Einstein Magazine

No. 85

Einstein Magazine

EINSTEIN MAGAZINE & All Micro News No. 85
****
published for users of Einstein (and other) computers
by Steam Computer Society. Chief Editor and Publisher:-
A E Adams, Ivy Cottage,Church Road,New Romney,Kent. TN28 8TY
****

NEWS FROM NOWHERE

We share with the other groups for long-discontinued 8-bit computers a steadily diminishing membership, and a reducing income which is now insufficient to fund a printed magazine.

All such groups are nowadays only able to produce a regular magazine for users with their active support, by eating up their reserves of cash -- except that we don't have any! -- by eking out their dwindling income by cutting costs in all directions, or by subsidies from one or more individuals in the form of raw materials, equipment & vast amounts of time.

One such subsidy is access to commercial printing equipment, which we enjoy in return for a lot of my highly skilled time being donated to the community as the unpaid parish printer.

A lot more of my time was invested in finding a workable method of converting reels of glossy commercial web-offset magazine paper into A4 sheets on a cottage-industry scale, & then printing the magazine on them when they are way outside the design parameters of both printing machine and ink. The resulting methodology is effective, but very time-consuming.

Two economies that we exercise are systematic recycling of incoming envelopes that are still in good enough condition to re-manufacture them into magazine-sized envelopes, and use of "clean one side" coloured paper for magazine covers.

The glossy magazine paper was to have been our standard for magazine & newsletter from now on, but Ted has paid for some copier paper for the newsletter, & the unexpected generosity of another member has paid for more of it for the magazine too. This is fortunate, since other demands on my time are already heavy, & look like escalating fast in the very near future. The copier paper is supplied cut to size, and the printing machine is designed to use it, so it dramatically reduces magazine production time. The down side is that we get fewer pages in the first postal weight step using copier paper, so we'll still use the glossy paper if time permits.

New coloured paper is rather costly stuff, so for this issue we're trying the experiment of printing the coloured cover sheets "wrong way out" - i.e. with the "used" side out, and the cover design overprinted on it. If successful, this will potentially give us two more usable pages in the magazine without incurring any additional paper or postage costs.

CONTENTS

THE WOEFULLY WONKY WORLD WIDE WONDER-WEB

Supporters of Louise Woodward patiently awaited the Judge's decision through the internet on the computer in their local pub, but the local newsboy on his pushbike read them the result first from the Birmingham Evening Post. The internet had crashed but the news got through on the steam telegraph!
****

LETTER from Thomas Spence, Edinburgh

As a passive member of the Einstein User Group for the past 10 years or so, I have decided that it is time to move on. About a year ago I bought a modern new IBM Aptiva computer (which is now outdated), an Epson Stylus Colour Printer (which is now outdated), and an Emerald Siscan 24 flatbed scanner (which I have no doubt is now outdated). I hope to use the scanner to transfer my Einstein material to my IBM, not being proficient enough to transfer data between them.

I have read my Einstein magazines avidly over the years, but I was unable to absorb enough technical information to become a "Boffin". I am sad to part with something that has become part of my life, but you can't stop progress.

I have passed on my "Albert" to my grandchildren to play with for its remaining lifetime, but I'm afraid they think that it is an antique, as all the school computers are the most up-to-date as money can buy.

All the best from a Canny Scot. Thomas Spence.

UPDATE FROM DUNCAN ELVIN -- HARD DISKS AND OTHER MATTERS

You can obtain various bits from RS Components or Farnell. Part nos. for Farnell (Tel. 0113 263 6311) are as follows:

5 Pin 'Domino' DIN Plug	610-0520	(1.30 ea.).
60 Way IDC Bump & Clip Pol. Plug	636-6041	(3.23 ea.).
60 Way Ribbon Cable	148-300	(1.77 per ft.)
40 Way Ribbon Lead	207-214	(4.64 ea.)
Power Y Adapter	231-150	(4.75)

The serial terminal idea will work with any serial terminal, but monitors (i.e. VGA etc.) have completely the wrong type of input. But a monitor connected to a PC with a serial port and a copy of Procomm (or other terminal program) running will do just nicely.

You are correct in your understanding: IDE-XT is a subset of IDE (the other being IDE-AT). Not many IDE-XT (relative to IDE-AT) drives were made, and they are hard to obtain. If you can get one, in theory it does simplify the interfacing (as it only has a 8-bit bus) but in practice it is still easier for the unskilled user to use the GIDE kit (which only works with IDE-AT drives) rather than building a simple circuit on veroboard.

Unfortunately I do not have the articles or the GIDE instructions. What I have is reports of the kit being used on an EPSON computer (in file GIDE.TXT ***). The articles can be obtained from TCJ (The Computer Journal) for a small renumeration (to cover postage etc.). ATA means the same as IDE-AT. Unfortunately, there is no universal way to identify a drive as an IDE-AT, but it should have a 40 way IDC header (but so has a IDE-XT). You could check with the manufacturer or in a freeware list such as TheRef (THEREF43.ZIP +++). New drives can be obtained from many sources: see Computer Shopper, PC Mart, second-hand check out the Stafford show.

The price I quoted for the GIDE was the price from TCJ in the USA who would make a quantity purchase. I have contacted Tilmann. His response is enclosed in file TILMANN.TXT ***.

[[*** = awaiting publication here if you're interested]]
[[+++ = Ted Cawkwell (our Technical Supremo) has a copy.

