Still Alive With Sir Clive!

ZXir QLive Alive!

The Timex/Sinclair NorthAmerican User Groups Newsletter

Volume 12 No. 3 Autumn 2002

MEMORY MAP

Bryce Canyon, Utah

ZXir QLive Alive! ©

Established 1991 The Timex/Sinclair NorthAmerican User Groups Newsletter

T/SNUG Information

We wish to support the following platforms: ZX-80/81, TS-1000, Spectrum, TS-2068, Z88 and QL. If you have any questions about any of these fine Sinclairs, contact the:

Chairman

Donald S. Lambert
738 Gunnar Ln.
Forsyth, IL 62535
(217) 875-8043
dslambert@email.msn.com

Vice-Chairmen

Tape & JLO PD Library

Luke Perry
3409 NE 62nd Ave. #187
Vancouver, WA 98661

Z88 Library

Dave Bennett (HATSUG)
1275 Timber View Dr.
Mechanicsburg, PA 17055-9146
717 732-4374
dave975@att.net

QL Hacker's Journal

Timothy Swenson
2455 Medallion Dr.
Union City, CA 94587-1914

TS-2068

Rod Humphreys (VSUG)
10984 Collins Pl.
Delta, BC V4C 7E6 Canada
604 583-2819

QL PD Library

John Donaldson (CATUG)
835 Foxwood Cir.
Geneva, IL 60134-1631
630 232-6147
goodolejohn@avenew.com

AERCO & Z80 Emulator

Keith Watson
41634 Amberly Dr.
Mt. Clemens, MI 48038

--==GATOR==--

Bob Swoger (CATUG)
613 Parkside Cir.
Streamwood, IL 60107-1647
630 837-7957
Rswoger@aol.com

ZXir QLive Alive!

T/SNUG's main goal is to preserve and encourage the use of Sinclair computers by providing an open forum for the exchange of knowledge, building and maintaining of software libraries. Providing vendors, repair service and members with free ad space.

Article Contributions

Send in your articles and inputs by disk, hardcopy mail, or e-mail to:

ABED KAHALE
432 WEST OAKS TRL
WOODSTOCK GA 30188-7358
E-mail: AKahale@juno.com

WEBPAGES

http://www.timexsinclair.org
http://groups.yahoo.com/group/ts2068/
ql-users@nvg.ntnu.no
ql-users@quanta.org.uk
www.geocities.com/NESQLUG1/

Message to Members

Good things do come to an end

I am sorry to advise you that I have ran out of material for the Newsletter.

I will use the moneys in the treasury until it is used up to mail you final issues for your archives.

Please do not send any more contributions

I do thank you all for your support all of these years.

Abed

Input/Output

by Abed Kahale

Dear Abed,

I had a TS-2068, Alphacom, Interface 1, two microdrives, snapshot and ZX emulator cartridge. I made by myself one Kempston Interface, too. That was in 1984. I moved my home several times and lost my dear TS and Printer. Now I'm looking for another TS, (Exist someones here in Argentina) but I need to get the circuit diagram of the **Twister** in order to connect the IF One. I'd like to get details of the Double ROM project, too. Can you **help** me? Thanks in advance, and excuse me for my English. Note: By the way, do you know any person that are selling some TS-2068?

Dear Abed,

Thanks a lot for your fast and kindly answer. If you know where can I find some TS, or who would sell one, in USA, please advise me. Thanks again.
Hector

Dear Abed,

Thanks a lot for your help. I sent a mail to Jack. (Boatwright) Is your Newsletter on-line? Until the next mail... Regards. My address is

Hector Picone
Avenida Santa Fe 348 - 11 E
CP 1642-Acassuso
Provincia de Buenos Aires
Argentina
dunkel_piky@yahoo.com
005411-49592218 Fx

Hi Abed,

Sorry that I can't help with the twister board. I sent all of the TS stuff to Jack Boatwright except the Z88 stuff which I kept but I have not done anything with so far.

I guess you will have to put that in ZQA and see if anyone knows anything about a twister board.
I have gotten into experimenting a little with electronics and doing a little soldering. I am working on a kit I got thru Poptronics magazine web site which is based in Australia. I bought a kit to have a variable power supply the is only lacking the transformer (which I had) and am now getting it into a case so a loose wire won't short it out. The power supply uses a 7805 and is variable from 5V DC on up to whatever the transformer will supply. To get a lower voltage I can put a 1N4001 diode in series with the output, or put more than one in series. This would be elementary stuff for you. My knowledge is still back in the vacuum tube days. Timex still lives,

Don Lambert

I remember that the Twister Board was developed by Nazir Pashtoon, but I do not know his present address or his URL and I forgot in what publication that article appeared. Editor

Visit ZX-TEAM

I decided to go to the meeting, so I send an e-mail to Peter with the question that I wanted to come to the meeting. Well I live in the Netherlands and I had to borrow a good car. I own a Suzuki Alto 0.8. So I borrowed the car from my wife's parents and we were ready to go to the meeting in Fulda so finally it became Friday and we left at 10.00. But my wife had to buy some groceries and we left at 1.00 with some arguments.

I drove to Germany so naturally, we got lost in Germany. But after some nice words of me and some km drive, we got on the right road. Later we got lost again in the dark because it was hard to find the road to Fulda. So finally we came to the right place a Youth hotel which was rented by Peter. I think it was 8 or 9 PM.

I was very eager to go to meet some more people whom also like the ZX81 Finally I saw Kai Fisher and his **ZX-2002** myself. There was a table in the middle of the room and in the surroundings there where many tables. All with ZX81 ZX-96 an ZX-2002s. Everybody was eating. The only problem was. (no problem), I brought my wife and they didn't expect it, so they had to make a room for us. Then. Peter walked towards me. I told him. I was Bjorn from Holland. He told me he wasn't sure if I was crazy. But he received an email from me the day before, telling him I was coming to the meeting.

He now was sure that I was crazy. Then I saw the **ZX-2002** standing, so I ran to it to see for myself about this wonderful piece of work. After seeing it and talking to Kai, I remembered my wife and went to her. After all she doesn't like the ZX81 at all. Then Peter explained the house rules and how did they arrange the money for the meeting. Later there was a room for me and we had to go make up to bed.

Then we went back to the room with the computers. After talking to Kai, some older guy interrupted me and took me to a table and showed me his **ZX-96**. this was standing on the machine. I told him I was sorry for Kai that I wasn't ready for talking, then he told me this was normal, because every body was talking to everybody at the same time (true).

I read before about the ZX-96 and it was difficult to build. So amazed. I noticed it was standing on the machine. I pointed the ZX-96 and he told me that it was hard to make the letters for his machine. But never speaking about the technical part witch is much more difficult (For me). It also contained a floppy drive and some more stuff.

Almost every ZX81 at the meeting had a floppy drive. Some time later my wife went to bed and I talked until deep in the night with every body. I also bought many books and a ZX81 with 1K RAM from Kai (now containing 32k). And there I also found Adventure A from Artic computing.

Later, I saw a ZX81 with a RAM chip on the ROM soldered and asked what was happening, with the ROM. No, no, no, it is a WOM! A WOM???', I answered. This was a nice project. so you can program the ROM.. (Write only memory) Very nice.

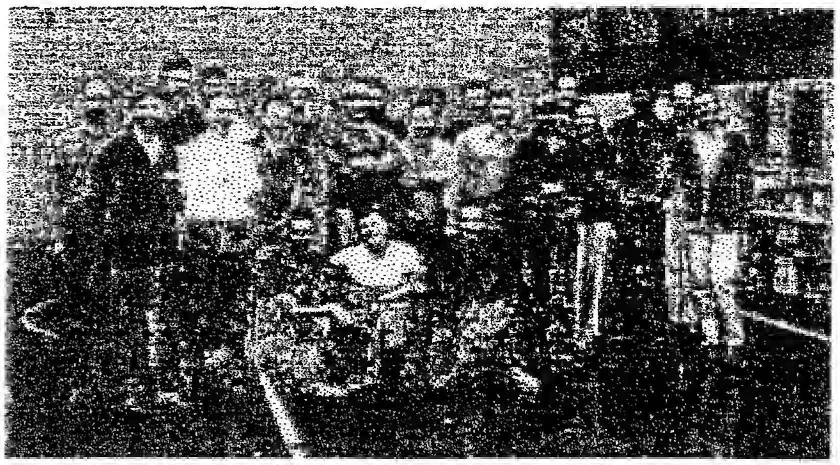
The breakfast was between 8 and 9 am. I was talking to some other guys. And soon, the ZX81 I brought from Kai was connected to an interface and for internal look into the ZX81. Now we could examine the ROM and we got a nice experiment. I learned some outputs which I now better understand. After some experimental, I searched for my wife and finally found her in the garden reading. She told me she liked to go home, (AAARGGGHHH.......

After some talking we decided to go to the town and walk there then she went early to bed. Then after 2 hours experimenting and (time fly's when you're having fun). By now there were almost 30 people inside.

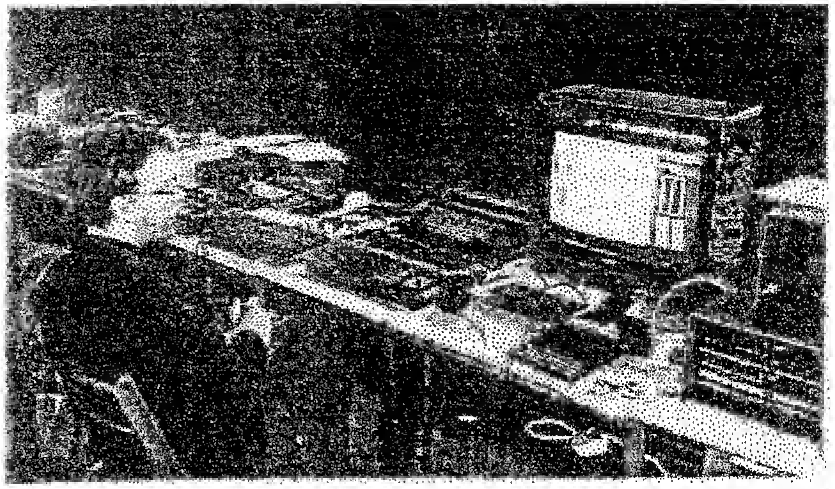
Also a guy from former DDR East Germany, who in the old time build a ZX81 himself, from TTL and without a ULA. I told Peter. that we would leave, and go to the towns and we would be back later that day. So we did and me in **trauma** because it just became fun, but we went together. Some guys from the ZX Spectrum organization, also where there, and they had connected a video camera to their Spectrum. They could now take some pictures of the people. Very nice and amusing. Later we went back to the meeting, and my wife soon went to bed, so I could talk all night about the ZX81 and the fun stuff about it and I did.

