to correct it if it proves to be so. If the latter is the case, I dont want to compete for FD-68 users, but would like to invite all to subscribe to SDU. If FD-68 User does not resume, then perhaps we can start a special section in SDU to serve the Aerco users. Again, WRITER SUPPORT will be needed and welcomed.

OUTLOOK: A publication builds upon the effectiveness of its content and the enthusiasm of the user group that it represents. Both equals success. I am enthusiastic and I project about 300 subscribers by mid 1988 and increasing. When we reach that point, a break even situation will exist and consideration will be given to expand coverage and to change to a bi-monthly schedule. BUT, First Year First! I am committed to a full year of publication of SDU before even checking on RED INK (losses). Then we will decide. Best O Computing to all.

WINTER COMPUTER FEST IN ORLANDO, FLORIDA: Right in the midst of Disney World, Sea World, Circus World, nearby to the East Coast Beaches and to the Space Center. What a great place to combine a Computer Fest and a Family Vacation! The details will unfold between issues of SDU. Watch for news in TDM, SYN, and TDM. Eric Johnson, 249 N. Hardin Ave, Orange City, FL 32763, is your contact. He needs your support for success.

By Bill Jones

Enthusiastic 2068ers

I wish that I could print the dozens of enthusiastic and supportive letters that I have received from dedicated 2068ers. One nice guy even sent an extra $15.00 "for the kitty". There is a real solid desire for a publication dedicated to TS-2068 support. THANKS to all of you well wishers! Soon we will see just how many serious 2068 users are out there in the USA and CANADA. Counting noses, Canadians are right in there at the stretch. One of you Canucks send me a little maple leaf decal to dress up the pages with. It'll go side by side with the Stars and Bars. Incidentally, I have had the priviledge of flying side by side with many RCAF Jocks and I can tell everyone that the Country to our North is a proud and sturdy neighbor having a tradition of integrity that is difficult to match.

Gaffs and Laffs

Defenitichions

Interface: Whats under the cold creme as the wife prepares for beddi-bye.

Dynamo: A small town in Missouri.

Data bus: Kissin and talking at the same time.

Syntax: A Colorado tax on brothels.

Peripheral: What one sees outta the corners of the eyes.

Optimist: One who believes the sky is rising with him in it.

Pessimist: One who believes the sky will fall upon him.

Free: What taxes provide to them that dont pay taxes.

Service: A crime committed by a male.

Consultant: One who can't do it but wants to tell you how for a price.

Coax: A small axe which has two cutting ends.

Telecomputing: Secrets told to a friend and relayed to others.

Trouble Free: It dont break as much as it usta.

Buffer: Where Ma keeps the Sunday table cloth.

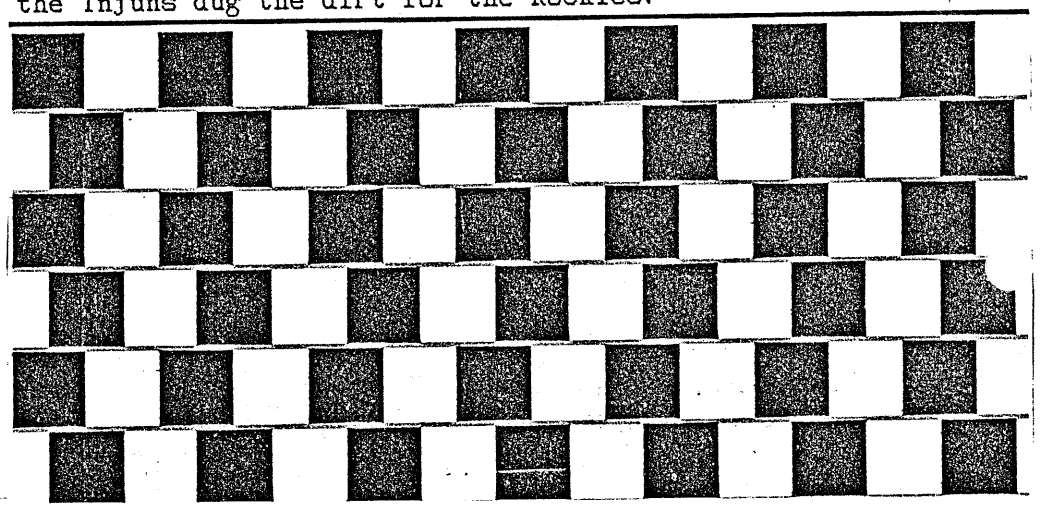
Grand Canyon: Where the Injuns dug the dirt for the Rockies.

Your eyes are fooling you.

All Lines are Straight.

All Squares are Square.

All are the same Size.

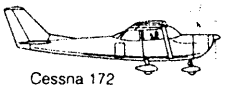


EDITORIALS

My Two Cents Worth

plus

Guest Editorials



Cessna 172

Eye Burn From Computer Monitors

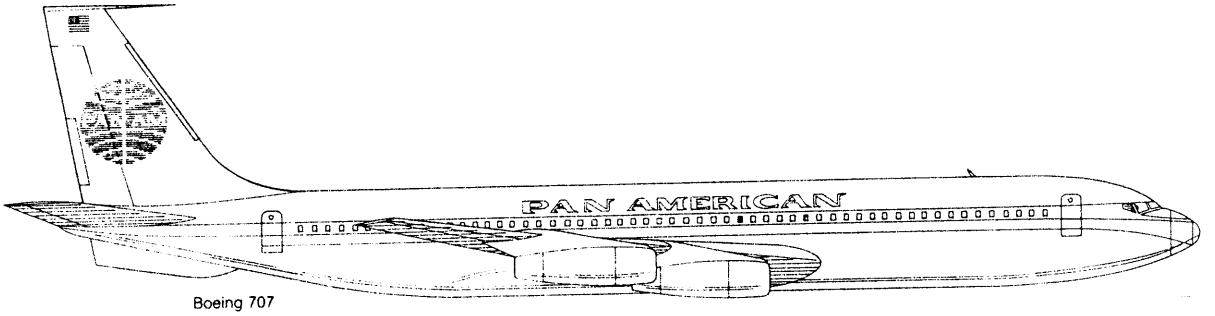
Computers have introduced a damaging phenomenon, "Monitor Burn". Monitor burn is similiar in effect upon the eyes as sunburn after a day at the beach or a afternoon drive toward the western sun. One's eyes can become quite irritated after hours before a computer monitor. Continued exposure can result in "dry eyes", and even scalding of the surface of the eyes, a severe condition that has lasting effects.

Color Monitors produce the most eye damage. Monitors that present black background and blue print are the easiest on the eyes. This is because blue light is of a lower frequency and has the most diffusion, while red light is a higher frequency and is concentrated as in a beam. Other colors fall in between as to their damaging effects.

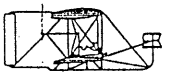
As much as we like to use our computers, we should be alert to "Monitor Burn". The early symptoms of Monitor Burn are: 1. Eyes dryer than usual and a need to blink more often. 2. Eye redness or irritation. 3. Itching of the eye or eyelids. When these symptoms occur one should quit for the day and until the symptoms disappear.

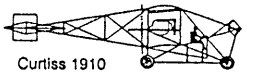
By the time one notices the symptoms of Monitor Burn, the eyes could be considerably burned and scalded necessitating a prolonged period of recovery. Recovery may require the continual use of "artifical tears" eye drops to overcome dryness, and or, a antibiotic salve to keep down bacteria infection of the eyes.

Those who operate computers should be conscious of the damaging effects of Monitor Burn and take these precautions: 1. Use sun glasses for long periods of viewing (especially color monitors). 2. Operate with a "blue on black" color combination if possible. 3. Keep eyes moist by blinking, or use of mild "artificial tears" eye drops. 4. Watch for early symptoms of Monitor Burn and discontinue until the symptoms disiappear.



Boeing 707


Wright Flyer


Curtiss 1910

Editorial

Here we go! Like the Wright Flyer, we're off the ground! This first issue of TS-2068 Safe Disk Up-date cost a bundle because of the insert cover and section separations. But maybe the red ink will change to green by the end of the first year. I wanted to provide a couple of pages for each section so that you would get acquainted with the organization. These sections will build as the issues come to you.

The pages are punched for insertion in a three ring note book in order to keep the issues together in one place. Each section will be a different color so that major subjects can be easily found. I've never seen a publication organized that way, but I've never published anything before. Anyway, it sounded like a good idea.

If you like this issue, please rave about it to all who have a TS-2068, members of clubs especially. Although we will always support SDOS, the subject matter will be applicable to the TS-2068, whatever the configuration. Please help the newsletter grow by recommending it to your friends.

This issue has several useful utilities plus two major programs. The second program "SDOS MAIL MERGE" is outstanding in its usefulness. SDOS MAIL MERGE will work right now as a printing mail list, using the Safe Disk printer driver routine. It will print labels and envelopes, or just listings of whatever nature you store in the data base. For the next two issues we will be adding more functions to Mail Merge. We'll add a couple more menus, a printer control group, auto printing of disk files, plus more. This software will grow.

The first program is a nice group of six interactive utilities. You can actually do "mixed math" using inputs of HEX or Decimal. Multiply FFF Hex time 123 Decimal, for example. Even fractions are handled. Of course there is not a way of expressing fractions in HEX, but this will work: Divide 102697 by FA5 Hex. The result will be printed in decimal with fraction. Direct conversions of Decimal to Hex or Hex to Decimal are handled up to FFFFFFh or 16777215 decimal. Thats 16 megabytes.

SDU DISKETTES: Lazy? Well, maybe just "busy". Whatever, if you dont want to key in the programs and utilities and debug key-in errors, you can order the "SDU DISKETTE". There will be a SDU DISKETTE for each issue. Each SDU diskette will have all programs and utilities of that issue on 5 1/4 inch diskette. Additional documentation will be provided when appropriate. All formats of 5 1/4 inch diskettes will be supported. You can order the SDU Diskette for a price of $16.00. See the order form in the yellow pages for each issue. A scheme has been concocted to share the proceeds of SDU Diskette sales with the authors of the programs and utilities. SDU will provide the diskettes and take 50% for expenses. The other 50% will be shared on a basis of the author's percentage of the total diskette programs. One wont get rich this way, but will at least receive something for one's efforts.

The BLUE section of SDU will be filled in as issues continue. It will be a complete "Electronics Data" section and will include all abbreviations, symbols, conversions, tables, and formulas. The Technical Section will contain all of the TIMEX schematics, Port pin assignments, Oliger schematics, and technical

data about modifications.

The Club: Well, actually it isn't a club, but lets think of TS-2068 Safe Disk Up-date as representing the TS-2068 user group. SDU encourages your participation, your questions, and your comments. We need useful utilities and short programs for publication. Please send in your creations. Lets use this analogy in computer syntax: < IF 100 subscribers SHARE utilities and programming TIPS THEN ALL MAY INCREASE computer literacy BY a FACTOR OF 100 > < IF NO OVERLAP >. Oh well, you get the gist of the idea. Lets have a participating group for the benefits of all.

The section titled "TS-2068 Computing With SDOS" is John Oliger's section, his rapport with Safe Disk Users. John will answer all of your questions. Send your questions to the newsletter, or direct to John. I'm sure that John already has a number of questions that he has answered to individuals, that would also be of interest to most users. I'll keep my hands off of this section and it will be what you, the readers, and John make of it.

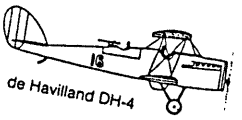
Subscribers FREE ADS: You may send in an ad to sell something that you dont need anymore, or ask for something that you dont have (Wanted), or TRADE. These free ads will be column printed and must be 400 characters or less in length. Please give your mailing address and phone number if possible, and state if the item is in working order.

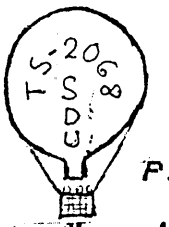
The Catalog Section: Consider this "the yellow pages". Some of the ads will be paid ads from vendors. Subscribers who have their own software or hardware to sell may "announce the product" FREE. Such free announcements must be sent in "copy ready" and shaped to 1/2 page in size. Use a good typewriter or computer print with a "letter quality" printer. Draft quality print wont copy very well. These subscriber "Free Announcements" are given to encourage the programmer to create good software for the users. It's tough out there to have to pay to advertise to a small user group. This may induce our qualified programmers to PRODUCE.

REVIEWS: We need volenteer reviewers. If you want to receive software and write reviews, please write in and get on the SDU writer list. Individuals or firms who desire that a product be reviewed, please send the software or hardware to the newsletter. We will expect that a software will be kept by the reviewer as compensation for the writing effort. Reviewers of hardware should be allowed to purchase the hardware at 40% off market price.

LETTERS: I will appreciate your comments. The editorial section will be kept brief because we dont want much "non permanent" data to clutter up the Safe Disk Notebook. Letters, while appreciated, may not be published, depending upon whether or not the comments are deemed to be of interest to most subscribers. QUESTIONS and informatives WILL be published. Complaints about vendors will be handled by direct contact with the vendor to attempt to satisfy the user, and will be published only if a vendor fails consistantly to resolve complaints.

So, thats my two cents worth for this October 87 issue. Hang in there and participate! We need that. (Ed)





How Goes It?

Now I'm gonna take a leaf from 'ole Chet's book and ramble a bit. SDU intends to cover ALL TS-2068 Disk Drive Systems. These are; The Oliger Safe Disk, The Aerco FD-68, The Larkin, and The TOS (Zebra). Eric Johnson (your contact for the Florida Computer Fest) has agreed to provide writer support for the LARKIN DISK system. By the January issue of SDU we should have a writer lined up to provide coverage of the TOS DISK system, and another for the FD-68 DISK system. Other good programmers and writers have promised articles and programs for future issues. Misters Tom Kalogerson, Robert Hartung, Dick Wagner, Roelof Mulder, and John McMichael, have all promised to help with the writing. I really appreciate this and am looking forward to publishing their works. As these and more writers join in, SDU will become the most exclusive source of up to date information about the TS-2068 and its disk systems.

Want to call a Bulletin Board and leave a message for all to read, or get info about the Orlando Computer Fest? Get your Modem up and running and let your computer dial 904-775-0093. Use 300 BAUD, 8 - 1- none, as the log on protocol (No Voice). This is Eric Johnson's BB (NW FLORIDA TSUG), headquarters for the 88 Winter Computer Fest.

Eric also has a NEAT, and inexpensive DOCK RAM board which he will write up as a building project for SDU. This fits in nicely with a EPROM programming article by Tom Kalogerson, and one that I am planning for conversion of the (Oliger) 64k RAM for the TS-1000 to a 64K Dock RAM to fit into the extra slot of the expansion bus. BASIC programming details will be given. These articles will be paced so that a similiar subjects do not take up a whole issue of SDU.

Planned are articles and programs about such subjects as: Dock Programming, EPROM Programming, Extra Memory Devices, Telcomputing, Interfacing, Disk Drives, plus more. The material presented in SDU will not be designated as "public domain", but will remain the property of the writer. Each subscriber is given the programs for individual use. Now, a bit on "self interest". Some of you know that I am the author of the software "Smart Text", and may be wondering if SDU is something to feather my nest. ST has been on the market long enough that almost everyone who wants it already has it, and I do not have, nor do I intend to have anything else to sell. Naturally, I believe that ST is a good and useful program, and I'll keep a "Subscriber Ad" in SDU offering it to any who wants a copy. You can do that also for any software or hardware that you have for sale. Free subscriber ads are given to encourage our software writers to produce. There are many expert programmers in our midst, but few are producing products for sale because of the costs of advertising to a small user group. Maybe some will release their products via the Free subscriber ads. So, "how bout it", you programmers? Lets get your good programs into circulation.

Plans and Organization

At this writing there are 45 subscribers to SDU. Dont laugh! These 45 came from a mail out invitation to only 100. All are SAFE DISK users. More subscriptions are trickling in each week, and another 100 mail outs to SD users will be done soon. The projection is for 2-300 subscribers by July 88. Publication of SDU is "by year of issues", which means that a subscription year is OCT to JULY, and each "mid year" subscriber will receive that current year of issues. This strange set up does two fine things. It provides the new subscriber with the back up issues that are supporting, and it makes each issue more uniform in distribution.

I expect that the publishing of SDU will cost me about $200.00 more per issue than subscription income- until the subscription base reaches about 300 (the break even point). The sales of issue diskettes can help this posture by bringing in small profits. I will commit to a full year of publishing and see how it goes for the OCTOBER 88 issue. Enthusiasm of the TS-2068 users is the essential factor. IF we have that, and IF the subscription base builds, THEN PERHAPS SDU can become a bi-monthly, beginning with the Oct 88 issue.

The Lonesome Dock

I mentioned that the two programs in this issue were designed for installing in a Dock Cartridge. To date little information has been published about how Dock Cartridge programs, Home Bank Memory, and Disk Drives can be "cross utilized". Picture a 64K Dock cartridge that has a major software, plus the two programs in this issue, plus TWO MORE major programs. The major programs are all edited to remove ALL screen presentations and replace with a simple disk call (LOAD /"title"SCREEN$). Each time that is done, the program length of the Dock Program reduces by 400 to 700 bytes. Cut 8 menus and screens and the program length is reduced by 3-6K.