The GIDE is just an interface and so it can handle any IDE-AT drive made, but in practice you should be looking at small drives (maximum 200Mb) as CP/M can only handle logical drives upto 4Mb, which would give us 25 logical drive partitions. As to the practical problems of using hard

discs, you have to partition them up into small chunks, otherwise they cannot be used (the allocation vector gets too big for a start) but that is fairly simple. You just adjust the number of system tracks you tell the O/S you have, to get it to skip over previous partitions.

Most CP/M utilities will work just fine, but backing up is a problem. Mostly I don't bother (slapped wrist!), I just save anything really important to floppy. There is no way to have tree of directories, like you have on the PC under MS-DOS, but you can use the "user number" scheme to give a one level 'tree'. ZCPR extends this by giving "user numbers" names.

****

WHY DO I? (WHY DON'T YOU?)
by Stephen Potts

Hello to all who read this, do let me tell you why it really is worth making the effort to join in and support the group whose purpose is to support you.

It seems to me we have some very promising changes, with Tony sorting out the paper and printing, and with Ted's new Technical Development Group and newsletter. Thank you for making such a difference. I try to keep my end up by making people aware that there is still life in the EINSTEIN yet, especially at Stafford and various other computer shows.

There are several 8BIT computers out there that are still perfectly serviceable for the job they were originally intended to do. One such is the AMSTRAD I'm involved with, and there are several clubs for the CPC 6128 and the PCW types. I belong to WACCI and UAUG. These both have regular magazines and good software services. WACCI is undergoing a problem at the moment with a fruitless power struggle in which some of the membership are pushing to make it a PC magazine instead of concentrating on AMSTRAD. This is dodgy ground, as we in the EINSTEIN user group know, since we lost out in exactly the same way a few years ago. As PC magazines fill the newsagents from one end to the other I see it as foolish in the extreme to debase our user groups & alienate the very person the user group is for. One chap Brian Watson is publishing a magazine called 8BIT. This is to support and encourage use of our type of machine. 8bit, z80 based, 3inch discs, etc. <<Ch Ed: Sadly, it's ceasing publication soon>>

I support and encourage the use of these machines because they work well and the internal programming is (although initially hard to grasp) soon second nature, and then you can design your own programs to suit you, and even publish them in a Homegrown PD.

In case you think I am biased, let me tell you I have a PC 486 DX2 66, and any of my PCW's, CPC's, or EINSTEINS knock it into a cocked hat for word processing. Do I need to buy a Formula 1 Williams to nip to TESCO's? "NO!" So when I want to write I use my PCW. The PCW9512 uses 3 inch and 3.5 inch discs, both in 720k format so I have lots of shuffling to get across formats to say EINSTEIN format. I'm dyslexic, so I have trouble spelling. So when I've finished writing I press F7 and spellcheck. Then to print press P....

Good, isn't it? On the 486 I first have to boot Dos... then Windows...launch applications... launch Word... then double click on start .. then a totally bewildering array of options loses you before a character is even typed. To get a letter printed is like pulling teeth..

The word processor I use and like best is LOCOSCRIPT. I have used it for a long time and the various types over the years from LOCO 1 through 2 and 3 now (May 1997) LOCO 4 is out. The only thing I have against it is it isn't pure CP/M, as it carries its own operating system that is almost the same. I suppose the same could be said for AMSDOS on the CPC, as this too is a front for the CP/M system monitor.

The way I do things involves a lot of swapping around the various types of disc formats used on different models within the frame of CP/M.

First I write the piece, then spell-check it, then print it out, all on my Amstrad PCW-9512. I save to 3inch 720k disk and 3.5inch disc, then I use the LOCOSCRIPT "Make ASCII" option. This makes the text OK for most word processors. I then take the 3inch disk and put it into my Amstrad PCW-8512. This has drives of two different types: A 720k B drive and an 180k A drive. Then over to my CPC 6128, and using WACCI disc PD82 CP/M utilities I alter the drive parameters to that of the Einstein, and save the file to the Einstein formatted 3 inch disk (or 3.5 inch disk) to send off.

ASCII is a plain text form that is truly a godsend when using different makes and models of computer, the transfer from one computer to another is possible due to Gary Kildall who was the head of Digital Research and came up with CP/M. This operating system has its opponents, but these can only be using it on one machine and comparing it to the operating system specifically designed for that model, to which a compromised transportable standard must come second by all the laws of nature.

There are lots of reasons for supporting 8BITs of varying types, not least I can just manage to follow the m/c code numbers with 8bits and double registers giving 16 bit numbers. If you wish to alter ,,,poke amend peek in memory to alter programs it is possible. You ought to try it with CoralDraw and see how you get on..

So there you have it. I support several 8 bit computers and CP/M, but don't follow Microsoft blindly, as you can achieve perfectly good work without Bill Gates having an open door to your bank account.

If you can use your computer at all, do think about what you would like it to do and make it do it. I still cannot believe how many people use their computer regularly and have not setup their boot disk to set the function keys to what they want.

Be creative. It's not hard to achieve. Get a book from the library on BASIC programming. It doesn't have to be specific to any computer make, they should all have the same core of commands or words. With very little practice you can create a question and answer program, then go on from there and you can do almost anything. Do let others see what you are doing and socialise with others in your area, or hold an evening of brain storming with new friends. I wish clubs would encourage local area meetings or publish a distribution map.

