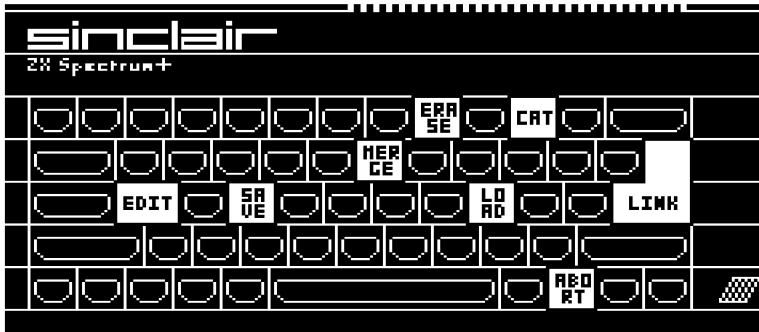




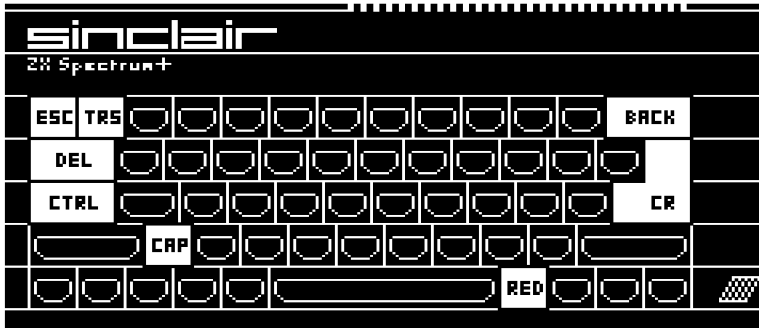
LINK

by
Enrico Maria Giordano
 for
ZX Spectrum 48/128K



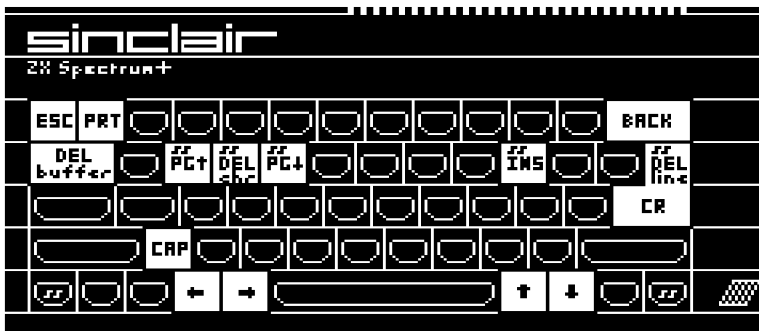
EXTERNAL COMMANDS

- | | |
|-------------------|-------------------------------------|
| ENTER | -> go into LINK |
| EDIT | -> go into EDIT |
| S (= "save" key) | -> save buffer to disc |
| L (= "load" key) | -> load file from disc into buffer |
| T (= "merge" key) | -> merge file from disc into buffer |
| 7 (= "erase" key) | -> erase file from disc |
| 9 (= "cat" key) | -> catalogue disc |
| ARROW DOWN | -> abort |



LINK COMMANDS

- | | |
|-------------|-------------------------------------|
| BREAK | -> backspace |
| ENTER | -> carriage return |
| CAPS LOCK | -> capitals lock |
| EXTEND MODE | -> control (CTRL) |
| ARROW UP | -> re-dial telephone number (Hayes) |
| DELETE | -> delete buffer |
| INV VIDEO | -> transmit buffer |
| TRUE VIDEO | -> exit |



EDIT COMMANDS

- | | |
|------------|-------------------------------|
| BREAK | -> backspace |
| ENTER | -> next line |
| CAPS LOCK | -> capitals lock |
| ARROWS | -> cursor left right up down |
| ss+Q | -> previous page |
| ss+E | -> next page |
| ss+I | -> insert/overtyp |
| ss+W | -> delete character |
| ss+ENTER | -> delete line |
| DELETE | -> delete buffer |
| INV VIDEO | -> print buffer from cursor * |
| TRUE VIDEO | -> exit * Seikosha GP-50S |

LINK

ZX SPECTRUM 48/128K

by
Enrico Maria Giordano
Via Trionfale 125
00136 Roma
Italia
tel (06)3252069

This text file has been translated in English
from Italian by:

Carlo Malantruccio
Viale di Villa Pamphili 59
00152 Roma
Italia
tel (06)5812334

WHAT IS LINK?

