

# SYDTRUG NEWS

## SYDNEY TRS-80 USERS GROUP NEWSLETTER

Volume.4 Issue.12 August 1984

### IN THIS ISSUE

Have you noticed the new frontpage format for the newsletter this month? It was printed using DOTWRITER 4.0, absolutely fantastic! A detailed review will appear next month.

We open this months' issue with more "ROMblings" from our secretary, Jim Whittaker. He has a few words to say about Membership, the Newsletter, Committee reports, Feedback from the membership, Correspondance and an error in a previous offer.

The Gamers corner, by Keith Black, appears again with a review of a slightly different "Invaders" game (an oldie but a goodie :Ed).

The Prophet and Oracle Speak column continues this month with news of yet another TRS-80 BBS in Sydney (that is three with another in the testing phase). Larry (Prophet) continues his series on programming in COBOL and finishes with some news tidbits.

This month we have re-printed a couple of articles from the U.S. newsletter NORTHERN BYTES, the first being a few Zaps to Newdos/80 ver. 2 for the Model III, I hope these are useful to those members using Model III Newdos (especially on the Mod 4).

Ben Barat tells us how the old faithful (Mod I) can be used to improve the (Rotten) APPLE II, with his contribution of a Delay Filter for operation under LDOS (Keep the good work coming Ben).

My contribution to this issue follows, with the details of how to create your own self-booting files on disk. This started out as another re-print from NORTHERN BYTES, but as the original method didn't work when I tested it, I was forced to develop a method that did (More on this in a later issue).

Random I/O is a new department by Errol Rosser, and this month includes a few helpful hints from Errol and Alan Johnstone.

We finish off with a very interesting article on a JKL screen print routine (for EPSONs), which uses an interesting technique of including the program as a SYSTEM overlay for NEWDOS/80.

Your Editor,  
Gary Bryce.

---

### MEETING NEWS

The special interest group meeting lectures were well attended and received an the Introduction to BASIC programming and Assembly language continue this month. Alans talk on his mods to Newdos/80 was very informative (all thanks to you Alan). I will take up the third session this month with a talk on Disk Directory structure and using Superzap.

As always the group meets at the rear of Pattersons Florists, Botany Rd, BOTANY (entrance from Chegwyn St.), on the second and third Saturdays of the month.

August	11th	Monthly General Meeting	September	8th
"	18th	Special Interest Meeting	"	15th

---

The contents of this publication are (c) 1984 by SYDNEY TRS-80 USERS GROUP. All rights reserved. Material appearing in this publication may be reprinted in similar computer club newsletters and nonprofit publications if accompanied by the following notice:

**Reprinted from SYDTRUG NEWS**

Secretarial ROMblings.

by Jim Whittaker.

Firstly I would like to give a warm welcome to all our new members. I so often forget this common courtesy. We are a self help group of enthusiasts, presently numbering 184, and everything we do is worked on the Honesty system. If you cheat the Club, you cheat yourself. The committee requests only two rules be kept at meetings. 1) Please wear your membership badge and 2) Please sign the membership book when you first arrive, this also includes a section for visitors.

**HOUSEKEEPING:-**

I hope you have all seen the change in format of this News Letter. We are now a REGISTERED PUBLICATION. We owe the most credit for this to Denis Pagett and Algis Bizys who have really done a lot of work behind the scenes. It has also halved our postage charges.

**Renewals:** Please pay promptly. Your renewal date is on your mailing label. If you are more than one month in arrears you will not receive your News Letter which would now be like cutting off your right arm.

I have received a request for a Year Book of previous News Letters. Would anyone else be interested. There would probably be a minimal charge. Advertisements are now being accepted for the News Letter. The first ones in will get the prime spots. This service is for your benefit.

**COMMITTEE REPORTS:-**

The committee has debated, at length, the question of student and interstate discounts. We have decided to leave our present pricing as is. We are organising a social PICNIC DAY outing to include the computer widows and offspring. The tentative date is SUNDAY 23rd SEPTEMBER. There is a token charge of \$1.00 per car load that will go to buying lollies for the littlies. Even if you dont turn up to any other activity please support this as I feel it will be a very important interaction within the group. If you need a lift then please see me to arrange something. All committee members will be taking names of those who are coming so please register early to allow us to plan some good activities.

We have decided to buy, for member use, a printer ribbon re-inker. If anyone has any good or bad reports then please let us know your recommendations. Regular columns on any subject for the News Letter are encouraged. The only restriction we have is that it must be REGULAR and presented on some magnetic media.

We are now actively going to support some research and development on modifications to our machinery. The designs are going to be useful and relatively inexpensive. Presently we are looking at

- a) A RAM disk - bank selectable memory
- b) Hard Disk Interface
- c) Speed up mod for the MOD III.

If anyone wishes to help or independantly invent some hardware mods, we will consider supporting you.

**FEEDBACK:-**

There has been some constructive criticism about the High Profile of the committee and the aggressive attitudes I personally take. I do not excuse myself for this. If the committee were to take a lower profile, they would end up in the subterranean mess of the previous group and the membership would again suffer. Finesse is directly proportional to the fee paid. If I am to be on the unpaid management team, I will do things my way. You will all have your chance to express your feelings at the AGM.

