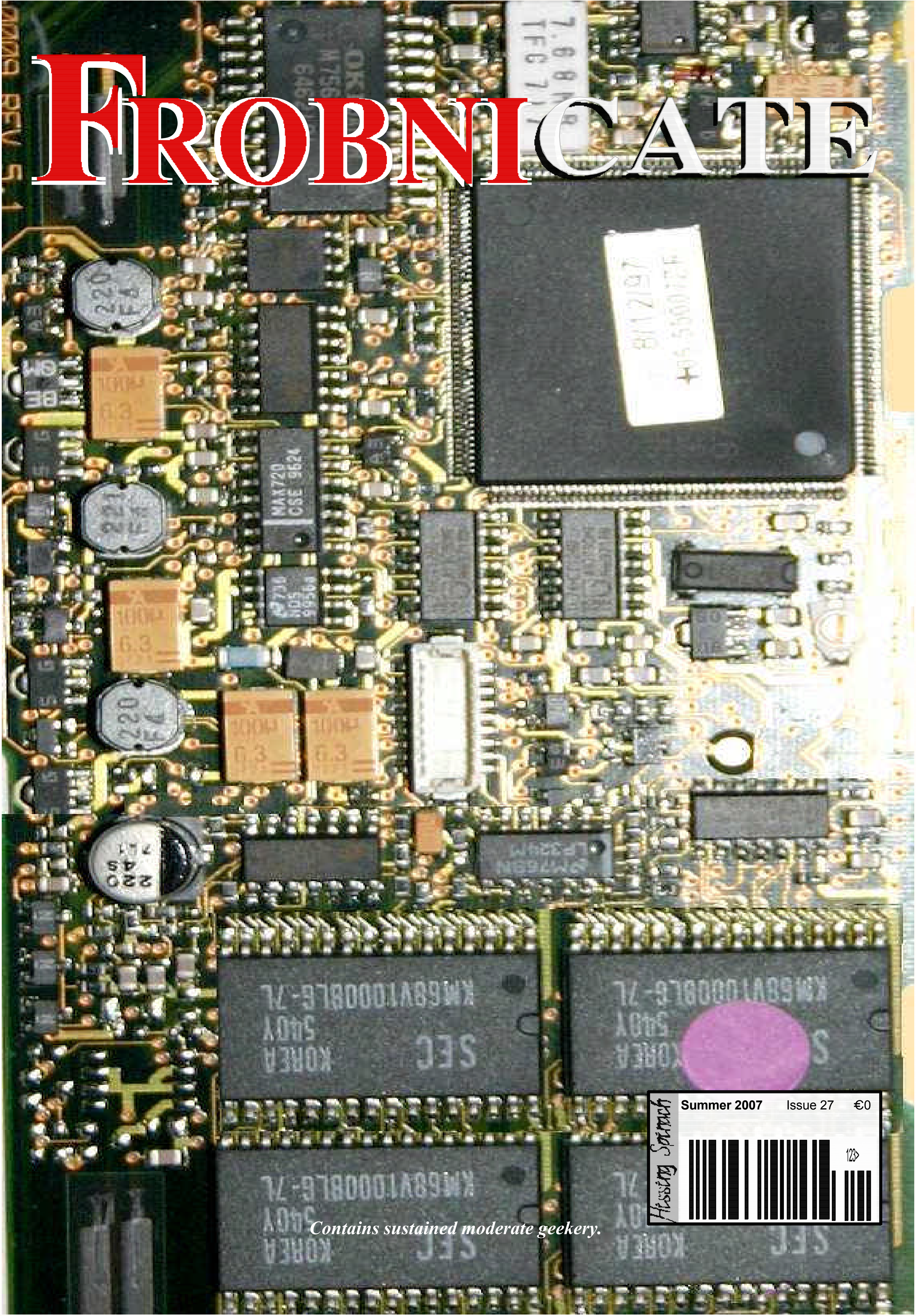
FROBNICATE



Hissing Spinach

Summer 2007 Issue 27 €0

123>

Contains sustained moderate geekery.

Index:

[#29, 2007/06/26 20:20 CET]

Chocolate pudding day

Credits:

Designed, written, and created by Richard Murray.
"Mom's Page" written by Stephanie-Jane Murray.

Images that are not my own have attribution.

a. **You may print this document provided it is unaltered.**

b. **This document may be freely distributed in an unaltered form.**
(if you wish to convert this document to a different format, please contact me *first*)

c. **You may not charge any fee for passing on copies of this document (in electronic or printed form) *except* for "reasonable" media/printing/postage fees** (which total no more than 5 euro; approximately £3.15 sterling or $5.60 US dollars). **Please note that nobody is 'authorised' to provide printed versions of Frobnicate, so obviously we are unable to control the quality of any such prints made.** Frobnicate incorporates colour images/logos. Don't settle for black & white unless it's *very* cheap!

d. **The contents of this document are Copyright © 2007 Rick Murray, unless otherwise noted.**

e. **All reasonable care is taken in the production of this document, but we will not be legally liable for errors, or any loss arising from those errors. As this document may be of a technical nature, do not do anything you are unsure of. Reliance is placed in the contents of this document at the readers' own risk.**

f. **You may quote sections of this document within other documents (either printed, electronic, or otherwise) for review purposes as is provided by European law. There is no requirement to ask for permission first, though it would be nice if you did in case I may be of assistance to you. This does *not* permit the reproduction of entire articles within *other* documents.**

g. **This document, and any legal issue relating to it, is governed by relevant European laws.**

h. If you wish to contact an author and an email address has not been provided, please send your email to the address below, and your message will be forwarded.

Our URL:

http://www.heyrick.co.uk/frobnicate/

A Hissing Spinach production
© 2007 Rick Murray

Keep in touch!

heyrick -at- merseymail -dot- com

Welcome to Frobnicate 2.0

As of recently, there has been an amount of "political" content within *Frobnicate*, and as you would expect this is praised and insulted, loved and hated, in roughly equal measure.

While I will continue to provide small amounts of commentary on current events – this time I refer to the apologies for the slave trade, and Global Warming (again), I feel it is high time to look to getting back to *Frobnicate*'s geeky "roots". Back to the techie stuff. No more insulting "Dubyah" just 'cos he is an easy target...

The only problem is that when *Frobnicate* was new and fresh, the Internet was not a widely known term and search engines did not exist. There is little point in me telling you how to, in VisualBasic, take a big picture from one "picturebox" and create a half-sized version in another... a few minutes spent Googling will reveal to you the API calls that it took me several hours of non-Internet wading through the API to figure out [hint: *StretchBlt* within *GDI*]. So there is little point in writing a "how to" when you can find lots of examples and descriptions on-line.

It will have to be smaller things then. With an emphasis placed on *why* rather than *how*. Any programmer can read the code, good code should be largely self-documenting. So instead of explaining what should speak for itself, you dear reader may get more benefit from explanations of why one procedure, method, or implementation was chosen over another. Now I don't think I'm perfect and I am able to write great code all the time. However I will attempt to justify my decisions, or at least to outline what is going on in my head as I am writing the code.

Hopefully from this you can pick up some ideas and learn a little. Even if you think "oh my God, how could you write such a stupid routine, I think XYZZY would work better...", well, you're thinking, improving, and just being a bit geeky.

That's what it's all about...

This issue sees another change from the norm. While *Frobnicate* will remain as *OvationPro* and *PDF* documents – I am currently rejecting an HTML version on the grounds of complexity – I am making available short "podcasts" of articles within the magazine.
Obviously this isn't terribly useful where code is concerned, how geeky do you have to be to listen to somebody reading: *if left-bracket workflag equals equals false right-bracket newline left-curly-bracket newline indent workflag exclamation equals workflag semi-colon newline indent processblock left-bracket right-bracket semi-colon newline right-curly-bracket newline newline* ? Exactly.