LINK is a public domain communication program written for the Sinclair ZX Spectrum by Enrico Maria Giordano, an Italian programmer.

It allows to link Spectrum (both 48K and 128K models) - through RS232 serial interface - with Bulletin Board Systems (by means of modem and telephone line) or with another computer (like Z88 or any computer equipped with comms program and RS232 serial interface) at 1200/1200 bps.

LINK exists in two versions for two different RS232 serial interfaces:

- * RS232 by Enrico Maria Giordano
- * Spectrum 128K/+2 RS232

The former is a serial interface suitable for all Spectrums 48K: it's very simple and economical, in a subsequent article I will explain how it works and how you can build it.

The main features of LINK are its screen display of 64 columns x 32 rows, instead of the usual 24 rows (every character measures in fact 6x4 pixels, instead of the usual "Tasword-like" 8x4 pixels) and its internal RAM buffer of 32K (from address 25000 to 57767, that is 512 rows or 16 pages of text).

HOW DOES LINK WORK?

LINK is structured in three main sections:

1. drive handling (SAVE LOAD MERGE ERASE CAT)
2. comms program (LINK)
3. full screen editor (EDIT)

Drive handling is made from Basic, so everyone could easily modify the syntax for his own drive interface (Microdrive, Opus Discovery, Disciple/+D etc.).

Both LINK and EDIT are written in machine code: they don't make any call to Spectrum ROM, but use original routines written by Enrico Maria Giordano.

LINK is based on a loop reading the keyboard and the serial interface: if a key is pressed then the relative function is executed, whereas if a character is coming in from the serial interface then it is displayed on the screen and stored in the internal RAM buffer.

EDIT is based too on a loop reading the keyboard: whenever a key is pressed, the corresponding function is executed.

For those lovers of technical specifications, I'll give a brief description of LINK/EDIT functions and general routines.

LINK FUNCTIONS

TCH
transmit the character corresponding with the pressed key

CPS
lock capital letters

RPT
repeat automatically the last telephone number (Hayes)

CNT
send a control character by pressing EXTEND MODE plus a key

TRS
transmit buffer in ASCII protocol

NEW
clear buffer

EXT
exit from LINK

EDIT FUNCTIONS

CHA
display and store into the buffer the character corresponding with the pressed key

CPS
lock capital letters

BS
delete the character to the left of the cursor

CR
go to the next line

PGU
go to the previous page

PGD
go to the next page

INS
toggle insert/overtyping mode

DEL
delete the character pointed by the cursor

DLL
delete the line pointed by the cursor

NEW
clear buffer

PRT
print text on Seikosha GP-50S printer

EXT
exit from EDIT

GENERAL ROUTINES

TXT
display a character on the screen at defined row and column (64x32)

KBD
update a variable with the code of the pressed key; it's handled by Mode 2 Interrupts

SCU
scroll up the screen in text mode

SCD
scroll down the screen in text mode

PRN
print text on Seikosha GP-50S keeping the same screen format and characters (64x32)

TXD
send a character to the serial interface

RXD
receive a character from the serial interface

HOW TO USE LINK

Firstly, both 48K and 128K LINK versions must be loaded in 48 BASIC mode.

This is necessary because LINK uses the "printer buffer" (23296-23551) to store temporarily its stack, but that area is used too by Spectrum 128K/+2 to store permanently some system variables: so running LINK in 128 BASIC mode will soon cause the computer crash, whereas it will not occur in 48 BASIC mode.

Secondly, both 48K and 128K versions include a "trap error" routine - called in line 10 with RANDOMIZE USR 23760 - that prevent the program from stopping and returning to Basic in case of error (when BREAK key is pressed, for example).

However Opus Discovery owners have to remove that instruction, because it causes the computer crash whenever a drive error occurs (like "write protected", "no room on disc" or "file not found"); it doesn't occur with Disciple/+D because that routine doesn't trap the errors of those interfaces.

When the program stops with an error report, you can start it again with RUN 20 without losing any text in RAM buffer.