By way of explanation of the above diatribe, the committee is really trying to work for you. The problem is lack of both feedback and direction. We are often stabbing in the dark, trying to find activities that are of interest to the majority of members. Our ideas and resources are limited and is dependant upon the support of you the member. We and especially I are not always right, and it is up to you to let us know. Contrary to appearances, I do not bite and will not belittle any suggestions or criticisms you wish to bring up. This is not to say I wont give my own definite views.

**OLD NEWS:-**

Some time ago, I explained on offer we had to set up a tandem BBS. I have had ONE reply, so we will not be setting up another BBS. There has been NO questions for our experts to answer in the Q & A column. There have been NO suggestions as to the goals or direction the club should take. NO one has approached me seeking nomination at the next AGM.

**CORRESPONDANCE:-**

We have received a News Letter called NORTHERN BYTES from the U.S.A. It holds a lot of specific information for our usage. Some of the articles will be reproduced in our News Letter.

The Sysop has been giving me a list every month of enquiries logged on the BBS. I send a copy of the News Letter and a joining form to each. Last month it was so good I had to scratch around for any old News Letters I had, including last December issues. This is good exposure for the club so we will now have to increase our production of the News Letter to 300 copies per month.

**MISTAKES:-**

Elliot Humphries (now a member) is offering a holiday to any members who can impart some computer info. to him. The trouble is I got the spot wrong. It is called WATTLERIDGE and is located just north of ARMIDALE. This is a commercial operation run by Elliot and I have the brochure available for any who wish to take up this FREEBY.

Well I've said a lot this month, but not all I wanted. I hope some of it will sink in and that I haven't frightened you off. We really are working for you and we have your best interests at heart. Unfortunately I get very busy at the meetings so please bear with me and you will eventually get my attention.

P.S. I passed my exams and I hope all others did the same.

---

**GAMERS CORNER**

by Keith Black.  
516-3673

Welcome to this months corner, this month I will review the game "Invaders +" by Larry Ashmun.

When you boot up "Invaders +" you get the theme from star wars and on the screen you get the message "get the invading forces before they get you !!!!!". You use the left and right arrows to move, and the space bar to fire, you are asked what level of play, 1 being a "piece of cake" and 9 being "suicide".

The invaders move slowly and there are few bombs in level 1 but if you try level 9 watch out faster invaders and more bombs. In play the bombs only drop above you and not all over the place so you have to watch for them, I found level 9 very hard as the invaders really move fast as you get rid of them.

If you are like me you get hit the four times in short time, then you get the star wars theme again and the screen shows shots fired, missed shots, your score and the high score, if your score is the high score then the word \*\*\*\* new \*\*\*\* will be under the high score, also the messages "your tombstone will look nice with that score on it" or "maybe you will have better luck in the next life" for low scores, and "how does it feel to be a hero" for new high score.

You can use the break key to reset the game at any time, the scoring is large invaders = 50 points, small invaders = 75 points, mother ship = 100 points, bomb = 150 points, if you miss you get minus 20 points.

I found this to be one of the best invaders games with 9 levels of play and good sound it keeps you on your toes. Well, good luck with the game and get some of those scores in. Until next month good gaming.

The Prophet & Oracle Speak.

by Larry Lewis & Rowan Evans.

In this edition we have the second part of the 'Using COBOL' series and some other interesting bits of news.

NEW BULLETIN BOARD

Members are invited to try 'The Prophet Remote Bulletin Board, the telephone number is 628-7030, please note that the system is still under development and is still a bit sparse.

The system uses 'The Bread Board System' (TBBS) which I purchased from the states on the basis of a rave review in the U.S. magazine Computer User (Feb 84), I don't usually buy software as reviews always seem to overstate the good points of the programs with none of the bad points aired. This review had so much to say about the system I just couldn't resist and bought it, now I'm waiting official confirmation (I already have it by telephone call from the President of the Coy) to distribute the software here, its that good!

USING COBOL - PART II - PROGRAM FORMAT

The format of a COBOL program is very rigid and all definition statements must be included, as none have a default setting (e.g. BASIC automatically 'CLEARs' an amount of string space).

All programs consist of various separate definition areas called 'DIVISIONs', although some versions of COBOL have additional Divisions the main ones are:

IDENTIFICATION  
ENVIRONMENT  
DATA  
PROCEDURE

The Divisions must be specified in this order and must be present in each program. Each Division is again separated, into 'SECTIONs', these Sections are where you define all occurrences of a parameter (e.g. File Attributes) for the whole program, the contents of the Sections of a program are explained later.

The reason for this rigidity is twofold, it allows the compiler to be more efficient both during compilation and with the object code (whether it is machine or psuedo code) due to having all common information and routines together, it also provides a documentation medium allowing a programmer to see at a glance the number of files used or what a field is defined as etc.

Furthermore (although not generally applicable to micro-computers) are what are call 'optimising' compilers which make use of the structure to reuse code etc to speed up program loading and executing (sometimes with strange side effects).

Now lets look at the Sections contained in each of the Divisions.

