

WAGEI

137

01 - Thanx & Stuff

02 - Fair Comment

03 - New Generation

04 - Exploring the PSG

05 - Programmers' Patch

06 - CPC Changed My
Life

07 - Cartography



Thanx & Stuff

The best-laid plans of mice and men aft gang agley. And you don't get much more embarrassingly agley than this

Ooops

Ok. This issue of WACCI was ready on time - give or take a few days. Plentiful (well, a few) articles from you lot. Fair Comment from Philip. No indication from Brian of how many pages the Aktiv ad would be, but this is par for the course. A few more evenings' work, and it'd be ready for the printers.

At this point, my lovely 20in Mac monitor decided that ganging agley sounded fun and it ought to try it. Having asked Rob Scott to repair my various computers for free once too often, I thought I'd do the decent thing and pay for a repair at a local computer specialist. So off it goes to G Electronics in Bicester. A few days later, they phone to say they can't figure out the connection on the back, and could I drop the Mac in, too, so they could test it? Sure.

Anyway, about a month later, they phoned to say it was far too agley for them to fix. So this issue of WACCI has been finished using a 14in screen borrowed from work. This might sound fairly good on the CPC scale of things, but Mac programs tend to have floating control windows, menus, and that sort of thing all over the place. It's like editing WACCI while looking through a letter-box, though with less chance of being arrested.

Still, you might even be able to read the text, as I've even bought a new toner cartridge specially for the occasion. (Laser printers are wonderful. With inkjet printers, you have to buy a new cartridge every two weeks, and end up remortgaging your house to pay Mr Epson. But this is the first new cartridge I've bought for my laser printer in five years.)

Another one of the nice things about CPCs is that they don't run Microsoft Word. Around 10% of the work for this issue has involved persuading Word to read Word documents, which you wouldn't have thought was too much to ask. For those of you with PCs - and I know it ill behoves a WACCI editor to be picky, but still - please, please, please don't send articles in Word format, or even Word-generated RTF, as Microsoft appear to have done their best to mangle that, too. Use 'text only' (i.e. ASCII). Or a piece of paper. Or cave paintings. Anything. Philip has started going on about carrier penguins, but he's a bit odd in the head.

The great inscrutable

Apologies to all and sundry, by the way, for not printing my phone number in the magazine. It's nothing personal - just that I'm generally too busy doing one thing or another (playing the church organ, editing various websites, drawing maps, restoring our old wooden narrowboat - even working, believe it or not) to spend hours on the phone to CPC users every night. Sorry. But do send me (or Philip) a letter.

And I'll let you know if I ever manage to get Widor's Toccata licked.

www.wacci.org.uk

has the contents of this issue on it once again. Tell your friends.

Blast from the past

"He also worked on the design of an extranet broker connectivity trading system which was later implemented and has been running successfully as a core component of Schroders Salomon Smith Barney's business." Alan Scully, apparently. It's amazing what you can find on the Internet.

My brain hurts

You will have noticed there are lots of techy programming articles in this issue. That's because it's all anyone has sent me (except for the next part of 'SCART Connections', held over until next time). If you'd rather read something else - have you all really given up on Protext? I didn't think so either - then you know the routine. And once again, thank you to everyone who's contributed this time. The next one will be ready in two months. Even if I have to use PowerPoint.

Dear committee

Please could you tell me how many pages you would like for the Aktiv ad, ideally before the magazine is meant to come out, not three weeks later. It kind of helps in laying out WACCI. Thank you very much.

Anyway, I'm off for a week's holiday on a boat in France. Hope you like the issue.

Richard Fairhurst



Fair Comment

with Philip 'Tigger' DiRichleau

From the letters editor

Fair Comment, unFair Comment, Very Little Comment. So what's happening, I find myself asking. Perhaps it would be best to introduce myself for those you who are not quaking in your boots, or having fits at the sheer mention of my name once more gracing the pages of WACCI. I am Philip DiRichleau (contrary to rumour, stupidity or downright falsehood this really is my legal name - if you wish to check it up, it's on the register of electors), one time Editor of WACCI and, on and off for the past few years, Fair Comment Editor of the same.

So, back to my original question. What's been happening? Very little. I can only attribute this to the general apathy that seems to have settled over the world of the CPC and, to be bluntly honest, WACCI in general. If you want this magazine to continue, then you're going to have to do it. I'm quite willing and able to waste my time and effort on it if you're willing to do the same. It's not like it's a difficult job to write a quick letter, or send an e-mail. You could tell us what a lousy job we're doing, you could tell us that you hate the way I have a go at certain programs (like Brunword, which I hate). Ah, I can see letters flying in from the Brunword Appreciation Society Targetted Against diRichleau's Dastardly Schemes.

I'll tell you this much though. Fair Comment editing is a no-no as far as I'm concerned. You have the right to reply to anything I, or anyone else for that matter, says about this mag, just as we have the right to reply over any complaints that have been made against us.

So, the point I'm making is a simple one. Write us a letter, send it by disc, or disk if you prefer, or e-mail, or jungle drums, or carrier penguin. I don't mind (I don't return penguins, make too good a Sunday lunch).

Now, in my personal opinion (note how I'm distancing the rest of WACCI from this bit), we need to do something now, not tomorrow; now, before the WACCI that I, and others, have worked so damn hard on really does turn into a sad, pathetic old lady who would be better euthanised than kept on an expensive life support machine. And if you don't like that statement, then write and tell me.

Philip DiRichleau
32 Arboretum Avenue, Lincoln LN2 5JE
dirichleau@netscapeonline.co.uk
philip.dirichleau@ntlworld.com

Yeah, I know, I'm greedy. I've got lots more e-mail addresses if you get bored with those two.

On with the very cut down Fair Comment.

A cynic writes:

Dear Richard/Philip,

I am delighted to have received the New Look WACCI and I must compliment you on the splendid team you have assembled. I am also pleased with the contents although I cannot refrain from commenting on your remark that "there won't be any PC content". As far as I can see every page was concerned with PC-related matters. But then, I'm an old cynic.

My setup is very simple. I have a CPC 6128 and a printer, a second 3 1/2in disc drive, and various

bits of software. No PC, although I am tempted and thinking of joining the rest of the members in the two-computer brigade.

The trouble is I don't know where to start. As an absolute beginner, should I go for a PC, or an iMac, or a set-top box? (I have ON-Digital, which has the facility for surfing via a set-top box.) I would be glad of a few pointers.

The running costs are also a factor. (I'm retired.) How much is the phone bill for, say, an hour? And an ISP, and anything else you can think of.

Emulators, what are they and where do you get them?

Incidentally, I was always at a loss to know where to get the excellent Disczines various people published from time to time. I have always thought that it would be a good idea if all the good people publishing them could be persuaded to let WACCI publish them, at a price if necessary.

Finally, greetings, and thanks to Dave Stitson for his valiant effort to keep WACCI going.

With best wishes,

Ron Hobday

43 Cody Road, Clapham, Bedford MK41 6ED

(01234) 290826

Good day to you Ron. Where to begin? Is it worth getting a PC or a Mac (I can see a fight developing between the Editor and Fair Comment Editor here). I would go for the PC if you really want one, if you will find a use for one, and if you can afford the monster (they are very expensive for what they are). [As ever, depends what you want to do with it. There are more games and CD-ROMs available for PCs, but Macs are better for design work and are easier to use. - Richard]

As for ISP (Internet Service Provider). If you live in an NTL cable area, then you can get Free Internet Access, and I mean free; you don't even pay for the call charges. Otherwise, there are plenty of other servers available which usually charge a base rate and no call charges; BT, I believe, do one like this. (Richard, please correct me if I'm wrong.) [No, you're spot on. I've got a rather good deal from ClaraNet, who offer 12 hours' free access a month for £40 a year - Richard] What you do need to watch for is those that say they're free and you only pay for the calls. 1p a minute at weekend, 2p a minute in the evening and 3p a minute any other time, doesn't seem that much... until the £200 phone bill arrives and gives you a prompt heart attack. (It was more probably too much smoking in my case, rather than the phone bill, but hey, it's a good excuse.)

Emulators! Great things. They make a £1500 Pentium 3 PC with more memory than sense, a massive harddrive and all the rest of it, act and behave just like a CPC. Why would anyone want to? Because they're damn good fun. WINape (by Richard Wilson) is the best! Accept no substitutes.