Somebody asked me if I was a user or a builder. User? Builder? I asked him, what do you mean? Well if I designed my own hardware, or if I used the ZX81, so I was a user. Glad because I had another idea about users. I also bought a little book from Peter. The price was to write a brief review about the meeting. Later I played some games, and made some friends. Peter and I, had a plan to make us millionaires and Bill Gate would polish our shoes. (Little joke)

I went talking until it was very late and my head full of ideas, I couldn't sleep of all the input I had about this little machine. The next morning, I left after the breakfast.



Short note: But if I write all the things and all the people I met, this would be a brief view of about a little book. and this is not the purpose.



We left at 9.00 AM and we where back by 17.00 PM with a speeding ticket. But it was all worth it. I had a wonderful time. I will always remember the nice time I had at the meeting. Many, Many thanks,

Bjorn

A Good Advice

On Sun, 4 Aug 2002, James Coles wrote:

Hello all. I have a Timex Sinclair 2068 with no **RF switch**. Would anyone know where I could get one from?

James Coles

I don't know if you'll find an exact replacement (these devices aren't nearly as common as they used to be), but it's easy to work around it. The output from the RF jack on the 2068 is the same as you'd get from a standard 75 ohm TV cable, only via an RCA plug instead of an F connector. You can get an RCA-to-F adapter or cable at Radio Shack, along with a switch, if needed. This is probably a better solution than the original switchbox anyway, since that device is oriented to 300 ohm connections (with two screw terminals), while most or all recent TVs are strictly 75 ohm (F connectors).

Even better would be to connect the Composite output of the 2068 to the video in on the TV; that will give you a better picture because it bypasses the modulation and demodulation (tuning) steps. At the time the 2068 was made, such connectors were not common on TVs, but nowadays they're nearly universal. Failing that, you can connect it to the video input on a VCR (even more common than on TVs), and pass it to the TV that way. (This may or may not be better than using the 2068's internal RF modulator.)

William McBrine <wmcbrine@telocity.com>
ts2068@yahoogroups.com

To: ts2068@yahoogroups.com

Subject: [TS2068] Twister Board

Hi,

Anybody can help me?

I´m trying to made a twister board for my TS2068. Most signals exist in ZX and TS.

But IORQGE (IORQULA) and ROSCS are present only in ZX bus.

Which ones are TS equivalents ? Thanks,

Hector Picone
<dunkel_piky@yahoo.com>

ROSCS is pin 28A on the 2068 backplane and pin 35 on the dock connector.

I'm not sure about the other. Hope that helps. See TS2068 Technical Manual for details. You can download it from a handful of web sites.

Scott A. Rossell
sarossell@cox.net

Offer, $100 Reward

for a TS2068 **emulator** that will work with a classic iMac, not a Spectrum. I do not want to spend hours making it work or have to learn Linux or Unix. I got enough trouble trying to work the iMac with the Dell and so far doing well with printer, SuperDrive, sound amplifiers and scanner alternately working with both.

I use **XP** on Dell so it will not accept a DOS antiquity to emulate either. The emulator does not have to be original, but I will not reward for what will not work. Maybe one of the you have such going--they would if they had ever played with the TS2068 instead of the Spectrum. Emulate from TS2068 to classic **iMac** using **8.6 OS** or emulate from TS2068 to DELL using **Windows XP** Home version.

Joan Kealy
hjkealy@rionet.org
PO Box 1439
Brackettville TX 78832-1439

This is another **QL** emailshot for a forthcoming show.

Bill Richardson:
<URL:http://www.firshman.demon.co.uk/ireland/eec.jpg>
Jochen Merz Software
<URL:http://smsq.j-m-s.com>
Qbranch:
<URL:http://www.qbranch.demon.co.uk/flyer.jpg>
Qcelt Software:
<URL:http://www.firshman.demon.co.uk/ireland/qcelt.jpg>
Saturday 24th August 2002
Roundwood Village Hall, Roundwood
Republic of Ireland

(There is likely to be an organized dinner afterwards, and plenty of Guinness in local pubs)
Tel: +353 (0) 404 45319 (Darren Branagh)

It is a lovely part of Ireland and is easily reachable from Rosslare or Dun Laoghaire (Dun Leary!) near Dublin by car. To quote Darren:

It is high in the beautiful Wicklow mountains. It is a PERFECT place for a quiet holiday - it's the second biggest tourist location in Ireland. It has a lot to offer including some of the ancient relics and dozens of the oldest standing buildings in the country, including one of the first Christian settlements. The scenery and countryside has to be seen to be believed - its stunning. Nearby attractions include Powerscourt Extate and waterfall, the monastic city of Glendalough, Wicklow Gaol (are you having us on Darren?), the Guinness Estate, and Bray seafront resort.

Darren Branagh is organizing this show of course. He will have his advertised products, including the first ever commercial QL CD ROMs. He will also be directing the filming for a unique DVD ROM of the QL scene, available later this year. We will have a selection, as usual, of software and hardware.

Tony Firshman
tony@firshman.demon.co.uk

Abed

You can add to my list of

FREEBIES

a **Gemini-10X printer** - replete with cables and manual and a **2400 Baud Modem**. Once again, no cost for shipping.

Ruth Fegley
ruth.fegley@Worldnet.ATT.net one.

Hello Fellow T/S Users....

Sadly, the subject pretty much says it. After 18 years of Timex/Sinclair I have a push-comes-to-shove situation on my hands and I have to get rid of a lot of stuff. I will keep the computers but I have a lot of commercial software, books and magazines up for grabs.

I will post a detailed list later but off the top of my head I have.

Your Sinclair Magazine
Ordi-5 (French language Sinclair computer magazine)
Time Designs
SincWare Newletter
Ottawa-Hull Timex Sinclair User Group Newsletter and a lot of similar stuff.

I have a numerous books, technical manuals, software manuals. If you are looking for something in particular give me a shout. First come, first served.

David Solly
k_david_solly@hotmail.com

If you have anything technical on the 2068 (other than the technical manual) such as a docuented Timex (not Sinclair) ROM disassembly, schematics or machine code associated with any peripherals or game cartridges, please let me know.

Scott Rossell
sarossell@cox.net

I'll buy all your magazine that relate to Sinclair and all book that I don't have it. Please give me the list and price. Thanks, Watchara

Don Dindang
kmitl22@yahoo.com

Let me know what Your Sinclair magazines you have. Maybe I can fill up the blanks that I don't have

James Coles
colesj@rogers.com

SuperBASIC

by Al Boehm

A good editor can make SuperBASIC coding and debugging much easier. In this article I also mean SBASIC since it is an extension of SuperBASIC. QD with QBASIC or Basic Linker allow a versatile editor along with Thing multitasking to provide a fine SuperBASIC development environment. Instead I like to use the TK2/SMS editor ED and the other built-in features. My reasons are:

1. Immediate syntax checking,
2. Ability to run parts of a program,
3. Good handling of GO TO and RESTORE. And
4. The option of adding various aids as desired.

Before I go any further, realize that these are personal preferences. However, I have written a lot of SuperBASIC and have used other types of programming environments. (A programming environment is jargon for a front-end program to write and debug programs; it is not the temperature in your office!) I am reluctant to use the mouse which requires good hand and eye coordination. When my eyes are blurry, I have trouble finding the pointer. Further, I can use CSIZE to make larger letters with ED; all the other editors (Please correct me if I am wrong!) use only standard size fonts. In any case, you may want to contrast my preferences by reading Tim Swenson's "BASIC Linker - A Review" in the July/August 2000 QL Today.

EDs strengths ED checks syntax as soon as you try to enter a line. The majority of my errors are due to misspelling or other syntax mistakes. I find it daunting to have to correct all the syntax errors all at once. Plus a simple syntax error, for example in a Dimension, can cause numerous errors through out the program.

In debugging, it is nice to be able to run a fragment of a program to see if that part works. If I want to see if lines 150 and 160 work, I just temporarily add a stop at 165 and Goto 150. Indeed, this fragment running is a big reason why I prefer the QL and SuperBASIC over other any other system for program development.

I avoid GOTO in programs since they tend to make it hard to follow what is going on. But sometimes a GOTO is just the best way to do something. Plus I often use parts of other people's code that has GOTOs. RESTORE sets the next DATA statement for read. When programs are renumbered using RENUM, the line numbers of GOTOs and RESTOREs are automatically updated to the new line numbers.

I like the ability to add keywords, ALTKEYs, or DOs to tailor the editing/debugging to my preferences. EDs weaknesses There are a lot editing facilities that are not in ED. For example, find, replace, move. However, keywords have been developed to do these things. I use Liberation Software's QREF which can print a list of a single or all variables or procedures and at what line number they are found. Also I found Simon Goodwin's REPLACE better then the find and change in any of the editors. For example, try replacing i to item in an editor. Every i in any word will be found. REPLACE only changes variable names. Want to edit a particular procedure but don't know where it is in a long program? Simon's Ed_Def% to the rescue. It lists all the procedures and functions in a temporary window, you pick one, and it automatically EDs to it.

I also use a modified version of Turbo's demo HOW_COME which prints the calling sequence of procedures and functions. This allows you to know when a procedure bombs where it was called from. There are certain editing sequences that I find myself doing frequently. I put the commands in a DO file and when I need to do it I DO it. For example, I have a MoveIt which moves a block of code from one location to another.

I also have a ListQ which list all the lines that have a given variable name on them - very handy for finding out if you really did initialize that variable somewhere. DO files are so easy to make, I sometimes write specialized versions for a specific program. For example, in a long running simulation, I hit CTRL/space to temporarily stop the program then type DO out which prints out the values of I, J, and K which are FOR indexes which tell me how far along things are.