Do remember to write instructions in a README.TXT file, and if you like submit it to a PD library. Remember, it doesn't need to be a program. It could be a book or helpful hints. Schumacker's Ferrari may look sleek, but what is it like at towing a caravan.

There is a wealth of information in the Einstein Magazine back issues, & I have ordered a set from Tony for reference and type-ins. I will try to keep coming up with new ideas for those brave enough to try them. Please do remember we all do this voluntarily and are all unpaid, so do appreciate this is not a business, & therefore do treat us accordingly.

If you require an answer or info do include a SAE and an extra stamp. Please be patient, we are here to help out but things can become a little demanding. Personally I travel around the shows displaying my EINSTEIN like a classic car and it costs me quite a bit. Recently I stood a show and lots of people asked for me to send them things, so I was well out of pocket. So if I ask for a donation towards out of pocket costs, please don't be offended.
See you at the next show?   Steve Potts.

<<Ch Ed: You can chat with Steve -- or better still help him with the stand -- at Stafford and various other shows where he demonstrates the Einstein, or he'll be glad to hear from you at 85 Thorold Ave, Cranwell Village, Lincs, NG34 8DS >>

****

INTRODUCTION TO AN INTRODUCTION TO MACHINE CODE GRAPHICS (2)
by Dave Salvage

I get the impression that the User Group is picking up a little bit over the last few months. Is that the case, or am I living in a fantasy world? Nevertheless, many thanks for keeping it going. I shall try and do my bit now in return for the immense amount of knowledge and enjoyment I have gained from the User Group in the last ten years. Anyway, welcome back to this series of introductory articles.

The FLAG register (F) is very important, and useful in MC programming. It has eight bits, of which four are of use to us since they can be tested for their state (set or reset). These bits are:

0	Carry (C)
2	Parity/Overflow (P/V)
6	Zero (Z)
7	Sign (S)

The carry bit is used to indicate whether an addition or subtraction has resulted in a carry or borrow. It is also used as a ninth bit for SHIFT and ROTATE operations (see below).

The parity bit has several functions, which will be mentioned when (and if) appropriate later on in this series.

The zero bit is set to 1 if the value of a byte which has been calculated, or is being transferred, is zero. It is also set when comparisons indicate a match. Otherwise, it is reset to 0.

The sign bit indicates the most significant bit of a result or a byte being transferred (bit 7), with it set to 1 if bit 7 is 1 which usually means negative, and it reset to 0 if bit 7 is 0 which usually means the number in the byte is positive.

These flags can be tested by instructions, usually in conjunction with a conditional JUMP to elsewhere in the MC program. More on this as we encounter such instructions.

SHIFT and ROTATE are instructions which affect only the register to which they are applied plus the FLAG register.

Pure rotation of a register, either to the left or the right simply moves the contents of each bit of the register to the left or right by one bit and transfers bit 7 to the carry flag and the carry flag to bit 0 or bit 0 to the carry flag and the carry flag to bit 7 respectively. Example are RR C, RL H.

The registers can also be rotated with "branch carry". In this instance, rotation to the left moves each bit along one to the left, with bit 7 moving to both the carry flag and bit 0. Rotation to the right moves each bit to the right with bit 0 moving to both the carry flag and bit 7. Examples are RLC D, RRC B.

Shifts of the registers have similar effects. The bits are moved one position to the left or right, but there is no rotation. An arithmetic shift left, e.g. SLA C, puts bit 7 into the carry bit and resets bit 0 to 0. This has the effect of multiplying by two, provided the contents of the carry bit are not forgotten. An arithmetic shift right, e.g. SRA D, puts bit 0 into the carry bit and replaces bit 7 with itself so bits 6 and 7 are the same. There is also a logical shift right, e.g. SRL E, where bit 0 is placed in the carry bit and bit 7 is reset to 0.

The Z80 processor has specific commands to test any bit of any register (well almost). The result of the test is stored in the zero bit - set to 1 if the result is zero, and reset to 0 if the result is 1. Examples are BIT 5,D; BIT 0,A; BIT 3,(HL).

Logical operators are another important facility in MC programming. They are AND, OR, XOR. The result is automatically stored in the accumulator so this does not need to be specified in the instructions.

AND will set to 1 those bits in the accumulator and the comparator (register or direct data) which are both set to 1. If only one of the two bits is set to 1 or both are reset to 0, that bit will be reset to 0 in the result. For example:

```
     0  1  1  0  0  1  0  1    binary data in A
     1  1  0  0  1  1  0  1    binary data in C
     ----------------------
     0  1  0  0  0  1  0  1    binary result in A of "AND C"
```

OR will set to 1 those bits which are set to 1 in either or both the accumulator and the comparator, and reset to 0 only those bits which are reset to 0 in both the accumulator and the comparator. For example:

```
     0  1  1  0  0  1  0  1    binary data in A
     1  1  0  0  1  1  0  1    binary data in D
     ----------------------
     1  1  1  0  1  1  0  1    binary result in A of "OR D"
```

XOR is an exclusive OR in which only those bits which are set to 1 in the accumulator or the comparator, but not both, are set in the result. Bits which are the same in both the accumulator and comparator, either set to 1 or reset to 0, will be reset to 0 in the result. For example:

```
        0  1  1  0  0  1  0  1     binary data in A
        1  1  0  0  1  1  0  1     binary data in E
        ----------------------
        1  0  1  0  1  0  0  0     binary result in A of "XOR E"
```

A similar instruction to these is COMPARE. The contents of the accumulator are compared to those of a register, an address held in the HL register pair, or a value. Neither are changed, and the result of the comparison is stored in the zero flag. The zero flag is set to 1 if there is a match, but otherwise is reset to 0. Testing the zero flag after the comparison allows us to determine the result of the comparison. An example of the format is CP (HL) where the accumulator (not specified) is compared to the contents of the memory address held in the HL register pair.