I also plan to introduce short "*Froblets*" which are *PDF* documents of two to four pages, perhaps monthly, which extend or revise topics covered in the magazine. Sorry to *OvationPro* users, but these are only available in *PDF* form as they are written on a *Psion* organiser. The reason for this is a conversion from what the organiser believes is a *DeskJet* printer into a *PDF* document will result in a *PDF* of maybe 30-50 kilobytes. The exact same thing written under Windows, in *OvationPro* or *Word* or what-have-you would run in at several *hundred* kilobytes. Size matters.

This is where ***you*** come in. As always your suggestions are not only welcome, they are positively demanded. Whether or not you read *Frobnicate* comes down to one of three factors:

1. A morbid curiosity to see what weird opinion or article I'll come up with next...
2. Somebody suggested this, or you *Google*d it by mistake and thought you'd have a read seeing as you downloaded it and all.
3. You like the magazine, its writing style, articles, etc.

Obviously, I would like you all to fall into the third category. So, please, tell me the sorts of things you would like to see. What you liked, and didn't. All suggestions welcome... •

Too damn sorry...

Well, I am very ***sorry***, but I refuse to apologise for the slave trade. Furthermore, I think it is reprehensible that we're all supposed to feel "sorry" for these heinous actions.

For at this moment in time we are so busy being sorry for all of our cockups and mistakes – we leave the telly on standby, we wear the wrong clothes, eat the wrong food, support the wrong charities, drive the wrong cars, use the wrong carrier bags, have the wrong light bulbs, give our kids the wrong butter, vote for the wrong party, watch the wrong TV programmes... and if all else fails, we are constantly accused of creating too many carbon emissions because we don't live like a dead corpse.

That is all too much, and to top it all off, we're expected to be all apologetic for the slave trade... It's beyond the pale. Enough is enough.

Let's get one thing absolutely straight here and now. ***I do not condone slavery.*** In no way, shape, or form can anybody be morally justified in stealing "blacks" (for they were treated as commodities, not as people) from their homes, shipping them around the place, to do the gruelling work the "white man" didn't want to do... in no way can anybody argue that such behaviour is justified.

However, there are some extremely good reasons why I refuse to be "sorry" for these actions. The first and foremost is that it happened two hundred years ago. Why don't I see the French apologising for the Norman invasion? The English apologising for rampaging around Scotland and Wales. While we're at it, they can apologise for India. Oh, and dumping a load of unwanted criminals on the continent of Australia. Let's see the Americans apologise en-masse. They were the religious weirdos that were so weird they got kicked out of Europe. So they found a new country and history recalls what they did to the natives of the place called America. Spain, among others, should apologise for plundering various parts of South America. The Vikings came from where? The Boers came from where? Israelis can apologise to who?

Point is, all though history people have taken to stomping on others. Sadly it is still happening today. Witness Iraq.

As if that isn't good enough reason not to feel obliged to apologise, we must also remember that 200-odd years ago, ***common people did not have the democratic power to vote***. They did what they were told. Black people were not the only slaves. Parentless children, bad children, girls that got pregnant out of wedlock... anybody who has ever read a Dickens novel will know of the concept of workhouses. There is even a word for the bleak forsaken industrialisation – "Dickensian".

The country was variously owned and managed by religious leaders, royalty, and the wealthy. Those who owned land tended to own the people farming the land.

You want apologies? Fine. Let's hear it from the descendants of the wealthy. Let's hear it from the royalty. Let's hear it from the Church of England (who can also apologise big-time for the witch hunts, among other sins). ***Don't expect to hear it from Joe Average***. People like myself and a great many of my readers. For our progenitors had no vote, no voice, no power.

Having said this, I think that apologising *now* is more ego-soothing than anything else. Say "sorry", and your karma will rest happy. Rather like paying a company to plant trees (carbon offsetting) because you feel guilty for not turning your heating down a degree. It doesn't really *help*, it just makes you feel better inside.

So let's ***not*** apologise. Instead, let's try to make a serious effort to make this a better world. Not equal opportunities for minorities (which, in itself, is a blatant discrimination), but true equality. The rules that apply for men apply for women. The rules that apply for whites apply for blacks, Asians, Martians, whatever. And none of this politically correct Human Rights crap. Just good, honest, proper equality. Both at home and in places where we can help overseas. *Comic Relief* is only the beginning. I think Zimbabwe needs to be rescued from itself. How about Darfur? In Iraq it is time European troops ditched associations with America and made an effort to make the country better instead of chasing the ghosts of supposed terrorists. And numerous other places.

For absolute certain it will be an uphill struggle, and it will be a hell of a lot harder than simply saying "sorry", but you will be able to feel good inside because you'll be helping to make things better.

Actions speak louder than words. Let's make it happen. •

Digital television

Introduction

A good friend recently posted me a Silvercrest SL65 satellite receiver. He had wanted a "freeview" box in order to receive the stuff that his Sky Digibox could not receive. [for my non-UK readers, the free channels on satellite and the free channels on digital terrestrial differ]. He got the satellite receiver, which was a mistake (though not for me!), however it is a mistake easily understood when you aren't really sure what all this digital television lark is about *and* people use terms such as "FTA" ***and especially*** since the freebie channels off satellite were (and often still are) called "freeview" which is also a name given to the terrestrial channels. As if that isn't enough, there are no less than *four* types of Silvercrest SL65. The first three are satellite receivers (old model, new model, 12V DC model). The other is a terrestrial receiver – what he would have wanted.

Is it any wonder he was confused by all of this? And judging by the emails I receive, he certainly isn't alone in not understanding it all.

So here's a (hopefully simple) guide to digital television...

Digital?

Once upon a time, the signal received by your antenna (or dish) directly corresponded to what appeared on your television screen. If we imagine this signal is a piece of string, and we make it wibbly like ocean waves, the result of how this would look on a television is vertical bands of light and dark.
Unfortunately it takes a lot of "space" to transmit signals like this. Back in the early days of Sky television you could receive maybe forty channels off satellite. Now there are almost as many televangelist channels, twice as many trying to sell you garbage, and *hundreds and hundreds* of other things to watch.

This is all possible because of "digital" television.

The way digital television works is it looks to see what changes from moment to moment. Instead of transmitting the entire complete television picture twenty five times per second (as old-fashioned television does), digital will send the complete picture maybe once a second, followed by directions on how to *alter* the existing picture in order to make what you see on your television look correct. When you realise that you can also "compress" these parts of the picture, as it is now computer data and not just a television picture, you can see how it is possible to get a hundred different channels in the space that used to carry ten.

Understand, though, that ***all*** future broadcasting will be digital. Starting this year, and ending in 2012, the older analogue channels (in the UK) will be switched off around the country. It's much the same in Europe.

No more analogue?

What this means is that if your television only accepts analogue signals – you have to wobble a switch or push buttons to tune in to a UHF channel (your tuning range will be marked 21 to 69), which is probably ***most*** televisions in use, then it will cease to work once the analogue signal is switched off. You will need to have a digital receiver, either a satellite dish, or a digital terrestrial receiver to hook into your old TV.

Digital Terrestrial Television (DTT)

Your existing television antenna is not useless. The end credits of *Coronation Street* (famed for a shot looking across a bunch of roofs with lots of aerials on chimney pots) will look the same. However instead of the old "big" analogue signals, it will receive digital signals. If you don't have a television that can accept such signals directly then you will need a box to do this for you – prices in the UK range from £20 upwards depending on quality and capabilities.

When purchasing such a receiver, ***do not*** be thrown by buzzwords such as “DTT” and “freeview”. Explicity ask, and ensure, that the equipment you are about to purchase is for ***terrestrial*** digital channels.

“Terrestrial”, by the way, means those signals broadcast on earth, travel through the air, and are received on earth. Like ‘normal’ TV. Not like satellite TV, which travels tens of thousands of miles!