By using a CONTINUE at the end of the DO file, the program automatically starts rerunning. There is very little if anything an editor can do that I can not do efficiently with ED and ED aids. In separate articles, I plan to describe in detail the ED aids that I have wrote. I admit the block move took some fancy programming which is worth sharing. God bless, Al MoveIt, A DO File To Move Blocks of SuperBASIC by Al Boehm In a previous article " ED Aids", I explained why I like to use the TK2/SMS SuperBASIC editor ED instead of a separate editor.

There is, however, one operation that is very difficult to do using only ED. Try moving a block of SuperBASIC lines from one section to another when there are intervening lines. Pretty tough work, but very easy to do on most any editor. The trouble is that editors do not keep track of GOTO and RESTORE line numbers. Thus, you have to go back and check and correct any of these. So I kept track of the commands that I needed to move a block of code and put them into a DO file.

However, there were at least five "challenges" that I had to overcome for the MoveIt to work properly.

CHANLENGE 1: Parameters To A Do How do you get parameters (in this case the line numbers specifying the block) to the DO file? I wanted something like: DO MoveIt 500,750,1100,10 where 500 is the start of the block, 750 the last line, 1100 is where to move it to, and 10 is the gap size to leave between lines when done. All my DO files are in win1_boots_, and I use PROG_USE win1_boots_ so I don't have to specify a drive with the DO. I don't know how to send parameters via a DO. Please let me know if you do. You could equate numbers to the parameters: start=500:last=750,PutAt=1100:gaps=10, since variables are available in the DO file. However, this is somewhat awkward.

A better way, I think, is simply to let the DO file INPUT the values. That way you can print prompts instead of having to remember the syntax. Plus you can use unusual variable names that are unlikely to be used in your program. (If you know how to do the equivalent of LOCal variables in a DO, let me know.)

Thus, the first section of MoveIt is: REMark MoveIt moves a block of SuperBASIC. 18 Oct 2001 by A. Boehm - public domain REMark _ before names to make use in main program unlikely CLS#0:INPUT#0,' Move lines ';_L;' TO ';_U;' To new location ';_A;' with gap=';gap$ IF gap$='' THEN _Gap=10: ELSE _Gap=gap$:END IF IF _L<_A THEN DO moveitup:ELSE DO moveitdown:END IF Note the use of gap$ to let ENTER alone specify the default, gap=10. I did not add any checks on the inputs since if they are inconsistent, the program stops with error: Out of Range, and it is easy enough to start MoveIt over again.

CHALENGE 2: LONG IFs Only inline IFs are allowed in DO files. The IF in the 4th line of MoveIt works fine. However, a long structure, for example: IF a<b THEN a=b ELSE b=a END IF will not work in a DO. The 5th line IF originally had too many statements in it to keep it readable. So instead I used the chaining properties of DO to call one or the other of the DO files: MoveItup or MoveItdown.

CHALENGE 3: MOVEING LINES PAST OTHER LINES RENUM easily changes line numbers but can not change the sequence of the lines. This limitation was dealt with by first SAVEing the intervening block of lines, then

DELETING them so RENUM could move the block past. Finally the intervening block is MERGEd back into place.

All GOTOs and RESTOREs are updated properly except any between the moved block and the intervening block of lines since the intervening lines are not in the program when the block is RENUMed.

MoveItup DO file: REMark Case 1 of Moveit - move a block higher REMark separate RENUMs allow known block limits no matter what gap was used RENUM _A TO;_A+_A-_L+2,1 SAVE ram1_TempLines,_U+1 TO _A-1 DLINE _U+1 TO _A-1 RENUM _L TO _U;_A,1 REMark Add to file command to run when Merged. _F=FOPEN(ram1_TempLines):PUT#_F\(FLEN(#_F)) REMark PUT sets file position to end PRINT#_F,'DO MoveItEnd' CLOSE#_F IF VER$='HBA' THEN MERGE ram1_TempLines REMark For non SMS, use MERGE in program only or DOs are fouled up.

CHALENGE 4: MERGE CAUSES DO TO STOP A MERGE causes a DO file to stop. How could I complete the remaining steps automatically? The process is continued by PRINTing - DO MoveItEnd - as a direct command after the lines to be MERGEd. This DO automatically activates the last section of the MoveIt files when the file is MERGEd. SAVE closes the file and leaves the file position pointer at the start. The PUT command in MoveIt's 8th line sets the position pointer to the end of the file. Then the DO MoveItEnd is PRINTed and the file is closed.

CHALENGE 5: MERGE BUG IN JSU AND MINERVA The MERGE with the added DO MoveItEnd in the file works fine with SMS but it fouls things up when I ran it with Minerva or the JSU ROM. Apparently with these OS, when MERGE is used from a DO file or even as a direct command, no more DOs will work!!?? The workaround for this is when MoveIt stops, ENTER: 1 MERGE ram1_TempLines Then ENTER: GOTO 1 assuming line 1 is not used. This works but is not slick. If you know a better way, please let me know.

Moving a block to a lower location is similar, but the block is temporarily SAVEd and the intervening lines moved up. MoveItdown DO file: REMark case 2 of Moveit - move a block lower SAVE ram1_TempLines,_L TO _U REMark Add to file final commands to run when Merged. _F=FOPEN(ram1_TempLines):PUT#_F\(FLEN(#_F)) REMark PUT sets file position to end PRINT#_F,'DO MoveItEnd' CLOSE#_F DLINE _L TO _U RENUM _A TO; _U+1,1 IF VER$='HBA' THEN MERGE ram1_TempLines REMark For non SMS, use MERGE only in program or DOs are fouled up.

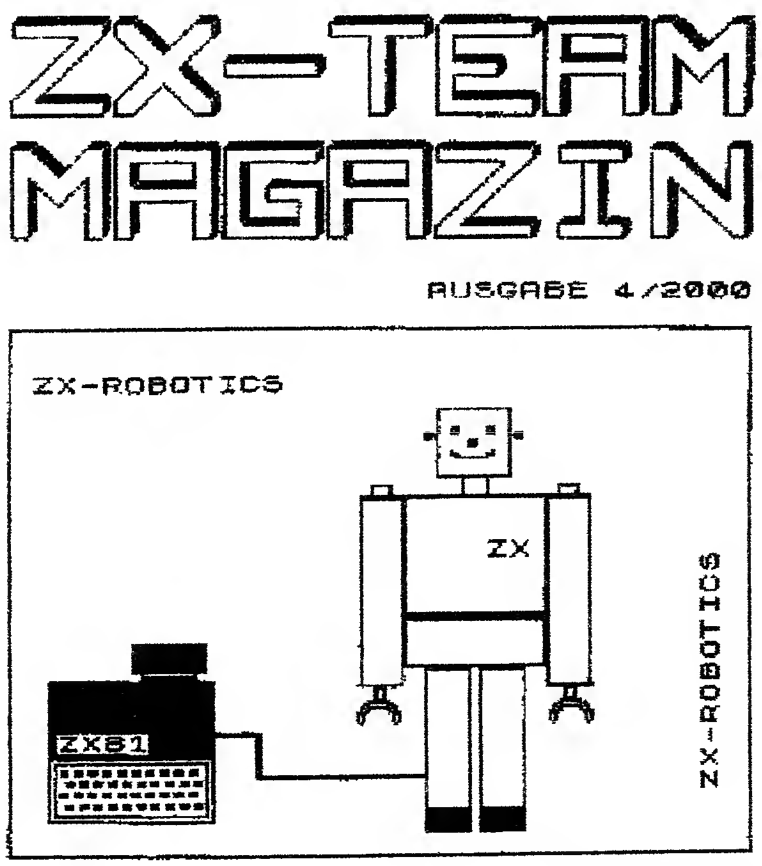
Cleaning Up And The Last Small Challenge

The last DO section, MoveItEnd deletes the temporarily ram file, RENUMs the block and the following program lines with the desired gap. It finishes by EDing to the new location of the block. The trouble is that sometimes you want to insert a block at an odd line number, then all following lines will have odd numbers. For example, if you insert at 513, then RENUM with a gap of 10, the line numbers will be 513, 523, 533 and so on. I'd prefer 520, 530, 540, etc. The MOD function in the 2nd line of MoveItEnd gets evenly divisible line numbers. MoveItEnd DO file: REMark MoveItEnd. Final DO file section of MoveIt RENUM _A TO; _A+_Gap-1-(_A-1) MOD _Gap,_Gap REMark Get divisible (e.g.100,110 not 111,121) numbers with MOD calculation DELETE ram1_TempLines ED _A

There is one more quirk I need to warn you about. RENUM doesn't work if the last line of the RENUMed block is just one less then the lines above it. This minor limitation is easily taken care of by making space for the block plus two lines as in the 3rd line of MoveItup.

This article is available on the NESQLUG web site: MoveIt is freeware and you are invited to use or change it anyway you wish. However, I do hope if you make any serious changes, you will add a REMark with your name so we know who to query when it bombs!

God bless, Al



LarKen Disk Drive

Spectrum Emulator Compatible

Larry Kenny

Introduction to LKDOS

The LarKen disk system for the Timex 2068 is fully Spectrum compatible and also can be used with AROS or LROS cartridge ROMs.

It supports all token keywords - CAT, ERASE, LOAD, SAVE, MERGE, OPEN#, CLOSE# and also GOTO and PRINT that were intended to be used with a external mass storage device. FORMAT and MOVE are supplied as programs that run in RAM.

The way these commands are implemented by the LarKen system is to precede them with a Rand USR 100: e.g.: RAND USR 100: CAT "",

For easier typing and a shorter command the PRINT #4: command can be used instead of Rand USR 100:. To use PRINT #4: you must first OPEN Channel 4 to the disk drive with RAND USR 100: OPEN #4,"dd"

You can now precede all LKDOS and Ex-BASIC commands with PRINT #4: e.g.: PRINT #4: LOAD "filename.ex"

If the PRINT #4: command is used before it has initialized, error 0 - Invalid Stream will result.

LKDOS File Names

The only other difference between a LKDOS command and a standard cassette command, is the file name.

LKDOS uses a file name that contains a program name up to 6 characters followed by a two character **extension**. A period separates the program name from the extension. The first letter of the extension tells the DOS what type of file it is . It must be an A, B or C

'A' for Array 'B' for BASIC 'C' for Code

This second letter of the extension can be any character except, if you are saving a sting array, it must be a $

The extension must be two characters long.