Another aspect of Dock Programs is the fact that the dock program contains NO VARIABLES. Just pure programming. Variables can be identified in a dock program, but the vars are STORED in Home Bank. The DATA and Vars in the home bank can easily be manipulated by the dock program. So can the Disk Drive. Program planning can easily result in the Dock Program manipulating at will, its own 64K of programs, plus 38K of FREE Memory in the Home Bank, plus a "Million K" of data in your diskette library.

So far, the TS-2068 community has not embraced the Dock Cartridge with the enthusiasm that it deserves. Articles about Dock Programming have been very technical, presenting the picture that the use of dock cartridges is very difficult to master. The programmer's job is to reduce complexity to "user friendliness", and this can be done easily. We will explore all of the above in future articles.

HELP IT FLY!



The Programs for This Issue

Actually this page was written after the outstanding editorial of page 1 and 2. After reading it I didn't think it did justice to the two programs of this issue, so I'll expound a bit, and be somewhat repetetive.

The two programs are in CRASH FREE BASIC. Both programs were originally designed to consolodate into a Dock Cartridge, along with another major software. Vars files of each are supportive. The disk drive is used to hold supporting menus for "SOOS MAIL MERGE", thus saving about 2K of program memory. We will have a future article about doing the Dock Cartridge transfer.

SOOS MAIL MERGE: If you have "Pro-File" and like it, you'll also like this software. There are two flexible data bases. One is the "mail merge" base which has up to about 120 fields of 7 lines each. The user decides the number of fields needed and sets the first dimension by prompt. Each of the seven lines are set for 31 characters each. You can make the lines longer or shorter by editing line 2006 and changing from 31 characters to whatever you want as a line length. The primary purpose of this data base was designed to be a Mailing List, using the first four lines of each group (field) for "address data lines", plus the "first name" of the individual as the fifth line. Then two more data lines are provided for ancillary information about the person or company.

Labels and envelope printing uses the first four lines only. Prompts allow you to set up both vertical and horozontal spacing to strike a series of labels or envelopes uniformly. Another prompt provides the option of printing your own return address label after each "out label" prints. Disk files of mailing lists may be ordered for loading and printing. Also, automatic SAVE to disk by discrete file number is a menu option. The data base is ideally suited for many other purposes, such as setting up invoice fields, inventory lists, magazine article listings, or just anything that one needs to retain as data. The menu allows a selectable number of lines of the fields to be printed. Automatic page control with page number is a option.

Sorting is provided by either first name or last name. When the data base is used for inventory the sorting can be by either stock number or noun. Sorting in BASIC is rather slow, but not awfully slow, and how often does one need to sort a list?

The second data base is "OUTLINE". This data base is entirely flexible and is set by input prompts for: "Number of Major Topics", "Number of Minor Lines of Each", and "Line Length". The printing is set by input of TAB, and Spacing. The outline prints to input TAB for the major topics and TAB+5 for the minors. Automatic page control and page numbers are optional. Centered Subject captions are also optional. The menu allows complete page management for the printing of either data base. No supporting printer software is required when using the SAFE DISK system, as the LET/P=0 or LET/P=T protocol is used in the programming. Prompts simply ask whether you want to print with a Centroncis printer or the TS-2040. "MERGE" is MENU DRIVEN, every function before your eyes in simple English.

THE J-UTILITIES



Boeing F2R-1

How can one name a program that has six diverse functions? J-Util combines most "number functions" needed in computing.

MEMORY MANAGER is the first 3 useful utilities for the Machine Code programmer. Call it "mini z" if you like. I originally designed it to avoid having to convert hex to decimal to poke a MC program to memory. Then evolution took hold. Two DATA lines are used to preserve a MC Code table. One data line is for a HEX program, and another data line is for a Decimal program. By placing a MC program in a data line, the program may be SAVED to diskette before attempting to run the MC program. Then the menu prompts for starting address, READS the data line, and installs the program into memory. THE MEMORY DISASSEMBLER works in reverse to PEEK any selected segment of memory and print the block of addresses. The contents of each address is given in Decimal, HEX, and Character code.

2. HEX-DECIMAL-BINARY CONVERTOR: Input a number in either data base and the equilivant is given in all three data bases. Conversions are provided from 0 to FFFFFF, or 16,777,215 (16 MBYTES).

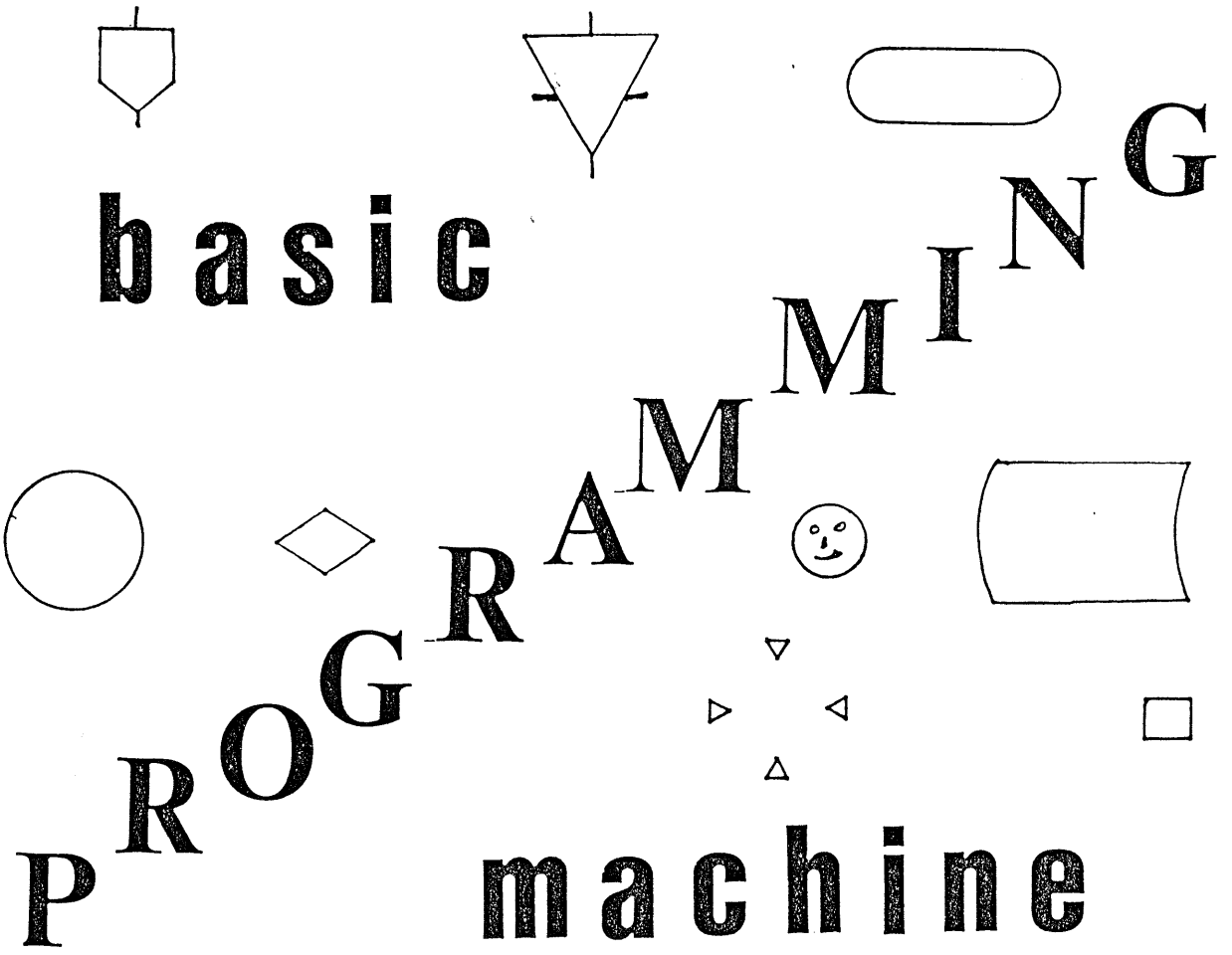
3. MULTI LATERAL MIXED MATH: Dunno what else to call it. What is FFA4h times 215.3 decimal? Or, 7649311 divided by BIN 110011? What is F43Ch minus F01Ah? Or FFFAE3h minus BIN 1111111111111? Whatever!! Who cares? Input any combo of mixed numbers and the program quickly gives the answer.

4: BIN/DECIMAL/HEX GRAPHICS: A college class on computer technology that I attended a while back got awfully hung up on this subject. A conceptual visualization of the three bases helps. A key punch brings on the graphic correlations.

THE KEY-IN PROJECTS

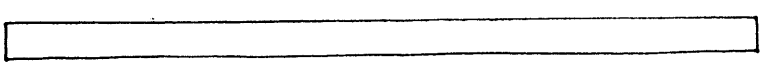
Most short utilities will slip right into computer without much de-bugging for key in errors. It is the longer programs that give problems. But the longer programs are usually more useful and enduring. Please be assured that all program listings are actual LLISTS of a operating program. SOV will not attempt to type out a program to make it fit into a page format, and risk typos. All will be actual LLISTS from the computer. These two programs will be difficult to avoid key in errors because of long and complex lines that employ Boolean expressions using brackets and math symbols. After a few hours one tends to give up.

These two programs are worth the key in effort required. Both compare with softwares for Apple and Commodore that sell for over $60.00 each. The "SOV ISSUE DISKETTE", at $16.00 is the bargain of the year. Who else can buy useful software at such a price? "Issue Diskettes" contain all programs and utilities in an issue of SOV. They are supplied on 5 1/4" diskettes in any format. The concept of "issue diskettes" is to provide the author with a little bit of compensation for his efforts, and to provide the user with useful programs and utilities, while saving lots of key-in time and frustrating de-bugging. Later, we may have to increase the price, depending upon cost factors. We'll see how it goes for at least one more issue.



...sdos...





About your Editor, plus Nostalgia

Way back in the turbulent '50's the USAF developed a gun sight which could, from 20 miles, automatically fly a fighter aircraft to a point in space to fire rockets, supposedly to destroy a enemy bomber. Thus came into being our first real computer application. Back then the transistor was a drawing board figment of engineering imagination. We used Vacuum tubes, and lots of um, thousands in a system. Computing was analog technology, using such mathematics principles as trigonometry, rotating vectors, scale factors, etc. You see, analog computing is based upon Voltage comparisons and measurements, rather than numbers and memory.

The F-89 Fighter aircraft was the largest fighter ever built, and it was a machine built around a electroncis package. The Radar and electronics package came in at about two tons, including the wiring. That was my baby, to fly and to maintain, and to look forward to advanced applications. The evolution of military computing led the electronics industry. By the '60's we were into Solid State devices, using diodes from small signal diodes to large high current diodes. High current diodes cut 600 pounds from a electronics package and reduced heat considerably. Analog computing was still the basis for computing.

The late 60's brought on "unreliable" transistors, affected by heat changes and vibration. Few military systems were developed to rely upon transistors. Then the micro chip came along- still very unreliable. But the industry began to lead the military and find civilian applications for the micro chip in such applications a cash registers and timing systems. It wasn't until the mid 70's that the military could package electronics systems that contained mostly micro devices. Still in Analog applications, the complex aircraft system that weighed in at two tons could be packaged down to about 500 pounds, most of the weight being in antenna and transmitter.

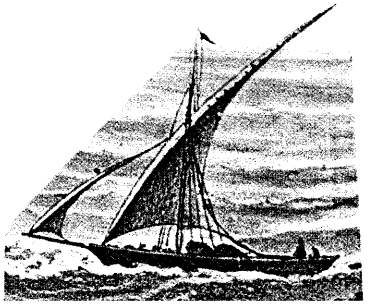
Military digital systems came in about the same time as the Sinclair ZX-81 in the 78-79 era. Then a system weight of 400 pounds would do more than a thousand times as many applications and much more accurately than my old F-89 (two ton) system. Most importantly, the electronics industry was "un-coupled" from military dominance. The world wide computer evolution began, and the industry took off on it's own power based upon civil consumption. Now our military systems are engineered and built with off the shelf items.

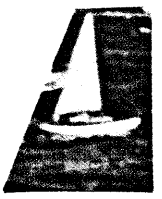
Being from the horse and buggy days of electronics, your editor was delighted to have the opportunity to buy a ZX-81 (kit) for the bargain price of $199.95. But what a transition! Now I had to learn digital, things like Binary, and Hex, and Programming languages! We who are involved in electronics are constantly faced with a problem called "Obsolescence". Take a year vacation and one comes back to be overcome by technology advances. One must be continually engaged in his own personal education up-date program. It is a matter of time sharing. In my case that is a tough thing to master, what with having too many hobbies.

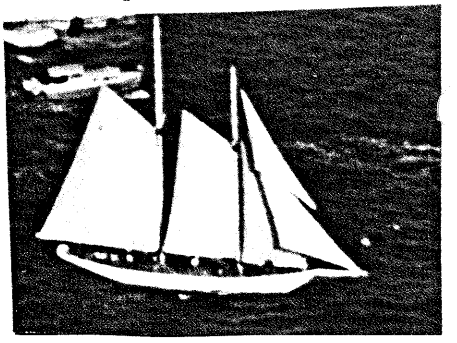
I am a avid TS-2068 computerist, a sailor, a commerical pilot and flight instructor, and I like to go golfing and fishing at least once or twice a week. I'm "between sail boats" right now, and scheming on a way of combining computing and sailing. (Anybody got a 38' ketch for sale under $20M?). I've ordered a spare 2068 to go aboard my next vessel and hook up to ship battery. I'm working on a program to navigate with the 2068 so that the sailor can get about 6 hours of sleep while the 2068 is at the helm.

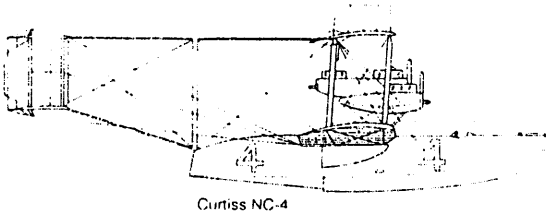
Now, time sharing as it is, I refuse to pour over the SDU text for typos, dangling participles, and split infinitudes. So, some typos will be fixed with pen and ink and if my smashing of the Kings English is still understandable, then so be it. If you become irkedover non-gramatics just come on down and go sailing. That's guaranteed to soothe any case of irritibilitis. I do believe that sailing and computing will mix well, so dont be suprised if the January issue of SDU is published at sea, twix the tending of a trailing fishing line. And, I absolutely guarantee at least 10 gaffs in English and several pen and ink changes in each issue of SDU. See you in the islands!