IDENTIFICATION DIVISION

Sections in this Division are:

PROGRAM-ID  
AUTHOR  
INSTALLATION  
DATE-WRITTEN  
DATE-COMPILED (Not applicable to TRS-80 COBOL)  
SECURITY

All of the entries except PROGRAM-ID are optional, all other entries are for commentary use only (ie the computer doesn't care what you put in the Section, it is for programmer use only, an example of this Division is:

Prophet & Oracle (cont).

## IDENTIFICATION DIVISION.

PROGRAM-ID.	EXAMPLE.
AUTHOR.	LARRY LEWIS.
INSTALLATION.	L.A. & H.M. LEWIS COMPUTER SERVICES P/L.
DATE-WRITTEN.	JULY, 20 1984.
SECURITY.	MUST BE AUTHORISED BY PERSONEL TO LOOK.

## ENVIRONMENT DIVISION.

The Environment Division serves two functions:

- (1) It identifies the computer to be used for compiling and executing the program (usually the same). This is done in the Configuration Section.
- (2) It relates programmer chosen filenames to data files (filespecs) on a diskette. This is done in the Input-Output Section.

Because the format of this division is very much machine dependant, I will describe the entries required using Ryan-McFarland COBOL on the TRS-80's.

## Configuration Section

The entries in this Section are:

SOURCE-COMPUTER  
OBJECT-COMPUTER  
SPECIAL-NAMES

The Source and Object computer statements are comment entries only and are as per the other comment entries described before.

The Special-Names entry performs a number of tasks, it enables the programmer to describe certain run time options (e.g. use comma instead of full stop as the decimal indicator viz 100,00 instead of 100.00), it also allows for the operator at runtime to indicate up to eight conditions to the program these are called Status switches (In mainframe usage, IBM, these are called UPSI settings (Universal Program Setting Indicators)), an example would be 'RUNCOBOL EXAMPLE (S=00011100)' this tells the program Example that switches 1,2,3,7 & 8 are off and switches 4,5 & 6 are on, the program can then act accordingly (or ignore the setting, of course).

Because this Section is so program dependant I will not explain the entries any more, I'll just give you an example:

## CONFIGURATION SECTION

SOURCE-COMPUTER.	TRS-80-MOD4.
OBJECT-COMPUTER.	TRS-80-MOD1.
SPECIAL-NAMES.	SWITCH-1 ON STATUS IS DISK-MOUNT-WAIT.

## Input-Output Section

The Input-Output Section is again very heavily machine dependant, in fact the TRS-80 only has a very restricted set of commands for this area, and they are variants of the basic instruction to the compiler about files. These instructions are very complex, depending on whether the file is a report print to a printer, a sequential file, and indexed file etc. The full organisation of this Section is far beyond the scope of an introduction, so I will show you a couple of examples and leave it at that (sorry you will find now that full explanations will be counter productive to this overview and if you require further information you will need to buy one of the books I describe at the end).

Prophet & Oracle (cont).

INPUT-OUTPUT SECTION.

FILE-CONTROL.

```

SELECT TEST-FILE,
      ASSIGN TO RANDOM, "TESTFILE/DAT.PASSWORD:0",
      ORGANIZATION IS INDEXED,
      ACCESS MODE IS DYNAMIC,
      RECORD KEY IS TEST-FILE-KEY-FIELD,
      FILE STATUS IS RETURN-CODE.
SELECT PRINT1
      ASSIGN TO PRINT, "PRINTER".
    
```

Just from those two examples you can see that the amount of information required is very much dependant upon what you are trying to get the program to do, which means more study than can be done here.

The remaining two Divisions will be covered in the next edition.

OTHER NEWS

For all those users with Model 4's Tandy has now (at last) released CP/M+ at \$199.95. Which will utilise the Tandy hard disk, sorry the disk can't then be used by TRSDOS/LDOS. Also the 15 Meg hard disk has been released at a cost of \$3999 (we think).

Bye,  
Larry (Prophet)

---

MODEL III NEWDOS/80 ZAPS

by Tony Domigan

These Zaps were extracted from NORTHERN BYTES (vol 5 iss 4), and are for the Model III version of NEWDOS/80, unless otherwise specified.

1. To boot in lower case:  
SYS0/SYS,11,2F change 28 05 to 00 00
2. To boot in 4MHz on the Model 4:  
SYS0/SYS,13,B4 change 00 00 00 00 00 00 00 00 00 00 00 00  
to 3A 10 42 F6 40 32 10 42 C3 B6 48 00  
SYS0/SYS,11,DF change C3 B6 48 to C3 A8 50  
  
Also execute the command SYSTEM,0,BJ=2
3. To correct date and time updating on warm boot (on Model 4):  
SYS0/SYS,02,20 change 1E 20 0D 36 1E 21  
to 19 20 0D 36 19 21
4. To allow the date in DD/MM/YY format:  
SYS0/SYS,12,2B change 20 ED to 00 00  
SYS0/SYS,12,3E change 20 ED to 00 00  
SYS0/SYS,13,4E change 4D 4D 2F 44 44 2F to 44 44 2F 4D 4D 2F
5. To allow DD/MM/YY in the FORMAT command:  
SYS6/SYS,12,45 change 32 to 34

(ED note: as I don't have access to a Model 4 or III these Zaps have not been verified - feedback noting any problems will be published if they come to hand.)