The only character that can't he used in a file name is a "^". This is used as a **wildcard** character for use in CAT searches.

Some examples of file names:

"Progrm.B1"	a BASIC program
"zeus.Cx"	a code file
"Name.A$"	a string array
"Numbrs.A1"	a numeric array

SAVE Commands

Before you can save your programs and data to the disk, your disk must be formatted. (See section on setup and formatting) Any formula or expression in a command can be used.

e.g.: PRINT #4: SAVE a$ (TO 6)+".CT" CODE Start, End-Start All variations of cassette commands are supported.

PRINT #4: SAVE "Prog.Bl"	BASIC program
: SAVE "Prog.B1 LINE 100	Basic Auto run
: SAVE "Prog.C1" CODE	Start, length Bytes save
: SAVE "Prog.C1" SCREEN$	Screen save
: SAVE "Prog.A1" DATA ( )	Numeric Array
: SAVE "Prog.A$:" DATA$ ( )	String array

Before saving make sure that the write protect notch on the disk is not covered by a protect sticker and that the disk is in the drive properly with the door closed.

LOAD

Similarly all cassette LOAD commands are supported. A special feature in the LOAD command will allow your BASIC program to continue even when a 'File Not Found' error has occurred.

If you POKE 23728,100 before doing a LOAD command, your program will not stop with error T if the DOS can't find the file. Instead, the DOS will PRINT "NO FILE" on the screen in the current position and return to your BASIC program. It will also put 101 in 23728 to indicate it didn't find the file.

This feature was added for programs such as word processors etc. that LOAD and SAVE text or work files from within them. The ON-ERROR command does not recognize LKDOS errors.

MERGE

The MERGE command differs from the cassette MERGE in a few ways. Programs aren't automatically stopped when merged like the cassette does. This allows 'basic overlays' which means that a BASIC program can be far larger than what can be loaded into the computer at once. You can have parts of a program merged into your main controlling program when necessary.

Two rules determine from what line # the program will continue at, after a MERGE command.

□ If a running BASIC program merges an other program into memory, the first BASIC program will continue and have control over the merged program. The merged program's AUTORUN line# will be ignored.

□ If a program is merged from an immediate command (not running), the merged program will start from its AUTORUN line #. If it is not an AUTORUN program it will not run.

Due to the fact that programs are stored on the disk in blocks, the maximum size of any line in a program to be merged is 1200 bytes. Merging from disk is a complex process and there will be noticeable difference in the speed of a MERGE compared to a load.

The MERGE command doesn't MERGE program variables only the program lines. This makes your BASIC overlay programs easier to program.

CAT (Catalog)

The Cat command displays the disk name, all the files on the disk along with the size of each file in blocks, disk, number of files and free blocks.

The syntax for the Cat in the 2068 is Cat " ", can be closed quotes or they can contain part of a string that you would like searched for, in the catalog.

E.g.: PRINT #4: Cat ".Bz", would PRINT all the files that had ".Bz" in them.

The ↑ is used as a wild card character and can be substituted for any character. e.g.: If a disk had files called progl.B* , Prog2.BB , Prog2.B* on it. And the command

was PRINT #4: CAT "Prog↑.B*"
The CAT command would find - Progl.B*   Prog2.B*

The Spectrum does not allow the search function since the syntax is only - PRINT #4: CAT
To send a copy of the catalog to the printer, you can OPEN channel 2 to the printer then do the CAT command.
E.g. OPEN #2, "p" for small printer
PRINT #4: OPEN #2,"lp" for large printer.
(note- use LKDOS Close command to close #2 . See Ex-Basic commands for use of OPEN command)

ERASE

The ERASE command deletes the file on the disk. The blocks used by the file will be made available to the free blocks. The erase syntax requires a comma after the file name.

GOTO

The GOTO command is used to select which disk drive is to be used for the next disk commands. The LarKen disk interface will support up to 4 drives, Numbered 0 to 3. GOTO 4 will select the RAMdisk.

The selected drive will stay selected until another GOTO command or the system is turned off.

Drive 0 is selected during power up. Pressing 'J' on power up will select RAMdisk.

PRINT

The PRINT #4: print "File" displays the file to screen or printer directly from disk without altering your program memory. It can be used to search a disk for a program or even to examine an NMI SAVE to see the contents of memory. The listing of BASIC programs will not print line numbers, but word processor files and text are fairly accurate. This command was inspired by the CPM **type** command.

LPRINT

The PRINT #4: LPRINT file command is BASICally the same as the PRINT command except the file is sent to the printer instead of the screen. If the file contains printer control codes (escape codes) then you should PRINT #4:POKE 16098,3 and also POKE 16093 ,32 to disable any filtering by the printer driver and LPRINT command.

NEW

Entering the command PRINT#4: NEW will cause the DOS to do a warm start. This can be used to reload an AUTOSTART program from the current drive. If an auto start is not on the drive then it will do a cold start. A warm start doesn't change any of the settings in the cartridge or any selected banks.

MOVE (Rename)

The PRINT#4: MOVE "oldname", "newname" command will rename a file on disk. This is useful for giving NMI saves a more meaningful name or changing the name of any file. It does not check for an extension (.B or .C etc.) so use it wisely. You can rename an NMI save with a .B extension so it can be loaded with out using the CODE token.

Sequential Files

A sequential file is a file that is read from or written to, one character at a time. Just as the way that you PRINT to the screen or the printer by sending one character at a time, you can now send characters or data to a file on disk the same way.

The 2068 uses channels 0 to 15 to send and receive data to the screen or printer or from the keyboard. Channels 2 to 15 can be also used for LKDOS sequential files.

(Note - The 2068 uses the command PRINT # to send data to a channel . E.g.: PRINT #2 ;"the brown fox" Do not confuse this with the 'PRINT #4:' that is used to precede LKDOS commands. In the following examples RAND USR 100: is used instead of PRINT #4 to precede LKDOS commands for clarity.)

Writing to Sequential Files

To begin, you must OPEN a file to send data to (as OUTPUT) by entering the command RAND USR 100: OPEN #C,"filename OUT"

(C = channel 2-15 , the file name can be anything up to 9 chrs , OUT or IN is the Sinclair token on the 'i' or 'o' keys . There should be 1 space between the filename and the OUT or IN token) After the file has been opened you can send data to it using the PRINT# . e.g.: PRINT #7; "HELLO"

The data sent should be separated into **records** by using a comma or a return character (13) as a separator. This will allow the data to be read back a record at a time. After all data has been sent to the file, you must CLOSE the file . E.g.: RAND USR 100: CLOSE #7
Here is a program to demonstrate OUTputing to a file.

```
10 RAND USR 100: OPEN #5,"Test.Cf OUT"
20 FOR A= 0 TO 10
30 PRINT #5; "RECORD -"; A
40 NEXT A
50 RAND USR 100: CLOSE #5
```

(Note that only OPEN and CLOSE commands are preceded by the LKDOS RAND USR 100: , not the PRINT command.)

You can also send a program listing to a seq-file by using LIST # (ch), and if the file is OPEN to channel 3 (printer channel) you can use LPRINT or LLIST.

Reading From A File

To read data back from a file, you must open it as input (IN).

E.g.: RAND USR 100: OPEN #7, "filename IN"
Then you can use the commands INPUT# or INKEYS# to read back the data in the file. e.g.: INPUT #7; a$; or LET a$ = INKEYS#7
This program will read back the file and print it

```
10 RAND USR 100: OPEN #5,"Test.Cf IN"
20 INPUT #5;A$; (or LET a$=INKEYS#5)
30 PRINT A$
40 PAUSE 30 ( delay to show data is read in by records)
50 GOTO 20
```

(note that INPUT or INKEYS are not preceded by RAND USR 100)
INKEYS# only reads one character at a time, but INPUT# reads an entire record.

If you read a file right to the end, you will get an END of FILE error and the file will be closed automatically. If you don't read to the end you must close it before any other LKDOS Commands can be used. LKDOS will send CHR$ 255 as its last character. If the file is read further you will get the EOF error.

Uses of Sequential File

Seq-files are normally used for storing data but are also handy for transferring text and data between different programs.

By opening the seq-file to channel #3 (the printer channel) you can **print** your data to a file instead of printing it to the printer. You could then for example, LOAD the file into a word processor for modification or read the data back a bit at a time by a program. You can also convert BASIC listings into ASCII text by listing them to a file, or print the output from an assembler or disassembler to a file. You can also read array, BASIC or code files using INKEYS.

Using a modem you can send and receive very large files (larger than 100K) directly to and from disk, eliminating the need for a large memory buffer.

Notes On Sequential Files

❖ When a file is OPEN, the only LKDOS command allowed is CLOSE. Be sure to use RAND USR 100: CLOSE and not just CLOSE.

❖ Input works very much like a normal keyboard input. It will not accept some characters such as quotes ' " ' or unprintable characters. You can use INPUT LINE to read in quotes. INKEYS will read in any character though. Also there will be a **click** noise for every character read using INPUT.

❖ You can have more than 1 variable in an INPUT statement but they should only be separated by a semicolon e.g.: INPUT a$; b$;

❖ Channel 2 is used by the 2068 to print to the main screen. If you open a file to channel 2, you can use ordinary PRINT commands without specifying a channel #. But it is tricky to use because no printing will be send to the screen.

❖ Do not change the disk in the drive when a file is open.

❖ If you PRINT #4:POKE 16098,3 then this will turn off the deTokenizing routine in the seq-file OUTput routine. This lets you now send all character 0-255 to the file without the LKDOS treating these as Timex Tokens . This means you can use seq-files to store machine code files. POKE 16098,0 to return to normal.

❖ The file name for a sequential file does not need an extension (e.g. .D or .C). This allows a lot of freedom but there are limitations. Such as you can not name it with a .B in the file name and LOAD as BASIC. If the file is not too big though you can LOAD it as code.

❖ Only one file can be opened at once, also data cannot be added to a file once it is closed. (These restrictions do not apply to the RAM based version of Seq-Files, available on disk)

Push Button SAVEs (NMI)

Located on the top left corner of the disk interface is a push button that will save any program running in memory to disk. It is known as a **snap shot** or NMI SAVE push button. The button triggers a Non-maskable interrupt in the Z-80 and LKDOS software in that area it SAVEs the program along with all registers and stack pointers. (the line on the bottom of the screen is actually these registers).