by: Barnacle Bill, the Sailor









Curtiss NC-4

"J-UTIL" LISTING

```
9800 GO SUB VAL "9884": INK OF: BORDER OO: PAPER OO: CLS : P
RINT FLASH OA;AT OD,OB;"MULTI LATERAL MIXED NUMBERS"; FLASH
 OO''TAB OB;"MEMORY ORGANIZER"''TAB OF;"DISASSEMBLER"'TAB OJ;
 "HEX DATA POKER"'TAB ON;"DECIMAL DATA POKER"''TAB OB; INK 3;
 "CONVERSIONS- HEX/DEC/BINARY"''TAB OB; INK 5;"MATH WITH MIXE
D DEC/HEX/BIN"''TAB OG; INK 4;"BINARY ANALYSIS TABLE": LET A
=1: FOR N=1 TO 40: BEEP INT ((RND*10)+1)/10,INT (RND*35): IN
K A: PRINT OVER OA;AT OD,OB;"                              ": L
ET A=A+1: IF A>7 THEN LET A=1
9801 NEXT N: CLS : PRINT INK oe; FLASH 1;"The ASSEMBLER/DISA
SSEMBLER"; FLASH oo;" is  designed to use DATA lines to  o
rganize a Machine Code program in either HEX or Decimal, the
n  SAVE/ the program to Diskette  before attempting to RUN
the MC program and chancing a crash.  A CRASHED program can
 be re-  loaded from diskette and worked on.  DATA lines a
re 9878 (DEC), and 9880 (HEX)."''"  The MC program is insta
lled  in memory at your selected startaddress."''"  Selecte
d segaents of memory  are peeked and disassembled as  desire
d."'' INK of; FLASH oa;"MULTILATERAL NUMBERS"; FLASH OO;":"'
'"The upper limit for conversions and math is FFFFFF h, 16,7
77,215decimal."'' INK oc
9802 PRINT "Always enter hex in CAPS."'' INK od;"You may do
 mixed math, multiply,divide, add, subtract.  Watch  the men
u for electives."'' INK oe;"For Division, enter largest
 number first."''"Accurate division dividend is  expressed i
n decimal fraction.  DISREGARD HEX DIVIDEND IF THE  DECIMAL
 IS IN FRACTION."'' INK of;"You may do MIXED addition using
 negative DECIMAL numbers, BUT  NOT NEGATIVE HEX.  If the SU
M isnegative, the decimal SUM will  be correct, but the Hex
 SUM willbe false."'' INK od;"Numeric expressions for DECIMAL
 input are O.K., ie: <PI * D>,  <2* PI *F*L> < PEEK 23635+2
56* PEEK 23636*-1>."
9803 INK of: PAUSE VAL ".14E4": CLS : PRINT AT OJ,OB;"PLEASE
 ENTER "; FLASH OA;"CAPS SHIFT"; FLASH OO;" MODE"''TAB OB;"T
HEN <"; FLASH OA;"CONT ENTER"; FLASH OO;">": STOP
9804 CLS : LET xy=oo: LET T=oo: INPUT "Select: 1. Peek and D
isassemble        2. Poke a Decimal Table        3. Poke a
 Hex table            4. Convert Hex-Dec-BIN          5. Mi
xed Math                6. Binary Table";M: CLS : GO TO (
m<oa AND m>of)*VAL "9804"+(m>oo AND m<od)*VAL "9834"+(m>oc A
ND m<OG)*VAL "9808"
9808 IF m=oe THEN LET xy=oa: GO TO VAL "9886"
9810 IF m=of THEN GO TO VAL "9934"
9812 INPUT "1. Dec to Hex, or 2. Hex to Dec 3. Quit?";X: IF
X=oa THEN GO TO VAL "9822"
9814 IF x=oc THEN GO TO VAL "9804"
9816 DIM x$(6): INPUT "HEX Group:";y$: LET x$=("00000"+y$ AN
D LEN y$=oa)+("0000"+y$ AND LEN y$=ob)+("000"+y$ AND LEN y$=
oc)+("00"+y$ AND LEN y$=od)+("0"+y$ AND LEN y$=oe)+(y$ AND L
EN y$=of): IF LEN y$>of THEN GO TO VAL "9816"
9818>LET D1=(1048576*(CODE x$(0
A)-Q) AND CODE x$(OA)<R)+(1048576*(CODE x$(OA)-P) AND CODE x
$(OA)>Q): LET D2=(65536*(CODE x$(OB)-O) AND CODE x$(OB)<R)+(
65536*(CODE x$(OB)-P) AND CODE x$(OB)>Q): LET D3=(4096*(CODE
 x$(OC)-O) AND CODE x$(OC)<R)+(4096*(CODE x$(OC)-P) AND CODE
 x$(OC)>Q): LET D4=(256*(CODE x$(OD)-O) AND CODE x$(OD)<R)+(
256*(CODE x$(OD)-P) AND CODE x$(OD)>Q): LET D5=(16*(CODE x$(
OE)-O) AND CODE x$(OE)<R)+(16*(CODE x$(OE)-P) AND CODE x$(OE
)>Q): LET D6=(CODE x$(of)-O AND CODE x$(of)<R)+(CODE x$(of)-
P AND CODE x$(of)>Q): LET DEC=D1+D2+D3+D4+D5+D6: LET M$=x$:
IF xy=oa THEN RETURN
9820 GO TO VAL "9826"
9822 INPUT "Input Decimal OR BIN Number: ";DEC
9824 DIM x$(of): LET A=DEC: LET B1=INT (A/1048576): LET A=A-
B1*1048576: LET x$(oa)=(CHR$ (B1+O) AND B1<10)+(CHR$ (B1+P)
AND B1>9): LET b2=INT (A/65536): LET A=A-B2*65536: LET x$(OB
)=(CHR$ (B2+O) AND B2<10)+(CHR$ (B2+P) AND B2>9): LET B3=INT
 (A/4096): LET A=A-B3*4096: LET x$(OC)=(CHR$ (B3+O) AND B3<1
O)+(CHR$ (B3+P) AND B3>9): LET B4=INT (A/256): LET A=A-B4*25
6: LET x$(OD)=(CHR$ (B4+O) AND B4<10)+(CHR$ (B4+P) AND B4>9)
: LET B5=INT (A/16): LET A=A-B5*16: LET x$(OE)=(CHR$ (B5+O)
AND B5<10)+(CHR$ (B5+P) AND B5>9): LET x$(of)=(CHR$ (A+O) AN
D A<10)+(CHR$ (A+P) AND A>9): LET M$=x$: IF xy=oa THEN RETUR
N : REM 13 IF THEN Conditionals this line
9826 CLS : PRINT "DEC=";DEC;TAB 15;"HEX=";
9828 FOR n=oa TO of: IF m$(n)>"O" THEN FOR n=n TO of: PRINT
m$(n);: NEXT n: GO TO VAL "9832"
9830 NEXT n: PRINT "Oh"'
9832 GO SUB 9946: GO TO VAL "9812"
9834 LET T=oa: INPUT "Start Adr? ";S: INPUT "END Adr? ";E;:
IF M=oa THEN FOR N=S TO E: LET A=PEEK N: LET X=N: GO SUB VAL
 "9874": GO SUB VAL "9856"
9836 IF m>oa THEN CLS : PRINT "Working": GO TO VAL "9842"
9838 IF N>=E THEN PRINT "DONE"''"Enter when Ready.": PAUSE 4
e4: GO TO VAL "9804"
9840 NEXT N
9842 LET RT=INT (S/256): LET GT=S-(RT*256)-oa: IF GT<oa THEN
 LET GT=254: LET RT=RT-oa
9844 POKE VAL "23731",RT: POKE VAL "23730",GT
9846 RESTORE (M=OB)*VAL "9878"+(M=OC)*VAL "9880"
9848 FOR N=S TO E
9850 IF M=ob THEN ON ERR GO TO VAL "9882": READ A: POKE N,A:
 NEXT N
9852 IF N>=E THEN GO TO VAL "9882"
9854 IF M=oc THEN ON ERR GO TO VAL "9882": READ X$: GO SUB V
AL "9876": POKE N,A: NEXT N
9856 LET A=PEEK X: PRINT X;TAB OG;A;TAB OK;"d";: GO SUB VAL
"9874": PRINT TAB OO;x$;" h";: ON ERR GO TO VAL "9862": PRIN
T TAB VAL "24";CHR$ A
9858 RETURN
9860 ON ERR RESET : PRINT : RETURN
9862 ON ERR RESET : PRINT : RETURN
9864 IF A<33 THEN PRINT " "
9866 IF T=oa THEN RETURN
9868 POKE X,A: IF N=A THEN PRINT "DONE": BEEP oa,OL: GO TO V
AL "9804"
9870 LET x=x+oa
9872 NEXT N
```

Cont Next Page

```
9874>DIM x$(OB): LE
T B=INT (A/16): LET C=A-B*16: LET x$(OA)=(CHR$ (B+0) AND B<1
0)+(CHR$ (B+P) AND B>9): LET x$(OB)=(CHR$ (C+0) AND C<10)+(C
HR$ (C+P) AND C>9): RETURN
9876 LET D1=(16*(CODE x$(OA)-0) AND CODE x$(OA)<R)+(16*(CODE
 x$(OA)-P) AND CODE x$(OA)>Q): LET D2=(CODE x$(OB)-0 AND COD
E x$(OB)<R)+(CODE x$(OB)-P AND CODE x$(OB)>Q): LET A=D1+D2:
RETURN
9878 DATA 84,104,105,115,32,100,97,116,97,32,116,97,98,108,1
01,32,105,115,32,115,104,111,114,116,46,1000
9880 DATA "54","68","69","73","20","6D","73","67","20","69",
"73","20","69","6E","20","6D","65","6D","6F","72","79","2E",
"STOP"
9882 ON ERR RESET : CLS : PRINT '''The message is in memory.
  The  starting adr ";S;".  ''''Use the MENU  to PEEK it out
.''''<ENTER>": PAUSE VAL "4e4": CLS : GO TO VAL "9804"
9884 LET oo=0: LET xy=oo: LET oa=1: LET ob=2: LET oc=3: LET
od=4: LET oe=5: LET of=6: LET OG=7: LET OH=8: LET OI=9: LET
OJ=10: LET OK=11: LET OL=12: LET OM=13: LET ON=14: LET OP=15
: LET OQ=16: LET OR=17: LET OS=18: LET OT=19: LET OV=20: LET
 O=48: LET P=55: LET Q=64: LET R=65: LET A=OA: DIM V(24): FO
R N=1 TO 24: LET V(N)=A: LET A=A+A: NEXT N: RETURN
9886 INPUT "Select Choice:
              <1> Add       <2> Subtract      <3> Multiply
  <4> Divide        <5> Home Menu ";g1: IF g1<oa OR g1>oe TH
EN GO TO VAL "9886"
9888 IF g1=oe THEN GO TO VAL "9804"
9890 CLS : INPUT "First Number HEX or DEC/BIN?    <1> Hex
<2> Dec/Bin";M1: IF m1<oa OR m1>ob THEN GO TO VAL "9890"
9892 INPUT "Second Number HEX or Dec/Bin?  <1> HEX  <2> Dec
/Bin ";m2: IF m2<oa OR m2>ob THEN GO TO VAL "9892"
9894 IF m1=oa THEN INPUT "INPUT First Number (HEX) ";y$: IF
y$="" THEN GO TO VAL "9894"
9896 IF m1=ob THEN INPUT "INPUT First Number (Dec/Bin)    ";
y$: LET n1=VAL y$: IF y$="" THEN GO TO VAL "9896"
9898 IF m2=oa THEN INPUT "INPUT Second Number (HEX): ";z$: I
F z$="" THEN GO TO VAL "9898"
9900 IF m2=ob THEN INPUT "INPUT Second Number (Dec/Bin):  ";
z$: LET n2=VAL z$: IF z$="" THEN GO TO VAL "9900"
9902 DIM x$(of): IF m1=oa THEN LET x$=(y$ AND LEN y$=of)+("0
"+y$ AND LEN y$=oe)+("00"+y$ AND LEN y$=od)+("000"+y$ AND LE
N y$=oc)+("0000"+y$ AND LEN y$=ob)+("00000"+y$ AND LEN y$=oa
)+("000000" AND y$=""): GO SUB VAL "9818": LET n1=DEC
9904 IF M2=oa THEN LET x$=(z$ AND LEN z$=of)+("0"+z$ AND LEN
 z$=oe)+("00"+z$ AND LEN z$=od)+("000"+z$ AND LEN z$=oc)+("0
000"+z$ AND LEN z$=ob)+("00000"+z$ AND LEN z$=oa)+("000000"
AND z$=""): GO SUB VAL "9818": LET N2=DEC
9906 IF G1=oa THEN LET DEC=N1+N2: GO SUB VAL "9824": PRINT y
$;("h" AND M1=oa)+("d" AND M1=ob);"+";z$;("h" AND M2=oa)+("d
" AND M2=ob);"=";DEC;"d";" and "
9908 IF G1=oa THEN FOR N=oa TO of: IF M$(N)>"0" THEN FOR N=N
 TO of: PRINT M$(N);: NEXT N: PRINT "h"': GO TO VAL "9912"
9910 IF G1=oa THEN NEXT N: PRINT "0h"'
9912 IF G1=ob THEN LET DEC=N1-N2: GO SUB VAL "9824": PRINT y
$;("h" AND M1=oa)+("d" AND M1=ob);"-";z$;("h" AND M2=oa)+("d
" AND M2=ob);"=";DEC;"d and"
9914 IF G1=ob THEN FOR N=1 TO of: IF x$(N)>"0" THEN FOR N=N
TO of: PRINT x$(N);: NEXT N: PRINT "h"': GO TO VAL "9918"
9916>IF G1=ob THEN NEXT N: PRINT "0h"'
9918 IF G1=oc THEN LET DEC=N1*N2: GO SUB VAL "9824": PRINT y
$;("h" AND M1=oa)+("d" AND M1=ob);" * ";z$;("h" AND M2=oa)+(
"d" AND M2=ob);"=";DEC;"d";" and "
9920 IF G1=oc THEN FOR N=oa TO of: IF x$(N)>"0" THEN FOR N=N
 TO of: PRINT x$(N);: NEXT N: PRINT "h"': GO TO VAL "9924"
9922 IF G1=oc THEN NEXT N: PRINT "0h"'
9924 IF G1=od THEN LET DEC=N1/N2: GO SUB VAL "9824": PRINT y
$;("h" AND M1=oa)+("d" AND M1=ob);" / ";z$;("h" AND M2=oa)+(
"d" AND M2=ob);"=";DEC;"d and "
9926 IF G1=od THEN FOR N=oa TO of: IF x$(n)>"0" THEN FOR N=N
 TO of: PRINT x$(N);: NEXT N: PRINT "h"': GO TO VAL "9930"
9928 IF G1=od THEN NEXT N: PRINT "0h"'
9930 GO SUB 9946: GO TO VAL "9886"
9932 GO TO VAL "9886"
9934 CLS : PRINT "1111 1111 1111 1111 1111 1111";AT OO,20; O
VER 1; FLASH 1;"          "; OVER OO; FLASH OO'''From Right T
o Left, the Binary  bit data is given for a 24 bit  BIN grou
p.  Left colum is bits 1-24 with bit content, GROUP    Cont
ent and progressive possibleTotal content.''': LET a=1: LET
b=0: LET C=OO
9936 FOR y=1 TO 24 STEP 4: GO SUB VAL "9940": PRINT TAB 10;b
'': NEXT y
9938 PRINT '''BIN-HEX-DECIMAL DIGIT AND GROUP ANALYSIS"'TAB
 2;"F";TAB 7;"F";TAB 12;"F";TAB 17;"F";TAB 22;"F";TAB 27;"F"
'''1111 1111 1111 1111 1111 "; FLASH OA;"1111"; FLASH OO'"^^
^^ ^^^^ ^^^^ ^^^^ ^^^^ ^^^^_1"'TAB 25;"^^^^_2"'^^^^^ ^^^^ ^^^
^ ^^^^ ^^^^ ^^_4 GP"'TAB 25;"^_8  1"'TAB 20;"^^^^^";TAB 28;"F
=15"'TAB 20; FLASH OA;"1111"; FLASH OO'"^^^^^ ^^^^ ^^^^ ^^^^
^^^^_16"'TAB 20;"^^^^_32"'TAB 20;"^^_64  GP-2"'^^^^^ ^^^^ ^^^
^ ^^^^ ^_128  F=240"'TAB 15; FLASH OA;"1111"; FLASH OO'"^^^^^
 ^^^^ ^^^^ ^^^^_256"'TAB 15;"^^^^_512"'TAB 15;"^^_1024  GROUP
-3"'TAB 15;"^_2048   F=3840"
9939 PRINT TAB 10; FLASH OA;"1111"; FLASH OO'"^^^^^ ^^^^ ^^^^
_4,096"'TAB 10;"^^^^_8,192"'TAB 10;"^^_16,384     GROUP-4"'TA
B 10;"^_32,768      F=61,440"'TAB 5; FLASH OA;"1111"; FLASH
OO'"^^^^^ ^^^^_65,536"'TAB 5;"^^^^_131,072"'TAB 5;"^^^_262,144
      GROUP-5"'TAB 5;"^_524,288         F=983,040"' FLASH O
A;"1111"; FLASH OO'"^^^^^_1,048,576"'^^^^_2,097,152       GRO
UP-6"'^^^_4,194,304        F=15,728,640"'^^_8,388,608"''"TOT
AL CONTENT=16,777,215= 16MBTS"'' FLASH oa;"ENTER"; FLASH oo;
" When Ready": PAUSE VAL "4e4": CLS : GO TO VAL "9804"
9940 LET C=OO: FOR n=OA TO OD: PRINT a: LET b=b+a: LET C=C+A
: LET a=a*2: NEXT n: PRINT FLASH OA;C; FLASH OO;
9942 RETURN
9944 FOR n=oa TO od: PRINT a: LET b=b+a: LET a=b*ob: NEXT n:
 RETURN
9946 LET M=DEC: DIM T(24): FOR N=24 TO 1 STEP -1: LET T(N)=(
1 AND M>=V(N))+(0 AND M<V(N)): IF T(N)=1 THEN LET M=M-V(N)
9948 NEXT N
9950 LET T$="": FOR N=24 TO 1 STEP -1: IF T(N)=1 THEN FOR Y=
1 TO N: LET T$=T$+STR$ T(Y): NEXT Y: GO TO 9953
9952 NEXT N
9953 LET M$="": FOR N=LEN T$ TO 1 STEP -1: LET M$=M$+T$(N):
NEXT N
9954 PRINT '''BIN=";M$: RETURN
```

— End —

up Arrows

Quick and Easy Routines

THE QUICKIE LIST: Sometimes the shorties are the most useful. This one was sent by Mr. Robert Hartung. Did you digest his articles in CTM? If not you should back up four issues of CTM and dig his series. (quote) To make a hard copy of a disk CAT for, labels, or an index, or a copying a screen: <OPEN #2,"P": LET /p=o> When CAT/ is entered the disk directory goes to the printer instead of the screen. Or, any PRINT statement will go to the printer. When finished, <CLOSE #2> to terminate. (Thanks Bob!)

YES - NO INKEY$ for use after a prompt requiring YES or NO: GO SUB 200 will cause line 200 to loop itself until a Y, y, N, or n key is touched. YES assigns z=1 or NO assigns z=2. <200 PAUSE VAL "4e4": LET T$= INKEY$: IF T$<>"y" AND T$<>"y" AND T$<>"n" AND T$<>"N" THEN GO TO 200> <202 LET z=(1 and T$="y" OR t$="Y")+(2 AND T$="n" OR T$="N": RETURN>

INKEY$ Prompt for any key: Any key touched will result in its character being input to T$. Numbered characters will be represented by variable Z. Ideal for use with <GO SUB> after menu that has several selection options. (200 PAUSE VAL "4e4": LET T$=INKEY$: LET z=CODE T$-48: RETURN>.