Now to start some programming!

You will have found very quickly in BASIC programming that LOOPS are useful for repeating operations a specified number of times. The same goes for MC programming.

The start of the loop is marked by a label which the compiled assembly program, i.e. raw hexadecimal machine code, knows as a memory address only. Labels can be called anything, but are followed by a colon without a space before or after it to tell the assembler that it is a label. For example LOOP1:. The first instruction of the loop follows immediately (no space!).

At the end of the loop, the program needs to be told to jump back to the label. If the number of bytes of compiled program is less than 127 backwards, a relative jump can be made. If 127 or more, a normal jump is necessary. These instructions are JR LOOP1: and JP LOOP1: respectively. Both types of jump can be conditional, and forwards as well as backwards. The format is JR condition, LOOP1: and JP condition, LOOP1:.

The conditions are	NZ	not zero
	Z	zero
	NC	no carry
	C	carry
	PO	parity odd
	PE	parity even
	P	plus (sign flag reset to 0)
	N	minus (sign flag set to 1)

Obviously, the jumps can also be made to specific memory locations rather than labels, and either directly or as held in the HL register pair.

When counting to perform a loop for a specified number of times, the Z80 has another special command. The B register can be used as a counter register. It is loaded with the number of times the loop is required to be performed. The range for this single register (one byte) is from 0 to 255, but by a little trick, this can be made to loop 256 times (far more useful than 255!). If the register is loaded with zero, when it is decremented, it will contain 255 and not zero so the loop will be performed a further 255 times, giving 256 in all.

The command at the end of the loop is DJNZ LOOP1:. This command only works on the B register. First it Decrements the register. Then it will jump to the specified address, in this case label LOOP1:, if the zero flag is not set, i.e. the value in the B register is not zero. Once the register has decremented to zero, the loop is finished and the program continues with the next instruction.

Also as in BASIC, subroutines are very useful in MC programming. Instead of GOSUB, we use CALL, and rather than a line number, a label can be very useful since it gives an indication of what the subroutine is about. For example, CALL PRINT: if the subroutine allows characters to be printed to the screen. As with BASIC, there must be a return from the subroutine to the point in the program immediately after the subroutine was called. This is simply "RET".

On the same theme, there are a large number of "subroutines" in Albert's MOS (machine operating system), and the DOS (disc operating system) which can be used from MC and save yourself a lot of programming. The MOS routines are employed by using two lines of commands. First is "RST 8" which is a specific restart address to access the MOS routines. Second is the hexadecimal number given to the routine you wish to use defined as a byte of data rather than a program instruction. The CPU knows to expect such a data byte after a RST 8 instuction so it does not get confused (even if we humans do) between program instructions and data. However, we do need to make the assembler realise that the byte which follows RST 8 is data and not an instruction. An example of the second line is therefore "DEFB &9E" (define byte as &9E), which is the hexadecimal number ascribed to the routine which outputs the character held in the accumulator (as the binary equivalent of the ASCII code) to the screen. A short program to illustrate this might be as follows:

```
LD A,"A"
RST 8
DEFB &9E
```

This would output the letter A to the screen at the current cursor position.

Lists of the MOS subroutines, also known as MCAL's, accessed by RST 8 can be found in Einstein User magazine Vol 1, No 4, the Glentop Einstein Assembly Language Course manual, "The Source" Appendix D and "Albert Revealed" section 2.3, to name the sources I have and know of.

Although machine code can be entered directly into Albert's memory as hexadecimal codes, it is necessary to have a list of all the codes to allow programming like this, and it is an extremely tedious process, particularly if there are errors in the initial program which require sorting out. That is why the use of mnemonics and an Assembler is recommended when writing machine code programs from scratch. Of those available, I think they are fairly similar to each other, but I have only one - the Glentop Einstein Assembly Language - which is what I am using to illustrate these articles. UKEUG Software Library volume 147 claims to contain assemblers and disassemblers, hopefully with disc-based documentation, so for those of you without an assembler who want to have a go, that would be worth a try.

Finally for this article ("Thank goodness", I hear you cry), here's how you load a machine code program into Albert. If it is completely "stand alone", which is unlikely to start with, it can be saved as a .COM file and loaded from DOS by typing the filename only. This loads the program starting at memory location &0100 and will run it as soon as it is loaded. More commonly, machine code routines are used from a BASIC program. These need to be loaded "above" the BASIC program in the memory, but below the "reserved" memory at the top. The precise locations of these depend on the BASIC program that has been written, but before any program has been written in BASIC, with XBAS loaded, it has to be above &3E03 and below &E73F. With a BASIC program written and loaded, the top and bottom memory locations available for machine code routines are given by PTR 17 and PTR 16 respectively.