After loading LINK, on the left bottom corner will be displayed the cursor waiting for a command.

EXTERNAL COMMANDS

Well, now you can:

1. go into LINK: to do it press ENTER key
2. go into EDIT: to do it press EDIT key
3. access a drive function:
to do it press

key "S" to SAVE the buffer to disc
key "J" to LOAD a file from disc into the buffer
key "T" to MERGE a file from disc with the existing buffer
key "7" to ERASE a file from disc
key "9" to CATalogue the disc

and you will see the command displayed on the left bottom corner: type now the suitable filename and press ENTER to execute the command.

If you have pressed the wrong key, you can abort the command by pressing the "arrow down" key (note: Opus Discovery owners who removed RANDOMIZEUSR 23760 in line 10 will exit in Basic: to restart simply RUN 20); in a similar way, if you have typed the wrong filename, you can erase it by pressing the "edit" key.

LINK COMMANDS

BREAK backspace
ENTER carriage return
CAPS LOCK capitals lock
ARROW UP re-dial telephone number
 (Hayes)
EXTEND MODE control (CTRL)
INV VIDEO transmit buffer
DELETE delete buffer
TRUE VIDEO exit

Here is the way to connect with a BBS.

After connecting Spectrum RS232 serial interface with modem and telephone line, load LINK and go into LINK section by pressing ENTER key. Now type ATD followed by the telephone number (don't worry if the first letter "A" will not be displayed on the screen: it's all right), then press ENTER: you'll hear the modem dialling the telephone number, and if a carrier will be detected you'll see displayed on the screen the fateful "CONNECT".

If you'll see otherwise displayed "NO CARRIER", press the "ARROW UP" key: the modem will re-dial automatically the telephone number.

Please note that all text displayed on the screen during the connection will be automatically stored in the internal 32K RAM buffer ("download"): exceeding that limit you'll lose the early text to store the new one.

If you wish to send a text to the BBS ("upload"), you have first to load it from disc or type it into the EDIT section: please note that every text will be transmitted in ASCII format (neither X-modem nor other protocols are available).

In the former case you have to exit from the LINK section by pressing TRUE VIDEO key and load the wanted file from disc by pressing the key "J" followed by filename and ENTER: please note that the buffer content will be deleted, so if you want store it you must first save it to disc by pressing the key "S" followed by filename and ENTER.

In the latter case you have to exit from the LINK section by pressing TRUE VIDEO key and go into the EDIT section by pressing EDIT key: now you

can type your text.

When the text to be sent is ready, go again into LINK section and press INV VIDEO key: the buffer (containing your text) will be transmitted to the BBS.

Sometimes the BBS asks for "control sequences", that is it has to receive from the computer some signals, which on an IBM compatible computer can be obtained by holding down CONTROL key (CTRL) and pressing another key (for example, Control-S to stop the text scrolling).

Well, inside LINK CONTROL key is replaced by EXTEND MODE key: however you must remember that you haven't to hold down this key, but to press it, then release it and press the other key.

EDIT COMMANDS

CAPS LOCK capitals lock
BREAK delete last character
ARROWS cursor left right up down
ENTER next line
ss+Q [<=] previous page
ss+E [>=] next page
ss+I insert/overtyping
ss+W [<>] delete character
ss+ENTER delete line
DELETE delete buffer
INV VIDEO print buffer from cursor
 (Seikosha GP-50S)
TRUE VIDEO exit

* ss = symbol shift

The EDIT section doesn't need any explanation: it's a very fast and flexible full screen editor that will allow you to type your messages off-line before linking the BBS.

Please note that the "print buffer from cursor" option (INV VIDEO) will work only with Seikosha GP-50S printer: to print a file through an Epson compatible printer read the following "Appendix B".

APPENDIX A: HOW TO ADD "SELECT DRIVE NUMBER" OPTION

As written above, all drive commands are handled from Basic: therefore is easy to add the "select drive number" option (for those lucky owners of twin drive systems); normally LINK uses drive no.1.

For example, to modify the LOAD command you have to rewrite line 60 in the following way (I'm using Disciple/+D syntax, but the method is the same for any other drive interface):

```
60 IF INKEY$="j" THEN CLS:
   INPUT "LOAD d";d;" "; LINE n$:
   RANDOMIZE USR 62129: LOAD dd;n$ CODE:
   RANDOMIZE USR 62098: RUN 20
```

NOTE: Opus Discovery owners have only to replace the first "d" of LOAD dd (not the numeric variable d) with a "*".