KEY LOCK after the above INKEY$ routine is used: Suppose that you have a MENU with 9 selection choices 1-9. <GO SUB 200 will wait until a key is touched, then assign "Z=key struck". BUT, we want to lock out all except 1-9. <90 menu print statements: GO UB 200> <102 If Z<1 OR z>9 THEN GO TO 90>. Better still, lets use one line to lock the keys, PLUS- take action on ALL 9 menu selections. <102 GO TO (z<1 OR z>9)*90+ (z=1)*150+ (z=2)*200+ (z=3)*300+ (z=4)*400+ (z=5)*500+(z=6)*600+ (z=7)*700+ (z=8)*800+ (z=9)*900>. In this case there are 10 IF THEN logicals in line 102 that sends the program to the line numbers following the condition enclosed in brackets. (BOOLEAN Logicals works just fine, and conserves lots of memory. We will publish an article soon about Mr. Boolean's 19th Century works and how the logic applies to BASIC Computer programming.)

THE SDOS PRINTER DRIVER: <LET /P=0> enables the built in Centronics printer driver. LPRINT commands then sends character streams through the Oliger or Aerco printer interface to a centronics printer. There are two TS-2068 memory addresses that one needs to know. The LET /P=0 safe disk command "zeroes address 23324, which is used as a switch to command a line feed. If PEEK 23324=0 then NO line feed is directed by the printer driver. So, if your printer requires a line feed, you must POKE 23324,10. Another important address is 23323 which contains "Maximum Line Length". LET/P=0 places 255 as the content of Adr 23323. For most printers this is O.K., even if the printer's maximum line is shorter than 255 characters. If you want to limit the printed line to any given length, say 60 characters, then POKE 23323,60. Then you must take TAB into consideration. For example: <LPRINT TAB 20;A$> would be O.K. if the length (LEN) of a$ were 40 characters. An additional character (41) would force a line feed and wrap print to the next line. A problem would occur if printing lines of a character array where the empty ends of the array lines are spaces. LLISTS in columns of 60 lines are easily printed by <LET/P=0: POKE 23323,60: LLIST>. If you want double spaced LLIST lines, add <POKE 23324,10> before the LLIST command.

SOFTWARE CHECK OF PRINTER STATUS: This may come as a "EUREKA! revelation" to some who are having trouble constructing program lines to COMMAND A PRINTER to change formats or print styles. When your printer is sitting there with lights on, it is in a READY STATUS. To be sure you can Type <PRINT IN 127 ENTER>. If the printer is READY, a "237" will appear on screen. When constructing program lines to COMMAND the printer to do such things as changing a print style or set EXPANDED characters, the programming is <OUT 127,number> then <OUT 127,another number> until all required "change commands" are sent. The problem is, a printer is slow to execute the commands, while the computer is fast in sending the commands. Most likely, the computer will finish sending three or more commands before the printer has executed the first one. As a result, the printer may not SEE some commands. A program loop is needed to make the software WAIT until the printer is in a READY status. Here it is: <100 IF IN 127=253 THEN GO TO 100>: RETURN>. When the printer is NOT READY, the command <PRINT IN 127 ENTER> returns the number 253 (BUSY). So, a <GO SUB 100> placed between each printer command will loop at line 100 until the printer is READY, and then RETURN for the next command. The COMMAND LINE would look like this: <50 GO SUB 100: OUT 127,27: GO SUB 100: OUT 127,60>. A printer BUSY status would hold up each command at line 100 until the printer status changes to READY (237).

PSEUDO HEX VARS TABLE with LETTERHEAD DATA

This vars file can be stored in disk for call up to your programs. Numbers 0 through 20 are represented by "double 0 vars" as, oo=0, ob=2, oc=3, through ou=20. When used in a program, each number 0-9 usage will conserve 5 bytes of memory. Each usage of the vars for numbers 10-20 conserves 6 bytes. N$ array is set up to service a LETTERHEAD, SIGN OFF, and SALUTATION BLOCK needs for letter correspondance. See the listing for designations.

```
 5 REM *** Type <CLEAR> then <GO SUB 10> then <SAVE/"pseu
"VAL>Note the letter support data forN$ array. Array lines
1-5 are reserved for Salutation block: 1=Company name. 2=D
ept. 3=Streetaddress. 4=City State Zip. 5= correspondants
first name. 10=input date. 11=spare line.
 6 REM ***Use <LOAD/"pseu"VAL>in a program line to initial
ize the variable table into a activeprogram.
 10 LET oo=0: LET oa=1: LET ob=2: LET oc=3: LET od=4: LET o
e=5: LET of=6: LET og=7: LET oh=8: LET oi=9: LET oj=10: LET
ok=11: LET ol=12: LET om=13: LET on=14: LET op=15: LET oq=16
: LET or=17: LET os=18: LET ot=19: LET ou=20: DIM n$(13,31):
 LET n$(6)="letterhead first line": LET n$(7)="second line":
 LET n$(8)="third line": LET n$(9)="fourth line": LET n$(11)
="Sincerely": LET n$(13)="Signature line": RETURN
```

Programming with SDOS BASIC Commands

<LET /D=0>: The range of disk selection is 0-3 to select one of up to four drives in a system, where disk number zero is the first disk and disk number three is the fourth disk. Disk drives are connected via two cables, the 36 conductor data/control cable, and the 3 conductor power cable. When a <LET /D=> command is sent, the SDOS sends "switching signals" to the "drive number selected by the LET/D" command". The signal "turns on the drive power". The disk door handle (most disk drives) is also a switch. When the door handle is open, the disk drive cannot be selected. When the door is closed, the drive may be selected. The power supply is passive, in that power is supplied to all disks, ready to be switched by SDOS.

The cables to the disk drives are very simple. Connections are parallel. Each disk drive and the SDOS controller uses the same 36 pin connector. These connectors are simply "pressed on" to the 36 conductor cable and spaced so that the connectors will reach the plugs of the disk drives. For more than one disk drive, the power is connected via "Y splitter cables". The parts to fabricate these cables are widely available and quite inexpensive.

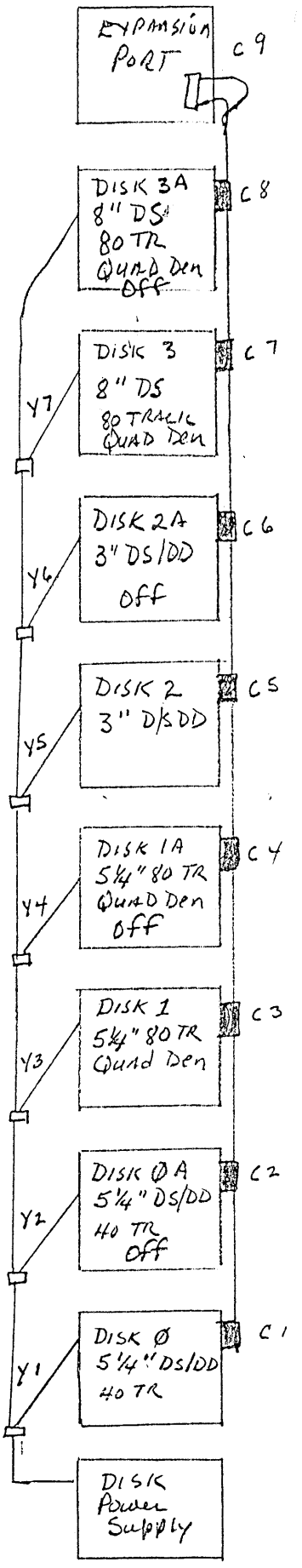
Most disk drive power supplys have sufficient power to operate four drives. If erratic operation is experienced while operating a multi drive system, the door latches of all drives except the one in use may be opened to relieve the power drain. Power supplies should be enclosed in a metal cabinet for shielding. The signals carried within the 36 conductor cable can be distorted by radiation of EMF from a un-shielded power supply. Also, diskettes placed within range of such fields can be made un-readable.

Although we are limited to four "addressable" disk drives in a SDOS system, more drives may be operated. At right is a "speculated" system of eight drives of different size and track formats. Such a system would allow the use of 3", 5 1/4" and 8" drives in a system, and two of each type drive. When more than four drives are connected, all except the one in use should have the drive doors open so that the parallel drive is dead. Parallel drives are two or more that have the internal drive switches set to the same "drive number". One other consideration is the "terminating resistor" should be in the active chain. I hope that someone will comment upon this, as I have operate my four drive unit with the "door open" on the drive that has the resistor, and no ill effects have been noted.

<LET /S=1> would be needed after a drive has been selected IF the disk is a single side disk (it has only one read/write head). The SDOS system will set itself for double side drive if there is no LET/S=1 command. DS/DD diskettes may be used for single side formats, but not vice versa. A diskette that is formatted with LET/S=1 can be written to or read from by a DS/DD drive, but a single side drive cannot read a disk formatted with LET/S=2.

<LET/T=40> establishes the format for 40 tracks. Normally a 40 track diskette will format to 41 tracks and give reliable read and write results. This will give 400K of disk free on a 40 track or 805K on a 80 track diskette. A diskette will accept LET/T=44, and will indicate a good format, however I have found that the capacity is really only 400K, and disk read/write errors occur if there is an attempt to store more than 400K. (LET/T=82 and 805K for 80 track drive).

I have had almost a year of trouble free disk operation with two 5 1/4" DS/Quad DENSITY drives purchased from BG Micro (price $89.00 each). One of the original drives would not read/write upon receipt and was replaced promptly and without question. (I thought some would like to know this.) SDU will discuss more about the SDOS BASIC commands in each issue.



Software and Hardware

ARTICLES

and

REVIEWS

About Disk Drives

There are many types of disk drives available. Commodore introduced a 3 1/2 inch drive with their Amiga computer. Now IBM has a new 3 inch drive. We Safe Disk users have a system that will work with almost anything that can be hooked up to a "Shughert Standard Parallel I/O" Disk Drive. Safe Disk uses a standard 36 pin data cable. Thats great, but we need to think about some things before dashing off and purchasing a disk drive. Of primary consideration is whether the drive that we purchase will be supported by most of our software suppliers. Another thought is the cost of the diskettes for the drive that we purchase. Still more; Does the disk format enough space to support "disk dependant software". Lets explore some of these factors.

Software Suppliers most generally supply their software on 5 1/4 inch, DS/DD, diskettes. Not many suppliers of TS-2068 software are willing to purchase another disk drive to support a few owners of 3 inch, 3 1/2 inch, 8 inch, drives. To the Safe Disk user this means that at least one 5 1/4 inch DS/DD drive will be needed in order to take advantage of available disk software. Fortunately, Safe Disk will easily accommodate "mixed" drives. That is to say, one can hook together a 3 inch (Portugual) disk drive with a standard 5 1/4 inch DS/DD drive, and work from one to the other. In such case, < LET /d=0: LET/s=2: LET /t=40: SAVE /"name" > would save a memory program to a "double side 40 track" (Double or Quad Density) drive in drive 0.

< LET /d=1: LET s=1: LET t=40: SAVE /"name" > will save a program to diskette in drive 1 when drive 1 is a "single side drive". In either case, the "size" of the drive is unimportant. These two examples are typical of a "drive mix" of a 5 1/4 inch DS/DD in drive Zero, and a "Portugual 3 inch" drive in drive 1. All other Safe Disk LOAD/SAVE/FORMAT commands work as well for such a drive mix. The beauty of Safe Disk is that it allows "program commands" to switch operation from one type of drive to another. Most disk operating systems are fixed in the interface firmware to operate with only one type of drive. How does one "hook up" two drives that are different? Thats simple too. Another 36 pin connector is simply "pressed on" to the existing data cable for hook-up to the back of the second drive. These press-on connectors are widely available. Also, a "Y splitter" power cable is attached to the existing drive power cable that connects to the disk drive. So, getting into business with a second disk drive "of another size" is very simple. What about the cost? Well, one firm "B6 Micro" sells TEAC DS/DD, 40 Track drives for $69.00. The same firm sells 5 1/4" 80 Track, Quad Density drives for $89.00 each. The "Y Splitter" and Data Connector can be purchased for about $3.00.

For about $75.00, one can add a standard 40 or 80 track drive to a Portugual Drive, have the best of two excellent systems, and load software supplied by our vendors. My drive configuration is: Two DS/DD, 40 track, plus Two DS, Quad Density, 80 track drives. The 40 track drives format to 395K capacity, and the 80 track drives format to 795K each, for a total disk system capacity of 2370K. The nice thing about using this mix of 5 1/2" drives is that the diskettes are interchangable. Programs on a 40 track diskette will LOAD in the 80 track drives. Also, 80 track diskettes will FORMAT to 40 tracks and become usable in the 40 track drives. Another good point is the cost of the diskettes, about 59 cents each from mail order sources.

Disk Dependant Software: The concept is to eliminate all menus and screens from the mother program. Put them in disk for quick boot in as needed. This saves about 1/3 of a program's length an allows many more functions to be added. Also, data files are dumped to disk before computer memory becomes full, and variable files are saved to disk for use by the main program. Next, with the new MERGE capability, entire segments of a program can be saved as "program cells" in diskette, to be MERGED to a software program as needed. The concept allows the software to have a "program length" that far exceeds the memory capacity of the computer's RAM. This requires lotsa disk capacity, more than a single side disk can hold. This is another reason that at least one disk in a system should be a double side disk. The single side disk just does not format enough disk capacity for such programs.

So, whether we like it or not, the "Standard" for most small computer systems is the 5 1/4 inch DS/DD Drive. The 5 1/4 inch diskettes are the least expensive. The drives themslves are less expensve. The DS/DD drive formats enough continuous disk capacity to accommodate extensive "disk dependant" software. And, all software vendors support the 5 1/4 inch DS/DD drives.

By: Bill Jones

SDOS MAIL MERGE

Complete This Issue

A Major Software for SDOS

The feature program for the October 88 issue is "SDOS MAIL MERGE", a major software that was designed specifically for this issue. There are several reasons why this software is special, and why almost all will surely want it. First, it is the ONLY "printing software" that will work AS IS with ANY Centronics printer, plus the TS-2040 printer. There is absolutely no monkeying around changing line numbers required to make it print with "brand y" printer. Second, it has many more features than just a Mailing List software. It is a "Dual Data Base" and printing software that can be used for many purposes. Third, the data products are simple Character arrays that can easily be transferred from diskette to other operating softwares, using the LOAD /"x"DATA x$() command.

The purpose for designing SDOS for SAFE DISK UP-DATE was two fold. 1. A good software of this kind was not available. 2. A vehicle was needed for illustrating the versatility of SDOS. I'll tell you a little tale on myself to make a point. A year AFTER buying a new Edsel, I discovered that the seat would tilt forward, as well as adjusting fore and aft. We have that nice SDOS system which has many many capabilities to discover. Safe Disk Up-Date will try to reduce the discovery time. This software can unveil many tricks of SDOS programming. AND, it will definitely fill a gap in our TS-2068 software needs.

FUNCTIONS

There are two main data bases, both flexible to the user. The "main" data base is for listings of all types, "Mailing List" being the common usage. A "field" consists of 7 lines of data. The data lines are organized by "field numbers" as follows: 1. Company Name. 2. Department. 3. Street Address. 4. City, State, Zip. 5. First name. 6. Telephone Nbr. 7. Other data. The operator is prompted to "Input Number of Fields". This allows from one to about 120 fields to be organized. Then prompts ask for inputs by line number. A listing can be discontinued at any time. Menu choice allows "adding to" the list at any field number. So, if one inputs 20 names and quits, the next entry can be started at "field 21" by prompt. Menu choices are: 1. Start New. 2. Correct. 3. ADD TO. 4. View. 5. Print/Labels/Envelopes. 6. SAVE. 7. DELETE Data. 8. Sort File. 9. LOAD File from Disk.