Satellite channels

The other option, and the option I feel is more flexible, is for satellite reception. I do not need to say much about this as Sky has been digital for years, so much of the equipment around is for digital reception. However every so often you will come across an analogue receiver – ***do not buy it!*** There is *nothing* on the Sky satellite that you can receive. On other satellites, you’ll find CNN (in English) and an ever-decreasing selection of German channels. Unless the price is good and you really want the dish, purchasing an analogue receiver these days will be a waste of money... especially seeing as a little friendly digital satellite receiver can cost as little as £30 – in the UK keep an eye on the special offers in a chain of shops called “*Lidl*”.

Which is best?

This is rather like asking “which mobile phone is best?”. It depends a lot upon what you want, and what you expect.

If you get a *good* analogue signal for BBC1, ITV, Channel 5 and so on, then quite possibly a digital *terrestrial* receiver will be the easiest for you to set up – it sits between your existing aerial and your television. Let it “autoscan” to find all the channels it can for you, simple.

You need very good or excellent reception.

If you would like a wider range of channels, though not necessarily a more useful range, then you should opt for satellite. This is more a necessity for those in weaker reception areas, and obviously for those living overseas. ***C4** and **C5** are not yet available for “free”.*

How about a one-size-fits-all box?

Well, given that digital terrestrial uses an encoding system called “MPEG2”, and digital satellite uses an encoding system called “MPEG2” and DVDs use an encoding system called “MPEG2”... you really would have thought that *somebody* would have put together a box that can play DVDs, tune to digital channels off digital on either terrestrial *or* satellite, and perhaps even throw in digital radio and MP3 playing too.
You’d have thought...
...so if you happen to know of such a receiver, please inform the rest of us!

HD? [High Definition]

This is available “for a price” on Sky. The BBC also broadcast an HD channel part time. At the moment, it looks like OFCOM (who licence the ability to broadcast on radio waves in the UK) are hell-bent on flogging off broadcasting space to line their own pockets, so *at the moment* there is ***no*** possibility of HD broadcasts via terrestrial means. Hopefully democracy will kick in and they’ll reconsider their position. It is an expensive outlay for not much in return (as there is only ***one*** free HD channel), so progression to HD in the UK will not be quick. •

What those terms mean...

analogue	Old, traditional way of broadcasting television.
codec	Geeky term for the algorithm used to determine *how* a digital receiver interprets the broadcast, it will be MPEG2 or MPEG4.
Digibox	The generic name for Sky receivers; note that the term “Digibox” applies *only* to Sky boxes and not to “any old digital satellite receiver”
digital	New-fangled way of broadcasting television
DTT	Digital Terrestrial Television
D(V)BS	Digital/Direct (Video) By Satellite
freeview	Has too many different meanings to be useful
FTA	Free To Air – can be watched by anyone
FTV	Free To View – you need a card and a compatible decoder, but you don’t pay a subscription. This is akin to getting Channels 4 & 5 on a Digibox when you aren’t a Sky subscriber.
HD	High Definition. Normal broadcasts are 576 lines in resolution; HD offers 720 or 1080.
MPEG2	The ‘codec’ used to encode and decode video for standard digital television (including DVDs).
MPEG4	The ‘codec’ used for HD television; an HD box will be able to revert to MPEG2 for standard TV. A non-HD receiver won’t understand MPEG4.

More at `http://www.heyrick.co.uk/ricksworld/digibox/`

Experiences with *Iyonix*

Iyonix?

The other day I was able to get to grips with an *Iyonix*, the latest and most powerful machine in the RISC OS arena. I believe the machine to be correctly updated, so I can be confident that I was using the latest versions at the time (mid-May 2007).

It was a bittersweet encounter.

Initial impressions

The machine was already booted, so I cannot give any impressions on what the boot sequence/speed is like. What I can say is that it was *exactly* like my RiscPC, only whizzing along at a blistering speed. *!SparkFS* took a bunch of stuff and zipped it up faster than it took me to create the archive in the first place.

Normally, with an ARM7 RiscPC, and sometimes with a StrongARM, the *one* main time where the machine appears to crawl compared to a PC clocking many times faster is when it comes to serious intensive number crunching – such as zipping with maximum compression. Not so with the Iyonix, blink and it is done!

The machine is *so* exactly like the older RiscPC that there is plenty of stuff that is familiar. The desktop looks practically identical, except for the fact that the Task Manager is represented by a blue blob – which is perhaps better than the RISC OS Ltd cog wheels, but no matter, I'm afraid I'd patch the 3D Acorn from one of the old RISC OS 3.1x Clan updates as the icon. Old habits die hard, you see...

Internet

Connected to an Orange *Livebox* ticking at 1 megabit, the connection was equally slick. Now, I'm used to short bursts of ~18 megabit at my local library, but the sheer fact of the matter is that the speed is only really relevant if you're the sort that wishes to download loads of songs and dubiously-sourced movies. Both 1 meg and 18 meg are somewhat faster than the average person's ability to read. If you need a download, it won't kill to fire it off and set it into the background. And, seriously, if 2 minutes instead of 15 seconds is *such* a big deal, imagine doing it on dial-up! :-)

The *Iyonix* is perfectly matched to decent-speed internet, unlike many older XP machines with service pack 2 (or later) that seem to need to "think" about it first.

This, sadly, is where the happy story ends. You see, there are three notable browsers on the RISC OS platform. There is the second incarnation of *Oregano*, there's the newer (easily upset) *Netsurf*, and then the incomplete *Firefox* port by Peter Naulls (I hope I've spelled that correctly!). If I was a web developer, I'd not give the time of day to either of the first, because they cannot do JavaScript. *Come on people,* the days of saying "only luser script kiddies use JavaScript" are well behind us. It is now, in some form or other, one of the base web technologies. And, yes, it *is* horribly abused. That doesn't justify skipping it.

The saviour? The most technically competent is the somewhat cumbersome "Bon echo". This is what *Firefox* calls itself as there was apparently some sort of 'issue' regarding the use of the name "Firefox". I am under the impression that the project has stalled as a result of this, which is a great (and I mean *hugely massive*) shame.

While my *FilmFour* movie reviews looked okay with *Oregano* and *Netsurf* (and likewise on my machine with *Fresco* and so-so with *Browse*), none of these browsers support the layer mechanism used to 'hide' the extended reviews that may contain spoilers. One browser *did* hide them all (*NetSurf?* I forget), but as it didn't support JavaScript, the "show extended" buttons did nothing.

The one that had the most potential is *Bon echo*, but sadly a peculiar bug rendered the background of the document in black... along with black text and, well, you get the idea.

USB support

Everybody has USB sticks of some description these days. Some are memory devices, some are MP3 players. Some of the more clued-up geeks use one as both. I was pleased when inserting two different MP3 players (*Technika ML-2* and *IItronics IMP-65*), the *Iyonix* recognised them as a removable memory storage device, and allowed access to the files – including supporting long filenames. I was worried about the *IMP-65* as it presents itself to my W98SE machine as a "*USB to IDE interface*" with a storage device hanging off of that. USB accesses are *incredibly* ***slow***. I think it is the handshaking that is slow, as transferring two multi-Mb files is quicker than three icons.

The ***big*** problem (yes, there's a serious problem) is that *writing* to the device is potentially disastrous. My PC ***cannot*** see the the files written, twice, on separate days. I need to run *Scandisk* to tidy the mess of recursive directory structures; copy *all* of the files off, and then use a sector-search-and-recover product to locate the lost files. I cannot get *Scandisk* to "convert lost clusters to files" as the last time this happened, *Scandisk* tried to convert a couple of Gigabytes (on a 256Mb device) to files, and promptly hung the machine. Once the lost files have been recovered, the device is then formatted (*Scandisk* ***cannot*** sanely recover from this problem), and the files are copied back again.

I wonder if there is some teeny tiny word-order quirk throwing things astray? It *almost* works. And, yes, I *did* dismount the device before unplugging it.

In summary

The *Iyonix* simply oozed power. With the right software it'd be a serious force to be reckoned with.