For a machine code program to run from a given memory location, all jumps, etc., need to be calculated from the start position of the program/routine. This is done in assembly language by setting the "origin" of the routine using "ORG memory address" as the first line of the

assembler program, for example ORG &E000. When such an assembly program is compiled to give the machine code, it is saved as a .COM file. This should be renamed as a .OBJ file. The BASIC program then loads the machine code program into memory as follows:

```
CLEAR &E000
LOAD filename.OBJ
```

Obviously, the .OBJ file must be on disc 0 when the BASIC program is run or a No File error is generated. To use the machine code routine from the BASIC program, use CALL &E000. To ensure safe return to the BASIC program, the machine code program must end with RET, as for any subroutine.

The memory address supplied in the CLEAR command sets the upper limit of memory available to the BASIC program so that the machine code routine is not overwritten when the BASIC program runs. Thus if the CLEARed memory address is too low, the BASIC program will not be able to run. If too high, there will either not be enough room for the machine code routine to fit below the "reserved" memory or the stack used by BASIC, which fills from the top of memory down, may overwrite the machine code routine. The latter can be prevented by setting a lower limit to the stack as a second address after the CLEAR command, e.g. CLEAR &8000,&E700, but if the BASIC program requires more stack than allowed by this command, then a Memory Full error will be generated.

I know all that sounds very complicated, but for short machine code routines run from a BASIC program, I have never yet encountered difficulties when using anything from &8000 to &E000 as ORG and CLEAR addresses.

Examples of Assembler-written machine code programs will appear in the articles to which these are an introduction, after a brief third introductory article to expand on some of the concepts already described. Hang in there until then, and suddenly you will be able to make Albert do things you never dreamed of!

****

CHRIS COXALL WRITES:-

Please find enclosed my membership renewal. Also enclosed is some material which may be of use to the magazine.

The Tekno Scrimping, Content Explained and Labelling is an idea I have been working up for myself so as to provide a structure, a format and a discipline for a written and pictorial record of my own projects. An autobiography, as it were, of my own achievements, so that if, and when they might be relevant to others, they can be shared. It is not exclusively applied to micro computers, but that seems to be its major preoccupation. I hope this helps to explain the way other material is presented & the motivation behind it.

The activity listing is something that I would like to share on a regular newsletter basis with other individuals involved in similar pursuits, so that if a solution to another's problem can be seen it can be communicated.

I believe one problem of asking readers to contribute material to a magazine could be that when through self discovery they accomplish something of great value to themselves, they put their all into presenting it to others, only to be told that their solution is a common one and has been well covered before. At the same time, another of their achievements is considered by them to be a minor accomplishment of straight forward logic not worthy of a mention, when it would in fact be of great value to others.

Sharing activity newsletters with others would, I think, be a way for me to avoid this problem. The articles are worked up to a standard where I am content to put them into my own Tekno Scrimp folder for a time where they might be relevant to others and can be shared, or I can sort and compile them into booklets to help sell off surplus micros or related equipment. If you can use them please do so. If you have the means to improve the quality in a way that alters the layout and format or generally edit them to your satisfaction please do this also. I have left them in A4 size for this purpose but they reduce down to A5 size on a photocopier and to my eye seem to improve. I am afraid my diminished art college skills are now only resourced with propelling pencils, cartridge paper and computer printout for text. The local library photocopier is used for reducing and enlarging pictorial material and all other copying. Paste-up is with scissors and Pritt-Stick glue. You end up with photocopies of photocopies and a lot of Tipp-Ex in between. It is at best, what we called (at college) a studio visual standard.

Some bad news my Einstein has ceased to be operational. It never did work properly and kept crashing out on me. But now, nothing at all. I did (for a time) use the 3 inch drives as a second one on a Spectrum+3, and the other on an Amstrad CPC-644 (until replaced by 5 1/4 inch). Albert's P.S.U. was used to power one of these until I found that the vibrations of a passing lorry would set the drive off on its own Dos routine to destroy all files and formatting. I do not have the same problem with a scavenged BBC PSU, so it must have been Albert's P.S.U. that killed him.

I hope the enclosed can be of use, and I look forward to receiving another year of Einstein Magazines.

TEKNO SCRIMP Activity Listing

Operational results:-

Adding two Teac 5 1/4 inch FD 55F-03-U as second drives to Amstrad CPC 644 & Spectrum+3.

Using power supply units from broken Einstein & BBC computer to power additional disc drives meant for internal use.

Exchanging a broken disc carriage of the Amstrad 644 with an external non working Amstrad FDD1 carriage. This meant swapping and using the original disc drives circuit board.

Cross wiring the RS 232 ports of Z88 & Spectrum+3.

Cross wiring the RS 232 ports of Atari STfn & Spectrum+3. (Illustrations being drawn up).

Writing Atari ST basic programmes for serial communication with Spectrum+3. Program (1) to use Atari as a RS 232 to Centronics converter for printing, (2) for storing Tasword II files in page lengths on Atari's 3 1/2 inch discs. Also Atari basic programmes to store serially transmitted Z88 file on disc. (Instructions and program listings being drafted).

Bodging an 8 pin Din plug using panel pins, plastic padding, metal from tin cans and toothpaste caps. (illustrated instructions being drawn up).