Ancillary functions are: 1. Centered Captions. 2. Page Management. 4. Directed Line Space. 5. Directed Page End. 6. Print a Centered Letterhead. 7. Print a Sign Off. 8. Single or Double line spacing. 9. Change Line Length. 10. Set Up a Outline. 11. Print Outline. The "Outline Functions"

Program Listing

```
 400 INPUT "Number of Mailing list Names?";h: LET k=h: LET b$="'
 402 INPUT "Sort by: <1> First name                <2> Las
t name";z
 404 IF z=oa THEN GO SUB VAL "1010": GO TO VAL "500"
 406 IF z=ob THEN GO SUB VAL "640": GO SUB VAL "1010": GO SU
B VAL "670"
 500 GO TO VAL "2000"
 640 CLS : PRINT FLASH oa;AT oj,oa;"Last Name Reverse- 40 Se
conds";AT ok,oa;"Line 640": FOR n=oa TO h: IF o$(n,oa)(oa TO
 od)="    " THEN NEXT n: RETURN
 642 LET b$=o$(n,oa)
 646 IF b$(LEN b$)=" " THEN LET b$=b$( TO LEN b$-oa): GO TO
VAL "646"
 650 FOR l=LEN b$ TO oa STEP -oa: IF b$(oa TO od)="    " THE
N NEXT n
 652 IF n>k THEN RETURN
 654 IF b$(1)=" " THEN LET o$(n,oa)=b$(1+oa TO )+" "+o$(n,oa
)( TO 1): NEXT n
 656 NEXT l
 658 NEXT n
 660 RETURN
 670 CLS : PRINT AT oj,oa;"Replace First name- 40 Seconds";A
T ok,oa;"Line 670": LET b$=""
 672 FOR n=oa TO k: IF o$(n,oa)(oa TO od)="    " THEN NEXT n
: RETURN
 674 IF n>k THEN GO TO y
 676 LET b$=o$(n,oa)
 682 FOR l=oa TO LEN b$: IF b$(l)=" " THEN LET b$=o$(n,oa)(l
+oa TO ): GO TO VAL "686"
 684 NEXT l
 686 IF b$(LEN b$)=" " THEN LET b$=b$( TO LEN b$-oa): GO TO
VAL "686"
 688 LET o$(n,oa)=b$+" "+o$(n,oa)( TO 1)
 690 NEXT n
 692 RETURN
1010 CLS : PRINT AT oj,oh; FLASH oa;"Sorting- 1 Minute";AT o
k,oh;"Line 1010": DIM v$(of,31): LET n=h
1020 LET I=N-oa
1030 FOR J=oa TO I
1040 LET K=J+oa
1050 FOR L=N TO K STEP -oa
1052 IF o$(l,oa)(oa TO od)="    " THEN GO TO VAL "1100"
1060 IF o$(L,oa))o$(J,oa) THEN GO TO VAL "1100"
1070 IF o$(l,oa)(oa TO od)<>"    " THEN LET b$=o$(L,oa): LET
 v$(oa)=o$(l,ob): LET v$(ob)=o$(l,oc): LET v$(oc)=o$(l,od):
LET v$(od)=o$(l,oe): LET v$(oe)=o$(l,of): LET v$(of)=o$(l,og
)
1080 IF o$(l,oa)(oa TO od)<>"    " THEN LET o$(L,oa)=o$(J,oa
): LET o$(l,ob)=o$(j,ob): LET o$(l,oc)=o$(j,oc): LET o$(l,od
)=o$(j,od): LET o$(l,oe)=o$(j,oe): LET o$(l,of)=o$(j,of): LE
T o$(l,og)=o$(j,og)
1090 LET o$(J,oa)=B$: LET o$(j,ob)=v$(oa): LET o$(j,oc)=v$(o
b): LET o$(j,od)=v$(oc): LET o$(j,oe)=v$(od): LET o$(j,of)=v
$(oe): LET o$(j,og)=v$(of)
1100 NEXT L
1110 NEXT J
1120 CLS : FLASH oo
1130 RETURN
2000 CLS : PRINT AT 2,6; INK 5;"MAIL";TAB 19; FLASH 1;("Yes"
 AND LEN o$(1,1)>1)+("No " AND LEN o$(1,1)<2); FLASH 0;" ML
DATA"'TAB 9;"MENU";TAB 19;"FREE="; FREE ;"    "'TAB 6; INK 6
;"<1> Start New"'TAB 6; INK 5;"<2> Add to List"'TAB 6; INK 6
```

Cont—

involve a second flexible data base to use for producing a outline with major and minor topics. Printing of the outline is in the outline format with selectable TAB for major topics. Minor topics are then indented five spaces.

OPERATION

Other uses of the mailing list data base are: Invoices, Inventory lists, Telephone lists, Magazine article indexes, or any data groups that one can set up in 7 line fields. The capacity of the mail/inventory data base is about 120 fields, as programmed. The fields may be dimensioned larger or smaller by simple line number adjustment (line nbr 2005).

Simplicity presonified! You just cannot get into trouble, and the software is crash proof. A mistake in operation will result in a "report code" stop. The cure is <GO TO FM> in every case. <GO TO FM> presents the Function Menu, and all program parameters are preserved. There is only one possible exception to this easy fix. If you have established a "outline data base" and then you select "Start New" at the Mail List Menu, it is possible to "over dimension" the mailing list data base by input of too large a number for "Number of Fields". When a command is given to DIMENSION a character array, and insufficient memory exists, the vars file is corrupted which results in a CRASH. Now this little blurb on operation is indeed skimpy, but what can one say when words are not necessary? The Menus provide ample guidance with literal selection choices, thus no need to study instructions.

The Mail/Inventory List Printer: A prompt asks for "Input Number of Fields to print". If printing labels or envelopes, "4" would be the correct input. Then "TAB, and Spaces between" are prompted. Inputs allow the printing to strike the labels or envelopes uniformly. The software differentiates between a three or four line address and spaces accordingly. Another prompt asks, "Twin Labels?". If Yes, then your own return address label prints between each "out label". When you want to print inventory or other data set-ups, an input of 7 will result in all 7 lines printing. Any number of fields 1-7 may be elected to print out.

Menu electives set up automatic SAVE or LOAD of the data bases OR a program that contains data. A prompt "Input File Number" makes each LOAD or SAVE discrete. LOAD selections result in the disk catalog presented on screen with a prompt "Input File Name". The Load is automatic and the Function Menu appears. At that time all software functions are available.

PROGRAMMING

SDOS Mail Merge utilizes the <LET /P=0> or <LET /P=T> functions of SDOS to print with any Centronics printer or the TS-2040. Inputs allow the configuration of desired line length for any pre-set print style. The built in SDOS printer driver is used for printing. The software itself


```
;"<3> View List"''TAB 6;"<4> Correct List"'TAB 6; INK 5;"<5>
 Print Lab/Env/Inv"'TAB 6; INK 6;"<6> Disk SAVE"''TAB 6;"<7>
 Delete Mail Data"'TAB 6; INK 5;"<8> Sort File"'TAB 6; INK 6
;"<"; FLASH 1;"9"; FLASH 0;"> Function Menu"''TAB 6; INK 5;"
<0> LOAD Mail File"''TAB 6;"Page=";pg;" ";TAB 20;"Line=";nn
2002 PRINT AT ot,ok;pg;" ";AT ot,VAL "25";nn;" ";AT ob,ot; F
LASH oa;("NO " AND LEN o$(oa,oa)<ob)+("Yes" AND LEN o$(oa,oa
))oa); FLASH oo;AT oc,VAL "24"; FREE
2003 GO SUB SQ: GO SUB I1: GO TO (Z<00 OR z>0I)*VAL "2000"+(
z=00)*VAL "3020"+(Z=0A)*VAL "2004"+(z>0A AND Z<0F)*VAL "2007
"+(Z=0F)*VAL "2600"+(Z=0G)*VAL "2058"+(Z=0H)*VAL "400"+(Z=0I
)*FM
2004 CLS : PRINT AT 0J,0F; FLASH 0A;"Confirm, START NEW"; FL
ASH 00''TAB 0F;"<y> Yes -or- <n> No": GO SUB SQ: GO SUB IK:
IF Z=0A THEN INPUT "Input Nbr of Names";mx: IF mx*VAL "217"
VAL "27000" THEN GO TO VAL "2004"
2005 DIM o$(mx,7,31): DIM O$(mx,7,31): LET IN=0A: LET ms=0A:
 CLS : GO TO VAL "2009"
2006 GO TO VAL "2000"
2007 IF Z>0A AND Z<0F THEN INPUT "Ent Strt Nbr ";MS: CLS : I
NPUT "Ent end Nbr ";MX: CLS
2008 IF Z=0E THEN CLS : INPUT "Twin Labels? <y> yes or <n> N
o ";t$: INPUT "Ent sps bet: ";SX: INPUT "Ent Tab ";TB: CLS
: INPUT "Nbr of Fields (1-7) 4 for labels";x: PRINT AT 0V,0B
;"Do you want Page Control?": INPUT "<1> yes or <2> No ";K:
 IF k=oa THEN PRINT AT 0V,00;"Ent ""PG"" for same page, or a
 nbrto change. ": INPUT PG: PRINT AT 0V,00;"Ent ""NN"" for s
ame line count,or nbr to re-set line count. ": INPUT NN
2009 FOR O=MS TO MX
2010 IF Z<0C THEN GO SUB VAL "2022"
2011 IF Z=0D THEN GO SUB VAL "2110"
2012 IF Z=0C THEN GO SUB VAL "2034"
2014 IF Z=0E THEN GO SUB VAL "2040"
2016 IF O>=MX THEN GO TO VAL "2000"
2018 NEXT O
2022 CLS : PRINT AT 0F,0F;"Ml No. ";O
2024 CLS : PRINT AT og,oo;o;AT 0J,00;"Input ";("Co Name " AN
D IN=0A)+("Dept    " AND IN=0B)+("St adr  " AND IN=0C)+("ct,
St,Zp" AND IN=0D)+("1st Name" AND IN=0E)+("Tel Nbr " AND IN=
0F)+("spare dta" AND in=og);"  or zz to quit.": INPUT T$: IF
 T$="zz" THEN LET IN=0A: LET O=MX: CLS : RETURN
2025 LET O$(O,IN)=T$
2026 IF IN=0G THEN CLS : GO TO VAL "2030"
2028 LET IN=IN+0A: GO TO VAL "2024"
2030 PRINT AT 0H,00;O$(O,0A)'O$(O,0B)'O$(O,0C)'O$(O,0D)'O$(O
,0E)'O$(O,0F)'O$(o,og)''"OK?  <Y> Yes, or <N> No": GO SUB IK
: IF Z=0B THEN LET IN=0A: GO TO VAL "2024"
2032 CLS : LET IN=0A: RETURN
2034 IF O$(O,0A)(0A TO 0B)="zz" THEN GO TO ml
2035 PRINT AT 0E,oo;"Ml No. ";o''O$(O,0A)'O$(O,0B)'O$(O,0C)'
O$(O,0D)'O$(O,0E)'O$(O,0F)
2036 PRINT O$(O,0g)
2037 PRINT '"Key ENT for Next": PAUSE VAL "4e4": CLS : RETUR
N
2038 IF LEN O$(O,0A)<0E THEN RETURN
2040 FOR y=oa TO x: IF y=oa AND o$(o,y)( TO ob)="  " THEN RE
TURN
2041 LPRINT TAB tb;o$(o,y): LET nn=nn+oa: NEXT y: GO SUB VAL
 "2046"
2042 IF t$="y" OR t$="Y" THEN FOR y=oa TO od: LPRINT TAB tb;
n$(y): LET nn=nn+oa: NEXT y: GO SUB VAL "2046"
2044 RETURN
```

Cont —

does not command the printer to change print styles. Style changes must be set manually with the printer's dip switches, or by direct commands entered in the "K mode". If the user desires software controlled printer style changes, it is practical to build a few more lines to do that. These custom printer control lines would be operated in the K mode by either <GO TO line> or <GO SUB line>.

"Now you see it, then you dont". Saving memory is a necessity in order to create comprehensive software for the TS-2068. 38K of FREE poses limitations as to the number of functions that can be stuffed into a program. The Safe Disk Operating System (SDOS) provides some interesting memory conservation capabilities. We conserve a tremendous amount of FREE Ram by first creating the needed MENUS with line programming, then SAVING the Screens to diskette, and then DELETING the program lines. A 700 byte Menu is then presented when needed by this simple 10 byte command in a program line: <LOAD /"fm" SCREEN$). Five such Menus are "canned in diskete" for a saving of about 3K of memory usage.

Other memory saving techniques are used as well. A variable file is first created in program lines. Then after the vars are imbedded in memory, the file is SAVED to disk with <SAVE /"mrg" VAL>. Then the lines of programming are deleted. Ordinarily the variables in a basic program are SAVED with the program and the vars ride in with the program when it is re-loaded. The vars file in diskette serves as back up if a program is inadvertantly CLEARED. Deleting the program lines saves nearly a K of memory. Most importantly, the vars file contains another memory saving device to be discussed next.

"Pseudo Hex" is a neat system that uses double character variables to represent numbers 1 through 20. Oo=0, oa=1, ob=2, oc=3, on through ou=20. Each usage of characters instead of numbers SAVES 5 bytes for numbers 1-9 or 6 bytes for numbers 10 through 20. There are 160 lines of programming and each will average 3 usages of numbers 1-20. The "pseudo hex" table saves between 4 and 5K of memory. The entire vars file is only 569 bytes only part of which is the pseudo hex table. Therefore some 4K of memory is saved this way. Pseudo hex lends itself to human memory. One can count on ones pinkies with it, using oa for one pinkie, ob for two, etc. Soon it becomes habit.

Another memory saving technique is the use of BOOLEAN expressions instead of IF THEN. Boolean expressions are chained in a line of programming, where IF THEN expressions would require single lines. For example, the two lines: 10 IF A=1 THEN GO TO 100. 20 IF A=2 THEN GO TO 200. A single line with BOOLEAN would be: 10 GO TO (A<ob)* VAL "100"+(A>oa)* VAL "200". The program makes extensive use of Boolean. Some lines contain as many a 12 IF THEN logicals expressed in Boolean. Memory saving is tremendous, up to 70% as compared with conventional IF THEN expressions. The average savings realized is around 30% of total program length. In this program some 3K of memory usage is conserved.


```
2046 LET DD=SX: IF K=OA AND SX+NN>=LN THEN GO SUB PE: LET DD
=OO: RETURN
2047 IF DD>OO THEN LPRINT : LET DD=DD-OA: LET NN=NN+OA: IF N
N>=LN AND K=OA AND NN+OD>=LN THEN GO SUB PE: LET DD=OO: LET
NN=OA: RETURN
2048 IF DD>OO THEN GO TO VAL "2047"
2050 RETURN
2051 NEXT N
2052 GO TO VAL "2000"
2058 DIM o$(oa,oa): GO TO FM
2060 GO TO FM
2061 CLS : GO SUB VAL "2586": GO SUB VAL "2572": CLS
2063 CLS : PRINT AT OH,OI;"Please Select"''TAB OD;"<1> TS 20
40 Printer?"''TAB OD;"<2> Centronics Printer": GO SUB SQ: GO
 SUB IL: IF Z<OA OR Z>OB THEN GO TO VAL "2063": LET PR=Z-OA:
 IF PR=OO THEN LET /P=T: LET lx=VAL "32": LET qq=oq: LET fd=
oo: GO TO FM
2065 CLS : PRINT AT OI,OF;"Does printer need a"'TAB OF;"line
 feed?:"''TAB OF;"<y> yes or <n> no": GO SUB SQ: GO SUB IK
: LET FD=Z
2066 CLS : PRINT AT OI,OD;"Turn Printer ON, and:"
2067 INPUT "Input MAX Line Length?";LX: IF FD=OA THEN LET /P
=O: PAUSE OL: POKE VAL "23323",LX: IF fd=oa THEN POKE VAL "2
3324",oj
2068 IF fd=ob THEN POKE VAL "23324",oo
2070 CLS : INK 6: PRINT AT 3,6;"FUNCTION MENU"''TAB 6;"<1> M
ail List Menu"'TAB 6; INK 5;"<2> Caption"'TAB 6; INK 6;"<3>
Format"''TAB 6;"<4> Page Management"'TAB 6; INK 5;"<5> Line
Space"'TAB 6; INK 6;"<6> Page End"''TAB 6;"<7> Letterhead"'T
AB 6; INK 5;"<8> Sign Off"''TAB 6; INK 6;"<"; FLASH 1;"9"; F
LASH 0;"> Set up Outline"'TAB 6; INK 5;"<0> Print Outline"''
TAB 6;"Page=";pg;TAB 16;"Line=";nn
2071 INK oe: LET gg=oa: PRINT AT ot,ok;pg;" ";AT ot,VAL "21"
;nn;" ": GO SUB sq: GO SUB il: IF z<oo OR z>oi THEN GO TO fm
2072 IF Z=OF THEN GO SUB PE: GO TO FM
2073 IF z=oe THEN GO SUB sp: GO TO fm
2074 GO TO (z=oo)*VAL "3324"+(z=oa)*VAL "2000"+(z=ob)*VAL "2
094"+(z=oc)*VAL "2083"+(z=od)*VAL "3004"+(z=og)*VAL "3370"+(
z=oh)*VAL "3384"+(z=oi)*VAL "3300"
2083 CLS : PRINT AT 5,11;"FORMAT MENU"''TAB 6;"<1> Single Sp
ace"''TAB 6;"<2> Double Space"''TAB 6;"<3> Line Length Chang
e"''TAB 6;"<4> Page Center Change"''TAB 6;"<"; FLASH 1;"5";
FLASH 0;"> To Function Menu"
2084 GO SUB sq: GO SUB il: IF z=od THEN CLS : INPUT "Input P
age Center";qq: GO TO VAL "2083"
2086 IF z=oc THEN CLS : PRINT AT OJ,OA;"key in lgth: ": INPU
T LL: LET /p=o: PAUSE OL: POKE VAL "23324",(OJ AND fd=oa)+(o
o AND fd=ob): PAUSE OL: POKE VAL "23323",LX
2087 IF z=oe THEN GO TO fm
2088 IF z<oc AND z>oo THEN LET sd=z
2092 GO TO VAL "2083"
2094 CLS : PRINT AT OJ,OD;"Type the Cap"''TAB od;"Line lgth
limit is: ";LX: INPUT m$
2095 LPRINT TAB QQ-LEN m$/OB;m$: LET NN=NN+OA: GO SUB bu: OU
T po,24
2096 GO TO FM
2110 LET IN=OA
2112 CLS : PRINT "Enter Field Nbr to Correct,     or ""8"" f
or next field."''"Ml No. ";O''"Fields"'"1. ";O$(O,OA)'"2. ";
O$(O,OB)'"3. ";O$(O,OC)'"4. ";O$(O,OD)'"5. ";O$(O,OE)'"6. ";
O$(O,OF)'"7. ";o$(o,og): INPUT "Field Nbr or 8:";IN: IF IN=o
h THEN RETURN
```

CONT—

Combining the above techniques reduced the memory length of SDOS Mail Merge from around 21K to 9K total program length. These facts are given both to help you understand the cryptic programming and to whet appetites for a later article about Boolean Programming. When studying the programming, "think pseudo hex" to understand the "o vars". Look at lines 2586 through 2590 for the variable tables. It will help to make a index card for reference to the variables. Later we will DELETE these lines and the vars will no longer be visable. While in that area, EDIT line 2590 and change the LETS for "N$()" to you own letterhead and sign off data.