And this, unfortunately, is the problem...

Where's the software?

Perhaps the bright shining future of the Iyonix is not so much in RISC OS, but in some version of Un*x? Or, perhaps, in a RISC OS/Un*x transition? It *needs* software. •

VisualBasic optimisation

Why optimise?

In this day and age, it may seem as if there is less and less reason to optimise code. After all, your Visual Basic program may run to a hundred kilobytes or so, and carry a few megabytes of dependencies (DLLs, ActiveX objects, etc). So what exactly is the point of saving a few cycles on a processor clocking in *giga*hertz?

Well, that's no excuse! Decent code is self-optimised, because there's a right way to do something, and plenty of wrong ways to do it.

Consider:

```
  For Posn = 0 To Len(Zone)
    ByteVal = Asc(Mid$(Zone, Posn, 1)) + _
              Val(Trim$(Me.txtKey.Text))
    ...now do something with ByteVal...
  Next
```

Two tiny optimisations are:

```
  KeyVal = Val(Trim$(Me.txtKey.Text))
  For Posn = 0 To Len(Zone)
    ByteVal = CByte(Asc(Mid$(Zone, Posn, _
              1))) + KeyVal
    ...now do something with ByteVal...
  Next
```

Can you see the difference? In one, we must read the string value in the text entry box *txtKey*, convert it to a numerical value, and then add this to the ASCII value pulled from *Zone*.

In the second time, we perform this operation once and assign the result to the variable *KeyVal*, which we then use that value each time we need it.

We also explicitly convert the value to *byte* type.

With *zone* being "wysiwyg" and the key being "22", a million iterations of the first routine takes my machine **47.12** secs. Optimised? **19.05** secs. *Wow*.

You may also think that it may be a further optimisation to assign the length of the *Zone* string to a variable, and use that instead of referring to the *Len()* keyword in the loop. Sometimes this is true – it depends upon how the language you are using is implemented internally. Tests with the above code (over a million iterations) show no significant differences either way – actually it's a ¼ sec *slower*!

If...

Normal programming languages tend to use short-circuit logic in *IF* statements. For example, if you:

```
  If (Smurf = BLUE) And (Uncle = BULGARIA)
```

then if the *Smurf* isn't blue, the value of *Uncle* won't even be looked at. *Visual Basic*, on the other hand, is not so logical. Both variables are examined. ***Always.***

The way around this is to split the *If* statement into separate lines; like:

```
  If (Smurf = BLUE) Then
    If (Uncle = BULGARIA) Then
      ...do whatever...
    End If
  End If
```

However this makes the code bulkier and harder to read. I'd suggest leaving this one until you require the speed improvement. For two million iterations, on my computer, it takes 26 seconds down to 13½ seconds.

You should put the clause most likely to be *false* on the outside of the *If*, as the first thing tested.

Assigning a blank string

Perhaps the biggest surprise is to assign a blank string. You *might* have thought thought this would be good:

```
  MyString = ""
```

The two million iterations takes 16¼ seconds. How about cutting this down to 4 seconds? Easy, like this:

```
  MyString = vbNullString
```

Note, however, that *the two are not the same*. The first is an empty string, the second is a null string. You should be aware of this if you want to call external functions (DLLs, the Windows API...) however as far as VB is concerned, they're the same.

Is the string empty?

A frequent string operation is to check if the string is empty. You'd usually write such a thing like this:

```
  If (MyString = "") Then
```

Two million iterations, it'll take 7¼ seconds. We can pull this down to 6½ seconds by performing a comparison with *vbNullString*.

A more widely compatible method (works with API functions) that is even quicker – 4¾ seconds – is:

```
  If (Len(MyString) = 0) Then
```

This is *so much* zippier.

Character to ASCII value?

We all have times when we want to get the ASCII code of a character. Like this, perhaps:

```
  MyVal = Asc(MyString)
```

If *MyStr* is the string “x”, then two million iterations will set you back 10 seconds of your life.

We can halve that, piece ’o cake...

```
  MyVal = AscW(MyString)
```

Again, these two commands to *not* perform the same function. The AscW version performs a “wide” character conversion, so some characters may return outside of the usual 0-255 range.

Always use dollars

It may look nicer to write code such as:

```
  MyStr = LCase(Left(Trim(InData), 1))
```

Sadly this will perform the following actions: Take *InData* and trim whitespace, returns as a ***variant.*** This is converted to a string for the Mid command, where we extract the leftmost character, returning a ***variant***. This is converted to a string for conversion to lower case, returning a ***variant***, which is converted to a string and assigned to *MyStr*.

For those who use sane programming languages, you may not have come across the dark anomaly that is the “*variant*” data type. It is (sadly) the ‘default’ type in *Visual Basic*, and it can be *anything*. You can assign a float to it, then string-add itself to itself. Frankly I regard this sort of thing in much the same way as people think of GOTOs.

Consider this *valid* code:

```
   MyVari = "22"
   MyVari = MyVari / 3
   Print Left$(MyVari, 2)
```

Aiyaiyai!!!

The way to remove all of these unwanted variants is to specify the code with dollar symbols, which gets VB’s functions to return strings in the first place:

```
  MyStr = LCase$(Left$(Trim$(InData), 1))
```

This can trim a third off of the time it takes to execute the command.

Use specific casting functions

You can always cast a string to a number using Val(), and a number to a string using Format(). This is *highly* inefficient. You see, *Val()* can support all sorts of numeric types (like floating point), plus cope sanely with stuff that isn’t a number. Likewise, *Format()* can output all sorts of fancy things, like outputting “Yes” and “No” for boolean input, or converting a number of days (since whenever VB’s time started) to a user-specified date format. Unfortunately you pay for such flexibility.

Consider the following:

```
  MyLong = Val("1234")
```

That, over two million iterations, takes my computer 75 seconds. We can do it in 35½ seconds using:

```
  MyLong = CLng("1234")
```

Likewise, you can use CStr() to convert something to a string quicker than Format(). There are lots of other conversion types – CInt(), CCurr(), call up VB’s help for more info.

There is a caveat. The following will fail:

```
  MyLong = CLng("")
```

This is because a blank string is not a number, so only replace *Val()* when you know the value input will always be a valid number.

Rounding weirdness

In writing my software to automagically generate my *FilmFour* reviews, I had a lot of problems where movies that scored something like 7½ (out of ten) were flagged as “favourite”, which is reserved for movies that score eight or more.

The following code did not work:

```
  If (RvwData!Score >= 8) Then
```

The reason this failed is simple – this comparison *rounds up*, so 7.1 would match against 8.

I point this out as it took all of thirty seconds of head-scratching to figure out what was going wrong, and to devise a cure. I am telling you as it may take you more than thirty seconds. Sadly, I’m getting used to VB doing the weirdly illogical, as this article demonstrates. Anyway, the fix is to convert to an integer, which *rounds down*:

```
  If (Int(RvwData!Score) >= 8) Then
```

Actually this isn’t quite true. The *Int()* function returns the “integer part” of a number passed. The help file doesn’t actually specify *what* the return data type is. Frankly, who cares. It clips off the fractional part without getting smart-ass about it, so it works like it should.
For consistant behaviour with negatives, use *Fix()*.

What have we learned?

The main lesson here is that *Visual Basic* is really *really* weird. However we have also learned some important optimisation tricks specific to this language. Yes, sadly, this article is only of use if you are a VB coder. It shows some of the things you may like to look into for eking the best from your own preferred language... and as this article shows, you may find some of the optimisations to be rather surprising.
Less so, usually, in C which has a closer correlation between source and generated code. Additionally most C compilers have an option to spit an assembler listing so you can actually *see* what the compiler is doing at every stage. I wouldn’t recommend hand-optimising a compiled C program, but in those rare cases when the compiler is less than optimal, you can usually replace the sub-optimal code with assembler and link it all into the final executable.