---

LDOS DELAY FILTER

by Ben Barat

Know how you get used to one kind of keyboard? Well I have 'grown' up with my TRS80 one - like it or not. When I bought an Apple last year, I could not type without looking at the keyboard all the time. There are no up and down arrow cursor keys (unless you use an escape sequence - yuk!). If I want to type anything for the Apple, I do it on the TRS80 and send it down the RS232 line. Unfortunately, the Apple often has a short 'think' after it receives a carriage return. This means that characters are lost from the beginning of the next line (most Apple serial cards were designed before handshaking was thought necessary).

I thought about it for a while, and came up with two solutions - build a handshaking serial interface for the Apple, and alternatively, introduce a time delay after each carriage return is sent on the TRS80. What follows is a short relocatable device filter for LDOS and a JCL driver to install it easily, which is an adaptation of the 'Add a LF after CR' driver found in the LDOS manual. This routine may also be of use to owners of older style printers such as the Teletype printer which are popular with limited funds hobbyists, where the print head takes a long time to return to the left margin.

The filter introduces a time delay of about one second. This should be ample delay for most devices. If you wish to alter the delay constant, change the loop counter initial value for the DE register. The shorter the delay required, the lower the hex value should be.

/JCL FILE

```
SET *CL RS232R
ROUTE *PR *CL
FILTER *CL DELAY/FLT
.Delay Filter routine installed
```

DELAY/FLT SOURCE CODE

```
00010 : PRINTER <CR> DELAY          20/6/84
00020 : FILTER ROUTINE FOR LDOS TO
00030 : DELAY SENDING BYTES TO THE
00040 : PRINTER AFTER A <CR> IS DETECTED.
00050 :
00060 : TO FILTER, USE FILTER *PR USING DELAY/FLT
00070 :
00080 : BY BEN BARAT 20/6/84
00090 :
00100 DLYCNT EQU 0FFFFH ;DELAY COUNT
00110 LF EQU 10 ;LINEFEED
00120 CR EQU 13 ;<ENTER KEY>
00130 EXIT EQU 402DH ;LDOS RETURN ENTRY
00140 ABORT EQU 4030H ;ERROR ABORT
00150 HIGH EQU 4049H ;HIGHEST USABLE MEMORY
00160 DPLV EQU 4467H ;DISPLAY MESSAGE
00170 LOGDT EQU 447BH ;DISPLAY & LOG MESSAGE
00180 DRG 5200H
00190 ENTRY LD A,(DE) ;GET DEVICE TYPE
00200 AND 2 ;MAKE SURE ITS AN
00210 JR Z,NOGOOD ;OUTPUT DEVICE
00220 PUSH DE ;SAVE DEVICE DCB
00230 LD HL,MSG ;POINT TO INITIALIZATION
00240 CALL DPLV ;MESSAGE AND DISPLAY IT
00250 POP IX ;RECOVER DEVICE DCB
00260 LD HL,(HIGH) ;REDUCE HIGH BY THE
00270 LD BC,LAST-START ;LENGTH OF THIS DRIVER
00280 XOR A ;CLEAR THE CARRY FLAG
00290 SBC HL,BC ;CALCULATE NEW HIGH
00300 LD (HIGH),HL ;DRIVER NOW PROTECTED
00310 INC HL ;POINT HL AT NEW START
00320 LD A,(IX+1) ;XFER DRIG DCB VECTOR
00330 LD (PUTBYT+1),A ;TO DRIVER CALL
00340 LD (GETBYT+1),A
00350 LD A,(IX+2)
00360 LD (PUTBYT+2),A
00370 LD (GETBYT+2),A
00380 DI ;NOT DURING UPDATE
00390 LD (IX+1),L ;UPDATE DCB VECTOR
00400 LD (IX+2),H ;TO FILTER ENTRY
00410 EX DE,HL ;XFER NEW START TO DE
00420 LD HL,START ;LOAD ADDRESS OF DRIVER
00430 LDIR ;MOVE DRIVER TO TOP
00440 EI ;ENABLE INTERRUPTS AGAIN
00450 JP EXIT ;RETURN TO LDOS READY
00460 : *****
00470 : ERROR HANDLING
00480 : *****
00490 NOGOOD LD HL,ERRMSG
00500 CALL LOGDT ;LOG ERROR MESSAGE
00510 JP ABORT ;ABORT THE REQUEST
00520 MSG DEFM '<CR> DELAY FILTER INSTALLED'
00530 ERRMSG DEFM 'OUTPUT ONLY - ABORTED'
00540 : *****
00550 : ACTUAL FILTER ROUTINE TO SHIFT UP TO HIGH
00560 : *****
00570 START JR C,GETBYT ;JUMP TO GET REQUEST
00580 PUTBYT CALL O ;OUTPUT TO DRIG DEVICE
00590 CP CR ;WAS CHAR A <CR>?
00600 RET NZ ;GO BACK IF NOT
00610 DELAY PUSH AF ;SAVE REGISTERS
00620 PUSH DE ;ON STACK DURING DELAY
00630 LD DE,DLYCNT ;DELAY COUNT
00640 LOOP DEC DE ;16 BIT DECREMENT
00650 LD A,D ;COMPARE BOTH REGISTERS
00660 OR E ;FOR ZEROS
00670 JR NZ,LOOP ;LOOP IF NOT ZERO
00680 POP DE ;RESTORE REGISTERS
00690 POP AF ;FROM STACK
00700 LD C,LF ;LOAD <LF> IN REG
00710 JR START ;GO BACK AND CONTINUE
00720 GETBYT JP O ;DON'T FILTER INPUT
00730 LAST EQU $
00740 END ENTRY
```