The LLIST Instructions

Key in the LLIST as given. This is a long and laborious task and nothing can be learned in the process that cannot be learned by studying the listing. It is likely that quite a number of key in errors will result which will have to be de-bugged. Why not save yourself the labor and order the "SDOS Issue Diskette" which is offered in the yellow pages of this issue. The diskette also contains the other utilities and program presented in this issue. At $16.95 it is quite a bargain. But whether you do or not, please follow through the logic of the following.

The program operates by <GO TO 2061>. The LLIST contains 5 programmed menus which will be SAVED as SCREEN$ files to diskette. Then the line numbers for the menus will be deleted and then each menu line will be re-created with a single <LOAD "name" SCREEN$> command. The Variable files of Lines 2586 through 2590 will be initialized, then the Vars will be SAVED to disk and those line numbers deleted. If you have completed the Key In project, Type <GO TO 2061> and follow the prompts until the FUNCTION MENU appears. If you cannot get that far, then you must re-check for key in errors in lines 2061, 2572, 2575, 2578,2579, 2578, 2579, 2586, 2588, 2590 and 2596. Continue by the following steps

1. Operate the program and exercise all menu functions. Create a mailing list, checking the menu selections of the Mailing list menu. Print the Mailing list in it's various offered modes. Test the "Print Labels" mode. Use a spare diskette to test the SAVE and LOAD functions.

2. Check all selections of the Function Menu. Create a "Outline Data Base" of at least 2 Major Topics and 2 Sub topics of each. Then PRINT the Outline. Print a Caption, a Letterhead, a Sign Off, and test the page management functions.

3. If any of the above fail, a report code will appear at the bottom of screen. You must then check the line number reported, edit the line and correct the key in error.

4. We will now assume that SDOS MAIL MERGE is operating correctly in all modes. Now use a spare diskette and <SAVE /"mmrg"LINE 2097> ENTER. This is a temporary save, using a different title and is for SAFETY in case you lose the program for some reason.

5. <CLEAR> the computer and type <GO TO 2061> ENTER. Respond to prompts until the Function Menu appears. <BREAK> and type <PRINT FREE> ENTER. Make a note of the figure

Program Listing (cont.)

```
2120 CLS : PRINT AT oj,oo;o$(o,in);AT ou,oo;"npt ";("Co Name
 " AND in=0A)+("Dept    " AND in=0B)+("St adr  " AND in=0C)+
("ct,St,Zp" AND in=0D)+("1st Name" AND in=0E)+("Tel Nbr " AN
D in=0F): INPUT T$
2122 LET o$(o,in)=t$: CLS : GO TO VAL "2112"
2550 PAUSE VAL "4e4": LET z=CODE INKEY$-48: IF z<oo OR z>oi
THEN GO TO il: REM * For Key 0 to 9
2553 RETURN
2554 IF PEEK VAL "23611"<VAL "221" THEN GO TO IK: REM * For
Yes or No.
2555 LET Z=PEEK VAL "23560": POKE VAL "23611",VAL "220": IF
Z<>VAL "78" AND Z<>VAL "89" AND Z<>VAL "110" AND Z<>VAL "121
" THEN GO TO IK
2558 LET Z=(0A AND z=VAL "89")+(0A AND Z=VAL "121")+(0B AND
Z=VAL "78")+(0B AND Z=VAL "110")
2560 RETURN
2567 IF IN 127=253 THEN GO TO BU
2568 RETURN
2570 CLS : PRINT AT oj,od;"INPUT FILE NUMBER": INPUT t$: LET
 t$="MMRG"+t$: SAVE /t$ LINE 2061: BEEP oa,oa: CLS : GO TO F
M
2572 CLS : PRINT AT 4,4; INK 5;"*** "; INK 6;"SDOS MAIL MERG
E"; INK 5;" ***"''TAB 3;"MAIL LIST"; INK 3;" LABELS "; INK 4
;"ENVELOPES"''TAB 6; INK 6;"INVENTORY  "; INK 2;"Outlines"''
TAB 13; INK 4;"WITH"''TAB 3; INK 4;"SAFE"; INK 6;" DISK"; IN
K 5;" OPERATING "; INK 2;"SYSTEM"''TAB 11; INK 3; 1984 by"
'TAB 3;"TS-2068 SAFE DISK UP-DATE"'TAB 6;"1317 Stratford Ave
"'TAB 5;"Panama City, FL 32404"'TAB 7;"rights reserved"
2575 FOR N=0A TO of: GO SUB SQ: PAUSE VAL "100": NEXT N: RET
URN
2576 INK xa: PLOT oo,oo: DRAW oo,VAL "175": DRAW VAL "255",o
o: DRAW oo,VAL "-175": DRAW VAL "-255",oo: LET xa=xa+oa: IF
xa>of THEN LET xa=oa
2578 INK xa: BORDER xa+oa: PLOT oh,oh: DRAW oo,VAL "159": DR
AW VAL "239",oo: DRAW oo,VAL "-159": DRAW VAL "-239",oo: LET
 xa=xa+oa: BEEP oa/oj,VAL "40": IF xa>of THEN LET xa=ob
2579 INK of: RETURN
2580 BEEP .1,40: PAUSE VAL "20": RETURN
2582 BEEP 0B/0J,0D: BEEP 0B/0J,0H: BEEP 0B/0J,0J: BEEP 0B/0J
,0D: RETURN
2586 LET po=127: LET oo=0: LET oa=1: LET ob=2: LET oc=3: LET
 od=4: LET oe=5: LET of=6: LET og=7: LET oh=8: LET oi=9: LET
 oj=10: LET ok=11: LET ol=12: LET om=13: LET on=14: LET op=1
5: LET oq=16: LET or=17: LET os=18: LET ot=19: LET ou=20
2588 LET tw=oo: LET xa=2: LET sq=2578: LET ln=54: LET nn=0:
LET pg=1: LET pr=1: LET bu=2567: LET zc=5: LET sq=2576: LET
fm=2070: LET sd=1: LET sp=2990: LET lh=2610: LET pe=3014: LE
T ik=2554: LET il=2550: LET ll=64: LET lx=80: LET qq=40: LET
 tb=8: LET fd=2
2590 DIM n$(og,31): LET n$(oa)="TS-2068 Safe Disk Up-Date":
LET n$(ob)="1317 Stratford Ave.": LET n$(oc)="Panama City, F
L 32404": LET n$(oe)="Bill Jones": LET n$(of)="904 871 4513"
: LET n$(og)="Sincerely,"
2596 RETURN
2600 CLS : PRINT AT oj,oj;"SAVE Menu"''TAB 4;"<1> Save this
Mail File"''TAB od;"<2> Save Program + File": GO SUB sq: GO
SUB il: GO TO (z<oa OR z>ob)*VAL "2600"+(z=ob)*VAL "2570"+(z
=oa)*VAL "2602"
```

2 cont- 2

presented. Next, Install a freshly formatted diskette and type: <SAVE /"mrg" VAL> ENTER. This saves the variable file to diskette. We will call this your "program diskette" and it will be used for all SAVES from now on. Next, DELETE lines 2586, 2588, and 2590.

6. The Function Menu should still be still on screen. If not, type: <GO TO fm> ENTER. BREAK and type: <SAVE /"fm" SCREEN$> ENTER. The disk should run and the Function Menu will be saved to diskette. To make sure, type <LOAD /"fm" SCREEN$. The Function menu should boot in from disk. If not, repeat the procedure of item 6. Next, ENTER a NEW LINE 2070: <2070 LOAD /"fm" SCREEN$>. Enter the line, which will replace the existing line 2070. Type: <GO TO fm> ENTER. The Function menu should boot in from diskette.

7. Type <LIST 2083 ENTER>. BREAK and type <GO TO 2083 ENTER>. The Format Menu should appear. BREAK and type: <SAVE /"fo" SCREEN$ ENTER>. Then type <CLS> and <LOAD /"fo" SCREEN$>. The Format menu should boot in from diskette. If O.K. then type a new line 2083: <2083 LOAD /"fo" SCREEN$> Enter the line. Then type <GO TO 2083 ENTER>. Again, the format menu should boot in from disk.

8. BREAK and LIST line 2572. Then add Line 2574: <2574 STOP>. Type <GO TO 2572 ENTER>. The title screen should appear. Type <SAVE /"cpr" SCREEN$ ENTER>. Type <CLS> ENTER. Type <LOAD /"cpr" SCREEN$. The title screen should boot in. Type an new line 2572: <2572 LOAD /"cpr" SCREEN$. Type <CLS> <GO TO 2572 ENTER>. The title screen should boot in from disk. Now DELETE line 2574 which has the STOP.

9. Type <LIST 2000 ENTER>. BREAK and type <GO TO 2000 ENTER>. The Mail List menu should be on screen. BREAK and type <SAVE /"ml" SCREEN$ ENTER>. Type <CLS> <LOAD /"ml" SCREEN$ ENTER>. The Mail list menu should boot in. Type a new line 2000: <2000 LOAD /"ml" SCREEN$>.

10. Type: <LIST 3004 ENTER>. BREAK and type: <GO TO 3004 ENTER>. The Page Management Menu should appear. BREAK and type: <SAVE /"pg" SCREEN$>. Type <CLS> <LOAD /"pg" SCREEN$ ENTER>. The menu should boot in from disk. Type a new line 3004: <3004 LOAD /"pg" SCREEN$ ENTER>. Type <CLS> <GO TO 3004 ENTER>. The screen should boot in.

11. BREAK and type: <SAVE /"SDOS MMRG"LINE 2061 ENTER>. This SAVES the main program to disk.

12. Type: <CAT ENTER>. The disk directory should show the programs: <mrg VBLS> <fm BYTES> <fo BYTES> <cpr BYTES> <pg BYTES> <SDOS MMRG>. Now we will install the file zero auto load line. NEW the computer. Then type this one line program: <10 BORDER 0: PAPER 0: INK 6: CLS: LOAD /"SDOS MMRG"> ENTER. Next type a direct command: SAVE /0 ENTER. Next, type: <LOAD ENTER>. The main program should AUTO LOAD and operate.

13. BREAK and type: <PRINT FREE ENTER>. Compare this FREE report to the earlier FREE report and you will see the savings of memory resulting from the use of disk for major parts of the program. You may go ahead and operate the software now and benefit from the extra FREE memory to use for data storage.

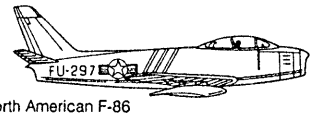
The concepts of "Disk Dependant" software should now be clear. The disk drive, employed with ingenuity, can

Program Listing (cont.)

```
2602 CLS : INPUT "Input Mail File Nbr:";t$: LET t$="m"+t$: S
AVE /t$ DATA o$(): GO TO fm
2604 IF Z=0G THEN CLS : PRINT AT 0J,0H;"INPUT FILE NBR": INP
UT t$: LET t$="m-"+t$: SAVE /t$ DATA o$(): GO TO VAL "2000"
2990 LPRINT : LET NN=NN+0A: IF NN>=LN THEN GO SUB PE
2994 IF sd=ob THEN LPRINT : LET NN=NN+0A: IF NN>=LN THEN GO
SUB PE
2996 RETURN
3004 CLS : PRINT AT 3,5; INK 6;"PAGE MANAGEMENT"''TAB 5; INK
 5;"Next Page is ";pg''TAB 5;"Lines to Page End=";ln-nn''TAB
 11;"Select"''TAB 5;"<1> Page End"''TAB 5;"<2> Adjust Line N
umber"''TAB 5;"<3> Re-number Page"''TAB 5;"<4> To Function M
enu"
3005 PRINT AT oe,os;pg;" ";AT og,VAL "23";ln-nn;" ": GO SUB
sq: GO SUB il: IF z<oa OR z>od THEN GO TO VAL "3004"
3006 IF Z=0D THEN GO TO FM
3008 IF Z=0B THEN INPUT "Ent new lines to end. ";NL: LET NN=
LN-NL: GO TO VAL "3004"
3010 IF z=oa THEN GO SUB pe: GO TO fm
3012 IF Z=0C THEN INPUT "Enter next pg No. ";PG: GO TO VAL "
3004"
3013 GO TO VAL "3004"
3014 IF PR=00 THEN LPRINT '''';pg'''''''''''''''''''': LET NN=
0A: LET PG=PG+0A: RETURN
3016 IF NN<VAL "59" THEN LPRINT : LET NN=NN+0A: IF NN<VAL "5
9" THEN GO TO VAL "3016"
3018 LPRINT TAB QQ;PG: OUT P0,0L: LET PG=PG+0A: LET NN=0A: R
ETURN
3020 CLS : PRINT AT oj,ob;"Install Data Diskette,"'TAB ob;"T
ouch ENTER": PAUSE VAL "4e4": CAT : INPUT "Enter TITLE ONLY
of C ARRY      ";t$: PRINT ''t$: LOAD /t$ DATA o$(): GO TO V
AL "2000"
3300 CLS : PRINT AT oj,ob;"Outline Fill": INPUT "Enter Nbr o
f Major Topics:";mj: INPUT "Enter Nbr of Sub Topics of each:
";sb: INPUT "Enter Column width:";lgt: DIM k$(mj,lgt): DIM v
$(mj,sb,lgt)
3304 LET a=oa: FOR n=oa TO mj
3305 IF n>mj THEN GO TO VAL "3320"
3306 PRINT AT on,oo;"Ent Major Line #"; INK oe;n; INK of;" o
r zz to quit": INPUT k$(n): IF k$(n)( TO ob)="zz" THEN LET k
$(n)="": GO TO VAL "3320"
3308 FOR y=oa TO sb: PRINT AT on,od;"Minor";AT on,oq; INK od
;y; INK of: IF y>sb THEN GO TO VAL "3316"
3310 INPUT v$(n,y): IF v$(n,y)( TO ob)="zz" THEN LET v$(n,y)
="": GO TO 3316
3312 NEXT y
3316 NEXT n
3320 LET n=mj: LET y=sb: CLS : PRINT AT oj,ob;"Print the Out
line now?"''TAB ob;"<y> Yes or <n> No": GO SUB ik: IF z=ob T
HEN LET n=mj: GO TO fm
3324 CLS : INPUT "Input Tab for line print:";tz
3330 FOR n=oa TO mj: IF k$(n)( TO ob)="  " THEN GO TO VAL "3
354"
3334 LPRINT TAB tz;k$(n)'': LET nn=nn+ob: IF nn>=ln THEN GO
SUB pe
3338 FOR y=oa TO sb: IF v$(n,y)( TO ob)="  " THEN GO TO VAL
"3346"
```

2 cont-2

mitigate the basic deficiency of the TS-2068, namely "insufficient FREE memory" to use for programming. As the Oliger EPROM developments continue, Safe Disk Up-Date will explore each new capability in detail. In the meantime, follow on issues will discuss subscriber questions and comments about this softare "SDOS Mail Merge". Feel free to write. A quick letter response will be sent and the questions comments will appear in the next issue.



North American F-86

Program Listing (cont.)

```
3342 LPRINT TAB tz+oe;v$(n,y)'': LET nn=nn+ob: IF nn>=ln THE
N GO SUB pe
3346 NEXT y
3350 NEXT n
3358 CLS : PRINT AT oj,obj"Delete the Data?"''TAB obj"<y> Ye
s or <n> No": GO SUB ik: IF z=oa THEN DIM k$(oa,oa): DIM v$(
oa,oa)
3360 GO TO fm
3370 FOR n=1 TO oc: LET m$=n$(n): GO SUB VAL "3380": NEXT n:
 LET m$=n$(of): GO SUB VAL "3380": GO TO fm
3380 IF m$(LEN m$)=" " THEN LET m$=m$( TO LEN m$-oa): GO TO
VAL "3380"
3382 LPRINT TAB qq-LEN m$/objm$: LET nn=nn+oa: RETURN
3384 LPRINT ''': LET m$=n$(og): GO SUB VAL "3380": LPRINT ''
': LET m$=n$(oe): GO SUB VAL "3380": LET nn=nn+oh: GO SUB pe
: GO TO fm
```

End

Printer Control Programming

<LET P=o> takes care of "sending LPRINT character streams" to a Centronics printer. The other half is sending "Printer Control Codes" to command the printer to do such things as change type fonts, set italic, and set underline modes. Most printers have a group of dip switches to manually set these functions, but ususally the dip switch panel is located in a difficult to reach area. What is needed is "software control". Lets explore how to build a software "printer control group". First, a MENU is needed. The menu would look something like the following:

1. Pica Style 2. Elite Style. 3. Condensed Style. 4. Italic Font. 5. Underline.

When the menu is presented, a touch of key 1-5 should set a variable to the value of the key touched. See the INKEY$ prompt routine in this issue. We will let the variable "ps" be our switch. After the menu and the INKEY$ prompt, this program line will do it: <900 LET ps=z>. For example; if Key 1 is touched, Pica style is selected and "ps=1". Before continuing, you should make sure that the printer actually responds to a LPRINT command. Type <LPRINT "12345" ENTER". If 12345 is printed to paper you may proceed. If not, then you must set up the printer correctly before proceeding.