Of course, the great thing about Emulators is that you can get all those bits you wanted (emulated of course), and play with them to your heart's content. You can pretend to have any CPC, with all the hardware (all the software is also available off the internet incidentally), and it don't cost a penny - except the money you spent on the hyperfast PC to make it run like a CPC.

Disczines? I'm afraid to say, and I hang my head in shame, that if I want an Amstrad Disczine these days, I get it from the Internet. So, in my normal manner I shall do this...

...Mr Fairhurst sir, where does people get them there Disczines from these days...

[Ok. Generally, they're written for people to swap amongst themselves: so if you know someone who knows someone who knows someone who has a copy of ATM 5, you can get a copy. This works well in Europe, where everyone knows each other, but in Britain - well, it sounds like just the job for the WACCI PD library. Right. I suppose I should send some to Ray, really, shouldn't I? And speaking of ATM, I'm sure Roy Everett will send you a copy if you ask nicely - Richard]

Blast from the past

Hi,

My name's Robin and I was a member of WACCI (about 2 or 3 years ago) and I found that all the people I came into contact with were very helpful and friendly, especially John Bowley and Christine Raisin.

Anyway, I'm just writing in to thank these people and say that, to be honest with you, I'm very surprised (but happy) that WACCI is still going strong even in the year 2001. I'm a member of Ebay (the Internet auction site) and a lot of Amstrad CPCs are still selling very well - in fact, I saw a cpc 464 the other day with the description of it being new and unused! It certainly looked in very good condition - it was still in the packaging! - and the manual looked in pristine condition. How is this possible?

Surely there cannot be Amstrads still lying around unopened in their packaging? They stopped making them 10 years ago! Anyway I might be tempted to buy it!

It's nice to see that there are still people who love their Amstrads. Long live WACCI!

Robin

somewhere

Peace be with you Robin, pleased to hear from you, but as an ex-reader I doubt very much that you're reading this.

There are still CPCs in mint condition out there. Why? I haven't the slightest idea, but the world does that to you at times. I think it's to catch us unaware or something like that.

Return of the PD librarian

Hi Richard,

It's good to see that you are still alive, and editing. Obviously I am still staggering on, with my creaky zimmer frame. It has to creak, so that I know that I am moving. Welcome back to WACCI, even if it is a move that throws a serious doubt on your sanity.

A member pointed out to me recently that he knows there are more than the 130 PD titles listed, but does not know what they are. I have given him the details, but he does have a valid comment. The missing titles are 131 Music & MIDI disc, and un-numbered 'The Protex Suite'. This comprises Protex on disc, Promerge, Prospell, Proprint (two discs), Prototype, Maxam, and Model Universe, all on separate discs. Unfortunately, Model Universe seems to be corrupt, and I am waiting for a replacement copy from Brian.

Incidentally, I do not have Sanko here. Other useful additions would be MakeBMP, Proclip, Multicon, and Mode 1-2. Apparently John had these to help in producing WACCI. Any chance of getting them? [Several of them are on the Robot PD Connectivity Disc. I'll find you a copy - Richard]

In the PD section, it mentions prices that include a 3.5" or 3" disc. Whereas I am more than willing to help out, I cannot supply free discs - although since have only had two requests for PD so far, and those from the same member who sent his own discs, this may not be a problem. Perhaps the fact that PD is now to be downloadable will make me redundant anyway. On the other hand, maybe the new improved list in the mag will cause a rush from those who, like me, have no access to the net!

One last thing. My house number is 36. This is a small village and I doubt if there are 1136 houses, as printed by Dave, or 136, as in the latest WACCI!

That's about it. I will try to come up with some sort of article for the mag in the near future.

Cheers for now,

Ray Powell

Sporle, Kings Lynn

Good Morning Ray, or Evening, it's me here. How are you my friend. Even as I took over Fair Comment (and started receiving no letters at all), I knew, I just knew, there would be a missive from your good self.

Sorry I wasn't able to attend our duel behind the gasworks, or defend your suit for Confirmation of Character, but I was busy falling up Everest in my dressing gown (which leaves a lot of people wondering just what old Tigger is going on about now).

Argggggg - ITGA

Dear Malnourished Vampire [Bunny is gonna pay - PdR],

Glad to see you back at the helm of Fair Comment. Richard's first issue of WACCI looked gorgeous. Nice to see him making use of the front cover a little more efficiently and that the magazine has finally officially gone bi-monthly, something I have been suggesting for years, as you probably recall.

The magazine content has been regurgitated in its spartan website format, in attempt to be compatible with as many machines and browsers as possible, including the new CPC browser.

Early feedback of the website is good and I hope that as many members as possible will head off to www.wacci.org.uk and take a look, just to see some of the diverse content we hope to be pouring onto the site over the next few months. I'd also love to know what members think, either by e-mail/letter to myself or via FC.

We, the website team, hope that WACCI having this kind of presence on the Internet will encourage new and diverse readers that simply wouldn't or couldn't subscribe to the magazine, as demonstrated by Julien Nevo from Arkos, and probably a similar aim to that of Euro-WACCI. We would like to encourage people to write for the website, which will result in articles for the magazine. We don't know how it will end up, but I'm quite proud of the momentum that the website, and Richard being appointed as Editor, has fired up, along with the thoughts, consideration and wonderful kindness those involved in the website have shown.

*Lots of love,
Angela Cook
Worthing*

Hiya Sweetness (not let's not start that rumour again, the wife would kill me)...

Keep up the effort with the Website, looking good. As for the rest, it's pretty self explanatory and I agree with most of it.

Most of it, I hear you cry in dismay. Well, in an ideal world, WACCI would remain monthly (I don't think I'm saying anything I haven't said before), but if bi-monthly is the way the Editor decides (it's his time being used up, not mine), then I will support him in that decision.

Hey, Angela, do you think I could write a letter to Fair Comment...

The man is schizophrenic

Pleased to see WACCI is back once more. Even more pleased to see Richard Fairhurst as Editor, rather Chuffed that Philip DiRichleau was once more asked to be Fair Comment Editor, and desperately hoping that WACCI will finally settle down into its rhythm once more.

*Philip DiRichleau
Lincoln*

Dear Philip,

Me too.

Philip DiRichleau

Fair Comment

That's it, the end of Fair Comment. Not impressive is it? Now I understand why letters haven't been arriving (that's a lie but I'm trying to be nice here), but please, get on with it people. I used to receive 10 a day before (and they were the light days).

New Generation: CPC Windows?

German programmer Prodatron is attempting to recreate Windows for the CPC. Richard Fairhurst finds out more

Yes, I know you've seen this before, in the form of PFD's April Fool many years ago. This one appears to be for real. Let me explain.

A handful of European ex-CPC coders are getting bored with PCs. When half the world uses the things, it's difficult to feel the sense of community, the competitive spirit, that characterised the best of the 'CPC years'. Since the CPC still offers the same mental challenge it always did, they're coming back to the old machine.

But their outlook, understandably, has been changed by exposure to the PC world. Computers work differently now. They have friendly menus; you're no longer expected to memorise a list of commands for each program. Users expect to be able to flick from one task to another in seconds. And they have 'helpful' paperclips which announce that "It looks like you're writing a letter" after you've hit four meagre keystrokes.

One such programmer is Prodatron. You might remember the name: in the early-to-mid '90s, he wrote the sound program Digitracker, a couple of disc fanzines (CPC Fastloader and Xtreme), and countless demos, of which the best-known is Voyage 93. After a few years programming PCs, he's returned to the CPC with a new project, SymbOS. Put simply, this is Windows for the CPC.

An unholy trinity

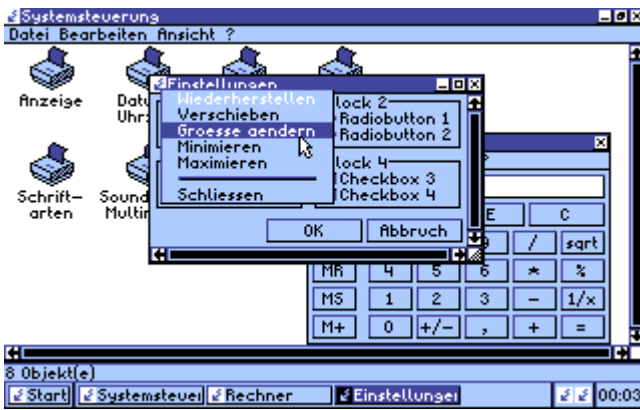
SymbOS promises three main advances: firstly, a Graphical User Interface, with windows, menus and a mouse pointer; secondly, 'pre-emptive multi-tasking', enabling several programs to run at the same time, with the CPC's resources shared equally between them; and finally, a new memory management system.