The same method is applied to the other commands (SAVE MERGE ERASE CAT).

APPENDIX B: HOW TO PRINT LINK FILES

As written above, if you own a Seikosha GP-50S printer you can print every text from EDIT section by pressing INV VIDEO key.

If you don't own that little printer, you can anyway print a text file loading the PRINT utility by Enrico Maria Giordano.

It's a small Basic program: line 0 contains a REM with m/c routine to calculate the text length; line 10 contains the control characters for an Epson compatible printer (to select left and right margins, full justification, skip over perforation and NLQ mode) and the instructions to prepare the binary output from the Spectrum: the control characters can be replaced to modify the print output, whereas the other instructions have to be replaced according to own printer interface.

On this last subject here are the needed instructions (you have to type the right one in line 10 after RANDOMIZE USR 23774):

```
* Disciple/+D ..... POKE @6,1
* Opus Discovery ... OPEN£3;"b"
```

NOTE: Opus Discovery owners have to add too in line 30 CLOSE£3 before RUN (to close the stream after printing) and modify line 10 by substituting PEEK 23804 and PEEK 23803 with PEEK 23811 and PEEK 23810: the instruction OPEN£3 in fact moves the Basic area up 7 bytes, modifying the addresses containing the "last text byte" position and causing consequently the text length miscalculation.

APPENDIX C: HOW TO CONVERT LINK FILES IN TASWORD 2 FILES

If you want to use text files received with LINK from a BBS or from another computer, you have to load them into a word processor: you all know that several word processors are available for the

Spectrum (Tasword 2, Tasword 3, The Last Word, The Writer, WordMaster etc.), but you all know too that the "standard" is TASWORD 2, because it was the first to appear on the market many years ago; so every subsequent word processor has been generally released with a "Tasword 2 conversion utility" allowing to modify and load Tasword 2 text files.

The LINK file format is like the TASWORD 2 one, that is every row of 64 columns is always filled up to the right end margin: however TASWORD 2 fills the gaps with spaces (CHR\$ 32), whereas LINK uses CHR\$ 0.

So Enrico Maria Giordano has prepared two utilities to convert files in both ways: they are named L2T (= Link to Tasword) and T2L (= Tasword to Link) and work replacing CHR\$ 0 with CHR\$ 32 and viceversa.

To use them you have simply to load the right utility and follow the screen prompts, after modifying - if necessary - the Basic syntax to suit your own drive interface.

APPENDIX D: HOW TO CONVERT LINK FILES IN FIDONET FILES

If you wish to transmit messages inside BBS of the International Fidonet Association you should remember that no empty lines are allowed: therefore LINK text files have to be converted so that every line contains at least one character.

For this purpose Enrico Maria Giordano has prepared a code named L2F (= Link to Fido) that inserts at the start of all empty lines one space (= CHR\$ 32).

To use it first merge to LINK Basic the following line:

```
95 IF INKEY$ = "f" THEN CLS:
   LOAD d1; "L2F" CODE:
   RANDOMIZE USR 58000: RUN 20
```

and resave the program with autostart at line 10.

NOTE: Opus Discovery owners have only to replace the "d" of LOAD d with a "*".

Whenever you have in the RAM buffer a text file to be transmitted (typed in EDIT or loaded from disc) you have only to press key "F" as external command: L2F code will be loaded from disc and automatically executed to convert LINK file stored in RAM in FIDO file.

Then go into LINK by pressing ENTER key and transmit the buffer to the Fido BBS by pressing INV VIDEO key.

RS232 INTERFACE

ZX SPECTRUM 48K

by
Enrico Maria Giordano
Via Trionfale 125
00136 Roma
Italia
tel (06)3252069

This text file has been translated in English
from Italian by:

Carlo Malantruccio
Viale di Villa Pamphili 59
00152 Roma
Italia
tel (06)5812334

The RS232 serial interface designed by Enrico Maria Giordano is suitable for all Spectrums 48K: it allows in conjunction with LINK - the public domain software programmed by the same author - to link Spectrum with Bulletin Board Systems (by means of modem and telephone line) or with any other computer equipped with RS232 serial interface and comms program (like Z88, for example).