**CREATE A SELF BOOTING DISK USING NEWDOS/80**

by Gary Bryce.

I found an article in an issue of a U.S. newsletter (NORTHERN BYTES vol.5 iss.4 by Joachim Kelterbaum) which showed how to create a self-booting disk, and decided to include it in this month's issue. After typing it in, I decided to try it out and "Lo and Behold" it didn't work! (it kept coming up with SYS ERROR after boot). Not to be deterred I investigated further and worked out how to do it. The procedure works for single density disks on a Model I and double density disks on the Model III.

Format a data disk (no system on it) - note that the TSR (Track Step Rate) for boot must be set correctly by PDRIVE before formatting the data disk. Now copy the /CMD file to be made self booting to the formatted data disk. Use the DIR d A command (where d is the number of the drive containing the data disk) to ensure that the file was saved in one contiguous block (usually it will have only one extent, as listed in the EXT5 column of the directory display, the exception being those files over 32 grans or 41K long).

Now use the DFS option of SUPERZAP to read relative sector 0 of the file, note the DRS (Drive Relative Sector) of this sector, return to the SUPERZAP menu (X), select the DTS option and enter the drive number and DRS, the resulting display will show the starting Track (TRK) and Sector (TRS) numbers of the file (for Mod I double density only, add 1 to the TRK). Using SUPERZAP, patch the following bytes of sector 0 in BOOT/SYS of the data disk in the following manner :-

**MODEL I -**

Byte 12 - Start Sector # of the /CMD file

Byte 13 - Start Track # of the /CMD file

Byte 4B - change from C8 (RET Z) to C9 (RET).

**MODEL III - (Boot/Sys sector 1)**

Byte 05 - Start Sector # of the /CMD file.

Byte 06 - Start Track # of the /CMD file.

Byte 3E - change from C8 (RET Z) to C9 (RET).

(Please note that I have not verified the Model III mod, as I have a Model I.) Now a few words about what types of files can be made self-booting. Generally speaking any file which has no calls into DOS would be suitable (e.g. Cassette based games, utilities).

**RANDOM I/O**

with Errol Rosser

This column has been set up to allow those of you who have little tid-bits of information; whether they be reports of 'bugs' or de-bugs, short-cuts, manual errata, or helpful hints; to get them distributed to others. If you give your info. etc. to me, I will put it in this column for others to read. With your help, YES - YOU, THE READER, this will not be a random column, but a regular part of the magazine, supplying help and info. to others.

The contributions this month are:-

From Alan Johnstone - re BASIC token 252 dec. 0FC Hex

This token is not documented in the Tandy Basic manual, but according to Alan, if it is placed in a BASIC program line, by POKEing it or by using a monitor etc., it prevents the listing of the of the line from the token onwards.

Also from Alan - zap for NEWDOS 80 V2.0

If you wish to display your directories without clearing the screen, then this zap is for you. Using ZAPPER or SUPERZAP etc.

change	SYS0/SYS	0,DE	from	CC 8F
			to	21 8F

Now one from myself to finish - Boolean Algebra in Basic

Whilst setting up a Basic program to simulate a logic circuit, I found that the NOT funtion does not give correct results in Boolean Algebra equations.

eg. 10 A = 1 : PRINT NOT (A) : '..... gives the correct result of " 0 "

but 20 A = 0 : PRINT NOT (A) : '..... gives a result of " -1 "

using the following function corrects that bug :-

```
5 DEF FN Y(Z) = ABS ( NOT (-Z))
```

so that instead of using NOT (A), use FN Y(A)



EPSON RX80/FX80 JKL ROUTINEAS AN OVERLAY MODULE FOR NEWDOS/80 ver.2.

by Joachim Kelterbaum.

reprinted from Northern Bytes.

Many NEWDOS/80 users find it quite convenient to be able to get a screen dump just by pressing the JKL keys. However, since the TRS-80 uses graphic blocks that are not usually included in the character generators of printers, it is not possible to make full use of this facility. Some printers though are able to do dot image graphics and this provides a means to simulate the TRS-80 graphics blocks. In the following article a method will be shown to implement this facility on NEWDOS/80 version 2 for the Epson RX80 / FX80 / MX80 III printers.