The last-named might sound arcane, but it's actually one of the main differences between a CPC and a PC. A 6128 might have twice as much memory as a 464 in name - yet you'd be hard-pressed to tell the difference. Your Protext document can't be any bigger. You can't run two 64k programs - say, GPaint and CP/M 2.2 - at the same time. It's just an overgrown 464.

If you have 128Mb of RAM on a PC, though, you can do a lot more than if you have 64Mb. (Note the x1024 multiplier!) Assuming that Windows will always take up 40Mb for its own workings, the 64Mb machine will have just over 20Mb in which to run a program - a simple word-processor, say. With the 128Mb machine, you have an extra 64Mb of memory to play around with. This means you can run three more programs at the same time.

This flexibility is what SymbOS promises. If you have a 128k machine, you'll be able to run a couple of simple programs at once. If you have a 256k expansion, you'll be able to run a couple of biggies together with a few small 'desk accessories' - perhaps a calculator-in-a-window and a note pad. Each program will communicate with you via its own windows: when you want to concentrate on one task, you'll be able to hide all the others from view.

And you'll need the extra memory. All of the standard 64k will be taken up with SymbOS code, as will much of the 6128's second bank. This will rule out serious use for the vast majority of CPC users.



The clever bits

A quick glance at the screenshots opposite will illustrate quite how far Prodatron has taken his aim of cloning Windows. You can 'maximise' windows (make them take up the whole screen) or 'minimise' them (collapse them to a single title at the bottom), just like you would on a PC. There's a SymbOS menu, which looks suspiciously like the Start menu in Windows, giving you instant access to all your favourite programs and documents. (A bit pointless if you don't have a hard drive, but never mind.) You choose program options by clicking clearly labelled buttons and 'check boxes', not by remembering the intricacies of Protex's SETPRINT command. (Again, it might help if you have a mouse.) Prodatron has also managed to create an appealing look for the program, which is half of the point of any Graphical User Interface.

It's all clever stuff, especially given that Windows was written by a cast of thousands.

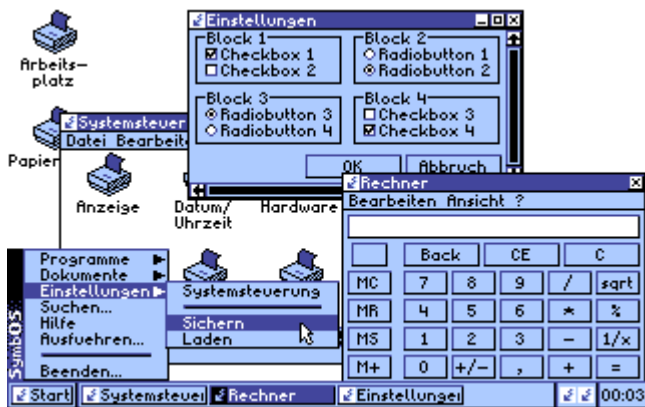


A strange case of deja vu

You may be forgiven for thinking you've heard all this before. DES, WOPS, MAX, Desk, Worktop (shh!), and so on have all attempted to recreate a GUI for the CPC. They have all failed.

DES was closest to SymbOS in conception, in that it wasn't just a disc manager program. Its services were available to other programs, and Comsoft encouraged programmers to write DES-friendly software. A few were released: Richard Wildey's DES-Text, my rather buggy Desktop Organise, and Comsoft's own disc of DES utilities. But that's as far as it went, despite the inevitable glowing reviews in AA and WACCI.

Perhaps DES could have succeeded if it were both free and seriously whizzy. SymbOS looks to be both of these, but is also several years too late.



Cloud cuckoo land

Though doubts have been expressed on the Internet, I'm pretty sure SymbOS actually exists. Prodatron has a decent enough pedigree for it to be believable.

Nonetheless, let's take a bit of time for a reality check.

Point one: If SymbOS programs are to run in windows, controlled by clicking buttons and pulling down menus, then they won't be anything like normal CPC software. In other words, none of your favourite programs will work under SymbOS, and Protext still won't be able to edit a 120k document. SymWrite will - if it's ever written.

Point two: It won't be. There aren't many people writing 'real' CPC software at the moment. The chances of programmers learning a whole new operating system, just for the benefit of a handful of curious people keen to try out this new wonder, are pretty slim.

Point three: A CPC simply isn't capable of running Windows. Multi-tasking is a nice idea, but the CPC isn't really fast enough to run simultaneous programs at a decent speed - and if one program goes wrong, it'll take the whole system down with it.

Does any of this matter? Probably not. Prodatron himself defines the project as "a demonstration of what could be possible with a CPC after 16 years". In other words: this is the CPC - it's supposed to be fun.

Where can I get it?

As yet, you can't. (This is sounding more and more like PowerPage Professional by the minute.) Prodatron is still working on the program, occasionally posting screenshots and progress reports to his website. At present, most of the brains of the system appear to be complete, but there are a couple of important features still to be written - notably the disc and printer handling.

We are assured that the finished version will be free, and that there'll be a preview available for a French CPC meeting later this year. We'll let you know as soon as we hear any more. CPC users with Internet access can find out more from the SymbOS webpage and mailing list at <http://62.26.220.31/symbos.asp>. (It might help if you can speak German!)

In future issues, we will examine CPC/IP, Mark Rison's software to connect your Amstrad to the Internet: the French CPC-NG project to design a new machine, based around the CPC, as a hobbyists' computer for the 21st century: and other new uses for an old Amstrad.



Exploring the PSG

James Hoskisson explains the intricacies of the AY-3-8912 PSG - otherwise known as the CPC's sound chip

Since you should now have at least some understanding of the PPI, the next chip on the list is the PSG.

James Gets it Wrong Again!

You may remember that last time I said that nearly all of the chips in the CPC use the Intel I/O convention; the PSG is yet another exception to the rule! In case you hadn't guessed, PSG actually stands for Programmable Sound Generator, in that it is Programmable and it Generates Sound.

The PSG was actually designed to be used with the CP1610 range of processors, which accounts for the strange I/O operation. This problem is negated within the CPC, since the PSG is attached to the PPI, which can generate any kind of signals.

Interfacing With the PSG

The PSG is actually quite easy to use once you figure out the slightly irregular addressing mode. If you have a decent grip on the commands associated with sound production in BASIC, then the PSG should be a doddle to use, since a lot of the functions of the PSG are equivalent to aspects of the BASIC sound commands.

The PSG is designed to work in both input and output modes. The direction of the data flow is determined by the state of the Bus DIRection (BDIR) pin. If the BDIR pin is high, then the PSG will be in input mode, if it is low the PSG will output data. Things get a little trickier when the other control pins are taken into account, but that's basically how it works.

There are two other pins involved in interfacing with the PSG. These are the two Bus Control pins, namely BC1 and BC2. These pins determine whether the PSG is receiving a register address, receiving write data, inactive, or outputting data. The four states are determined by checking the status of all three pins as follows:

BC1	BC2	BDIR	Function
0	0	0	Inactive
0	0	1	Latch Address
0	1	0	Inactive
0	1	1	Read from PSG
1	0	0	Latch Address
1	0	1	Inactive
1	1	0	Write to PSG
1	1	1	Latch Address

And now, to explain:

- *Inactive* - This is where the PSG does nothing. This may seem unimportant initially, but the PSG must be put into the inactive state when it isn't being used, otherwise any write data for

the selected register could be interpreted as a register select, which could lead to some strange effects!

- *Latch Address* - The latch address is used to select which register is used to read and write to. There are 15 registers in the AY-3-8192, numbered from 0 to 14. Each register controls a different aspect of the PSG. The registers are selected by putting the PSG into the latch address mode, and then putting the address to latch on the data lines.
- *Read from PSG* - This, rather obviously, indicates to the PSG that you want to get some data from it. When it is put into read mode, the data lines hold the contents of the selected register.
- *Write to PSG* - This indicates to the PSG that the data on the data lines is to be written to the currently selected register.