When the NMI push button is triggered. the computer will play a tune and then PAUSE. The entire program and screen can be saved by pressing keys 1 to 5 which will name the file "NMI-S (1-5).CM" . If you do not need to SAVE the screen and want to conserve disk space as well, you can press Caps Shift while pressing 1 to 5 and this will do a shorter NMI save that is also compatible with Versions 1 and 2 of LKDOS. The display file can he saved as "SCREEN.CM" by pressing 'S'. Also the "a" key will attempt to stop the program by a RST 8.

You can just press Enter to return you to your program if you decide to not do a SAVE.

To reload the saved program you need to LOAD it as a CODE file.

e.g. : PRINT #4: LOAD "NMI-S4.CM" CODE

AUTOSTART Programs

A BASIC or machine code program can be made to AUTORUN when the computer is turned on if the Enter Key is held down.

To save an auto start program, you should first reduce the size of disk space used by lowering RAMtop with the CLEAR (address) command so that when you type PRINT FREE you get. 150 to 200 free bytes. Then enter RAND USR 102: RUN or RAND USR 102: GOTO (line#). This will cause an NMI type SAVE. After the tune plays, press "d". This will save your program from the start of the attribute file to RAM top.

This program will auto run when the computer is turned on with the Enter key pressed. The memory above RAMtop will be clear, no UDG's. These can be loaded if necessary as a code file. If the Auto Start program is going to LOAD in a larger program it should first reset RAMtop to a high value. The lines on the bottom of the screen is actually all the Z80 registers, interrupt information and stack pointers.

Some Spectrum emulators need to be turned on by the command OUT 244,3. To make this type of emulator he able to AUTORUN a program you need to add this small machine code routine.

The best place for this routine is in the printer buffer at location 23300.

Assembly Language		Decimal Equivalent
CALL 102	(decimal)	205, 102, 0
LD A,3		62, 3
OUT (244), A		211, 244
RET		201

After this program has been loaded, you can save your AUTORUN program by RAND USR 23300: RUN

AUTOSTART saves in 2068 mode will not run in Spectrum mode and vice versa.

User Defined NMI Routine

Advanced programmers can LOAD their own NMI routine and have it run when they press the 'F' key after pressing the NMI button. Your code should be loaded into the cartridge in an unused Location and 8214 should he poked with the start address of the routine. The routine should not do any Sinclair ROM calls or disturb the registers stored at the bottom of the display file. This routine could be a screen copy, disassembler, monitor, etc. Code can he loaded into the cartridge using standard LKDOS LOAD code commands.

Kempston Compatible Joystick

A Kempston compatible joystick port is on the left side of the disk interface. It is compatible with most software written for the Spectrum. It is a good idea to anchor your joystick cable before it attaches to the board. This will prevent damage to the disk interface in case you

accidentally pull the joystick cable.
The joystick can be addressed by the command IN 31.

Extended BASIC Commands

These commands consist of extended graphic, additional PRINT and I/O device Channels and up to 3 scrolling windows on the screen.

Graphic Commands

❖ PRINT #4: DRAW width, height, pattern (This is a versatile box fill or clear command that starts at the last plot command that fills in the selected pattern (0 to 9 (or 10 = user defined pattern that consists of 8 bytes starting at address 23540).

❖ PRINT #4: CIRCLE x,y, pattern (This is a region filling command that fills from right to left, selected blank areas of the screen in the patterns described above. It will fill a line until it finds a set pixel) x, y sets the starting pos.

❖ PRINT #4: INK x (immediately changes ink color on screen)

❖ PRINT #4: PAPER x (immediately changes paper color on screen)

❖ PRINT #4: POKE a,b This does a double POKE for 16 bit numbers. 'b' can be 0 to 65535. This command can be used to POKE into the LKDOS cartridge if the address is below 16384. This can be used to change parameters for printer windows or files.

PEEKing Into The Cartridge

The method for peeking into the LKDOS cartridge EPROM or RAM is to first PRINT #4: POKE 8200, (address to be peeked). Then USR 110 will return the peeked value. e.g.: this program will print the first 10 bytes of the LKDOS EPROM.

```
10 FOR A = 0 TO 10
20 PRINT #4: POKE  8200, A
30 LET B = USR 110: PRINT A, B
40 NEXT A
```

(do not peek or POKE addresses 96 to 111 or the cartridge will crash)

Channel And Window Commands

❖ PRINT #4: OPEN #n , "device" This attaches a channel to a display device. (n can be channel 2 to 15) device can be "w0", "wl", "w2" (window 0 to 2). (See other uses for the OPEN command for sequential file or Large printer elsewhere in this manual.)

❖ PRINT #4: CLOSE #n (n can be 2 to 15). This is used for closing channels opened with the above command. Note: never use the Sinclair CLOSE command (unpreceded by LKDOS switch) to close a channel opened by LKDOS)

❖ PRINT #4: INPUT#(window), top, left pos, right pos, bottom pos.

This command defines a windows size, position and color. "window" is 0-2, other parameters are standard PRINT positions in absolute numbers (0 to 20 or 31) e.g.: PRINT#4: INPUT#1, 0, 10, 20, 8

This would define window #1 to have its upper left corner at print pos 0, 10 and the window would be 10 units wide (20-10). The bottom of the window would be at the 8th line. The screen colors that are being used when the window is defined will stay with that window.

PRINT #4: CLEAR w This will clear the window (0-2) and restore the print pos to the top of the window . It will also recolor the window.

Window Character Sets

When you turn on your 2068 the LKDOS loads the 2068 character set into the cartridge for use with the windows. The character set Loads into the cartridge at address 15200 and is 768 bytes long. You can LOAD in 42, 51, 64 or proportional spaced character sets for use with the windows also. The first byte in each character in the char set determines the width of the character. The number of pixels set in the first byte (top pixel row) starting from the right sets the width of the character. No pixels set would indicate a full 8 pixel wide character.

E.g.: This is the character pattern for a 'Y' character that is 5 pixels wide. (you could print 51 of these wide on a screen).

```
00000111 - This sets a character width of 5 pixels
10001000 - Character data
01010000 - Character data
00100000 - Character data
00100000 - Character data
00100000 - Character data
00100000 - Character data
00000000 - Character data
```

Using Windows

To use a window you need to OPEN a channel to a window and then define the size and position with the PRINT #4: INPUT command.

After that, to use the window (or "lp") you can use the standard commands in the 2068/Spectrum (not preceded by the LKDOS switch) such as PRINT #n, or LIST #n. e.g.: if channel #10 has been opened to a window then PRINT #10; "TEST' would print to it. Or LIST #10: would list your program to it.

The computer uses channel 2 as the channel for standard PRINT commands to the main screen. You can OPEN channel #2 to a window or printer. Then all standard PRINT or LIST will be directed to the selected device.

When any PRINT #4: OPEN # command is used, 50 bytes (total) of memory is added to the channel table. This may cause a problem for a program that has machine code in a REM statement. This is not reclaimed by the PRINT #4: Close command.

Large Printer Driver

The command PRINT #4: OPEN# 3, LP will allow you to use the standard LPRINT and LLIST commands for your large printer. It can be used with AERCO, Tasman or A&J printer interfaces or a user defined printer interface.
You can change line length, left margin setting, and specify line feed and it prints graphic characters as underlined ASCII chrs. TAB and comma ',' are supported also.

On power up the default settings are AERCO interface, line length 63, line feed with cartridge return and margin of 0.

To change settings you will need to use PRINT#4: POKE Add, Setting. These addresses are in the LKDOS cartridge RAM and are not affected by NEW.

POKE 16090, Maximum width of print. This is set at 64 on power up. **16092,0** if you don't want a linefeed with carriage return, 10 if **LF** wanted with **CR**. It is set at 10 on power up.

16094, Left Margin Setting. This is set at 0 on power up. Do an LPRINT after changing this.

16096, 0 for AERCO CPI, 1 for Tasman CPI, 2 for A&J CPI, 4 for user defined interface routine. For 4, POKE 8216, address of the routine.

You can open **LP** to any channel but channel 3 is normally for printer since it uses LPRINT and LLIST. You can use both Large printer and 2040 at the same time by opening the **LP** to a different channel than 3.

Copying And Moving Files

On the disk (or tape) supplied with the system there are disks programs that are loaded in for copying disks and moving single files FORMAT.Bx is used to format disks (see Formatting) and to copy entire disks using 2 drives.

MOVE.B1 is a program for moving 1 or more files between 2 drives or even just 1 drive. It is very easy to use if you only have one drive, enter drive 0 for both source and destination.

Adding a RAMDISK

A 256K RAMdisk can also be controlled by LKDOS as well as 4 floppy drives. The LarKen 1/4 Meg. nonvolatile memory board attaches to the rear of the 2068. It acts exactly like the other floppy drives but is very fast.

GOTO 4 selects the RAMdisk. Then all floppy commands can be used. Before using, you must format the RAM with the PRINT #4: FORMAT "n", command. (n = number of RAMchips on the board). Users of the LKDOS for the AERCO FD-68 can use the AERCO RAM.

The LarKen RAMdisk comes with complete documentation. Hardware experimenters can add a small RAMdisk by adding 32K of RAM in the upper 32K of the dock bank.

User Defined Command

Advanced machine code programmers can add an extra command to LKDOS by using the PRINT #4: DATA a, b, c... command. You must LOAD your new command into an unused area in the LKDOS cartridge and the POKE 8214, (address of command).

The LarKen Disk Editor includes documentation on how to use MC with the DOS and how to interpret BASIC lines.

LKDOS Errors

In addition to a number of Sinclair errors used by LKDOS, a number of extra errors are added.

The On-Error command in the 2068 does not recognize these extra errors .

S - Protect Error Disk cannot be written to. It has a write protect sticker on it.

T - File Not Found: The requested file is not in the Catalog. This can be overridden. See LOAD command

U - Disk Full: There is not enough room on the disk to save the file or more that 100 files in Catalog

V - Wrong File Type: The file type in the extension doesn't match the syntax of the command. If the token CODE or DATA is missing or the wrong array type.

W - Invalid Command: The command following the PRINT #4: is not used by LKDOS.