The interface consists of two circuits - input and output - connected with the Z80 plus a power circuit: all the voltages are supplied by the Spectrum through the peripheral connector, so it isn't necessary external power supply.

The interface can read only one bit at a time, so the Z80 has to do all the work: parallelization of received data and serialization of transmitted data. It means that the interface really works in half duplex, even if practically there is no difference with full duplex.

Decoding of the port address is made with the reduced system used by the Spectrum, providing only one address byte's test; it means that the port address is obtained setting all the bits at 1 except A1: so the port address is 11111101 = 253.

Both input and output have the same address and bit: DO.

COMPONENT LIST

- n. 2 resistors 1 K0hm
- n. 2 resistors 2.2 K0hm
- n. 1 resistor 4.7 K0hm
- n. 2 electrolytical capacitors 4.7 uF 16 V
- n. 2 diodes 1N4001
- n. 1 Zener diode 4.3 V 1 W
- n. 1 transistor BC 109
- n. 1 transistor BC 179
- n. 1 I.C. 74LS04
- n. 1 I.C. 74LS27
- n. 1 I.C. 74LS125

CIRCUIT DIAGRAM

The circuit diagram consists of three sections:

1. TX DATA - output signal
from Spectrum to RS232 interface
2. RX DATA - input signal
from RS232 interface to Spectrum
3. -12V - power circuit -12V

The **TX DATA** section consists of:

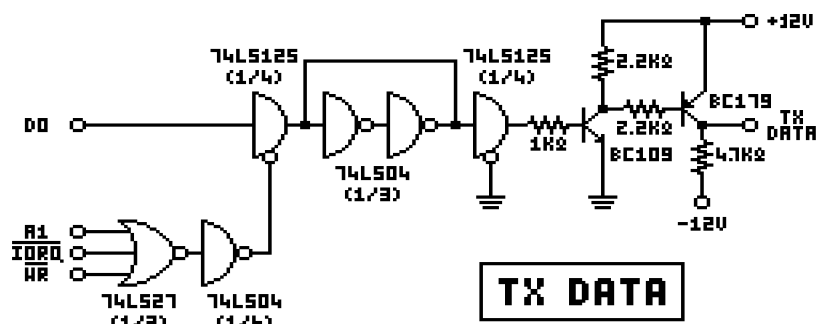
I.C. 74LS27 (1/3) and I.C. 74LS04 (1/6) decoding the address of the used port (253)

I.C. 74LS125 (1/4) linking in tristate the Z80 bus

I.C. 74LS04 (1/3) working as flip flop to store the output data

I.C. 74LS125 (1/4) working as buffer for the flip flop output

transistors BC 109 and BC 179 working as level translator to convert the output signal from TTL (0/+5V) to RS232 (+12V/-12V)

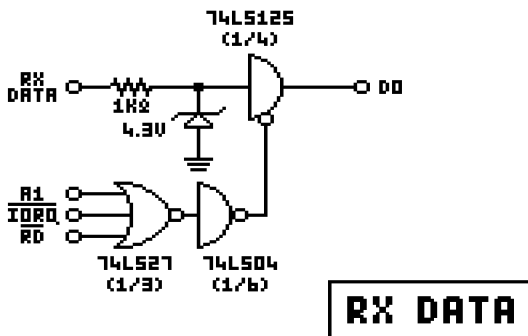


The **RX DATA** section consists of:

Zener diode 4.3V working as level translator to convert the input signal from RS232 (+12V/-12V) to TTL (0/+5V)

I.C. 74LS27 (1/3) and I.C. 74LS04 (1/6) decoding the address of the used port (253)

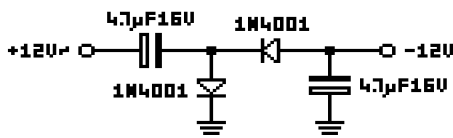
I.C. 74LS125 (1/4) linking in tristate the Z80 bus



RX DATA

The **-12V** section consists of:

two diodes 1N4001 and two capacitors 4.7 uF 16 V converting the 12V alternating current taken from the Spectrum through connector in -12V direct current, necessary for the TX DATA level translator



-12V

Note: the other required voltages - that is +12V for the TX DATA level translator and +5V for all the I.C. 74LS series - are taken directly from the Spectrum through connector.