Using dressmakers snap fasteners and plastic cut from milk cartons to make pop wiring interface connections. (illustrations to be drawn up)

Speculative Ideas For The Future:-

Three disc drive versatility. Using centronics switch box to run 3", 3 1/2" and 5 1/4" drives off the one second drive select line. (4 way centronics switch box Cirkit catalogue No. 10 25019)

3" & 3 1/2" drive 'A' option for Amstrad CPC & Spectrum+3. Using 2 way centronics switch box. Connect internal drives

I.D.E. cable to switch box input then one out put back to the internal drive the other to a 3 1/2" drive. [Wanted by Author the 26 line configuration of Amstrad's & Spectrum's I.D.E. interface so to match it to standard 34 line configuration before adding centronics plugs and sockets. This would not be a problem with an Einstein which uses a 34 line configuration.] (2 way centronics switch box Cirkit catalogue No. 10 25018.)

Use an acoustic coupler (original modem) to put serially transmitted files from a Z88 laptop onto a cassette recorder and also over a telephone line to an answerphone. (Wanted by Author where an acoustic coupler can be found or how to make one up.)

Full Idiot Proof Interface for Z80 Chip Computers .

Ambitious project for the future where the expansion sockets of Z80 micro computers can be wired to a common interface with an established standard on a text book layout basis. All inputs to be covered by light sensitive devices all outputs covered by light emitting devices. Purpose so that novices can experiment safely with parallel communication between computers and experiment with building their own computer operated mechanical constructions without destroying computer circuitry.

Also Wanted By Author:-

Where to get CP/M system disc (and manual) for a Spectrum+3.

I have 48k Spectrum+ with manuals and the Spectrum Interface 1 without manual. Wanted basic fitting instructions and programming just to see if its working. Its RS 232 wiring configuration is wanted for an illustration .

Items the Author knows where available but has not yet got around to doing so:-

Disc & comms utility software for Spectrum+3, Amstrad CPC and Atari STfn.
Other Non Computer Pursuits:-
Interest in D.I.Y. Solar Heating
D.I.Y. trailers, side cars and battery powered kits for push bikes.

--- *** ---

Please find enclosed a cheque for £20 for the full sert of currently available back numbers of EM. I have enclosed a sticky address label for your convenience.

Also find enclosed two three and a half inch discs holding the same TEKNO SCRIMP text that you requested from my Atari peripheral. You will see that the first copy I made was on a high density disc which reformat and work on the Atari's disc drive. When I realised this is what I had done I made another copy on a new double density disc just in case your equipment would not operate in the same way. The discs contain of course only the text of what I have previously sent you on paper. There is no way I can think of to get the illustrated material onto disc.

I have not really put any time into seriously looking at the Atari's potential, as I have been content with just using it as a serial to parallel converter for printing and storing Z88 files. When I was using the the Job Club my portable Z88 proved essential for speculative letters and c.v.'s, just plugging it into the printer they had there or waiting until I got home to use my own, but for the long term storage of its files it is hopelessly inconvenient on its own and needs the Atari. The two basic programmes I use are also on the discs enclosed.

I will have to find an Atari user and pester him for even the basics of operating. I am completely confused. The basic I use is booted from a ST TM disc which was boxed along with its comprehensive source book and tutorial in mint condition and it works perfectly on my Atari ST FM computer, but I have no luck at all in getting Atari TM cover discs or PD disc programmes into the computer. Unlike Z80 chip computers I do not want to get to involved with Atari's and write programmes for them. I just want to be a lazy software user and without the original manual for the ST FM computer I am at a loss to know what is what.

With the exception of the two Atari ST basic programmes on the discs sent, the other programmes are word processor text files transferred from my other computers (stored in A4 page lengths) so they must be in the ASCII format you requested.

I have enclosed a stamped addressed post card that you can tick or cross to let me know if you have successfully read them so as not to take up too much of your time.

The file names correspond to the printed matter already sent, as below:-

Act1 & Act2:- page 1 & 2 of the activity listing
Content:- the one page of Content Explained
EUGLT.P1 & EUGLT.P2:- pages 1 & 2 of my previous letter.
LABELLIN:- the one page of Labelling Of Projects

PERIPH.P1 & PERIPH.P2:- page 1 & 2 of Three Micro Computers or One System (The CPC 644 has been corrected to CPC 664. I am thankful for scrutiny. The 644 confusion came from the Amstrad CMT 644 Monitor.)

SCRIMP1.TAS & SCRIMP2.TAS:- page 1 & 2 of Tekno Scrimping

RSTFL.BAS:- is the BASIC program I use to put imported serial files onto disc. It has a bug. Although the first file after loading the programme records to the disc o.k. it does not display itself properly on the screen as it does so. The second and subsequent files do do this as I intended.

PRTSER.BAS:- is the basic programme which turns the Atari into a serial to parallel converter for other computers to print out on the Mannesmann Tally printer.

Since last writing to you my circumstances have changed, and I am now in temporary employment. It is boring repetitious production line work in a factory. It's the type of work that should be done by robots, but as the management does not have them, and chimpanzees and other apes will not do it, poor human beings such as myself have to stand in. When the management ask why I will not do any overtime, I tell them " It is time for me to go home and use my brain. "

The good news is that I have some money. Not enough, of course to be 'a plug it all together according to the manufactures instructions PC Clever Dicky', but enough to buy some equipment and experiment with what I think is theoretically possible.

With a Centronics data switch box and a new 1.44 3 1/2 inch drive adding to the two 5 1/4 Teac drives, along with the two three inch drives from the dead Albert that I already have, experiments show that having three different size drives on drive select A and three on drive B is possible.