**Function of the Overlay.**

Certainly you know that NEWSDOS/80 is a sophisticated operating system. Such a degree of sophistication would by no means be possible if all of the system's functions were located in RAM at all times (you could do that but there would be little room left for users programs). The solution to this problem is the overlay technique. This is accomplished in the following manner:

There is only one part of the system (SYS0/SYS) permanently resident while NEWDOS operates. In addition, there are approximately 20 overlay modules (SYS1/SYS to SYS21/SYS), of which only one is resident at a time. There are two overlay areas (the DOS overlay area at 4D00H-51FFH, and the secondary overlay area (used mostly by Basic) at 5200H-6FFFH). Each time a certain overlay is needed by the system, the DOS overlay loader (part of SYS0/SYS) will load this module to the overlay area (if that module is not loaded already). The module loaded before will simply be 'overlayed'. I suppose that you can well imagine that this particular function of the DOS is of vital importance to the system, this might be why the function of the overlay loader is hardly documented in the manual.

Obviously the authors of NEWDOS/80 - as well as those of the systems do not want us fooling around with the system. On the other hand, this overlay area is an ideal place to put programs which do not use up any room in the user RAM and which are practically invisible to the system. But how can one make use of this area?

The answer is surprisingly simple. There are only two instructions needed: LD A,<code> ; RST 28H, <code> is a one byte constant which must meet the following conditions:

At least one of the higher order bits (bits 7, 6, 5) must be set, the lower order five bits (bits 4 through to 0) tell the system where in the directory the needed module is located.

To make it easier to understand the following you should turn on your computer and use the DFS (Display File Sector) option of SUPERZAP to display File Relative Sector (FRS) 0 of DIR/SYS. Normally (depending on your particular PDRIVE setting) this is disk sector 170 on a single-sided, single density 40 track diskette.

FRS 0 is the GAT sector telling the system which tracks are formatted and which ones are allocated to files. It also contains the disk name, data, and the auto command (if used). FRS 1 is the HIT sector. This is used by the system as a hash table for quickly finding a certain file on the disk. Starting from the FRS 2 sector the directory entries begin. In the relative position 0 (top row) you always find the BOOT/SYS. If you turn to FRS 3 you will find DIR/SYS in the top row having relative position 1. The top row of FRS 4 will show the file SYS0/SYS (relative position 2) and so on. Once you have reached FRS 9 of the DIR/SYS file (relative position 7) - in the case of a disk that was formatted using the standard format parameter of DDGA=2, this is the end of the DIR/SYS file, you can start over again with FRS 2. Now you can count the entries in the second row, starting with relative position 8 (SYS6/SYS will be found there). Exactly these position numbers are the values you have to use in the constant <code> (low order 5 bits). As there will be only five bits decoded by the overlay loader you can only address up to 32 overlay modules this way actually only 30, since BOOT/SYS and DIR/SYS occupy the first two positions). Let me give you an example of an overlay call:

If you wanted to call SYS5/SYS (DEBUG), you'd find this entry at position 7 in the directory. If you remember to set one of the high order three bits (bit 7 in this case), your value for code should be 1000 0111B = 87H. So it is sufficient to execute the following instructions for an overlay call to DEBUG; LD A,87H ; RST 28H.

NEWDOS JKL OVERLAY (cont).

Exactly this method will be used to load our self written JKL module. You will find the details in Part III of this article, but before that I must say a few words on what conditions a /SYS module must meet in order to be loaded correctly by the overlay loader :

The module is an ordinary machine code routine which to reside in one piece on disk (no additional extents allowed). The format of the module is the normal load file format which for example is produced by EDTASM. There has to be a start address stated in the END statement. This address will be jumped to after loading the file via RST 28H. The file need not be positioned within the overlay area (of course, if you choose to have it load elsewhere, you must sure that that area of memory is somehow is 'Reserved', so that your overlay module won't overwrite another program already in memory). If you end your module with a RET instruction, a return will made to the calling routine (i.e. the adress on the top of the stack).

If what you have read encourages you to experimemnt, please, do so on a BACKUP SYSTEM!!!!. IT is quite likely that you will get some errors in the beginning. These errors will sometimes be 'honoured' by a destroyed system. SO BE CAREFUL!!!

**Function of the JKL module**

You'll find the source code of the JKL module at the end of this article. First the printer is initialised, so that the standard settings of 10 CPI and 12 dots linefeed will be used. After this the first line of the video RAM is loaded into the two buffers BUFTXT and BUFGRF. The buffer BUFTTXT will now be modified to contain only printable ASCII codes -i.e. graphic codes above 7FH will be replaced by 20H and control codes 00H - 1FH will be shifted up by 40H so they'll appear on the printer just like the character being diplayed by the TRS-80 character generator (the one normally supplied with the Radio Shack Model I lower case modification, which duplicates the upper case character set for ASCII codes 00H - 1FH). In the buffer BUFGRF all codes below 80H will be replaced by 80H.. This way we have a seperation of text and graphics.

Now the program functions in the following manner: If a line only consists of text codes, this text will be printed and a 12 dot line feed will be done. If there were graphics then first those graphics will be printed. As it is not possible to do a reverse line feed with the MX80 III, only the upper two blocks of that row are printed, then a one dot line feed is done. After this the text is printed and a seven dot line feed is done. In a third phase the lower blocks of the graphics are printed and a four dot line feed is done. This makes a full 12 dot high line of mixed graphics and text.