X - Cat Data Error: The Catalog cannot be read properly so the command is aborted.

Y - File OPEN: A sequential file is open and the only command allowed is CLOSE.

Z - Name Exists: When renaming a file with MOVE this will occur if the new name is already used.

Crc ERROR (num): A Crc error will be printed on the screen when the DOS cannot read a block after 10 tries. The number following the Crc error is the block number.

Notes On Using The LKDOS System

- Never turn the power to the disk drive ON or OFF when there is a disk in the drive with the door closed. This can glitch a disk and cause loss of data.
- It is a good idea to keep write protect stickers on important disks at all times except when you want to save or delete a file.
- You don't have to open the PRINT #4: command every time you use the LKDOS . If or example you just want to see the Cat of a disk, it is faster just to type PRINT USR 100: CAT "".
- But in BASIC programs, it is a good practice to use the PRINT#4 command. It uses 3 less bytes that PRINT USR 100: and looks neater. Your BASIC program should contain the OPEN #4, "dd" in it at the start. of the program so that, Error 0 - Invalid Stream doesn't occur if the program is just loaded and RUN.
- Never LOAD or save code over addresses 96 to 111. This will cause the LKDOS cartridge to be turned off during a command and cause a crash. This also applies to PRINT #4: POKE and using PEEK USR 110.
- When you remove and install your disk interface from the 2068 inspect the connector to make sure no pins are bent and also that the locating key is in place. A damaged or missing locating **key** can cause damage to both the 2068 and the disk IF.
- If you plan on distributing some software that is on LKDOS disks, using disks formatted as single sided will be compatible on more systems than double sided disks. (48 TPI is most common)

The LKDOS cartridge is also available for the RAMEX, OLIGER and AERCO disk systems. Your Disks are compatible with any of these systems.

Addresses Within LKDOS That can be poked to change parameters (Using PRINT #4: POKE or PEEKed by USR 110)

8195 - Dvsel - this is the current drive selected. 2 = drv0, 4 = drv1, 8 = drv2, 16 = drv3, 1128 = drv4 (RAMdisk)

8200 - PEEK - holds the address of the address to be peeked by USR 110.

8214 - USERad - holds the address of the user NMI function or the address of the User command (DATA)

8216 - PTdrv - holds the address of the user installed printer interface routine .

8218 - Chars - holds the address of 256 Window Character set

Window POKEs

Each window uses 20 bytes to keep its info. Window 0 starts 16000, W1 at 16020, and W2 at 16040. Addresses below are for W0. Add 20 for W1 and 40 for W2. You may have to PEEK the next address and then add 256* its value to the value being POKEd when using PRINT#4: POKE.

16004 – Wscrol – POKE this with 255 to increase scroll counter just like in the 2068.

16005 - Xpos – Xpos of the next print pos. (0-255). Don't go out of range of window.

16006 – Ypos – Ypos of the next print pos. (0-176). Don't go out of range of window.

16014 – Wcol – Ink and Paper color byte for window.

Printer And Detokenize POKEs

16090 – MWIDE – maximum printer width. 255 maximum

16092 – LFEED – if 10 then a line feed will be send with each carriage return. If 0 then no line feed will be sent.

16093 – PPAS – if poked to 32 then all characters LPRINTed will be sent directly to the printer with no checking. This is useful for sending control chars or graphic info to the printer. POKE it to 0 to return to detokenizing and filtering. Re POKE 'MARG' after poking this one because PRINT #4: POKE does a 2 byte poke.

16094 - MARG – specifies left margin width. Be sure MARG + MWIDE does not equal more than your printer can print (80 chrs)

16096 – PTYPE – Printer Interface type. 0 = AERCO , 1 = Tasman , 2=A&J, 4 = User defined. POKE USERad with address of user code.

16098 – PFLAG – POKE with 3 to defeat detokenizing and filtering in LPRINT "file" command and in printing to a seq-file. POKE with 0 for restore to normal.

Free Memory in the cartridge for USER routines. Code can be loaded by the standard LOAD CODE LKDOS command.

13600 - There are 1400 bytes here that are only used when the MERGE command is used.

16100 – There are 283 bytes here free at all times.

FORMATting

A floppy disk, unlike a cassette must be formatted before data can be saved on it. Formatting initializes all tracks and installs a track map, directory and disk parameters on track 0. Any errors will be reported as CRC errors followed by number indicating the track number. A good format should have no errors reported .

The Format command is not on the LKDOS Cartridge but is on the supplied cassette. LOAD the cassette and put a disk in drive 0 and close the door. The first thing to enter is the Disk Name. Usually the disk name is a short name (less than 1 line) but LKDOS allows the Name to be up to 1000 characters. This can be used for a disk description etc. The DELETE key (shift 0) can be used to correct mistakes but the editing is limited. If you make a big mistake you can just BREAK the program and reRUN it.

To end the entry of the disk name press STOP (symbol shift A). The drive number can be drive 1 to 4. Next is the head speed. It must be 6, 12, 20 or 30 milliseconds. Most modern drives can run at a 6 ms head speed. Some older single sided drives such as the SA400 Shugart run at a slow 20 ms. (Some newer LarKen disk interfaces may have a 1772-02 Floppy disk controller chip. The Format program supplied with it will have a head speed selection of 2, 3, 6 and 12 ms.)

The number of sides should be 1 or 2. (Single sided or double)

The next parameter to enter is the number of tracks. This number should be 35, 40, or 80. Most drives are 40 track. 35 track drives are obsolete now and rarely used. An 80 track drive is also known as a Quad density drive.

Type "y" if the drive is ready to format the disk. The drive should turn on and the drive should step from track to track until the format is completed. The format program can now be saved onto the disk for future use.

BREAK the program and type RAND USR 100: CAT " ". You should see a directory of the disk with all the disk info displayed. The term BLOCK refers to one track. The total Blocks on a disk is the (number of sides) X (the number of tracks)

The DOS numbers the blocks starting from track 0 side 0. Block 1 is track 0 side 1. Block 2 is track 1 side 0. Block 3 is track 1 side 1 and so on .

The Format routine creates a track map in the directory (block 0) that corresponds to the number of tracks and number of sides that were entered.

When you format a single sided disk, the format routine creates a track map on the disk that has all side 1 tracks set as in use.

Because the DOS reads tracks from side to side, you cannot read a double sided disk with a single sided drive. Also because of track density, disks formatted on an 80 track drive can't be read from 35 or 40 track drives.

You can however read single sided disks with a double sided drive. So if you have a double sided drive and you would like to transfer files to someone with a single sided drive, you can format your disk as single sided.

LKDOS Operation

The LarKen disk system consists of a Cartridge that contains the software for controlling the drive. This is called the Disk Operating System (DOS). On the cartridge is a 8K EPROM mapped in the 0-8K area of the DOS bank and an 8K RAM that is mapped in the 8-16K area of the DOS bank.

The DOS bank is activated when the cartridge senses a USR call (M1 cycle) At address 96 to 110 (decimal). For LKDOS commands only the address 100 is used (also 102 is used for SAVEing an AUTOSTART file). The other addresses are used by the DOS as entry points for the PRINT #4 command, PEEK and windows.

The DOS bank is deactivated when a read of write is made to this area of memory.

The command – RAND USR 100: OPEN #4,"dd" is used to initialize the PRINT #4: Command to point to the LKDOS cartridge. The PRINT #4: is easier to type than RAND USR 100: and there is no chance of accidentally typing a wrong USR address. The "dd" means "disk drive".

The LKDOS operating system manages all disk space allocations by using a track map. The map is created by the format program and resides in track 0 along with the catalog. By using the map the DOS keeps track of used and unused areas of the disk.

The catalog can contain up to a 100 filenames. The blocks used by each file are kept in the Catalog also.

Full information on LKDOS operation and accessing the DOS from machine code is supplied with the LarKen Disk Editor.

The rear mounted board is the Disk Drive Interface. This board controls the drive motors and the data transfer to and from the disk drive(s). It can control up to 4 drives. (3" 3.5" or 5.25" drives) The disk format used by LKDOS is double density with 10 sectors of 512 bytes per track (5120 bytes per track).

On a 40 track Double sided drive this will give a capacity of over 400K per disk. On a single sided 40 track you will have 200K and on an 80 track double sided you will have 800K per disk.

On each track 5090 bytes are actually used to hold the saved data and the remaining 30 bytes are used by an information header .

Also on the board is a push button that will trigger the NMI line on the Z-80. This is used to **capture** any program to disk.

A Kempston joy stick port is also on the disk interface for use with most of the software that is produced for the Spectrum.

Setup Procedure

Your new LarKen disk system contains some chips that can be destroyed by static electricity. This is why the boards are wrapped in aluminum foil during shipping. Observe antistatic precautions when handling these boards. Always make sure that all power is turned OFF when installing or removing either of the boards.

In addition to the LarKen boards, you will need a disk drive(s) , a disk drive cable and a disk drive power supply. The disk interface can control any floppy drive except an 8" drive .

The power supply needs to supply 5 volts and 12 volts DC. The current rating depends on the number of drives, but a general rule is 1 amp. (on 5V and 12V) per drive.

When using more that 1 drive, set the drive select jumpers on each drive, as drive 0 to 3 respectively. Also make sure that all terminator resistor packs are removed from the drive EXCEPT the drive that is connected closest to the end of the drive cable.

Connect the cartridge and interface to the 2068. Turn on the 2068 type PRINT USR 100: You should get the error message – W Invalid Command .

Connect the drive cable to the interface and drives and power up the drives and 2068. The drives should not turn on. If they do, you may have the drive cable reversed.

To test to see if drives are connected properly, insert a disk into drive 0. Type PRINT USR 100: Cat "". Drive 0 should turn on (LED and motor) and then after the approx 15 seconds it should have turned off. To test the other drives, use the command PRINT USR 100: GOTO drive number (0-3) then use the PRINT USR 100: Cat "", to activate the drive.

If you did not purchase the disk drive cable you can make your own using 2 feet of 34 conductor ribbon cable and 2 or more 34 position ribbon cable edge card connectors. The best way to install the connectors on to the cable is to use a **vise** to squeeze the connector. The connector on the interface end should be mounted opposite to the drive connector(s). If you use more than 1 drive, space the drive connectors 3" apart.