It all depends on how the jumpers are selected on the drives, and whether a terminating resistor block should be used. I hope soon to have two disc versatility cabinets which with two leads can simply be plugged into the Spectrum+3 or Amstrad CPC's. I have solved the mystery of the 26 pin configuration of Amstrad's and Spectrum's internal disc cable, and by making up a little adapter out of dressmakers pins, and setting it into plastic padding bought from a auto part store, I have an interface plug that joins a 26 IDE socket to a 34 IDE socket.

The hold up at present is that I am relying on scrimped second hand cabling and juggling edge board and centronics fittings the best I can. A bus trip to my nearest Maplins at

Thurrock is on the cards for one weekend. To run the 3 1/2 inch Mitsubishi MF355C-18UC from the Spectrum+3, by the way, requires cable line 34 & 33 to be pulled together.

Out of curiosity I tried one of the edge board to centronics fittings I made up for the disc versatility cabinet on the CPC 664 (not 644) and my old Epsom printer, and this cable worked.

It would not, though, work the Mannesmann Tally printer, but the 664 has now been replaced by a CPC6128, which (with the same cable) does work the Mannesmann. At the moment things do seem to be coming together successfully.

The bad news is that although I now have some money I have not got the same amount of time I would like to have for word processing or getting out the drawing board to add to my Tekno Scrimp folder.

I have received a letter from Ted Cawkwell as an active member of the EUG and I will be putting forward my best effort to send him every thing I can. At the moment, I am afraid, this will not be much more than copies of what I have already sent to you, but for the next week or so I intend my evenings and weekend to be for paper work and corresponding.

--- $$$ ---

Please find enclosed MS DOS formatted disc. This is not the one you sent me for the Atari. I am still working on that.

I have in the last few weeks acquired a Compaq Deskpro 386 PC and -- not without a great deal of frustration in writing a Qbasic programme for the serial transfer of files -- have put the Tekno Scrimp files onto the 3 1/2 " disc enclosed. These Type and copy to the console ok.

The files with the .DOC extension are Wordstar 6.0 copies which type and copy to the console, displaying operational codes. I cannot seem to get the Wordstar facility to create ASCII files to work. All non .DOC extension files were loaded in to Wordstar as "non-document" files, to use the spell checking facility. Some of the files have double spacing, and I have not found a way around this yet. I hope you will be able to read this disc. In the mean time I will try to unravel the mysteries of the Atari.

Failing this I should soon again have an Einstein. Unfortunately this is coming from an EUG member who also has just become the owner of a 386 PC but unlike me is going to let his membership lapse. I still intend to remain amongst the "8 Bit Die Hards".

With a new PC scrimped together with out manuals and an Einstein on the way I have not really had time to set my computers up for the convenience of use let alone familiarise myself with PC operating or using Wordstar software but I hope the disc enclosed will be ok.

Chris Coxall

****

TRADER'S JOTTINGS (Drives & More Drives) Ron Dunnett

It is not very often that I get the time to write articles for the current QL Magazines, but as the months of June and July have been rather quiet, regarding sales, and the current Editor of QUANTA has been crying out for copy I decided to write something.

The following might be of some interest, I don't know, so here goes.

IDE ZIP, CD-ROM and LS120 Drives

Some QUBIDE users have enquired if the new IDE version of Iomega's ZIP drive will work connected to QUBIDE. The simple answer is 'NO'. The reason for this is because the ZIP drive is not a true IDE or EIDE device. The ZIP drive along with other devices that purport to be IDE such as CD-ROM drives and the new so called replacement for the 1.44mb Floppy Drive known as LS120 or the 'A Drive' are actually ATAPI IDE.

Most of the PC suppliers that sell these drives do not have a clue as to the difference between ATAPI and normal IDE, they are in the PC market and are more often than not just 'box shifters'. I contacted the UK distributors of the LS120 to ask them about their drive, but even they didn't have a clue and couldn't answer my questions.

I then looked elswhere for information regarding ATAPI, and with the help of Phil Borman and Dave Walker we tracked down the specifications for ATAPI. It turns out that ATAPI is a subset of SCSI 3, and commands are sent in packets along the IDE cable to the ATAPI device which is then able to interpret the commands. So basically the IDE part is really just being used as a transport mechanism for the 'Packet Information'.

With all this information about ATAPI I asked Phil if it would

be possible to get an ATAPI CD-ROM drive to work on QUBIDE. Phil kindly said he would have a look but couldn't promise that it would be possible. I purchased an ATAPI CD-ROM drive, they are not too expensive, and gave it to Phil to investigate. Unfortunately at the same time the SyQuest EZ135 IDE (that's proper IDE) was available in the market place. After several months of intense investigation into the CD-ROM it was decided that it was a waste of time as the drives at the time were extremely slow and there wasn't really anything QL related to be found on CD-ROM's anyway, also the EZ135 was becoming a far better alternative. We then decided to call it a day but would offer the information regarding CD-ROM's to anyone who was interested in writing a driver themselves. In the age-old QL tradition only one person has enquired for this information.

Since the demise of the popular SyQuest EZ135 IDE drive, we have been at a loss where to find a replacement. An alternative could be the Iomega ZIP (ATAPI IDE) drive but, and this is a very big but, this would involve a device driver being written. Like all software this takes time and costs money. Don Walterman of the USA enquired via Email if a driver could be written and my reply was 'Yes' but are you prepared to pay for it ????