**APPENDIX A:
HOW TO LINK SPECTRUM &
MODEM**

To link Spectrum 48/128K with a BBS (= Bulletin Board System) you need:

- Spectrum RS232 serial interface
- Spectrum comms software
- modem
- telephone line
- two cables

Spectrum 128K/+2 is equipped with its own RS232 serial interface; Spectrum 48K can be equipped with the Enrico Maria Giordano's RS232 serial interface described above.

Comms software is Enrico Maria Giordano's LINK, available in two versions suitable for 48K & 128K RS232 serial interfaces.

Modem must support CCITT V22 protocol (= 1200/1200 bps) and should be preferably Hayes compatible, that is support AT commands.

Finally two cables are necessary: one to connect modem with the telephone line (normally it's a cable with two RJ11 jacks at the ends, provided with the modem), the other to connect RS232 serial interface with modem.

That's how to build the latter: you need three wires and two male connectors, one DB-9 (48K) or BT serial (128K) for RS232 Spectrum end and one DB-25 for RS232 modem end.

SPECTRUM 48K RS232 PINS	MODEM RS232 PINS
TxD 2	2 TxD
RxD 3	3 RxD
GND 7	7 GND

SPECTRUM 128K RS232 PINS	MODEM RS232 PINS
GND 1	7 GND
TxD 2	2 TxD
RxD 3	3 TxD

**APPENDIX B:
HOW TO LINK SPECTRUM & PC**

To link Spectrum 48/128K with a PC IBM compatible you need:

- Spectrum RS232 serial interface
- Spectrum comms software
- PC RS232 serial interface
- PC comms software
- cable

That's how to build the cable: you need three wires, one male DB-9 connector (48K) or BT serial connector (128K) for RS232 Spectrum end and one female DB-25 connector for RS232 PC end.

SPECTRUM 48K RS232 PINS	PC RS232 PINS
TxD 2 _____	2 TxD _____
RxD 3 _____X_____	3 RxD _____
	4 RTS _____
	5 CTS _____
	6 DSR _____
GND 7 _____	7 GND _____
	8 DCD _____
	. _____
	. _____
	. _____
	20 DTR _____

SPECTRUM 128K RS232 PINS	PC RS232 PINS
GND 1 _____	
TxD 2 _____ _____	2 TxD _____
RxD 3 _____ _____	3 RxD _____
	4 RTS _____
	5 CTS _____
	6 DSR _____
	7 GND _____
	8 DCD _____
	. _____
	. _____
	. _____
	20 DTR _____

As you can see, you have to bridge together pins 4-5-8 and pins 6-20 at the PC end.

If you don't own comms software for the PC, you can use the following DOS commands to transmit and receive ASCII files:

1. from PC to Spectrum:
TYPE filename > COM1
or
COPY filename COM1
2. from Spectrum to PC:
TYPE COM1 > filename
or
COPY COM1 filename

Please note that you must send too an "End Of File" signal (CTRL Z) to indicate the end of trasmission from Spectrum to PC: to do this type

on Spectrum keyboard EXTEND MODE Z.

NOTE: the DOS command CTTY COM1 - typed on PC keyboard - allows to use Spectrum as PC terminal.

APPENDIX C: HOW TO LINK SPECTRUM & Z88

To link Spectrum 48/128K with a Z88 you need the same items listed for a PC IBM compatible: please only note that the cable is different and that Z88 has its own comms software in ROM (application Terminal and popdown Import/Export).

That's how to build the cable: you need three wires, one male DB-9 connector (48K) or BT serial connector (128K) for RS232 Spectrum end and one male DB-9 connector for RS232 Z88 end.

SPECTRUM 48K RS232 PINS	Z88 RS232 PINS
TxD 2 _____	2 TxD _____
RxD 3 _____X_____	3 RxD _____
	4 RTS _____
	5 CTS _____
GND 7 _____	7 GND _____
	8 DCD _____
	9 +5V _____

SPECTRUM 128K RS232 PINS	Z88 RS232 PINS
GND 1 _____	
TxD 2 _____ _____	2 TxD _____
RxD 3 _____ _____	3 RxD _____
	4 RTS _____
	5 CTS _____
	7 GND _____
	8 DCD _____
	9 +5V _____

As you can see, you have to bridge together pins 4-5-8-9 at the Z88 end.