The way the PSG is wired up to the PPI makes using it a little counter intuitive. This is because the data on the PSG data lines must be present BEFORE any of the control signals change state. This means that the data to send has to be written to &F400, before the control signals are changed by OUTing to &F600. This may make more sense with an example:

```
LD BC,&F400 ;Put the data, for the register
OUT (C),C ;to select, on the data bus.
LD BC,&F6C0 ;Tell the PSG that you want
OUT (C),C ;to select register 0.
LD BC,&F600 ;Put the PSG in the inactive state.
OUT (C),C
LD BC,&F4FF ;Set up the data to write
OUT (C),C ;to register 0.
LD BC,&F680 ;Tell the PSG that the data
OUT (C),C ;is to be written to register 0.
LD BC,&F600 ;Return the PSG to the inactive state.
OUT (C),C
```

The example only writes 255 to register 0, but it is quite lengthy, due in part to the extra INACTIVE state selects. Without telling the PSG to ignore the data on the data bus, the new value, that is to be written to the register, would be used to select a register (which would probably be 7 in this case, since there isn't a register 255.) This could cause many problems because the new data would be written to the wrong register.

PSG Shortcuts

If you looked closely at the table you may have realised that some of the commands are repeated! If you looked even more closely you may have recognised that one of the BC signals is redundant, since all of the different modes of operation can be accessed with BC2 high. This is no coincidence. The chip was designed this way so that the decoding circuitry could be simplified, if it wasn't being used with a CP1610 processor. Amstrad took advantage of this in designing the CPC, and tied the BC2 pin to +5V, which meant only the BDIR and BC1 signals would have to be routed to the PPI.

The Useful Information

Right, now all the academic stuff is out of the way, let's get down to operating the chip.

As you already know the PSG is solely attached to the PPI (except for the power, address, and clock lines, of course!) All of the data lines of the PSG are attached to port A of the PPI. The two controls lines, BDIR and BC1, are attached to the top two bits of port C. You should already know how to set up the PPI to input and output the data from the ports, so I won't bother explaining it again.

As I already mentioned there are 15 registers in the PSG. These are:

Register	Function
----------	----------

0	Channel A tone period (fine tune)
1	Channel A tone period (coarse tune)
2	Channel B tone period (fine tune)
3	Channel B tone period (coarse tune)
4	Channel C tone period (fine tune)
5	Channel C tone period (coarse tune)
6	Noise generator
7	Mixer control
8	Channel A volume/Hardware envelope enable
9	Channel B volume/Hardware envelope enable
10	Channel C volume/Hardware envelope enable
11	Hardware envelope period (fine tune)
12	Hardware envelope period (coarse tune)
13	Hardware envelope shape
14	I/O Port data (Used for reading the keyboard)

Tone Generators: Registers 0-5 control the pitch of the note. The value is exactly the same as for the BASIC SOUND command, so the values can be looked up in the CPC handbook.

The lower register of each of the sound channels is 8 bits, with the upper register being 4 bits. The lower 4 bits (i.e. bits 0-3) of the most significant byte of the tone registers are used, with the upper 4 bits being ignored. This gives a range of 4096 possible tone values for each channel.

Noise Generator: This 5-bit register determines the amount of "noise" there is. The channel(s) that the noise is played on can be set using the mixer register, but the noise value is the same on each channel i.e. it is not possible to have two or three different noise values sounding simultaneously.

The value written to this register is exactly the same as the "noise" parameter in the SOUND command. The noise value can be anything from 0 to 31, with the lower values giving more of a hissing sound, and the higher values a grating sound.

Mixer Register: This is a 7-bit register which controls the I/O port, which channels have noise, and which channels should be sounding. The functions are allocated as follows:

Bit 7:	Not used.
Bit 6:	I/O Port control.
Bit 5:	Channel C enable/disable noise.
Bit 4:	Channel B enable/disable noise.
Bit 3:	Channel A enable/disable noise.
Bit 2:	Channel C enable/disable tone.
Bit 1:	Channel B enable/disable tone.
Bit 0:	Channel A enable/disable tone.

Whether the I/O port operates in Input or Output mode is determined by bit 6. If it is 1 then the port is set to output. If it is 0 then the port operates as an input (as used for keyboard scanning).

Bits 5-3 control which channels output the noise. If any of the bits are 1 then the noise is disabled for the channel. If they are 0 then the noise is enabled on the channel. Therefore, to disable noise output completely, all three of the bits would be set to 1.

Bits 2-0 control which of the channels output the tones that are programmed in to registers 0-5. If the bits are 1 then the tone is disabled, if they are 0 then the tone is enabled. These bits are useful if three notes are to be played at once, because the tone values can be set up in registers 0-5 and then all the channels can be enabled simultaneously.

One thing that isn't made clear in the PSG datasheet is that once a channel is enabled, the tone that plays will continue indefinitely, until the channel is disabled, or the tone is changed. This also applies to the noise output.

Channel Volume Control: As the name suggests these channels control the volume level for each of the channels, as well as whether hardware enveloping is to be used on the channel. Each channel has a dedicated volume register, so the volume of the different sounds can be controlled independently of one another.

Each of the 5 bit registers are arranged as follows:

```
Bit 7:  } Not used
Bit 6:  }
Bit 5:  }
Bit 4:  Amplitude mode
        1: Enable hardware enveloping,
        0: Disable hardware enveloping.
        (Use amplitude specified in bits 3-0)
Bit 3:  } Amplitude (0-15)
Bit 2:  }
Bit 1:  }
Bit 0:  }
```

The Amplitude Mode bit determines whether the hardware envelope is to be used with the channel. If it is 1 then hardware enveloping is enabled. If it is 0 hardware enveloping is disabled.

The lower 4 bits (i.e. bits 0-3) are used to set the volume value for the channel. The value is exactly the same as in the BASIC SOUND command. The volume ranges from 0 (inaudible) to 15 (loudest) and is arranged on a logarithmic scale, with the difference in volume increasing more between the larger values.

Hardware Envelope Length: One of the main concerns, if you are trying to reproduce a sound, is the length of the hardware envelope. Bearing that in mind, I should probably tell you what a hardware envelope is! The hardware envelope is just like the ENV command in BASIC, in that it allows the volume of the sound to alter throughout its production. This is especially useful when you are trying to copy instruments, such as a Piano, which has a sharp attack and a relatively short sustain.

The length of the hardware envelope determines how long one cycle of the envelope will take. This in turn determines how long each step in the hardware envelope is, since the number of steps in each envelope is constant. Confused? It may be a little clearer after you've read about register 13.

The two 8 bit registers, R11 and R12, control the length of the hardware envelope. The time that the hardware envelope will take to complete one cycle is calculated thus:

```
Cycle Frequency = (PSG clock/256)/register value
Cycle Period = (4*Register value)/15625
Register Value = (15625/4) * Cycle Period = 15625/(4*Cycle Frequency)
```

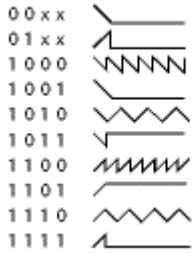
The bottom equation is the only one that you'll really need, but I thought I'd save you a little arithmetic! As you probably guessed, the register value has to be an integer number, but that shouldn't cause too much of a problem - who's going to notice the odd few ten thousandths of a second?

I should also point out that the PSG clock frequency in the CPC is 1Mhz! You may need to know that to calculate the Cycle Frequency!

Hardware Envelope Shape: This is a 4-bit register that indicates to the PSG what kind of envelope will be used. Since it is a hardware envelope, you are only allowed to specify one at a time, unlike the BASIC ENV command. This means that more complex envelopes will have to be produced by altering the volume values 'manually' by means of registers 8-10.

The pattern of the lower 4 bits of Register 13 determines the shape of the envelope. Unfortunately, there are only 10 different envelope shapes, so some combinations are obsolete. The whole selection looks something like this:

Bits 3-0 Envelope shape



'x' indicates that the value is ignored.

The 'Envelope Shape' patterns determine the sequence with which the volume level changes. The peaks on the pattern show the greatest volume, and the troughs show the lowest. It might also be handy to know that each pattern is supposed to have sixteen states per cycle, so you should be able to calculate the length of each state pretty easily.

To clear up any confusion, the patterns which appear to be repeating in the diagram will repeat ad infinitum, whereas those that flatten out don't repeat, once they have completed one cycle. I was a little unsure of this at first, so I thought it might be wise to clarify it for you.