Installing the cable on backwards will not cause any damage, but the drives will turn on.

ADDING a LROS Cartridge chip to the LKDOS Cartridge

Don't attempt this unless you have a proper soldering iron and experience in delicate soldering Parts needed – 28 pin IC socket , 74HCT37 IC

Use a low power soldering pencil that is properly grounded. Temporarily remove the EPROM from the LKDOS cartridge. Bend pins 20 and 22 on the socket outwards. Note – all Ics have their pin 1 (dot or notch) towards the center of the cartridge.

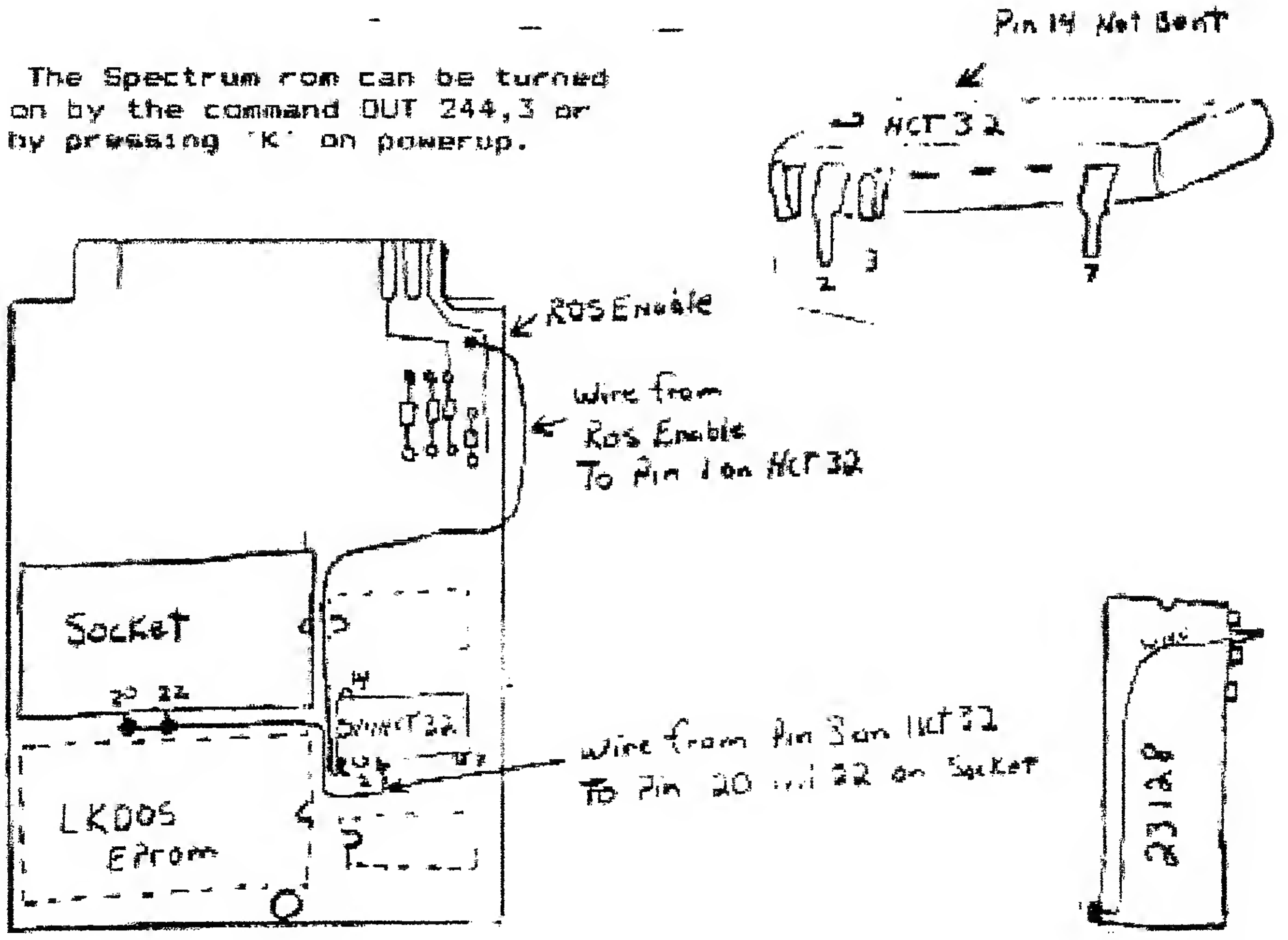
Install the 28-pin socket on top of the 8K; RAM IC (the 28 pin IC that is soldered to the board). Carefully solder all pins except 20 and 22. (you only need a tiny bit of solder). Carefully remove all leads from the 74HCT32 except pins 1, 2, 3, 7, 14. Bend pins 1 and 3 up (fold the lower half back up). Stack this chip on top of the middle 14 pin IC- (pin 1 faces in) and solder the pins 2, 7 and 14 to the same pins on the IC beneath.

Run a small wire from pin 3 on the HCT32 to pins 20 and 22 on the 28 pin socket. Run a small wire ROM pin 1 on the HCT32 to the ROS enable connection near the buss. That's it !

Check for any shorts or solder bridges. Reinstall the LKDOS EPROM and test it in the 2068 to see if it still works OK.

Remove the EPROM for your OS-64 or Spectrum emulator and insert it into the socket. If it doesn't work, recheck your work.

If you have a Spectrum ROM chip (not EPROM), part# : 23128 it may not work unless you bend pin 27 out (so it doesn't go into the socket) and ground this pin (connect a small wire from pin 14 to pin 27)


The Spectrum rom can be turned on by the command OUT 244,3 or by pressing 'K' on powerup.

The Spectrum ROM can be turned on by the command OUT 244,3 or by pressing K on power up.

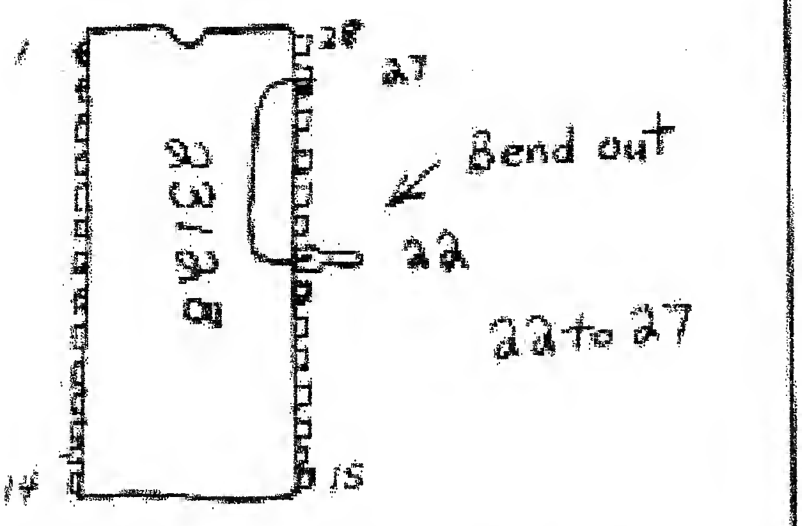
The Spectrum 23128 ROM

The 23128 ROM (NEC and others) has 3 chip selects which cause problems (it can't be deselected by LKDOS) when installed in the 2068 ROM socket or in the optional EPROM socket on the LKDOS board.

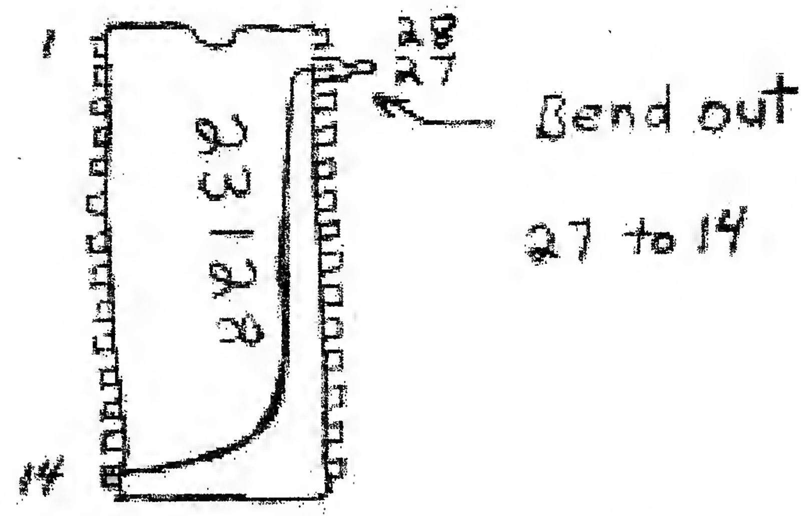
If you want to remove the 2068 ROM and install the Spectrum ROM, you must bend out pin 22 so it won't go into the socket and solder a small wire from it to pin 27.

If you want to install the Spectrum ROM into a socket added to the LKDOS cartridge you must bend out pin 27 so it doesn't go into the socket and solder a small wire from it to pin 14.