Z88 Panel popdown must be set in the following way:

```
Transmit baud rate 1200
Receive baud rate 1200
Parity None
Xon/Xoff No
```

Remember that when you insert the cable into the

Z88 RS232 its Transmit/Receive baud rate should be set at 9600 (default), otherwise the Z88 keyboard could not work: after inserting the cable change baud rate from 9600 to 1200.

Z88 User Guide (2nd edition) explains in Appendix C (pag.192) how to store received files and how to transmit stored files.

"Material received by the Terminal application can be spooled to a file S.Sgn automatically by typing []+S before commencing the transfer. Spooling can be disabled by typing []-S.

A convenient way of sending a file to a remote machine using the Terminal is to send it to the CLI with the following prefix:

```
£V
.J
... remaining text
```

The prefix can be added by editing the file using PipeDream, and then saving the file as plain text. It can then be executed with the Filer Execute command, <>EX.

The £V invokes the Terminal, and the .J prevents any special characters in the file, such as £,;', |, and '"', from being interpreted by the CLI."

Another simple way of transmitting a text file from Z88 to Spectrum is to load it in PipeDream and use the "Print" option <>PO: before printing, reformat text at 64 column width and set the Options Page in the following way:

```
Page length 0
Margins: Top 0
        Header 0
        Footer 0
        Bottom 0
        Left 0
```

If you wish an easier way to transmit/receive files, then buy Z-Term (by Wordmongers), a general purpose terminal package designed for communicating between Z88 and the outside world: it has many advanced features, between which a "file capture" facility.

Finally, a note about CR (= carriage return) interpretation made by the two comms programs.

LINK interpretes every received CR (chr13) as CR+LF (chr13+chr10): so if you press ENTER key on Z88 keyboard the LINK cursor displayed on Spectrum screen will go to the beginning of the next line.

Terminal interpretes every received CR only as CR:

so if you press ENTER key on Spectrum keyboard the Terminal cursor displayed on Z88 screen will go on the beginning of the same line.

Typing on the two keyboards could help you to verify the right connections of the cable and the correct working of the comms programs.

NOTE: as you'll probably notice, Spectrum 128K TxD/RxD connections are reversed compared to Spectrum 48K ones: why?

Well, in general communication terms devices like computers, printers, modems etc. are considered as DTE (= Data Terminal Equipment) or DCE (= Data Communication Equipment).

Normally a computer is considered as DTE, so pin 2 TxD (= Transmitted Data) is an output line, whereas pin 3 RxD (= Received Data) is an input line. Consequently to link two computers you need a crossed wiring (2-3 & 3-2), whereas to link a computer with a modem you need a direct wiring (2-2 & 3-3).

Well, Spectrum 128K/+2 RS232 is considered as DCE (as you all know, Sir Clive Sinclair didn't ever like the word "standard"): so its pin 2 TxD is really an input line and its pin 3 RxD an output line!

Now you can understand the reason of the reversed wiring.

TIP: could any expert user of Kemsoft PCB Designer design a printed circuit board, component layout and circuit diagram of the Enrico Maria Giordano's RS232 serial interface using that software package?

This handbook was written and paged up using a Spectrum 48K and DTP Pack by P.C.G. (61 School Street, Barrow-in-Furness, Cumbria LA14 1EW, England) then printed using a Star LC24-10.

ERRATA CORRIGE

=====

On the LINK manual you read:

"All the text displayed on the screen during the connection will be automatically stored in the internal 32K RAM buffer ("download"): exceeding that limit you'll lose the early text to store the new one."

It's wrong: when you exceed the 32K limit no other character will be stored in memory and you will see the cursor size reduced from 6x4 pixels to 2x4 pixels.

"Type ATD followed by the telephone number (don't worry if the first letter "A" will not be displayed on the screen: it's all right) ..."

Well, what you will see on the screen when you type ATD and the telephone number depends on your modem: you could even see nothing!

So if you want to be sure about what you are typing go first into EDIT and type ATD followed by the telephone number, then exit and go into LINK: by pressing INV VIDEO key you will transmit the internal buffer to the modem so dialling the desired number.