Don’t waste time trying to imagine doing this with *VisualBasic* – a language that doesn’t document the *StrPtr* command is hardly likely to provide support for writing code in assembler! This is probably just as well, because my staying with VB means that I have not had to delve into getting an application running under Windows (VB handles this) and much more usefully, I have *not* had to blight my brain with x86 code which... I’m trying to think of an adjective to completely describe what I think of x86 code, I’m afraid all the ones I’ve come up with so far are not PG rated. `:-)`

How I did this

I created a test program. Basically I called the *Timer* function to get a count of ‘seconds since midnight’. I then ran a loop of performing the requested code two million times. Call *Timer* again, subtract the original time value to turn the result into a duration.
I did this over and over, trying different things and observing how long it took to do each of these things. Be aware, of course, that these results are *highly specific* to my computer. A 466MHz Celeron machine running Win98SE (plus unofficial patch pack) in 64Mb. However one thing we can be sure of is that the results will scale. If X takes me 10 seconds and Y takes me 20 seconds; it may take you 4 and 8 seconds respectively. In any case, something twice as slow is twice as slow however fast it is running[†].

Because the results will be dependant on system events at the same time (*WinAmp*, network data transfer, Windows tidying its swap file, etc etc) what I did is to run the tests *three* times and average the results. They were fairly consistant, anyway.

The *Blue Peter* bit:

I have put together a program (executable and VB5 source), plus an extended analysis of these concepts, which you may find at:
`http://www.heyrick.co.uk/software/vbopti.html`
Give it a whirl on your system, I’d be interested to see how it works on newer hardware...

I hope this article has been of some use to you.

Rick, 2007/06/08 •

[†] This is not a sentence I’d fancy translating into French! Awww heck, how about:

> *Dans tout cas quelque chose deux fois plus lent est deux fois plus lent cependant rapide il marche.*

Or, maybe you’d prefer:

> *En todo caso, algo dos veces más lento es dos veces más lento no obstante cómo rápido él marcha.*

Any takers for translations into other languages? Italian? German? Swedish? Japanese? Greek? Klingon? Polish? Latin? Icelandic? Breton? Ursk?
Go on, surprise me – you know the email address!

:) Go figure! (:

a ramble around Rick's mind...

Bwoop! Bwoop! Midlife Crisis Alert!

So I am tidying up my reviews of movies on *FilmFour* and correcting some really stupid typos when Bananarama's "*Love In The First Degree*" comes on *The Vault*. For those without access to FTA channels of the Sky birds, this channel plays 'old' music videos – stuff from "the vault", get it? Anyway, this song is playing and I'm bopping away in my chair thinking "hey, I kinda liked this when I was younger" and then on-screen comes the caption... It managed #3 in the charts in 1987. Erm... Erm... Nineteen ***eighty*** seven. ***Twenty*** years ago. When I was 13. Aaaargh! Aaaargh!

Worse still, *Who you gonna call? Ghostbusters!* From 1984. Or how about the bouncy *Tiffany* with her (only?) song "*I Think We're Alone Now*" (recently covered, badly, by *Girls Aloud*. Reached #1 in 1988.

Another one: I was around a friend's place and talking to the teenage daughter about something on TV. It's some fancy art-drawing program. So I tell her that nobody's a patch on *Tony Hart* and *Morph*. She said she hadn't heard of it. I was all like "You know, *Hartbeat*". I yanked Tony's website through their Livebox and played the theme sequence video. An astonishingly '80s style animation played. After asking me what date this programme was broadcast, Becky replied, quite simply, "*I wasn't born yet!*".

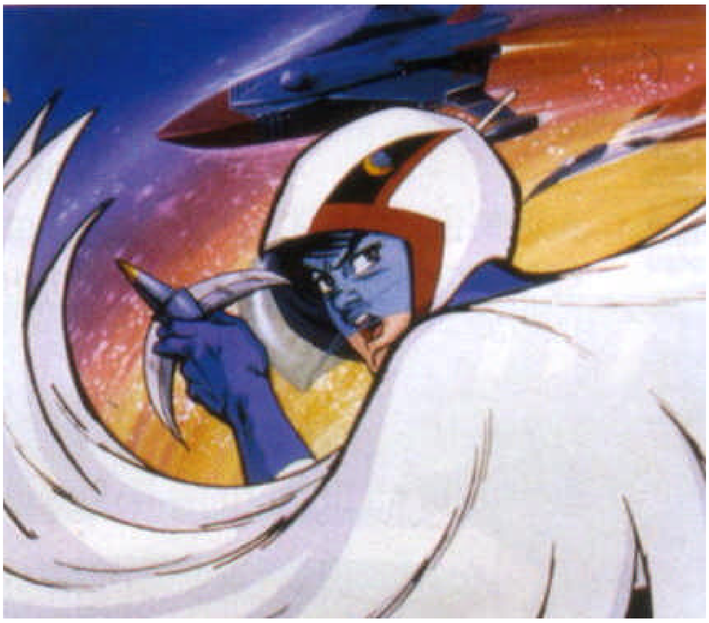
There's absolutely ***nothing*** you can say to that!

That sci-fi cartoon

In a recent issue I asked if anybody remembered an animated TV series from the early '80s with a blue/red (and yellow?) plane that could also 'fly' underwater. I had a few suggestions and URLs, but nothing pointed me to the correct thing.

I was reading an issue of *Coyote* magazine, a Japan/manga specialist publication, and I came across something that looks remarkably like what I remember. It is talking about the French release of a 1978 film called "*Gatchaman: The Movie*". If this then spawned a series, the timing would be about right. However I remember the likes of *Ulysses 31* and *Dogtanian* (*He-Man, Thundercats, etc*...) and I can tell you that the girl in *The Mysterious Cities Of Gold* is called "Estibalitz" (or something very similar, I'm not sure of the spelling); so I would have thought I'd have remembered a name like "*Gatchaman*".

Well, here's the piccy, does it ring any bells for you?



Scanned from Coyote magazine #22 (mai/juin 2007) p23

The political bit

We can keep this short and simple – can somebody explain why the British police have new powers to stop and interrogate any citizen (before, you *could* refuse to say what you were up to and they'd need a reason to pursue it) while *in the same week* the self-same government voted in (on the technicality of something like five *hundred* abstentions) to exempt themselves from the Freedom Of Information laws so they don't need to disclose how they're frittering away taxpayers money. [looks like the Lords killed this one]

Of course, people will say "if you have nothing to hide than you have nothing to fear". This isn't even approaching a valid answer. I mean, you could have heavily armed police on the street able to open fire on criminals. But don't worry, if you've nothing to hide then you've nothing to fear. Funny, that doesn't sound terribly convincing now, does it? And worse, while the police need "reasonable suspicion" in order to haul you off if you refuse to cooperate with somebody nosing around in your life, such "suspicion" is easily manufactured. *It need not exist.* Simply say evidence cannot be disclosed for "reasons of national security". Those in power are so worked up over terrorist phantoms that such excuses will wash for a while yet; especially if some sort of deranged twit decides to leave the guts of a toaster in a backpack in a crowded place (airport car park, train station, etc) and phone in a "warning" to the *BBC* (thus ensuring the media get the story ASAP). In fact, scrub that, phone it in to *Sky News*, they're more likely to portray it as the imminent end-of-the-world. The result? Instant chaos. Okay, it probably won't take long to discover it was a toaster all along, but people will be only too happy to stab each other in the back over the "shocking lapses in security" in order to ensure the blame doesn't land in their lap. Somewhere along the way, a back-bencher will call for a total ban on wearing backpacks in public places. *Think I'm joking?* I can see this. Can't you?

Reality TV, again...

Since the last issue, the world now knows the name *Shilpa Shetty*. Or maybe "*Shilpa Poppadom*" (one of the more polite insults given) – oh, and give the braindead perpetrator of that comment Brownie Points for actually knowing what a poppadom is. Wow.