Now, assuming that a menu has been constructed, we will build a printer control group. LINE 900: <900 LET ps=z: GO TO 1000>

Line 1000 should contain the programming needed to CANCELL all special printer set-ups to make ready for commands to establish the print styles selected at menu. Some printers have a single command code to do this, BUT many printers will also RESET TOP OF FORM when the RESET command is used. That would disturb a page format after a print style change has been directed. Line 1000 should contain groups of CANCEL commands for every conceivable printer set-up, WITHOUT resetting TOP of FORM. Then you'll be ready to set the print style selected at menu.

Before going to the next lines, we will discuss how control codes are sent to a printer. Lets use a typical control code group as would be given in a printer manual. (27,100) (80,102). To send such a group, this programming would be needed: <line GO SUB BU: OUT 127,27: GO SUB BU: OUT 127,100: GO SUB BU: OUT 127,80: GO SUB BU: OUT 127,102 >. Explanation: <OUT 127> sends the control code to the printer, BUT- the printer must be READY to receive a control code. <GO SUB BU> is a line to assure PRINTER READY>. We will now construct "line BU as line 2000. <2000 IF IN 127<>253 THEN GO TO BU> <2002 RETURN>.

To explain the above: When a printer is READY to receive a control code, <IN 127> produces character code 237. If NOT READY, <IN 127> produces the character code 253. So, line BU (2000) will loop itself until the printer is READY. Now, using the examples given, line 1000 can be constructed: <1000 GO SUB bu: OUT 127,27: GO SUB bu: OUT 127------ continue all control codes needed to CANCEL ALL printer set-ups--->. Refer to your printer manual for the correct control codes.

Next, <IF THEN programming is needed to set the print styles selected at menu. Typically, these would be:

<1002 IF ps=1 THEN GO SUB BU: OUT 127,------the printer control codes needed to set pica style. Use GO SUB BU between each.>

<1004 IF ps=2 THEN GO SUB BU: OUT 127,-----the command codes needed to set ELITE style. Use GO SUB BU between each OUT 127 command.>.

Continue building program lines as above for commanding the printer to set the other menu selections. When finished, the last line should be <line GO TO the line of your major menu>. Try out the "print style menu" and the program lines that have been constructed. Chances are that the print style changes selected at menu WILL be made at the printer, BUT, extraneous line feeds will occur. If so, check the printer manual to find the "Cancel Buffer" control code. Usually this is character code 24, but some printers use an extra command code. To eliminate extraneous line feeds, the Cancel Buffer code is inserted after each group of control codes for a special function. Dont forget to use <GO SUB BU> between each OUT 127 command.

TS·2068 Computing

▷ with S.D.O.S.

By John Oliger



** Language Discussions

** Questions and Answers

** Theory of Operation

** Dock and Extra Memory

** What's Next?

SAFE QUESTIONS AND ANSWERS

This column will answer questions of general interest about the Oliger 2068 Disk I/F and JLO SAFE DOS. If you have questions you feel other users of this system would like to see answered, send them to me at: 11601 Whidbey Dr. Cumberland, IN 46229. Because my schedule can at times be simply too demanding to always find the time, I cannot promise to ALWAYS have this column ready for each new issue of Safe Disk Up-date. I will only do the best I can...

-John T. Oliger

What is the latest version of SAFE and what new features does it provide?

The current version of SAFE as of today (8/16/87) is V2.34. The last change to SAFE (V2.33 to V2.34) corrected a minor bug in the FOR /n TO m command. The change from V2.32 to V2.33 also corrected a minor bug. The features it added to SAFE (from V2.2) were: 1)A fast new FOR/NEXT loop structure. 2)The ability to save a SCREEN$ via the NMI pushbutton along with all the previous functions by use of this pushbutton.

What does the different numbers represent when referring to SAFE's version?

The first number (the "2" in V2.34) says that this is the second major version of SAFE. SAFE V1 only supported a total STATE SAVE like V2's SAVE /"filename" ABS. The second number (the "3" in V2.34) is updated every time a new fuction, command, or group of functions and commands are added to SAFE. When the RESTORE /"filename" TO "newfname" command was added, SAFE's version number went from V2.01 to V2.10. The last number in SAFE's version is incremented any time the code in SAFE has been changed or modified, but no new commands or functions were added. This might be done to fix a bug or to make a portion of the code more efficient. An example of this would be when the FOR /n TO m routine was changed to fix a bug, which changed the version of SAFE from V2.32 to V2.33, and another to make it V2.34.

SAFE gives me occasional toots when LOADing, but the programs LOAD O.K. This seems to happen most often with SCREEN$ files. What does this mean?

Anytime SAFE sounds a short toot, it is reporting to you that an error has occurred, it knows it, and it will try again. SAFE will try 5 times to read a particular cylinder on the disk, and give a short toot after each try if an error occurs. If after 5 trys it still cannot read a particular cylinder, SAFE will start the load or save over from the beginning. It will do this up to 3 times (with 5 single cylinder tries) before it will finally just give up and stop with an error report "T" (Disk I/O Error). If you hear some toots while SAFE is trying to SAVE or LOAD, it is simply a report to you that SAFE IS having trouble. If it finishes without an error report, then it finally was sucessfull...but you have been told it DID have trouble so can make another back-up of important files, etc. Most DOSes will retry when an error occurs, just like SAFE does, but will not let you know it is having any trouble...until it cannot load the file at all!

A last minute Addendum

This page came in late from Mr. Oliger, so I'll add it plus a report on a extensive check out of the "SDOS MAIL MERGE" program. I'll quote John's note to me and then give his article. (Quote) I wrote this little article tonight as info for the disk system. I plan on putting it in the manual some day, but for now everyone who has the system does not have this information. (second note) Another unexpected bug has cropped up in some versions of SAFE V2.33 and all copies of V2-34. A BASIC program load in the 2068 mode will stop with an "OUT OF MEMORY" report when plenty of FREE memory remains. Advise everyone having this problem (only V2.33 and V2.34) to return their SAFE EPROM to me for FREE replacement with V2.35 which is now the current version. SAFE V2-32 is O.K. and the free replacement applies only to V2.33 and V2.34. (end quote) Ed note: The V2-30 and UP series of EPROMS have Snapshot NMI screen SAVE, a FAST COUNTER addition to BASIC, and Disk File Name Change. The next version will be V2-40 and will include the MERGE function in the disk language.

CORRUPTED DISKS???

Occasionally you might come across a disk that has become corrupted in its file storage pointer area (track zero) that will simply refuse to load any file at all. What will usually cause this is a glitch on power up of either the drive or computer, with the disk installed. Occasionally a bad WD1770 controller chip has also been found to cause this problem. ALWAYS remove your disk from your drive when powering up either your computer or the drive, unless the disk in the drive is something that you do not care if it is destroyed (or is one that you have a good back up for). It is best to simply get into the habit of doing this all of the time, regardless of the disk that you have installed. BUT EVERYONE can forget sometimes so I will present a small program that MIGHT bring your disk back to life, or may kill it completely. If it's gone already, you have nothing to lose by trying it and you might be able to recover the corrupted disk.

The first thing to try if you have a corrupted disk is to re-save file zero. File zero uses sectors 1-3 of track zero, and when SAFE loads ANY file it ALWAYS loads all ten sectors of track zero. Because the glitch could have only corrupted one of the first three sectors, we have a 30% chance that simply re-saving file zero will fix the problem. Enter a single line program such as: 10 PRINT and save it to file zero with the command <SAVE / 0 ENTER>. Now try a CAT or try to load a regular file. Perhaps we have fixed the disk already! If not....

If not, then after the computer stops with its error T report, enter the command <PRINT IN 175>. The sector number that is causing the problem will be printed on the screen. Because the CATalog area starts with sector 4 and ends with sector 10, and re-saving file zero into sectors 1-3 did not fix the problem, you will find that the actual bad sector will usually be in the range of 4-10. If the sector giving the problem is NOT sector 4, then it is VERY likely that we can restore the catalog area by using a little utility program. If it IS sector 4, you can still try it and it may work. The utility program is given below. Key it in and do a CAT (even though it stops with a error), then RUN the program.

```
10 RESTORE 30: FOR N=32000 TO 32006: READ A: POKE N,A: NEXT N
20 RAND USR 32000
30 DATA 205,10,0,205,142,10,223
```

This little program re-saves the CATalog back onto the disk. If the error was actually in a later portion of the catalog area that was not being used (the CAT area can hold 177 file, and will not be full unless you have thai many files on disk!), then the data re-saved will be correct and the disk will be completely recovered. You will find that over 90% of the time this program will restore access to your disk. To those interested in what this program does at the MC level, I provide the following annoted assembly listing. It is a very simple little routine.

```
CALL 000A  ;turn on Oliger Disk B bank
CALL 0A8E  ;save the contents of the B bank buffer to track 0.  Data will be
           ;put there by using CAT before runing this program
RST 18H    ;turn off B bank, turn on interrupts & return to basic
```

John L. Oliger 8/87

More About SDOS MAIL MERGE

A thorough Check Out

Next issue two more functions will be added to SDOS MAIL MERGE. One will quickly let you flip through the listing to find a name or stock list. Presented will be the "Field Number and first line of the field". These lines will scroll until you find the one you are looking for, then the menu can be used to do something with that particular field of information. The second function will be to INPUT a KNOWN LAST NAME or "NOUN" of a stock list. Then the field having that noun will be found and the seven line field will be presented on screen. If more than one field contains such a noun, you may continue until each field is presented.

Also, next issue we will use "TIMECHINE" to assemble the three search routines to make them ligntening FAST. By the way, after printing the LLIST of Mail Merge, I began to use the software extensively to produce mailing lists for mail out of invitations to subscribe to SDU. About 600 TS-2068 users were entered into six individual lists of 96 each. "96" was selected because that is a multiple of 24, and my Smart Text software uses mailing lists of 24 fields. Thats another story that we will take up in the next issue- how to split a large character array into several shorter arrays that can be used in other printing softwares.

BLEEPS and GLITCHES: Yep, I found a couple! Hey, there aint nobody thats are perfect! The following line numbers should REPLACE those given in the LLIST.

```
2005>IF z=oa THEN DIM o$(mx,7,31): DIM O$(mx,7,31): LET IN=OA: LET ms=OA
: CLS : GO TO VAL "2009"

2061>PAPER 0:: CLS : GO SUB VAL "2586": GO SUB VAL "2572": CLS

2068 IF fd=ob THEN LET /p=o: POKE VAL "23324",oo: POKE 23323,lx
```

I also found some subscript problems when SORTING BY LAST NAME, "when there are not TWO nouns in the name". A few line changes as follows cleaned up this little problem. Enter the following program lines. Some replace existing lines.

```
685>IF LEN b$=oo THEN GO TO VAL "689"
 686 IF LEN b$>oo AND b$(LEN b$)=" " THEN LET b$=b$( TO LEN
b$-oa): GO TO VAL "685"
 687 IF LEN b$>oa AND b$(oa)=" " THEN LET b$=b$(ob TO ): GO
TO VAL "685"
 689 LET o$(n,oa)=b$+" "+o$(n,oa)( TO -1)
 690 NEXT n
 692 RETURN
```

The "SDU ISSUE DISKETTE" will contain all of the above line number corrections, plus the two new added functions. See you next issue.

Update

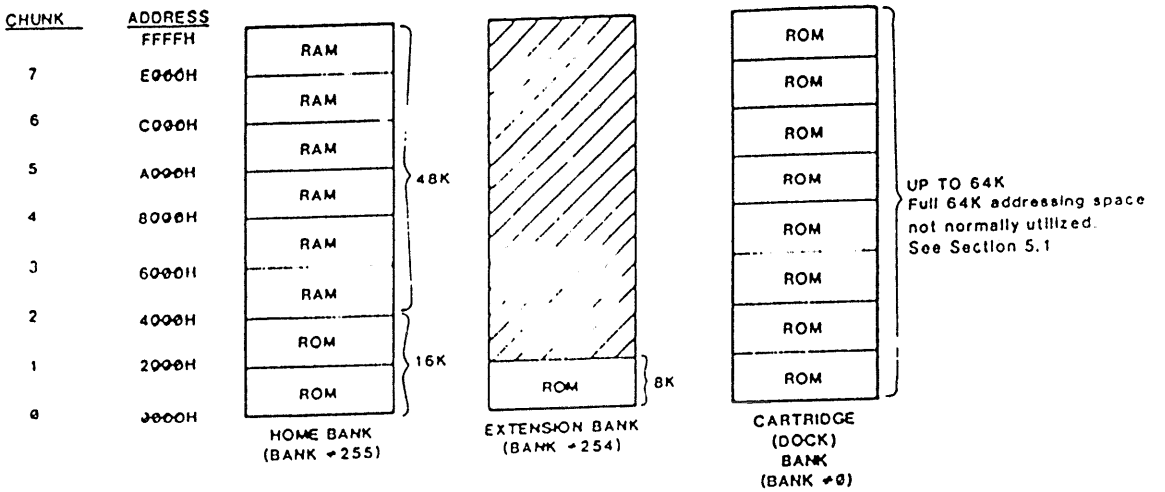
TECHNICAL SECTION

Oliger Schematics

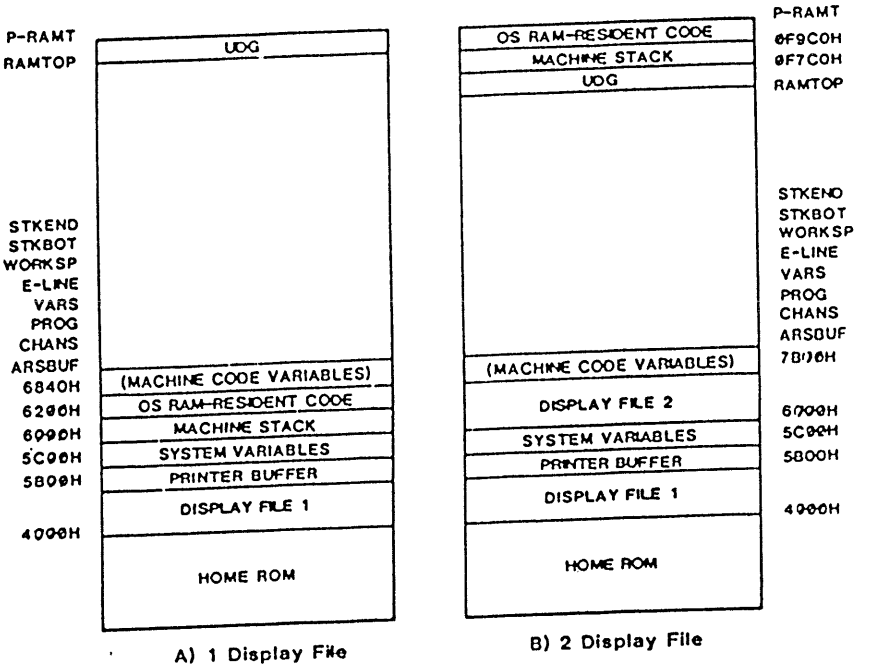
Timex Tech Data

Special Projects

TS 2068 STANDARD MEMORY CONFIGURATION



HOME BANK RAM



Catalog Section

PERMANENT SUPPLYERS : PLUS USER FREE AD'S

wanted

Software

Trade ?

Hardware

For Sale!

Livestock

Barter

fishbait

Cont-



The New England Moose

and

His White Face Harem

Everybody needs somebody.

ALLEGRO: A experimental musical composition by a tin eared Welshman. There are three <GO SUB routines>. <GO SUB 100> produces a indefinite musical chord which is interrupted by touching any key. <GO SUB 102> gets a 10 second musical pause. <GO SUB 122> brings prompts to adjust the chord dwell and time duration of a musical interlude. Line numbers 116 and 118 are data lines that contain the absolute frequencies of the musical scale. These notes are played in groups of three. Arrangement of a melody involves the sequencing of scales in the two data statements, using audio frequencies of the notes. Insertion of BEEPS in line 104 can produce other effects.