"To use them (L2T and T2L) you have simply to load the right utility and follow the screen prompts, after modifying - if necessary - the Basic syntax to suit your own drive interface."

Well, L2T and T2L have been changed: now they aren't Basic programs but code files starting at address 58000 (like L2F). This modification has been made to accomodate to a standard for any "executable file" inside LINK, that will be loaded - only when needed - and run by RANDOMIZEUSR 58000.

So we recommend to merge to LINK Basic the following line:

```
98 IF INKEY$ = "x" THEN CLS:
   INPUT "EXE "; LINE n$:
   LOAD d1; n$ CODE:
   RANDOMIZEUSR 58000: RUN 20
```

and resave the program with autostart at line 10.

NOTE: Opus Discovery owners have only to replace the "d" of LOAD d with a "*".

Whenever you have in the RAM buffer a text file to be converted from LINK to TASWORD 2 (or from TASWORD 2 to LINK), simply press

key "X" as external command and type "L2T" (or "T2L") at the EXE prompt: the code will be loaded from disc and automatically executed to convert LINK file stored in RAM in TASWORD 2 file (or viceversa).

Then save the converted file on disc in the normal way.

XASCII

=====

XASCII is a coding system created by Enrico Maria Giordano that allows to transform a binary file into an ASCII file and viceversa: it allows to exchange all kinds of files like normal ASCII messages.

You can normally transmit to a BBS in a text message only ASCII characters (CODE 32-126) but neither executable programs nor data files, because they could contain numbers ranging from 0 to 255.

Well, XASCII protocol allows to transform a file into a sequence of ASCII characters that can be handled like a normal text message: XASCII really means "Xfer ASCII", i.e. ASCII protocol for file transfer.

Now you can prepare messages containing both programs and instructions and send them to a BBS: a file coded XASCII appears like a sequence of lines - each 60 characters long - containing strange incomprehensible characters.

XASCII's main features are the following ones:

1. Coding of groups of 6 bits from a byte with XRLE compression.
2. Fastest processing entirely in RAM, independent from storage medium.
3. Simple check of file integrity by checksum.
4. Output format on 60 columns, compatible with Enrico Maria Giordano's comms program LINK.
5. Automatic relocation of files to their original start addresses after decoding.

The package to handle XASCII files consists of 6 programs:

C2X

It allows to transform a CODE file into a XASCII file.

X2C

It allows to transform a XASCII file into a CODE file.

B2C

It allows to transform a BASIC file into a CODE file.

C2B

It allows to transform a CODE file into a BASIC file.

NOTE: the last two utilities allow to transform a BASIC file into a XASCII file using an intermediate step: first from BASIC to CODE, then from CODE to XASCII (and viceversa).

XBL

It allows to break a CODE file longer than 8192 bytes into

XASCII compatible blocks of 8192 bytes.

XLD

It allows to reconstitute a CODE file from XASCII compatible blocks of 8192 bytes.

NOTE: the last two utilities allow to handle CODE files longer than 8192 bytes.

INSTRUCTIONS

C2X

1. Load and run C2X.
2. Type CODE filename at the FILE prompt and press ENTER. Please note that the file must not to exceed RAM areas 16384-23551 and 25000-65249 and must not to be longer than 8192 bytes (otherwise you have to use XBL).
3. Type XASCII filename at the MSG prompt and press ENTER.

X2C

1. Load and run X2C.
2. Type XASCII filename at the MSG prompt and press ENTER.
3. Type CODE filename at the FILE prompt and press ENTER.

B2C

1. Load BASIC program to be transformed into CODE file. Please note that the file must not to be longer than 16384 bytes.
2. Load B2C CODE and RANDOMIZE 60000.
3. Type CODE filename at the FILE prompt and press ENTER.

C2B

1. Load and run C2B.
2. Type CODE filename to be transformed into BASIC program at the FILE prompt and press ENTER.

XBL

1. Load and run XBL.
2. Type CODE filename to be broken into blocks of 8192 bytes at the FILE prompt and press ENTER.

XLD

1. Load and run XLD.
2. Type CODE filename to be reconstituted from blocks of 8192 bytes at the FILE prompt and press ENTER. Please note that the blocks must have the same filename followed by a progressive number.
3. Type the whole number of the blocks to be reconstituted and press ENTER.