Modified For 2068 Rom Socket



Modified For Lkdos socket (This is eprom compatible)

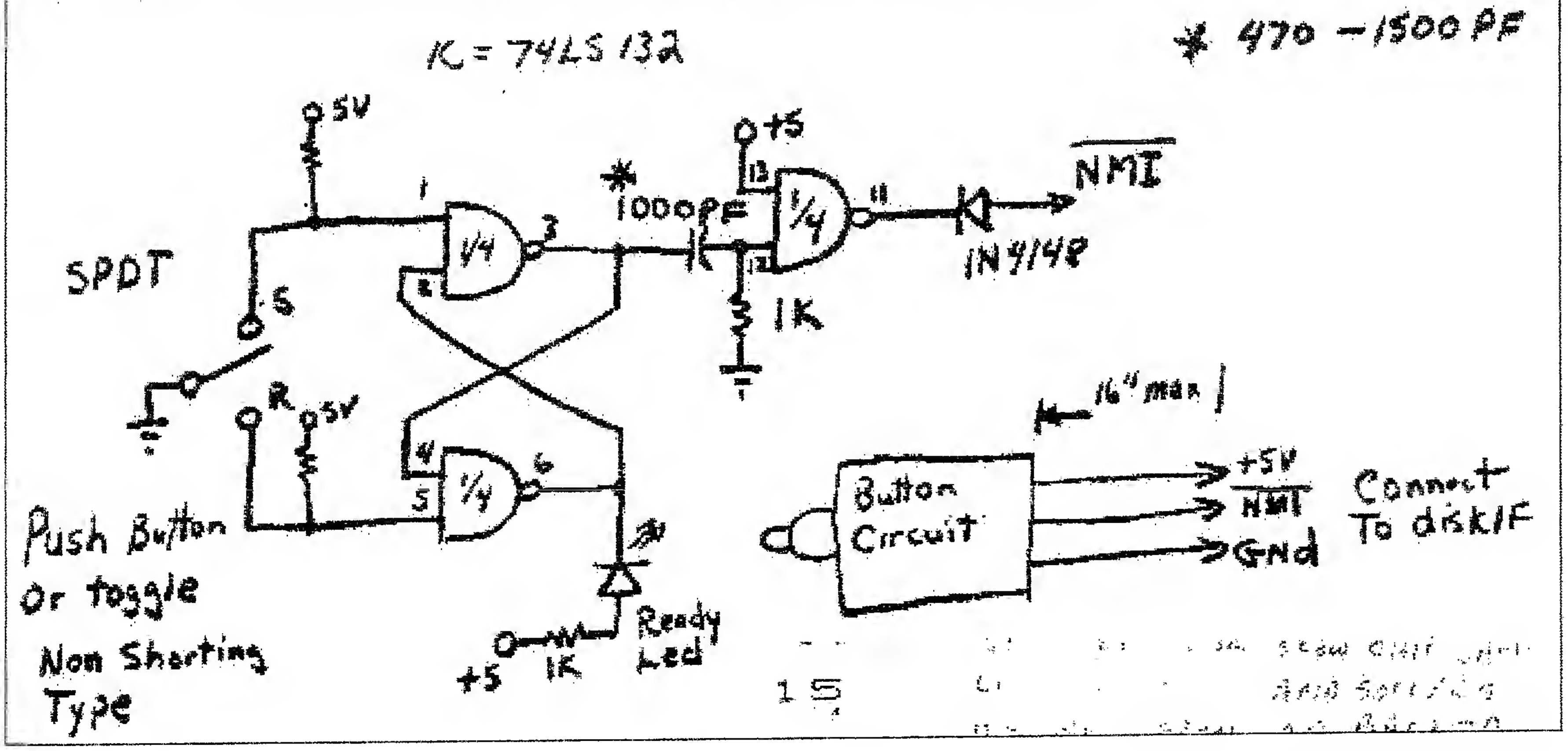


CIRCUIT for NMI SAVE Push button —— (Aero Ramex or Larger Only)

For the NMI save to work properly a push button circuit with no bounce and a short pulse is critical. This circuit is only a simple idea but you should use a flip flop or a latch like this.

A capacitor, schmitt trigger type debounce circuit should'nt be used because it doesent produce a single pulse.

To test the nmi button you can trigger it within a program and the press enter. The tune should only play once (start and end) and the program should continue. If the tune repeats its self or the program crashes then the switch has bounced.



LarKen Electronics
RR#2 Navan Ontario
Canada K4B-1H9

---==GATOR==---

Unclassified Ads

Place your ads here, it is FREE!

Please inform and/or **update the Editor of any changes in your ad/s**

QL, Spectrum, ZX-81 and Z88
www.members.tripod.com/hes_computing/hes1.html
E-Mail 74601.1535@compuserve.com
Phone 210 661-4376

Home Electronics Service
John R. Rish
5222 Kazen Dr.
San Antonio TX 78219 USA

The John Oliger Co.
11601 Widbey Dr.
Cumberland IN 46229
The John Oliger Floppy Disk System
FOR THE TS-2068
joliger@mindspring.com

NESQLUG
NEWS
New England Sinclair QL Users Group
Ed Kingsley, Editor
16 Highland Avenue
Saugus MA 01906
(781) 233-3671 EdK4@aol.com

WOOD & WIND
ARCHIVE Based QL Software
Bill Cable
Wood & Wind Computing
RR3 BOX 92
Cornish NH 03745 USA
Phone (603) 675-2218

Domino Cubes
Z88
Hardware & Software
352 7th Ave. 15th Flr.
New York, NY 10001
Phone 212 631-7563
Domino.cubes@excelsior.net

QL Hacker's Journal

Supporting All QL Programmers

Timothy Swenson, *Editor*
2455 Medallion Dr.
Union City, CA 94587-1914
swensontc@geocities.com
http://www.geocities.com/SilconValley/Pines/5865/

ZX-TEAM MAGAZIN
Peter Liebert-Adelt
LUETZOW STR 3
D-38102 BRAUNSCHWEIG
GERMANY
Email: p.liebert@t-online.de
http://home.t-online.de/home/p.liebert/zx-team.htm

Software for the Timex\Sinclair Computer
MC, VISA, American Express. Phone 717-748-1747

Keith Electronics
224 North Grove St
Lock Haven, PA. 17745

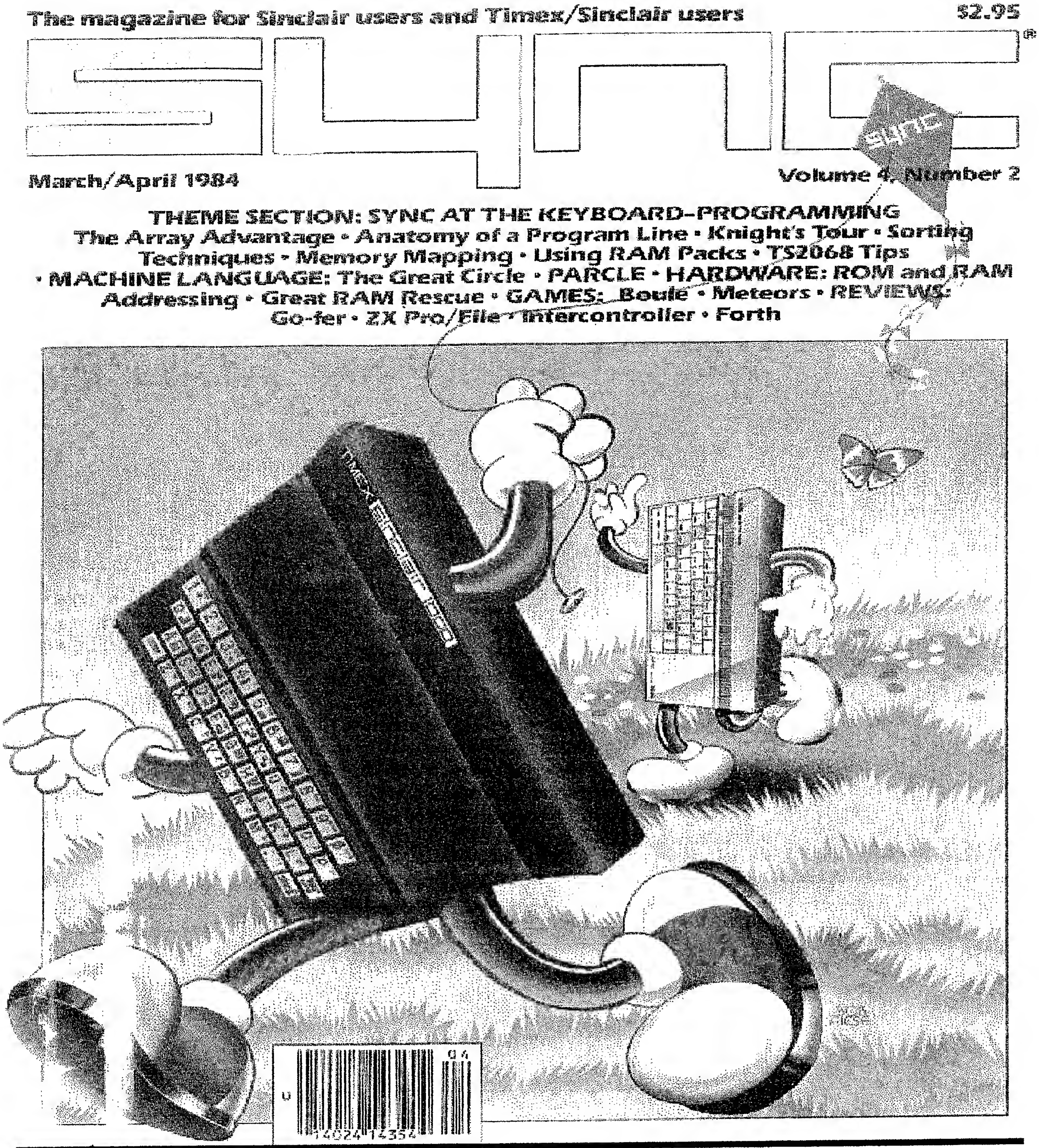
Surplus T/S Inventory

JOHN J SHEPARD III
281 130th ST
OGDEN IA 50212
< jshepard@wccta.net >

Mostly QL & TS-2068

JACK BOATWRIGHT
67325 FRYREAR RD
BEND OR 97701
< jboatno4@outlawnet.com >

Mostly ZX-81/TS-1000 & TS-2068



The magazine for Sinclair users and Timex/Sinclair users $2.95

SYNC

March/April 1984 Volume 4, Number 2

THEME SECTION: SYNC AT THE KEYBOARD-PROGRAMMING
The Array Advantage • Anatomy of a Program Line • Knight's Tour • Sorting Techniques • Memory Mapping • Using RAM Packs • TS2068 Tips • MACHINE LANGUAGE: The Great Circle • PARCLE • HARDWARE: ROM and RAM Addressing • Great RAM Rescue • GAMES: Boule • Meteors • REVIEWS: Go-fer • ZX Pro/File • Intercontroller • Forth

Abed, if it is not too late for the Sept issue, the following announcement could be included:

NESQLUG will have their 2002 annual weekend Fall meeting on 26-27 October at Bill Cable's unique hand built home in New Hampshire. The date was selected to coincide with the nearby machinist model show going on that weekend. Any QLer interested in attending should contact Bill for details:

cable@cyberportal.net

I hope someone from TSNAUG can attend the meeting to discuss coordination between our groups in such things as next years QL show. God bless,

Al Boehm <albertboehm@juno.com>