I/O Port: The I/O port is dead simple to use, once the data direction has been selected from the mixer register. Register 14 acts as an intermediary stage between the data lines of the PSG and the I/O port lines. Whatever is written to register 14 will be latched on to the output lines and whatever is on the I/O lines will be relayed to register 14.

In the CPC the I/O port of the PSG is connected up to the keyboard, so it is used to carry out keyboard scanning. The port can be used as an output, without damaging anything, which allows the Joystick Port to be used for other things apart from input devices. It is possible, for example, to make a parallel interface, which can be attached to the joystick port. More details on keyboard scanning and the I/O port can be found in the next issue.

The End - but Not Quite!

Well, that explains the PSG in all its glory - almost! The PSG is used for keyboard scanning, but the PPI is also, so I've decided to dedicate the next article to this, as well as the Centronics Port Latch. Judging by the size of this article it might be split into two. If this happens (which it already will have, if you're reading this in the second part of the article!) you'll have to wait until the issue after that.

Okay, I'm going for a lie down! [And me too... no program I could find would read the supposedly RTF file James sent this as, so it's been reconstructed by hand, and all the tables, which didn't come through, rewritten with help from Kevin Thacker's excellent website. - Richard] Bye.

Programmers' Patch

Matthew Phillips takes a look at the most popular assemblers that the CPC machine code programmer can choose between

Last time I was trying to transfer some vital files from the Amstrad CPC to the Acorn, using some very unorthodox methods! Since then, the replacement drive belts ordered from that marvellous man Brian Watson have arrived, and once fitted the old three incher has been working fine again. So, have I got any further with the job of developing the patch for CP/M Plus that I was talking of two years ago? Funnily enough, I have not...

This month, I thought we would do something rather more useful, by looking at some assemblers which are available for Z80 coding. The assembler is a vital part of the programmer's kit, which converts assembly language mnemonics into a machine code binary file. For any other language, the equivalent tool is called a compiler, but for some reason it is called an assembler for assembly language, which is the very lowest level of coding.

Raw power

The first assembler I ever used was one which was published as a BASIC listing in Computing with the Amstrad, July 1985. It's not bad for what it is, and various modifications were published in the letters pages of that magazine over the following months. I wouldn't on the whole recommend it now, as there are much better alternatives available free of charge, but the course on machine code which ran from January 1985 to December 1986 in that magazine is still one of the best introductions to machine code on the Amstrad.

I would also recommend Mastering Machine Code on your Amstrad 464/664/6128 by Clive Gifford and Scott Vincent, Interface Publications, 1986. Of course, it's no longer in print, but should you see it anywhere, it's worth a read. (Seconded - Richard)

The worst thing about RAW, as the assembler was called, was that it was very slow. Being written in BASIC, if you were assembling anything moderately big, it would take half an hour or more. The worst annoyance was when, towards the end of the second pass, it would find that a relative address was out of range, and you would then have to go back, alter the code, and assemble again. And that was before you got onto the debugging.

ZMac and fries please

After four years of extremely slow assembling, I joined WACCI, and ordered a disc full of machine code tools. Almost all of the WACCI PD library was CP/M stuff in those days. There were several assemblers available, but the one I picked on was called ZMac, which announced itself as "SSD Relocating (and eventually macro) Z80 assembler ver 1.07". I don't think it ever reached the macro stage!

Assembling was a two-stage process. First you use ZMac to convert the assembly language source files into relocatable object files, and then you use ZLink to join the object files together to produce your CP/M executable .COM file. This concept of a linker is very common when you are using "real" programming tools. It means that the source code can be divided up into manageable chunks. Also, if you have only altered one of the source files, you only need to pass that one through ZMac, rather than reassembling all of the object files.

I wrote a little CP/M application (assembled with ZMac and ZLink) called ZBin, which took the place of ZLink and produced Amsdos binary files. This is still available on WACCI PD disc 2 (I believe) or on disc 131, the Music and MIDI disc. I used this combination of ZMac and ZBin until very recently. It is pretty quick, and the only disadvantage is that ZBin will not link more than one object file together. You also have to use CP/M to assemble the code, which might put some people off.

Obligatory Arnor bit

A couple of years ago I got a second hand copy of Maxam 1.5. This is a ROM version for use with Protext. I had been using Protext for writing all my code for several years, so Maxam is the logical assembler to use. It is now available from WACCI. The last details printed on how to get hold of the ex-Arnor products were printed on page 23 of WACCI 132 (July 2000), but I think it is planned to make some of them available on the new web site. Perhaps it's time for an update article from somebody.

I have not done much CPC coding since I got Maxam, so I haven't really got into it yet, but it is certainly the best assembler available for the CPC. It does not have a linker, so when you assemble the code, you have to assemble all of it, every time. You can at least divide up the source code into several files, which makes things more manageable, and these subsidiary files can be inserted into the main code using an assembler directive. (Or, if you like, you can simulate a linker in your own code by writing two or three separate programs, which communicate with each other via a jump-block. This is how I wrote RoutePlanner PCW - using Protext and Maxam 1.5 on the CPC! - Richard)

Maxam has various features which put it in a totally different league to the assemblers I used to use. The most important of these is conditional assembly. This allows you to make several different versions of software all from the same source code. To take an example from last month, where I had three different routines to reset the different MIDI interfaces, you could instead have a code fragment like that in Listing 1.

At the start of the code you would set the labels RAM and EMR according to whether you were wanting to produce code for the RAM Music Machine, the EMR interface, or the DHCP one. The following would set things up for the EMR interface:

```
RAM EQU 0
EMR EQU 1
```

When the assembler comes across the 'if' statement, it evaluates the expression and then includes the block of code after the 'if' only if the expression is not zero. If you work through Listing 1, you will find that by setting RAM and EMR appropriately, you can get the three routines featured in the assembly listing last time under the labels .ramreset, .emrreset and .dhcpreset.

It is important to remember, though, that these choices have to be made when you are assembling the code. After that the user is stuck with them, making it a technique which is best reserved for fairly major alternatives to support different kinds of hardware.

By careful use of conditional assembly, a software house could even maintain CPC and PCW versions of the same package using one set of source files. To give another example, you could produce a specially restricted version of some software for free distribution, allowing you to assemble the full commercial version from the same code.

In general, it is a good idea to re-use code. If you have a piece of code that works, guard it carefully, and use it whenever it is appropriate. Writing a new piece of code each time to do the same job only increases the chances of bugs getting in again. Modularising code by dividing it up into manageable files and using conditional assembly techniques help you to re-use code.

Going native

The main reason I have not used Maxam much since getting it is that I now have something better! Now those of you who have been reading closely may have noticed that I said that Maxam was the best assembler... for the CPC.

The assembler I now use does not run on a CPC, but nevertheless produces Z80 machine code, and even Amsdos binary files. It is what is called a cross-assembler: an assembler running on one computer system which produces code for use on another computer system.

This particular cross-assembler is called Zmac, but has nothing to do with the CP/M program of the same name. This Zmac is written in C, and so can be compiled to run on a large number of

different machines. It has been used on DOS, Windows, Unix, and RISC OS, and probably a few other types of system.

It was originally written in 1978 by Bruce Norskog (whoever he was!) and has been modified and improved by several people over the years (including a small contribution from me). It is now maintained by Mark Rison, who uses it to develop CPC/IP, the CPC internet system that I have mentioned before.

Zmac is free, and available from www.nenie.org/cpcip/, the home page for CPC/IP. You can also download the CPC/IP source from there which is a good example of how to use conditional assembly and other advanced features.

Note that the version on this webpage is source only. You will have to compile it with a C compiler in order to use it on your chosen system. That should not be much trouble, though, if you have a C compiler. A pre-compiled version for RISC OS is available from my web site at www.sinenomine.freemove.co.uk/software/ which includes a nice graphical front end. (And if you want one for the Mac, I've put one up at www.systemeD.net. It's not very pretty. But it works - Richard)

The good thing is that this Zmac is extremely compatible with Maxam, and most Maxam source code will compile straight off in Zmac. The main problem will be if you have used colons to put more than one assembly command on a line, as Zmac does not allow that. (But with a little clumsy hacking, it can be persuaded to - Richard)

See the list in the box for the main differences between Maxam and Zmac. As you can see, Zmac has quite a few advantages over Maxam.