The whole procedure is repeated for all 16 video lines. Finally, the printer is reset to normal again and a return to the calling program (uaually DOS or BASIC) is done. Though this procedure of a JKL dump sounds quite complicated, it works suprisingly fast.

**Hints for the Installation of the Module to the NEWDOS/80 System**

After you have typed in the osource code using EDTASM or a similar program, save it as SYSJKL/CMD on a working diskette. Now make a copy of your (naked) system on a different diskette (COPY,0,1,,FMT,CBF,/SYS or similar). Boot this system. Now you have to instal a file at a particular position in the directory (see Part I of this article). We will use position number 1DH =29 in this example. To create a file entry in position 1DH proceed as follows:

(The method described here is not the most elegant one, but it is the safest). Create several files using the CREATE command (CREATE S010; CREATE S110 etc). Now use SUPERZAP's DFS option again to display DIR/SYS. Find the entry at position 29 (this will be found starting at FRS 7, byte 60H). Write down the name of the entry. If there is no entry at this position continue to CREATE files. Now boot your system again and rename that file at position 29 to SYSJKL/SYS.

Finally you can purge all those files that are not needed any longer. If you execute a DIR 0 now, you'll find that SYSJKL/SYS is present as a normal visible file. This does not look very proffesional for a /SYS module. Use SUPERZAP again to change the first two bytes of the directory of SYSJKL/SYS to 5FH,20H. Now your file will only be displayed by DIR 0 /SYS. Copy the file SYSJKL/CMD to SYSJKL/SYS onto your new system diskette and your all set.

NEWDOS JKL OVERLAY (cont).