Now? Some girl called Emily has landed poor old Channel4 in the quagmire (again) for referring to somebody as what ITV's teletext delightfully wrote as "`n*****`".

It happens!

This does not excuse racism, but I'm afraid it beggars belief that people want "reality TV" and then they get all freaked out when people start acting "real". There are some whites that say unpleasant things about blacks. There are some religious people who say unpleasant things about the followers of a different belief system. Thin people insult fat people. Everybody insults smokers. *It happens, deal with it!* Aren't the public supposed to be able to vote people off the show? I don't know – I only watched it *once*, when Jack Dee won. I don't even get C4 on my setup. And if I did, I'm afraid I've more important things to do than watch a bunch of people in a house. It's like the Victorian idea of "paying to see the asylum inmates" crossed with an ant farm.

Emily should have been left in the programme. If her racist comment was that offensive, then just ring up and vote her the hell off! The message'd be clear. But please, *reality isn't politically correct*. The United Kingdom doesn't have a video camera for every 14 people because the country is a sweet place where everybody loves everybody else. Reality can be a blast. And sometimes reality can suck. And you know what, sometimes you'll meet a person with views you don't like. People who use words like that out of cluelessness or intentional malice. People that support the National Front (which is basically politically sanctioned racism). Simply exorcising this stuff from the media isn't going to make the problem go away.

Thanks John!

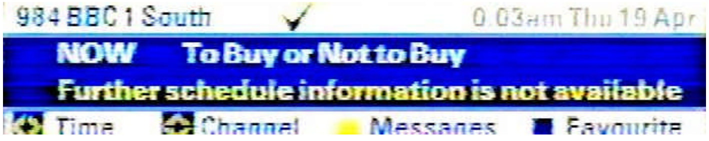
Just a quick *thank you* to John Williams... My old Presario monitor (Samsung chassis) started to slowly get dimmer and dimmer. For a while I could compensate by twiddling the HT BRIGHT but that'd only work for so long. In the end, it wasn't viable to use as a display device. I suspect there may have been a problem with the supply to one of focus anodes, but having no tools for this sort of thing, it isn't exactly a home repair job. John loaned me an old AKF53. An Acorn-badged Microvetic. It's a weird critter. It can *always* sync to 800×600 at 60Hz. *Sometimes* it can manage 75Hz. It can't do 85Hz. *Sometimes* it can manage 1024×768 with the edge being a black bar down the middle of the screen and the sides back to front (i.e. horizontally displaced by around 50%). There are also a lot of bizarre clicking noises (relays?) when the computer starts and the display mode changes, for DOS then for Windows boot, then DOS again, then Windows boot again, then finally the working display mode. However for a monitor of this age, and unused for a while, the colours are good and the display is stable. That's all I ask. *Thanks John!*

Demonising the geeks – *a tad more politics (sorry!)*

Well? What's this with wanting to clamp down on global warming created as a result of computers? That these computers do as much damage as all these dreaded airplane flights?

Over recent years, computers have made staggering leaps in efficiency. My first 14" monitor was rated at 1A (does anybody have a formula to convert this to watts?), the second at 0.7A, and the one I'm using now (John's one) is 0.5A and it is probably 15-odd years old. For more efficiency and a bigger display, I could get an LCD panel, a device with many attractions that is rapidly replacing cathode ray tube devices. The Acorn RiscPC reputedly was not able to quality for a "GreenPC" label as it had no low-power mode – it was not seen as worthwhile considering that full operating mode was less than many PCs of the time used in 'standby'! My laptop uses even less power.

Furthermore, isn't this somewhat hypocritical in a country where the "OFF" button is fast disappearing? There is *no* way to turn a Digibox off without pulling the plug. When it is in 'standby', it isn't really. There's just no sound and no picture. The tuner, MPEG decoder, processing... all *fully functional* in order to keep the programme guide up to date. In fact, this is such an expected part of the operation of the box that if you put the thing in standby and disconnect the dish, the time-of-day clock simply freezes. Upon reconnecting, because the EPG updated before the clock did, note the time...



984 BBC 1 South ✓ 0.03am Thu 19 Apr

NOW To Buy or Not to Buy

Further schedule information is not available

Compare this to a simple "free to air" digital satellite receiver such as the *Silvercrest SL65*. When you press the standby button, most of the unit powers down. The quoted power consumption in standby is ***three*** watts. You could run twenty such receivers in standby for the equivalent cost of a 60W light bulb, which is five or six for the equivalent consumption of an energy-saving bulb.

Additionally, the Silvercrest has an actual on/off switch on the back.

So what gives? The simplest way to deter excessive power consumption would be to increase the tax on electricity. This would not only target people who leave stuff on standby but also old inefficient fridges, those who have yet to turn their washing machine down to 30°C like those polar bears say we ought to, and generally the regular thoughtless waste of electricity most of us are guilty of at one time or another.

But no. They intend to demonise the nerdy side of life. Why? The answer is simple. *Revenge.* They really didn't like, and probably won't live down, the massive anti-road-charging response on their website – you know, the one that proves that democracy is alive and well in Britain... and those that make the rules have done such a good job of demonising SUVs that they hope they can damage the geek culture in much the same way. It may work for Joe Average, who doesn't get the scene anyway, but for the rest of us – we're on-line. We're connected. We talk to each other. We know what a bunch of useless hypocrites they all are – what with foisting digital television on us, which isn't a bad thing per se, but it'll require every household to have at least one set-top-box which no doubt will suffer the exact symptoms that they are making such a fuss of, and then to complain about these things. Such audacity!

How could we better improve the efficiency of these set-top-boxes? *Quite easily.* When the box is switched into standby, the tuner and MPEG stream decoder is disabled. The device then schedules itself to turn on briefly every half hour to update the programme guide. If tuned to a dedicated channel (such as `11.778V` on Sky), the update process should take about 30-40 seconds. If a regular channel (or DTT), the update should take a few minutes at most. There should be a "bad weather trap" that if it takes too long, it'll give up instead of wasting power trying. I mean, if reception is too bad for updating the EPG, it's probably too bad to watch TV. For firmware updates, the box will obviously stay on for the update.

So now we're talking a couple of minutes an hour instead of always-on. An improvement, no? • (2007/06/11)

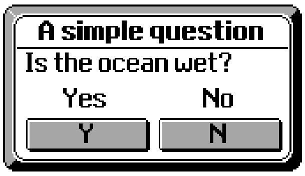
Hacking OPL programs

What the heck is OPL?

OPL, short for *O*rganiser *P*rogramming *L*anguage, is a BASIC-like programming language supplied on the Psion organisers... from the chunky series 2's to the series 3's (and Acorn's PocketBook range), the series 5's, *Revo*s and so on...

The program is procedure-based. Written in the word processor (faking a 'code mode'), and then *translated* to a form of P-code, OPL is a popular way to to write applications for the Psion machines. This is certainly enhanced by the fact that the OPL environment offers ridiculously simple access to the API. There are several limitations (parts of a dialogue cannot alter themselves depending upon the settings of other parts), however in return for these limitations, we can pop up a Yes/No box with the following code:

```
dINIT
dTEXT "", "A simple question", $302
dTEXT "", "Is the ocean wet?"
dBUTTONS "Yes", %y, "No", -%n
r% = DIALOG
```



A simple question
Is the ocean wet?

I *was* going to ask if the sky was blue, but I'm sure there is some smart-ass willing to say "no, it is grey, it is raining!".

Why hack OPL?

The problem, these days, is the organiser suffers from memory issues. The one I am using has 256Kb (*kilo*bytes) on-board, plus a 128Kb plug-in solid state disc (SSD).