Read the script

Perhaps the biggest feature is that Zmac has macros. Macros are a powerful feature which are only present in the CP/M version of Maxam, Maxam II. At the simplest level, a macro allows you to define a shorthand form for commonly used pieces of code. You might define a macro called 'exbcde':

```
exbcde macro
    PUSH BC
    PUSH DE
    POP BC
    POP DE
endm
```

and then you could just type 'exbcde' to exchange the contents of the BC and DE registers.

But macros are far more powerful than this, as you can use parameters and all the features of conditional assembly to produce some really clever stuff. I've given a quick example of a macro called 'copymem'. This is for copying memory from one location to another.

Usually, you would just use LDIR to do this - but if the block of memory you are copying from overlaps with the memory you are copying to, things can go wrong, as you may have already overwritten some of what you are copying before you get to the end of the block. In these cases, LDDR is required, with the copying starting from the end of each block. The copymem macro in Listing 2 will work out which is needed, and insert the appropriate code. The last few lines show how the macro is called from the main body of the source code. Listing 3 shows the code that the assembler produces as a result.

The other great advantage of Zmac over Maxam is that it is much faster, because you can use it on more powerful modern computers. It now makes a lot of sense to develop CPC software using tools like Zmac, keeping the source code on the hard disc of a PC or Acorn rather than on low capacity floppy discs on a CPC. You can assemble the code faster, and test it easily using an emulator.

Next time we will turn our attention to looking at disassemblers, unless anyone e-mails me on progpach@sinenomine.freemove.co.uk to suggest a different topic!

Using conditional assembly to cater for hardware differences.

```
.resetmidi
if RAM
    LD BC,&F8EC
else
    if EMR
        LD BC,&F8F2
    else
        LD BC,&F8E0
    endif
endif
LD A,3
OUT (C),A
if RAM
    LD A,&15
else
    LD A,&16
endif
OUT (C),A
RET
```

Listing 2
A macro for safe memory copying.

```
;This macro corrupts no registers
copymem macro from,to,length
PUSH AF
PUSH BC
PUSH DE
PUSH HL
LD BC,length
if from<&5200
7007 11FF55 LD DE,&5200+&400-1
700A 21FF53 LD HL,&5000+&400-1
700D EDB8 LDDR
        else
        endif
700F E1 POP HL
7010 D1 POP DE
7011 C1 POP BC
7012 F1 POP AF
7013 endm

;copy memory from &5200 to &5000
7013 copymem &5200, &5000, &400
7013 F5 PUSH AF
7014 C5 PUSH BC
7015 D5 PUSH DE
7016 E5 PUSH HL
7017 010004 LD BC,&400
0000 if &5200<&5000
        else
701A 110050 LD DE,&5000
701D 210052 LD HL,&5200
7020 EDB0 LDIR
        endif
7022 E1 POP HL
7023 D1 POP DE
7024 C1 POP BC
7025 F1 POP AF
7026 endm
```

Major differences between ZMac and Maxam

- ZMac allows ' and " strings to be specified using " and "". Maxam only allows " and ', which ZMac also allows.
- ZMac will truncate symbols to 40 characters. Maxam has no such restriction.
- Maxam allows strings to be terminated by end-of-line. ZMac does not.
- When using the READ directive, Maxam will take the first non-whitespace character as the filename delimiter. ZMac only allows ' or " as a filename delimiter.
- AMSDOS restrictions prevent Maxam from supporting nested READs, which ZMac allows. AMSDOS restrictions also prevent Maxam from supporting directories.
- ZMac has a pretty decent expression parser. Maxam does not (no parentheses, no precedence).
- ZMac handles multiple ORGs by inserting padding bytes in the .bin file, but of course retrograde ORGs are not possible.
- Maxam allows multiple statements on the same line, separated by colons. ZMac does not.
- ZMac has many synonyms for operators, directives, labels, hex constants, etc., which Maxam does not support.
- ZMac has macros!



How the CPC changed my life

Angela Cook explains how a DMP-1 printer transformed her love life. Honestly.

A while ago I was in Rob Scott's computer shop (STS Computers, no less). The subject of printers arose and we explained a brief history of the invention to the young employee. Rob mentioned the Amstrad DMP1 printer and we both sighed with the fond memories that it bought forth, whilst the young employee looked naive and wondered what we were talking about, which is nothing unusual.

We were talking about our time with the CPC. It got me thinking to just how much that innocent charcoal grey box has influenced some people's lives, including my own. When I reflect back now I wonder how my life would have been if that purchase in a large computer chain store had been for an Amiga and not the Amstrad that was plumped for. What if we hadn't bought a computer at all in our family?

We got our first CPC, a 464, around about 1988, which would have made me about ten. Within a couple of years, Dad (Arthur) had started buying and selling the odd bit of computer equipment and supplies, wheeler-dealership being a Cook family trait. I started helping Dad with this, handling much of the administration whilst he concentrated on the buying and repairing and things I usually refer to as "yukky and technical".

By 13 I was showing signs of being a rather mediocre writer, and Uncle Clive gave me my first big break. Yes, if you go back to WACCI 53 there's a rather embarrassing couple of columns proudly penned by myself. Once I'd seen it in print, I was hooked.

Write-off

The buying and selling got bigger and bigger (and until recently was still growing, but that's a whole other piece of CPC history) and I enjoyed the work immensely. I also wrote much more for WACCI and eventually worked my way up to the great heights of Amstrad Action. I hounded Dave Golder, then editor, for ages and finally he relented and asked me if I could write a monthly BASIC programming guide. I immediately answered in the positive and was incredibly happy, until it dawned on me I had the faintest clue about BASIC.

Enter Rob Buckley stage left. How I knew Rob, of Radical Software fame, is now lost in the dusty areas of my brain, but he knew about BASIC. I set about hounding him monthly for programming type-ins that didn't work and were constantly way past the deadline.

Work on Amstrad Action grew, and from there I worked on other magazines - mostly computer related, some not - and still do the odd bit of freelance work to this day. WACCI, for me, is directly related to the beginnings of my writing career. Oh, what a lot it has to answer for!

Of course I attended school and various colleges in between times, though my teachers wouldn't always agree. I trained in computing and as a scum of the earth journalist and then decided I didn't actually want a job in London.

Web wide world

So instead I decided to go down the rocky path of unemployment - sorry, I mean self-employment. I wrote freelance, I briefly worked for Rob at STS Computers (www.stscomputers.co.uk) and I gave

private computer tuition.

Then Rob tried to design his own website. Being trained a little in page layout and such (though not nearly as wonderful as Richard) I decided I could go one better. This, along with advice from AA's Simon Forrester, led on to becoming the website designer I now am.

And quite recently I landed a job, almost by accident, in a Further Education college teaching Website Design/Management modules of various courses. These things involve several skills I have learnt in the past years: the discipline of writing and general journalism, elements of the computer industry, being in business and dealing with the public, all sparked off by Merline-Serve and Radical Software.

Through the last 13 odd years I've picked up one or two scallyrags too. Notably: the rather mad Rob Buckley; Richard Wildey and Richard Fairhurst from my Amstrad Action days - when I first talked to Richard Fairhurst he spoke very fast (he's mellowed a bit now) and I was embarrassed at quite how easily a Multiface really does work; John and Carol Bowley from right here within WACCI; and Rob Scott and Paul Fairman from a rather bizarre argument.

Ho ho ho

Of course there are many more friends and acquaintances lost in between, not to mention one or two more adult and romantic (well, maybe not romantic) relationships. But perhaps we'd better stop there.

New Adventures in Cartography

With the right art package, you can create attractive maps on your CPC, as Richard Fairhurst explains

One respondent to the WACCI reader survey said that they'd like an article about making maps on their CPC. As chance would have it, I've recently been getting into all things cartographical, recently enjoying a day at the Society of Cartographers' Summer School. So here's a brief exposition of simple cartography and how you can do it on your CPC.

The British Cartographic Society's website proclaims that "Cartography is one of two professions in the world that is still fun". I'm not sure what the other one might be, but the point still holds.

This is because cartography is as much a creative art as an information science. No map, after all, can ever be a 100% accurate representation of what lies on the ground: if you wanted that, you'd use an aerial photograph. You'd have to scale the photograph down to a manageable size, of course - and in doing so, you'd make details like footpaths, streams and mini-roundabouts too small to be discerned.

That's the art bit: devising a scheme by which such small features are included on your map. The science is in ensuring that the map remains as accurate as possible, not deviating from what the map user will find 'on the ground'.