My reply to Don was also picked up by several other QL users who were also interested in seeing an ATAPI device driver written. I then decided to keep a list of all those people that were willing to put their money where their mouth is. The list isn't very big, only 8 people have come forward, and one of those is myself. This is very disappointing but not unusual especially as the collection was only from Emails that I had received. With only this small amount of people willing to pay then I'm afraid the ATAPI device driver will not get written.

I therefore issue this request:- If you would like to see an ATAPI device driver written and are prepared to pay for it then contact me (address below) and I will add your name to the list (Do not send any money). Once I have a big enough list I will approach Phil and ask him if he could look into writing an ATAPI driver for the Iomega ZIP drive.

FLOPPY DISK DRIVES (DS/HD 1.44mb)

Many QL Users are turning to DS/HD drives when their DS/DD drives have stopped working, but they are encountering problems. One of the biggest problems is that the new drive/s that they have purchased doesn't have any jumpers or switches to set the drive to work as FLP1.

The reason for this lack of switches or jumpers is because the drive manufacturers believe that there is only one computer on the face of the earth now and this is the PC, so they manufacture with the PC only in mind. The PC requires the drive to be set as DS1 (Drive Select 1), on the QL this would mean that the drive would be FLP2. I also suspect that this is a cost cutting exercise too.

There is a very easy way round this problem, but it does involve the ability to remove and re-connect the 34 way connector found on the ribbon cable associated with the drive.

This is how it's done:-

Remove the 34 way connector from the ribbon cable, locate wires 10, 11 and 12 of the ribbon cable, cut the ribbon cable so that wires 10, 11 and 12 can be separated, twist wires 10,11 and 12 so that wire 10 goes where wire 12 was and vice versa, re-connect the 34 way connector. The finished product can now be re-connected to the drive and the drive will now be set for DS0 (Drive Select Zero) which is equivalent to FLP1 on the QL.

Don't blame me if you cock it up.

Well that's all for now folks. If I get a chance or if there is something else you would like me to write about, please let me know, I'm a mine of useless information.

Ron Dunnett QUBBESoft P/D, 38 Brunwin Road, Rayne, Braintree, Essex. CM7 5BU. Tel/Fax: +44 (0)1376 347852 Email: QUBBESoft@aol.com

USING SUBDIRECTORIES WITH THE PSION PROGRAMS.

Mark Knight

I've seen a lot of questions about this over the years and recently in QUANTA it cropped up again. Although I don't use the Psion programs any more I thought I would dig them out and write a bit of helpful text on how to persuade Quill to use subdirectories. Although all the Psion suite are hostile to subdirectories and long filenames they can be tricked into it with a little help from some clever device drivers.

If you change from microdrives to floppy disks the sheer number of files you can pack onto a disk makes the use of subdirectory files a great idea, especially if you use HD or ED disks. On the Gold Card and Super Gold Card subdirectories are available on the floppy disk and ramdisk drivers, and on the Miracle and QUBIDE hard disks the same facilities are present. On hard disk it is a very poor idea to keep files without using subdirectories as you may end up with thousands of files and finding the one you want in a list that long could be a nightmare.

JOHN MARRIOTT WRITES:-

Sorry about my corrupted text files regarding DISK DRIVE UPGRADING (EM84/21) - if anyone wants them I can send the file(s) either on 3.5" PC disk (SAE with 1.44M formatted 3.5" disk) or reams of paper (SAE for 10 A4 sheets, plus 2*2nd Class stamps to cover my paper/ink!) to John Marriott, 121 Hill Barton Road, Exeter, Devon EX1 3PP.

A comment on Duncan Elvin's article - HARD DRIVE (EM 84/13). With the overall cost of the project, how about a little sideways thinking here? The Z80 chip is now quite cheap, and the easiest way to remove a soldered-in chip is to cut each "leg" as it comes out from the plastic case, using a pair of angled toe-nail scissors, and ensuring that the blades are parallel to the PCB (thin, vulnerable tracks abound!) Then un-solder each leg -- preferably holding the cut end with lock-tweezers (TANDY!), so you KNOW that it has NOT fallen down under another IC!

I use a needle with a dowel "handle" to "open up" that "plated-through" hole -- place the needle tip into the hole, and heat the needle close to its point whilst "twizzling" the dowel handle back and forth, (but practice on a defunct PCB first!) As the needle goes "through" the hole, keep twizzling, but remove the heat.

The result is plated-through holes ready to take a GOOD QUALITY IC socket. Now the whole uprate HARD DRIVE interface can be where it should be -- inside the case.

Incidentally, you DID have the computer switched off, unplugged, AND the PCB out of the case, didn't you?

****

A (LATE) CHRISTMAS MESSAGE FROM YOUR BELOVED CHIEF EDITOR

You didn't get a Xmas card from me, but if it consoles you at all this issue is being put together and printed while the rest of you are munching mince pies, knocking back the festive hooch & watching 60-year-old b&w films on the idiot box for the umpteenth time round. Actually I find printing the magazine just as much fun, so I'm not complaining any!

****

NEW HOMES WANTED:- Contact Commodore PET librarian Ken Ross at 26 Redinham Hse, Tangley Grove, Roehampton, SW15 4ED if you can provide a good new home for Commodore 8023 or 8024 printers, Olivetti 102-key PC keyboard, various other bits.

Hopefully we're getting an old HP laser printer from him to produce the mag with -- UNLESS Chris Coxall beat us to it!