Of the 256Kb on-board, between 86Kb and 120Kb (or more) is taken by the "Desktop" application. This means screen buffering, OS workspace, etc. At the minimum, you'll always lose around 86Kb. In addition, the unkillable "Time" application claims 8Kb. Not so much, but we've just lost half of our memory capacity to system management. That leaves half to be split between a faked internal 'disc' (so the machine is not useless with no SSDs installed) and the programs it'll run. The organiser communicates with SSDs using a custom serial/clock system not unlike I^{2}C. It offers a rate of data transfer equivalent to the ST506 harddiscs of its day, or older IDE drives. Fairly fast, but not fast enough to use for program execution. Note: Other models of series 3a/3c exist with larger amounts of memory on-board: 512Kb, 1Mb, or 2Mb.

So there are times when you'd like to remove a few unnecessary features from a program in order to re-translate it into something smaller. An alternative, and more dodgy, excuse for hacking OPL is to register the software to unlock the full feature set when you look at the help file requesting payment in a currency that no longer exists, with your being pointed to some CompuServe address for performing the registration. Typing the name/software into Google returns only the copy that you downloaded in the first place...

Whether or not you do stuff like this depends upon yourself and your conscience.

What you'll need

You will require *three* to *five* things:

Required:
- Some version of *RevTran*.
- A program called *Opodump.exe*
- A hex editor – such as *Hextreme*

Optional:
- The series 3a (or 3c) *emulator*
- The OPL translator *S3atran.exe*

If you have memory, you can do most of this stuff on the organiser, however running stuff on the PC is often quicker and easier and benefits from having two things on-screen at once. I shall assume we are decoding using a PC.

Beginning

Copy the `OPA` program that you wish to hack to your harddisc and place *Opodump* in the same directory.

Now run *Opodump* directing all output to a file. If we assume the program is *Jot3a.opa* then the command is:

```
opodump jot3a.opa > jot3a.txt
```

Spotting what went wrong

The most common way to trip up code decoders is by using the fact that the decoder steps through the program instruction by instruction, while OPL goes as directed in the code.

So in the program we can write something like this:

```
IF 0
  PRINT "@:~?£ (^!%' "
ENDIF
```

And then, in the translated program, we would locate the byte that describes the length of the string to print, and set it to a value other than that which is correct. OPL will completely ignore the bogosity because the instruction is *never* executed. The decoder, on the other hand, will accordingly treat the following *n* bytes as part of the string, and then panic upon attempting to reconcile what follows as valid code.

The example shown above is important, as it was part of a program in which you inserted procedures with a specific template, and it would go through and 'obscure' your programs.

The *Opodump* program would report a munged string like this:

```
0402  4F 00          w          PUSHw 0
0404  5B 13 00       empty      Jump if zero to   0417
0407  2B 00          s          PUSHs ""
        *** STACK MISMATCH POPPING w
        *** STACK UNDERFLOW POPPING w
0409  40             w          OP[=]w
        *** STACK MISMATCH POPPING r
        *** STACK UNDERFLOW POPPING r
040A  3A             w          OP[>]r
040B  7E             w          *** Unknown opcode 0x7E
        *** STACK MISMATCH POPPING s
        *** STACK UNDERFLOW POPPING s
```

This is *very* important for 'hacking' OPL because there is virtually *no* guesswork once we understand what is actually going on here. Let's examine it in detail:

`PUSHw 0`	push '0' on to the stack
`Jump if zero`	if '0' is on the stack, jump
`PUSHs ""`	push a blank string to the stack
error error error	lots of errors

This can be mentally translated into:

```
if (0) then jump to 'x'
some crappy broken code
'x', OPL would resume execution here
```

The fix is to load the software into a hex editor and examine it...

```
0400: 01 80 4F 00  5B 13 00 2B  00 40 3A 7E  3F A3 20 28   ..O.[..+.@:~?£ (
0410: 5E 21 25 27  20 8B 92 05  B7 02 29 03  C3 CB 90 85   ^!%' ...).......
```

If we look at the failed decode, we'll see all is okay until after +&0407. The instruction at that location is one to push a blank string to the stack (OPL works by chucking stuff on a 'stack' and then calling functions that expect to find stuff stacked and ready – you'll see much of the P-code is concerned with putting stuff on the stack, or taking stuff off the stack.

Looking at the hex dump, it is funny that the failure point is followed by a bunch of *printable* characters. There are no control codes or high-ASCII. Not until that &8B.

So let's count this up. There are twelve bytes between the &28, &00 (string push) and that &8B (high ASCII), so let's try assigning '12' as the string length. That is &0C in hex, so the dump now looks like:

```
0400: 01 80 4F 00  5B 13 00 2B  0C 40 3A 7E  3F A3 20 28   ..O.[..+.@:~?£ (
0410: 5E 21 25 27  20 8B 92 05  B7 02 29 03  C3 CB 90 85   ^!%' ...).......
```

Saved, *Opodump*ed again, the resultant dump reads as follows:

```
0402  4F 00          w          PUSHw 0
0404  5B 13 00       empty      Jump if zero to   0417
0407  2B 0C 40 3A 7E s          PUSHs "@:~?£ (^!%' "
      3F A3 20 ...
0415  8B             empty      PRINTs from stack
0416  92             empty      PRINT line end
0417  05 B7 02       a          PUSH ADDRl local 695
```

In other words, ***success!***

What remains is to locate all of the other occurrences of this 'problem', and patch in the correct string length. This is where the use of an automated tool runs into problems. *All* of the failure points were &28, &00 followed by a 12 byte string. Always the same, for twenty or thirty of them.

Time consuming to patch, but not something that required much attention. Actually I was paying more attention to *The Bangles* on *The Vault* singing about their *eeternal flaaaaaaaame*!

Once you've patched the string length markers, run *Opodump* again – make sure you've not missed one.

Now time for the real decode

If you have lots of memory to spare on your organiser, copy the *patched* program on to the machine and run *RevTran* on it.

Otherwise, time to close down *Windows* and reboot into DOS. Good luck if you have XP or Vista in actually *finding* DOS.

Load up in "*Safe Mode Command Line*", no point loading drivers we won't be making use of. Then run the emulator. Sadly the emulator does not save state, so you'll need to 'install' *RevTran*. No biggie. Install it, run it, say "wow gosh!" at how fast it runs!

Why couldn't we *RevTran* in the first place?

Because it is somewhat less helpful, the generated program would look like:

```
PROC Init:
!! Bad structure in command 3 at &4AF, 1 missing
item; procedure skipped. !!
```

(and so on for every affected procedure)

Note, meanwhile, that it appears as if an Acornite may have written *RevTran* because the Psion parlance for hex values is '`$`' (along with Commadore and Pascal), it seems Acorn is unique in using the '`&`' ampersand symbol.

We've made it this far, so...

If all went according to plan, you will have, in your OPL directory, a decoded program. Local variables will have lost their names, so will be assigned automatic names in the form "`lc70z%`", however all global variables and the functions retain their names. Additionally, the code is usefully indented to make it easier to follow.

Simply modify it as you see fit, and then re-translate it!

As for our test, simply removing the dopey 'protection' takes the translated program size from 12.4Kb to 11.6Kb. And there is plenty of room for further improvement – procedure and global variable names could be shortened, for example. Each procedure holds a reference of the globals it accesses, so shorter globals, less space required to reference this stuff. •

Rick 2007/06/12

The Frobnicate Quiz

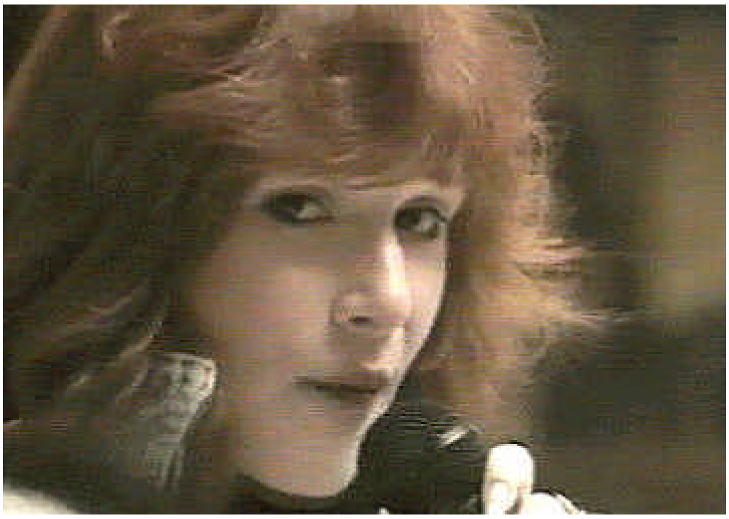
Image from Katherine's videos on ClassicFM TV

In the last issue, I asked "Who is this?". The correct answer is... *Katherine Jenkins*, and the picture was digitised from *Classic**fm** TV* from the video of her song ***L'Amore Sei Tu*** ("*I Will Always Love You*" in Italian).