But such decisions lie in the future. Let's start from first principles - how do you research your map?

Copyright & field surveys

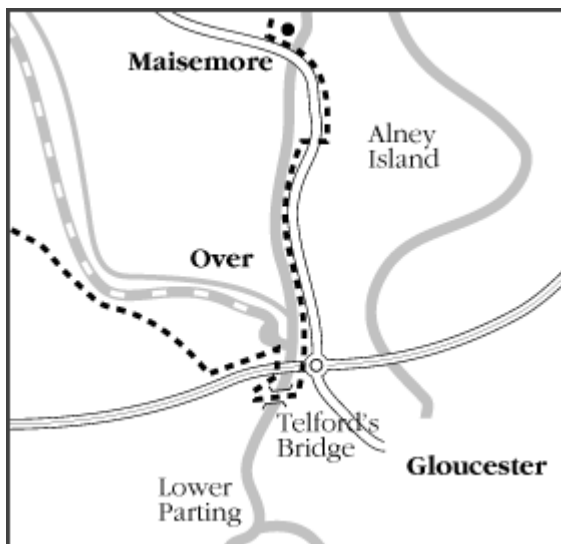
Plagiarise, plagiarise, remember why the Good Lord made your eyes. But always call it research," as the great Tom Lehrer once said (sang, probably). Fine advice - except if you're a cartographer. Before you draw your map, you have to know what it is you're going to draw. It sounds obvious. But you can't draw a road, for example, without knowing where it starts, where it ends, and how it bends between those two points. You could just guess. But then you'll end up confused when you start adding more intersecting roads, and find they don't meet in quite the places you expected them to.

The easy solution is to get the tracing paper out and 'borrow' the lines from an Ordnance Survey map. Don't. The OS is famously litigious, and recently landed the AA with a multi-million pound legal bill for borrowing such data for use in its own road atlases. The AA's downfall came because the OS, like other cartographers, footprints its maps - adding small, deliberate errors to entrap the unwary plagiarist.

You could buy a GPS set. These whizzy little boxes will tell you where you are in the world to an accuracy of several metres. That way, you can trace the co-ordinates of key points along a road, draw them on a grid, and join the dots to get your map of the road. Foolproof, accurate, but laborious.

Or you could borrow the data from an out-of-copyright source. Copyright usually expires 70 years after the death of the author, or (in the case of a map produced by a company) after the end of the calendar year in which it was produced. With OS maps, however, it expires only 50 years after the end of said year, so you can use any maps produced in 1950 or earlier. Not a lot of use if you're mapping the M6, but fine for local roads, railways, canals, and the positions of towns and villages, none of which have changed much.

All of this is a lot of hassle, and if you're just producing a map for friends, the parish newsletter, WACCI, or what-have-you, you're probably best off just guessing.



The computer age

Seventy years ago, the apprentice cartographer Phyllis Pearsall was wrestling with a fearsome card index listing every street in London. 'Mrs P' had decided that London needed a street atlas. Getting hold of the maps was not a problem: she chose to 'borrow' them from the Ordnance Survey (and yes, she did get clobbered for it).

But that only got her a load of lines on a sheet of paper. A street atlas names the roads, and equally importantly, indexes them alphabetically, so you can find where it is you're supposed to be going. In 1933, Phyllis Pearsall needed a card index. But today, she'd do it by computer.

Computers have transformed the field of cartography. Geographical Information Systems are huge databases of points on a map, including those which form discrete entities (such as a church or phone box), areas (such as city boundaries), and lines (such a road or river). From this database, the computer can draw the map itself. And from the same database, it can automatically generate an index.

Of course, a skilled cartographer is still needed to add the "art". A simple computer drawing from a GIS database would probably be messy, cluttered, and difficult to use. The cartographer's job is to tell the computer how to make a map from the database, by deciding how much detail to include, what style to draw each piece of information in, what scale to use for the map, and so on. Should towns with a population over 10,000 have their name in bold type? Should rivers be included at all? What colour should the roads be? Should north be at the top of the map? These are the decisions that have to be made.

Here lies another great benefit of the GIS approach. You only need to do the research once. From then on, you can create any number of different maps, just by altering the instructions you give the computer.

As an example that some of you might have seen, take RoutePlanner. It draws a (very) simple road map on screen to illustrate your chosen route across Britain. The map isn't stored as an Art Studio picture, or anything like that. Rather, RoutePlanner has a database of 5000 towns and road junctions in its memory. For each town, it stores its co-ordinates on a nationwide grid, together with a list of the roads that emanate from there, and where they go to. The map is automatically drawn from this information.

It's a very rudimentary map (the PCW version looks a little better). All the roads are straight lines: all the town names are displayed in capitals. But have a play, and you'll see that the basics of a GIS are in there. When you've zoomed in to the greatest extent, you see B roads, A roads, the lot: but when you've zoomed out to fit the whole of England on one screen, you only see the motorways.

At the other extreme, the Ordnance Survey has recently embarked on a project called the Digital National Framework, whereby the whole of Britain will be tabulated in one big GIS. Their computers will then generate each map automatically, drawing each feature according to a set of

instructions (or what they're calling a Style Guide).

But even if you're drawing a map by hand, you still need to make decisions on what you're going to include, and how you're going to depict each feature. As an exercise, take a few maps of your local area - perhaps a road atlas, an OS Landranger map, and a street plan - and see how they differ.

Generalisation

By now, you should be able to understand that maps are designed, that you have to sit down and work out (say) whether you want to show churches on a four-miles-to-the-inch road atlas. (You probably don't.)

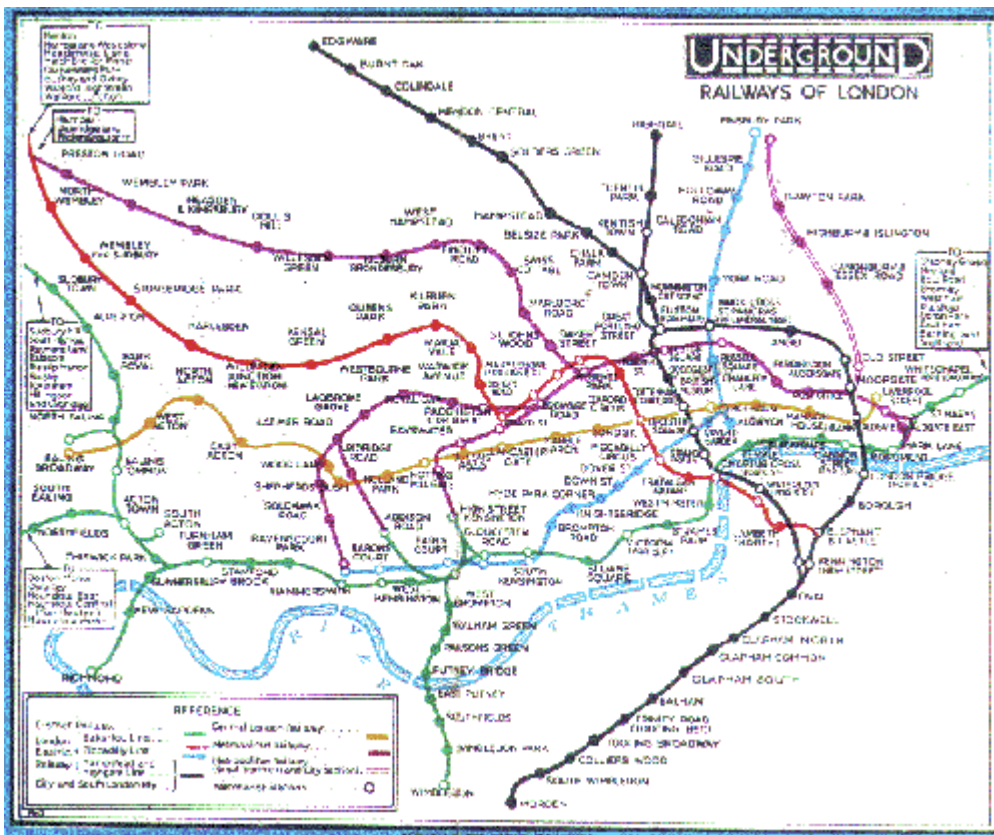
But that's not the end of the artistic bit.

Maps lie. Take a glovebox road atlas, and measure the width of a sample A road - probably a couple of millimetres. Now multiply its width up by the scale of the map. It looks like it's a quarter of a mile wide, and that's not actually true even of the M25.