The SYS module is now installed but your system will not recognise that it is there. In the MODEL I of NEWDOS/80 the normal JKL routine is located in SYS3/SYS, FRS 4, starting at byte 96H (byte 73H on the MODEL III version of NEWDOS/80). At this place we apply the call to our own module: LD A,9DH ;RST 28H, which is 3EH,9DH,EFH. Once you have zapped those three bytes your module will work.

```

00100 :*****
00110 :# JKL5V5/SAC #
00120 :# #
00130 :# #
00140 :# NOTE DUE TO SPACE LIMITS #
00150 :# ( AND SORE FINGERS ) NO #
00160 :# COMMENTS HAVE BEEN INC- #
00170 :# LUDED. #
00180 :*****
00190 :# DRG 4000H #
00200 START EXX #
00210 CALL INIT #
00220 LD BC,200H #
00230 CALL 60H #
00240 LD A,00H #
00250 CALL PRT #
00260 LD A,00H #
00270 CALL PRT #
00280 LD HL,3C00H-64 #
00290 LD (ZEIAD),HL #
00300 LD BC,16 #
00310 ZEILE PUSH BC #
00320 LD BC,64 #
00330 LD HL,(ZEIAD) #
00340 ADD HL,BC #
00350 LD (ZEIAD),HL #
00360 LD DE,BUFTXT #
00370 LDIR #
00380 LD BC,64 #
00390 LD DE,BUFGAF #
00400 LD HL,(ZEIAD) #
00410 LDIR #
00420 LD HL,BUFGAF #
00430 LD B,64 #
00440 BERGAF LD A,(HL) #
00450 CP 80H #
00460 JR NC,SKIP #
00470 N360 LD A,80H #
00480 LD (HL),A #
00490 SKIP INC HL #
00500 DJNZ BERGAF #
00510 LD HL,BUFTXT #
00520 LD B,64 #
00530 LD A,0 #
00540 LD (FLG),A #
00550 BERTXT LD A,(HL) #
00560 CP 20H #
00570 JR NC,N0CTL #
00580 ADD A,40H #
00590 JR CTLSF #
00600 GAFK LD A,20H #
00610 CTLSF LD (HL),A #
00620 UK INC HL #
00630 DJNZ BERTXT #
00640 JR WEITER #
00650 N0CTL CP 80H #
00660 JR C,UK #
00670 LD A,1 #
00680 LD (FLG),A #
00690 JR GAFK #
00700 WEITER CALL TEST #
00710 JP Z,FERTIG #
00720 CALL ESC5 #
00730 LD B,60 #
00740 LD A,0 #
00750 INDENT CALL PRT #
00760 DJNZ INDENT #
00770 LD B,64 #
00780 LD HL,BUFGAF #
00790 Z1GR LD E,0 #
00800 LD A,(HL) #
00810 LD D,A #
00820 BIT 0,A #
00830 JR Z,N01 #
00840 LD A,E #
00850 DR OFOH #
00860 LD E,A #
00870 LD A,D #
00880 N01 BIT Z,A #
00890 JR Z,N02 #
00900 LD A,E #
00910 DR OFH #
00920 LD E,A #
00930 N02 LD A,E #
00940 CALL PR3 #
00950 LD E,0 #
00960 LD A,D #
00970 BIT 1,A #
00980 JR Z,N013 #
00990 LD A,E #
01000 DR OFOH #
01010 LD E,A #
01020 LD A,D #
01030 N013 BIT Z,A #
01040 JR Z,N033 #
01050 LD A,E #
01060 DR OFH #
01070 LD E,A #
01080 N033 LD A,E #
01090 CALL PR3 #
01100 INC HL #
01110 DJNZ Z1GR #
01120 LD A,27 #
01130 CALL PRT #
01140 LD A,'A' #
01150 CALL PRT #
01160 LD A,1 #
01170 CALL PRT #
01180 LD A,00H #
01190 CALL PRT #
01200 LD B,10 #
01210 LD A,20H #
01220 IND2 CALL PRT #
01230 DJNZ IND2 #
01240 LD HL,BUFTXT #
01250 LD B,64 #
01260 ZTXOUT LD A,(HL) #
01270 CALL PRT #
01280 INC HL #
01290 DJNZ ZTXOUT #
01300 LD A,27 #
01310 CALL PRT #
01320 LD A,'A' #
01330 CALL PRT #
01340 LD A,6 #
01350 CALL PRT #
01360 LD A,00H #
01370 CALL PRT #
01380 CALL ESC5 #
01390 LD B,60 #
01400 LD A,0 #
01410 IND6 CALL PRT #
01420 DJNZ IND6 #
01430 LD B,64 #
01440 LD HL,BUFGAF #
01450 Z2GR LD A,0 #
01460 LD E,A #
01470 LD A,(HL) #
01480 LD D,A #
01490 BIT 4,A #
01500 JR Z,N011 #
01510 LD A,E #
01520 DR OFOH #
01530 LD E,A #
01540 N011 LD A,E #
01550 CALL PR3 #
01560 LD E,0 #
01570 LD A,D #
01580 BIT 5,A #
01590 JR Z,N015 #
01600 LD A,E #
01610 DR OFOH #
01620 LD E,A #
01630 N015 LD A,E #
01640 CALL PR3 #
01650 INC HL #
01660 DJNZ Z2GR #
01670 LD A,27 #
01680 CALL PRT #
01690 LD A,'A' #
01700 CALL PRT #
01710 LD A,4 #
01720 CALL PRT #
01730 LD A,00H #
01740 CALL PRT #
01750 FERTIG POP BC #
01760 DEC BC #
01770 LD A,B #
01780 DR C #
01790 JP NZ,ZEILE #
01800 CALL INIT #
01810 EXX #
01820 XOR A #
01830 RET #
01840 :
01850 TEST LD A,(FLG) #
01860 CP 0 #
01870 RET NZ #
01880 LD B,10 #
01890 LD A,20H #
01900 LEER CALL PRT #
01910 DJNZ LEER #
01920 LD HL,BUFTXT #
01930 LD B,64 #
01940 ZX LD A,(HL) #
01950 CALL PRT #
01960 INC HL #
01970 DJNZ ZX #
01980 LD A,27 #
01990 CALL PRT #
02000 LD A,'2' #
02010 CALL PRT #
02020 LD A,00H #
02030 CALL PRT #
02040 XOR A #
02050 RET #
02060 :
02070 INIT LD A,27 #
02080 CALL PRT #
02090 LD A,'A' #
02100 CALL PRT #
02110 LD A,0 #
02120 CALL PRT #
02130 LD A,27 #
02140 CALL PRT #
02150 LD A,' ' #
02160 CALL PRT #
02170 RET #
02180 :
02190 :
02200 :
02210 :
02220 :
02230 :
02240 :
02250 ESC5 LD A,27 #
02260 CALL PRT #
02270 LD A,'K' #
02280 CALL PRT #
02290 LD A,188 #
02300 CALL PRT #
02310 LD A,1 #
02320 CALL PRT #
02330 RET #
02340 :
02350 PR3 PUSH BC #
02360 LD B,3 #
02370 PTG CALL PRT #
02380 DJNZ PTG #
02390 POP BC #
02400 RET #
02410 :
02420 PAT PUSH HL #
02430 LD HL,37E8H #
02440 CHK 7,(HL) #
02450 JR NZ,CHK #
02460 OUT (PAPORT),A;MOD #
02470 LD (HL),A;MOD I #
02480 POP HL #
02490 RET #
02500 :
02510 FLG DEFB 0 #
02520 ZE1AD DEFW 0 #
02530 BUFTXT DEFB 64 #
02540 BUFGAF DEFB 64 #
02550 PAPORT EQU OF8H #
02560 END START

```

Notice:- The Inboard Disk and Printer Interface for the System-80 is now available. The cost, including fitting and testing, is \$150:00 for Sydtrug members and \$180:00 for non-members.

Contact Errol Rosser on (02) 709-7646.

---

Wanted:- CTR-80 or CTR-80A cassette recorder(s), any condition considered.

Contact Geza Dujmovich on (047) 74-1685.

---

#### DISCLAIMER

No Patent liability is assumed with respect to the use of the information contained herein. While every precaution has been taken in the preparation of this publication the Sydney TRS-80 Users Group or it's appointed office bearers assume no responsibility for errors or omissions. Neither is any liability assumed for damages arising from the use of any information contained herein.

---