ALLEGRO

```
100 RESTORE VAL "116": LET xx=VAL "4e4": GO TO VAL "104"
102 RESTORE VAL "116": LET xx=VAL "600": GO TO VAL "104"
 104 GO SUB VAL "108": SOUND VAL "7",VAL "56": SOUND VAL "0"
,a1;VAL "1",a2;VAL "8",VAL "12": PAUSE VAL "20": GO SUB VAL
"108": SOUND VAL "2",a1;VAL "3",a2;VAL "9",VAL "13": PAUSE V
AL "20": GO SUB VAL "108": SOUND VAL "4",a1;VAL "5",a2;VAL "
10",VAL "12": PAUSE xx: SOUND VAL "0",VAL "0";VAL "1",VAL "0
";VAL "2",VAL "0";VAL "3",VAL "0";VAL "4",VAL "0";VAL "5",VA
L "0": RETURN
 106 RETURN
 108 READ f: IF f=VAL "10" THEN RESTORE VAL "116": READ f
 110 LET x7=VAL "1.75"/(f*VAL "10"): LET x7=x7*VAL "1e6"
 112 LET a2=INT (x7/VAL "256"): LET a1=INT x7-INT (x7/VAL "2
56")*VAL "256"
 114 RETURN
 116 DATA VAL "880",VAL "932.32",VAL "987.76",932.32,VAL "98
7.76",VAL "830.608",VAL "58.27",VAL "65.406",VAL "130.812",V
AL "440", VAL "880",VAL "1760",VAL "92.498",VAL "69.296",VAL
 "138.592",VAL "97.998",VAL "110",VAL "130.812",VAL "146.832
",VAL "82.406",VAL "87.308",VAL "97.998",VAL "146.832",VAL "
155.564",VAL "164.812",VAL "195.996",VAL "103.826",VAL "138.
592",VAL "1396.928",VAL "1567.968",VAL "1318.496",VAL "2092.
992",VAL "2217.472",VAL "2349.312",VAL "1174.656",VAL "1318.
496",VAL "1396.928",VAL "493.88",VAL "523.248",VAL "587.328"
,VAL "233.08",VAL "246.94",VAL "61.624",VAL "10"
 118 DATA VAL "3136.936",VAL "297.856",VAL "3520",VAL "698.4
54",VAL "783.984",VAL "659.248",VAL "55",VAL "87.308",VAL "7
3.416",VAL "174.816",VAL "194.995",VAL "207.652",VAL "349.23
2",VAL "391.992",VAL "329.642",VAL "10",VAL "10"
 120 LET xx=VAL "4e4": GO SUB VAL "104": GO TO VAL "120"
122 RESTORE VAL "118": INPUT "Note Length (seconds)? ";a: L
ET xx=a*30: INPUT "Interlude in Minutes? ";b: LET yx=xx*b: F
OR n=VAL "1" TO yx: GO SUB VAL "104": NEXT n
```

A PRINT USING ROUTINE and DEMONSTRATION

This interesting batch of string slicing and concantations does a nice job of aligning decimal places and simulating the print using functions. There are shorter ways of doing it but this one is included for a study of its tricks. TYPE RUN.

```
 10 CLS : DIM a$(10,10): FOR n=1 TO 10
 30 LET a$(n)=STR$ (1000000*RND*RND*RND*RND): IF VAL a$(n)<
.0000001 THEN GO TO 30
 50 IF RND<.5 THEN LET a$(n)=a$(n, TO 5)+"     "
 60 IF a$(n,5)="." THEN LET a$(n,6)=STR$ (1+(9*INT RND))
 70 PRINT n,a$(n): NEXT n: PRINT ,,"The numbers will now be
 placed  in the first column and rounded to two decimal plac
es.          TOUCH ENTER"
 100 INPUT x$: PRINT AT 0,0;"";: FOR n=1 TO 10: GO SUB 240:
NEXT n: PRINT '"The numbers in left column will be placed in
 right column and  trailing zeros and commas added.TOUCH EN
TER": INPUT x$: PRINT AT 0,0;
 180 FOR n=1 TO 10: GO SUB 290: NEXT n: PRINT AT 21,3;"Touch
 ENTER to REPEAT"
 220 IF INKEY$="" THEN GO TO 220
 230 RUN
 240 LET a=VAL a$(n): LET a=INT (100*a+.05)/100: LET a$(n)=S
TR$ a: PRINT a$(n): RETURN
 290 FOR i=10 TO 1 STEP -1: IF a$(n,i)<>" " THEN GO TO 320
 310 NEXT i
 320 LET c$=a$(n,1 TO i): IF c$(LEN c$-2)="." THEN GO TO 380
 350 IF c$(LEN c$-1)="." THEN GO TO 370
 360 LET c$=c$+".0"
 370 LET c$=c$+"0"
 380 LET a$(n)="          "
 390 IF LEN c$>6 THEN LET c$=c$(1 TO LEN c$-6)+","+c$(LEN c$
-5 TO )
 400 LET a$(n,11-LEN c$ TO 10)=c$: PRINT AT n-1,16;a$(n)
 420 RETURN
```

HELP !

Safe Disk Up-Date needs both Utilities, Programming Tips, and Major Software Programs. The material in this issue came from the Editor's own treasure trove. The bottom of that basket will soon be uncovered. PLEASE HELP by sharing your useful programming. See the editorial about sharing the proceeds of "SDU ISSUE DISKETTE SALES". We all need particiption by all.

End 2

** SDU Quarterly Diskettes **

The programs and utilities that are presented in each quarterly issue of TS-2068 Safe Disk Up-date are assembled in one Diskette and offered at a price of $16.00 (post paid). Generally the text of the articles in SDU will be sufficient documentation. When needed, additional documentation will be furnished. Letter questions about the programs on SDU Diskettes will be published the following issue. Call for quick answers.

SDU Diskettes are available in 5 1/4 inch diskettes, any FORMAT. For other size drives, a cassette is provided. Please use the order form and check off your disk format or cassette.

SDU Diskette, October 87 issue contains the following:

1. SDOS MAIL MERGE. Create, Store, and Print Mailing Lists, Inventories, Telephone Lists. Auto Print listings. Auto Print Mailing Labels.
2. UTILITY SEXTET- Hex/Decimal, BINARY, MIXED MATH, ASSEMBLER, DISASSEMBLER.
3. PSEUDO HEX VARIABLE TABLE. Use with all BASIC Programs to conserve program memory and standardize use variables. (LOAD /"ph"VAR) in a program line merges the vars to any BASIC program.
4. PEEK K STATE PROMPTER. Three GO SUB prompt routines for use after menus.
5. VISUAL ASSORTMENT. Four GO SUB routines to provide color screens.
6. ALLEGRO. Three GO SUB routines to give a variety of musical effects.
7. PRINTER STATUS. Two alternate GO SUB routines to check the printer status.

SDB Diskete Order Form

Please send SDU Diskette (October 87 Issue).

Please Check

_____ 5 1/4 inch DS/DD 40 track Diskette.

_____ 5 1/4 inch DS/Hi Density or Quad Den, 80 Track.

_____ 5 1/4 inch Single Side, DD, 40 track.

_____ Cassette for loading and transfer to my disk system.

Name and Address

_____

_____

_____

_____

Advertisement

S.D.O.S. Software !!

** Smart Text SDOS **

Smart Text is designed for SDOS. It is a word processor, text editor, data base manager, and desk top publisher. Smart Text uses the disk drive as if it were a part of the computer memory. The Mail Merge annex allows one to build exceptionally large disk mail files and use the mailing lists for "personalized" form letters, printing telephone listings, inventories, labels and envelopes on tractor forms. I use it to publish this newsletter and to print the mailing labels. All subscribor data is created with Smart Text and saved to diskettes. Pages of the newsletter are filed away in like manner.

Before I got smart and bought a copier, I printed 200 copies of the 70 page Smart Text manual, with this software. 200 copies of three pages at a time on one uninterrupted print run is a pretty good job for any software. But thats not all! One can print 1000 letters, each to a different name and address, pick off paragraph 1 of one disk data file, paragraph 2 of another disk data file, and use a file in memory for paragraph 3, "for each of the 1000 letters". You may not need to do that. How about this: You have 30 data files stored in a diskette, each about 4 pages in length, and you want to "print the manuscript". Thats easy! Just elect a format and print style, punch up the Repeat Print Menu and elect "Print Disk Data". Then the whole 360K of data prints out UN-INTERRUPTED (about 120 pages).

Maybe all of the above is too convoluted. Want to "Type and Print", kinda like one does with a typewriter? Thats good. The difference is you have a chance to READ and EDIT first. Also, you can elect a different style of print between paragraphs, or insert a subject title (header). Or, you could elect to print an outline. The data bases are so flexible that short or long paragraphs may be tucked away in memory, or filed in diskette. Then your printing options are many. Print the first paragraph last, the third paragraph first, etc, etc.

Letterheads center print automatically. The business block of a letter formats and prints automatically. Letter sign off is automatic. Printing is centered on page. Pages are numbered. MUCH MUCH MORE! Smart Text is programmed for Epson and Seiksosha compatable printers. It is easy to adapt to other centroncis printers. The diskette contains a three hour tutor that explains how to match to any printer. Also, the TS-2040 can be used.

The policy is to provide continuing after sales support via telephone and correspondance. Previous buyers of Smart Text may receive the V2-32 version and manual up-date for $8.00.

TO ORDER "Smart Text SDOS", send $26.00 Check or MO to Bill Jones, 1317 Stratford Ave, Panama City, FL 32404. Diskettes are supplied in 5 1/4 inch, any format. Please specify either Single or Double Side, and whether 40 track or 80 track. "Satisfaction or FULL Refund", cant beat that with a stick! Also you should specify the "V number" of your Safe Disk EPROM. One version of Smart Text is for V-20 or lower and the newest is for the V2-32 EPROM.

Programmers!

Place your FREE AD Here!

-----Continued from inside front cover-----

f=Frequency; force
ff=Flip-flop
fm=Frequency modulation
4pdt=4 pole double throw
fw=Full wave
g=ground
gnd=ground
g=Gate
hfo=High Frequency Oscillator
hv=High voltage
if=Intermediate frequency
incr=Increment; Increase
inp=Input
I=Current (amperes)
IGFET=Insulated-gate field effect transistor
jct=Junction
JAN=Joint Army Navy
JFET=Junction Field effect transistor
l=inductance; inductor; luminance
lfo=Low frequency oscillator
lkg=Leakage
lp=low pass
lr=Load resistor (relay)
LSHI=Large scale hybred integration
m=Magneflux; mode
mag amp=Magnetic amplifier
mc=Momentary contact
mgn=Magneto; magnetron
MOS=Metal oxide semiconductor
nc=No connection; normally closed
nf=Noise figure; noise frequency
no=Normally open
oc=Overcurrent
ohm=Ohmmetter
osc=Oscillator
out=Output
pa=Pulse amplifier
pc=Printed circuit
pct=Percent
pec=Photo electric cell
permb=Permeability
permb=Permeability
pfm=Pulse-frequency modulation
ph=Phase
pk=Peak
plyph=Polyphase
pos=Positive
preamp=Preamplifier
prv=Peak reverse voltage
fdbk=feedback
fil=filament
4p=Four pole
4pst=Four pole single throw
FET=Field effect transistor
gdlk=Grid leak
G=Gain
hf=High Frequency
hp=High pass
hvr=High voltage regulator
impd=Impedance
inf=Infinite; Infinity
inv=Inverter
IC=Integrated circuit
jb=Junction box
jk=Jack
JANAF=Joint Army Navy Air Force
k=Dielectric constant
lf=Low Frequency
lim sw=Limit switch
lo=Local oscillator
lpo=Low power output
lsb=Lower side band
LSI=Large scale integration
mag=Magnet; magnetic
mag mod=Magnetic modulator
mg=Magnetic armature
mic=Microphone
MOSFET=Metal oxide semiconductor field effect transistor
neut=Neutral
nmag=Non magnetic
nol=Normal overload
oco=Open-close-open
opr=operate
osmv=One shot multivibrator
p=Pole; probe
pam=Pulse-amplitude modulation
pcm=Pulse-code modulation
pdm=Pulse-duration modulation
pelec=Photoelectric
pf=Power factor; Pulse frequency
pf=Power factor; Pulse frequency
pF=Picofarad
pim=Pulse-interval modulation
pls=Pulse
pnp=Positive-negative-positive
pot=Potentiometer
pri=Primary
pu=Pickup

Continued on next page.

Continued from preceeding page.

pwr=Power
qtz=Quartz
rcv=Receive
rect=Rectifier
reg=Regenerate
resn=Resonate
rf=Radio frequency
rfi=Radio frequency interference
rms=Root mean square
rot=Rotate
rtr=Rotor
R=Resistance (Ohms)
RL=Resistance-inductance
RTL=Resistor-transistor logic
schem=Schematic
sec=Secondary
semicond=Semiconductor
shf=Super high frequency
sig=Signal
slp=Slope
snr=Signal to noise ratio
sp=Single pole
spkr=Speaker
sqw=Square wave
ssbo=Single swing blocking oscillator
stalo=Stabilized master oscillator
sw=Switch; Shortwave
swp gen=Sweep generator
swr=Standing wave ratio (Voltage)
syn=Synchronous
sys=System
SCR=Semiconductor controlled rectifier
SWG=Stubs Wire Gage
tc=Thermocouple; Time constant
thms=Thermistor
thrm=Thermal
thyr=Thyristor
trf=Tuned radio frequency
tty=Teletypwriter
uhf=Ultrahigh frequency
undf=Underfrequency
UJT=Unijunction transistor
v=Vertical; Voltage
vam=Voltammeter
varistor=Variable resistor
vco=Voltage controlled oscillator
vdet=Voltage detector
vf=Variable frequency; voice frequency
vhf=Very high frequency
vidamp=Video amplifier
vlf=Very low frequency
vo=Voice
vom=Volt-ohm milliampmeter
vrly=Voltage relay
Pwr sply=Power supply
Q=Merit of a capacitor or coil; Quantity of electricty
rcvr=Receiver
ref=Reference
res=Resistor
rev cur=Reverse current
rfc=Radio frequency choke
rly=Relay
rmt=Remote
rpt=Repeat
rtty=Radio teletypwriter
RC=Resistance-capacitance
RCL=Resistance-capacitance-inductance
sb=Sideband
scr=Short circuit ratio
sel=Selector
sens=Sensitivity
shtc=Short time constant
sig gen=Signal Generator
siwl=Straight line wavelenght
sol=Solenoid
spdp=Single pole, double throw
spst=Single pole single throw
ssb=Single side band
st=Sawtooth; Schmitt trigger
svo=Servo
swp=Sweep
swp intg=Sweep integrator
sym=Symbol
sync=Synchronize; Synchronizer
S=Signal power; voltage standing wave
SH=Shield
t=Temperature; Time
tft=Thin film resistor
3ph=Three phase
thymo=Thratron motor
tpho=Telephotograph
tt=Teletype
uf=Ultrasonic frequency
undc=Undercurrent
usb=Upper sideband
USG=United States Gage
vac=Vacuum
var=Variable; Varistor
vc=Voice coil
vd=Voltage drop
vern=Vernier
vfo=Variable frequency oscillator
vid=Video; visual
vidf=Video frequency
vm=Voltmeter
vol=Volume
vr=Voltage regulator
vsm=Vestigial sideband modulation

Continued on next page.

Cont. Conversion Factors

To Convert	Into	Multiply by	Conversely, Multiply by
Steres	Cubic meters	1	1
Stokes	Sq meter per second	10^{-4}	10^{-4}
Tablespoons	Cubic cms	14.79	6.761×10^{-2}
Teaspoons	Cubic cms	4.929	0.203
Tonnes	Kilograms	10^{3}	10^{-3}
Tonnes	Pounds	2204.63	4.536×10^{-4}
Tons (long)	Pounds (avdp)	2240	4.464×10^{-4}
Tons (metric)	Kilograms	10^{3}	10^{-3}
Tons (short)	Pounds	2000	5×10^{-4}
Torrs	Newtons per sq meter	133.32	7.5×10^{-3}
Varas	Feet	2.7777	0.36
Watts	Btu per hour	3.413	0.293
Watts	Btu per minute	5.689×10^{-2}	17.58
Watts	Foot-lbs per minute	44.26	2.26×10^{-2}
Watts	Foot-lbs per second	0.7378	1.356
Watts	Horsepower	1.341×10^{-3}	746
Watts	Kilogram-calories per minute	1.433×10^{-2}	69.77
Watt-seconds	Gram-calories (mean)	0.2389	4.186
Watt-seconds	Joules	1	1
Webers	Maxwells	10^{8}	10^{-8}
Webers per sq meter	Gausses	10^{4}	10^{-4}
Yards	Feet	3	.3333
Yards	Varas	1.08	0.9259

METRIC SYSTEM

Linear Measure

10 millimeters = 1 centimeter
10 centimeters = 1 decimeter
10 decimeters = 1 meter
1000 meters = 1 kilometer

Area Measure

100 square millimeters = 1 square centimeter
100 square centimeters = 1 square decimeter
100 square decimeters = 1 square meter

Equivalent Numeric Representations in Decimal, Binary, Hexadecimal, and Octal Codes

Decimal	Binary	Hexadecimal	Octal
0	0000	0	0
1	0001	1	1
2	0010	2	2
3	0011	3	3
4	0100	4	4
5	0101	5	5
6	0110	6	6
7	0111	7	7
8	1000	8	10
9	1001	9	11
10	1010	A	12
11	1011	B	13
12	1100	C	14
13	1101	D	15
14	1110	E	16
15	1111	F	17

Volume Measure

1000 cubic millimeters = 1 cubic centimeter
1000 cubic centimeters = 1 cubic decimeter
1000 cubic decimeters = 1 cubic meter

Liquid Measure

10 milliliters = 1 centiliter
10 centiliters = 1 deciliter
10 deciliters = 1 liter

Weight Measure

10 milligrams = 1 centigram
10 centigrams = 1 decigram
10 decigrams = 1 gram
10 grams = 1 dekagram
10 dekagrams = 1 hectogram
10 hectograms = 1 kilogram
1000 kilograms = 1 metric ton