Now turn to the page showing the South Wales valleys. You'll probably find three roads running parallel up each valley, a railway, a river, maybe a derelict canal, and a load of houses. When put next to each other, the map probably shows them as a mile or two across. Fine. But the valley isn't really that wide. The map-maker has distorted the geographic surroundings of the valley to fit everything in, at the expense of less valuable information - such as the countryside on the ridge either side. You probably haven't noticed this before. You almost certainly wouldn't notice while using this road atlas for the purpose for which it was intended. Now that's an art.

This process of altering the lie of the land to aid legibility is called generalisation. The ultimate example is the Underground map. This bears little relation to the above-ground geography of London, or to the distance between stations - but for getting around the Tube, it's much more understandable than a simple geographical representation will ever be. (Compare the earlier, less generalised version on the next page.)

Computers don't generalise very well unless told how to do it. This is one of the reasons why GIS will never completely replace the human cartographer.



Here comes the CPC bit

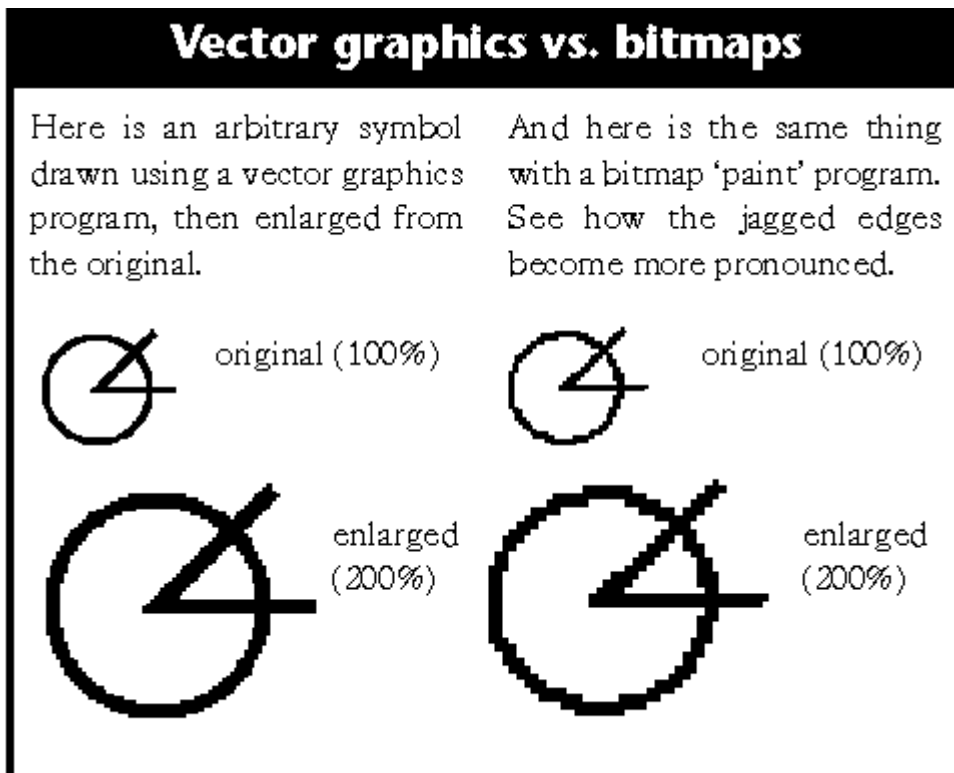
This should be enough information for you to decide what you want to show in your map, and how you're going to show it. So let's turn the CPC on and start to draw a map.

The CPC, it must be said, isn't ideally suited to the task. Maps are best drawn with 'vector graphics' programs, also known as drawing programs. When you draw a line in one of these, it records it in memory as something like "line from co-ordinates 53,10 to 83,40". When you draw a circle, it says "circle with centre 28,9 and a diameter of 10".

All these instructions added together make your picture. This has two great benefits. Firstly, you can zoom in or out of the picture without it looking jagged, because it's redrawing it to your instructions every time. (See the illustrations for more.) Secondly, you can make changes without affecting other parts of the picture. If you want to move the circle, you do so. The computer just changes that particular record to "circle with a centre at 25,6 and a diameter of 10", without affecting any of the others.

There aren't really any programs to do that on the CPC. (Model Universe does, I believe, but that's intended for a 3D environment rather than a 2D map.) Instead, you're restricted to 'paint programs' like Art Studio, MicroDesign and PowerPage. These work, as we all know, by inking in dots - or pixels - on a giant grid. If you've drawn a circle and want to move it, you have to rub out the first circle, redraw anything else that was underneath it, and then draw the new one.

In practice, this means you should draw your map on paper before you fire up your CPC.



Which program?

I'd suggest MicroDesign. It's a nice MODE 2 technical drawing package anyway, easy to use and fast. It gives you a large page area to draw on. And its icons feature is ideal for pre-defined symbols which you'll want to use repeatedly, like churches, phone boxes, pubs, and picnic sites.

PowerPage 128 is efficient when you're used to it, and it has good text features for adding annotations to your map. It only draws straight lines, which isn't a lot of use unless you're mapping Roman Britain, but a succession of them together will give you a passable arc or curve.

SD Microsystems' Picasso isn't a very well-known program, but it has one feature which will aid your cartography enormously: free rotation. You can rotate an area of your drawing by any angle

between 0 and 360, not just through right angles. This enables you to write road names along the roads themselves.

Art Studio, of course, is a famously capable art package capable of operating in all three screen MODEs, but its approach is better suited to freehand drawing than accurate cartography. GPaint may be a better choice if you want to use MODE 1.

How to do it

Pick your MODE. MODE 2, of course, is best if you're ultimately aiming for printed output - but if you have a colour printer, or if your map will be displayed on screen, you may appreciate the colour of MODE 1.

Most of your map will be drawn with straight lines and arcs. You can simply draw one line for each road or path, of course, and that will provide a basic, readable map. Many professional maps, however, use a technique known as casing. This means that the line has differently coloured edges, so an A road will be shown as green with black edges. If you're working in MODE 2, there's nothing to say that the road shouldn't be white with black edges.

There are several ways to do this. You could draw two lines, one for each edge of the road, then fill in the middle in a different colour using the Fill option on your art package. If you have the option of drawing lines of different widths, then draw (say) a thick black line to start with, then a thinner green or white line on top of it.

Don't forget that your art package may offer an option for dotted or dashed lines. These are a great way of adding meaning and differentiating between different types of road: on an OS map, for example, footpaths and county boundaries are shown with various combinations of dots and dashes. I'm sure the European constituency boundary means something very rude in Morse code.

Use a fill pattern or a different colour to illustrate areas. A fine checkerboard pattern in black and white will look like grey, and is well suited to showing built-up areas. You could even design a more complicated pattern for forests, based around a repeated tree symbol.

Choose a nice, legible font for labelling your map - ideally one without serifs (twiddly bits at the edge of letters). PowerPage comes with a good one, designed by Tim Blackbond.

If you're using MicroDesign, make full use of the icons feature for often-used symbols. But if not, you could try redefining unused characters in your font to give the symbols you require.

Alternatively, just draw them at the side of the map, and copy to the right position whenever you need one.

And have fun!

Common line symbols

A simple line can tell a dozen stories – even in a black-and-white map. Here are a few examples you could reproduce on your CPC.

Dual carriageway	=====
Single carriageway	====
Unfenced road	-----
Footpath	-----
Railway	#####
Canal	-----

GIS on your CPC

As demonstrated by RoutePlanner, there's no reason why you can't implement a simple GIS on your Amstrad. A practiced BASIC programmer will be able to grasp the fundamentals easily, so I won't expound further here - except to say that I'd be fascinated to see anything that you might come up with.

With thanks to the nice people at GEOprojects of Reading, who provided much help for my mapping feature in Canal Boat magazine, December 2000, and hence indirectly for this article. My on-going railway map project can be seen at www.systemeD.net/atlas/.

The British Library in Euston Road, London, is currently showing a splendid-sounding exhibition called the Lie of the Land, illustrating how maps have deceived and misled over the centuries.

Books about cartography aren't easy to find except in the most academic of bookshops (Blackwell's in Oxford is pretty good), but if you can locate it, I'd suggest a weighty-ish tome called Elements of Cartography as a good introduction to the subject.