So here is the question for this issue:

Who is this?

Bonus points if you can say where the picture is from!

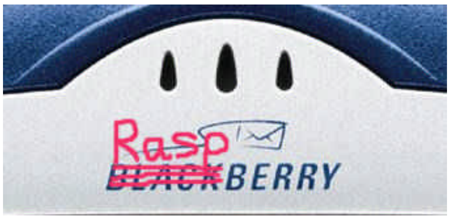


Answer in the next issue.

VeroDes, my veroboard (stripboard) design program, has been released. You can download it *for free* from:
http://www.heyrick.co.uk/software/verodes/

Does anybody have a Psion series 5mx (English) that they would be interested in selling?
heyrick *at* merseymail *dot* com

Mom’s page...



Hello everyone out there in the ether . . . it’s Rick’s Mom back again, after a long break. This was partially due to health reasons. I had a mundane athlete’s foot infection which turned quite nasty indeed. Some doctors have a problem believing their patients can diagnose anything, even something as simple and obvious as a very allergic reaction to the medicine they prescribed.

April was such a beautiful month here – the vegetable seeds all germinated and grew away, the swallows returned three weeks early, we sat outside and sunbathed in unbroken sunshine, everywhere had maximum “wow factor”. That was until May arrived. It seems as if we had June in April, April in May and October in June. As I’m writing this, the west wind is whipping along at Force 7, and Radio 4’s weather person (it sounded like Thomasz) has said that the Met Office has issued yet another severe weather warning for up to 100 mm of rain tomorrow in the North of England. It is actually Midsummer Day today, but you could have fooled everyone, including the swallows, the tomato plants and anyone who has wanted to sunbathe in the past seven weeks. Meanwhile, not a million miles east of here, in Bulgaria and Rumania, they are having one of the worst heat and dry spells ever. This is “the breadbasket of Europe” and wheat prices are going to go up and up. I read that the retail price index in France has gone up by 28% since the introduction of the Euro.

What did you think of the French elections? It is certainly going to be interesting watching M Sarkozy try out his new, up-to-the-minute style of government. We’ll miss the Chirac-Blair stand-offs, though, especially the pictures in the newspapers where both are grinning such totally false grins at one another. M Sarkozy is equipped with an entirely different range of facial expressions.

On Friday morning I went to one of the quarterly meetings at the head office of the association for which I work. Something went manifestly wrong, within the first half hour people began getting up and walking out! This was quite bizarre. My friend Marie-Luce said we were going, too, so we did. On the way home she asked if I had understood what was happening. I said no, I can’t understand the new lady in charge at all. Marie-Luce said that it wasn’t surprising, the lady speaks very fast, very softly and has an accent that is practically incomprehensible to the native French speakers as well. At least I didn’t feel quite so stupid, although I don’t yet know why these normally passive ladies took to “voting with their feet” in such spectacular form. It ought to make interesting reading in our monthly newsletter.

Our cockatiel celebrated his 33rd birthday in mid June (he wasn’t much impressed with the weather, either). He has fallen in love with a colour photograph of Katie Holmes, but please don’t tell Wee Tom that he has competition!

This is being written on Rick’s *Acorn Pocket Book II*, a marvellous innovation for those of us who have bad backs. Instead of sitting at the desk, typing on a megalithic computer (you know, it runs on pulleys, weights and water), I can sit anywhere comfortable and type away on the midget keyboard. The main beneficiary of this magic is my friend Julia in Australia. She now receives a weekly e-mail of about 20Kb.

This issue’s Good Read is “*Blood Memory*” by Greg Iles. It was published in 2005 by Scribner. It will appeal especially to anyone who lived through the Vietnam conflict era. The main character in the story is female, although the book was written by a male. This certainly gives an interesting slant on the female psyche. It’s about addictions, family secrets and loyalty. It’s also a cracking whodunnit, or rather, who is doing it.

One last thing: I’d like to thank Maurice White-Socks for taking such good care of Nikki, my very old Renault 5. You’re a real star! Long may your acetylene torch burn!

That’s it for now. Take care, and I’ll be back soon. •

Me. at ten past ten on longest day, 2007.

Review: *Iltronics IMP65*

Powered by a single AAA cell, the IITRONICS IMP-65 offers an easy to use MP3 player from the cheaper end of the market. That said, the player offers numerous useful features and a clear display which can show file names (long filenames supported) as well as ID3 tag details from the MP3 files.

The battery life is good. Between six to nine hours from a rechargeable. The audio jack is a standard 3.5mm socket (unlike the 2.5mm used on the Technika MP-906). This connects to a pair of generic earphones supplied with the player. Using better semi-pro headphones, the audio quality is good enough – marred only by the fact that the various predefined equaliser modes are not quite “my thing”. I’ve often said that these players should offer a three or five band equaliser that the user can set as they wish.

Switching the player on takes you to a menu with pictures and descriptions (choice of English, French, Chinese (traditional and simplified), German, Spanish, Italian, and Portuguese). Here you can set up the behaviour of the player, check the memory, switch between voice mode and music mode, and browse for music files as they are laid out on the device (including within “folders”).

The equaliser options, as previously said, are lacking. The only one that offers good bass has somewhat muted treble. The “rock” option doesn’t sound very *bright*. So I just leave it on normal.
Much better are the playback options. You can play normal sequentially, repeat, random order, or random order repeat. But it gets better. You can apply these choices to songs in the current folder, so if you have a folder with *Evanescence* and a folder with *Hayley Westenra*, you can choose to listen to the genre the best matches your mood, instead of a sort of culture clash as *serious rock* meets *classical transition* (aka “*popera*”).

Of the user interface, top-left is a slidey-switch. This is used to ‘lock’ the device so you don’t mess up playback by accidentally knocking a control. This switch also serves to *write protect* the device if connected to a computer. The top-right button is a –/+ choice for setting playback volume. Along the bottom, a “Rec” button to begin recording, an “M” button to call up and use the menus, and a –/+ button to change tracks or menu options. On the front, a button that offers On/Play/Pause/Stop/Off.

The IMP-65 can record via a little microphone opposite the headphones jack. It records fairly clearly. The settings quote the recording in Kbps. I have found 64Kbps to be a good trade-off between size and quality, but I’ve not yet loaded any files to see what this means in terms of sampling rate.

The driver disc contains a Windows98 driver that puts the little “thingy connected” icon on the system tray and requires you to have to choose to “Stop using removable media device E:”. This is very useful as I don’t much like the “if it doesn’t appear to be in use, unplug it” approach taken by many, for not visibly appearing to be in use (i.e. LED not flashing) does not account for files being open, etc.
Interestingly, the driver disc contains full binaries for a roll-your-own firmware update, plus firmware tools – though sadly none of the tools recognised my player.

The complaint? One. The backlight. Seven colours (red, green, yellow, blue, magenta, cyan, white). It would be *nice* if you could pick a colour. Unfortunately the colour changes every second, so it can be difficult to read in the dark... not to mention looking like a Christmas tree!

In short: It isn’t an ipod, but it tries hard to be more than its price bracket normally offers. •

Going *digital* part 1

Introduction

The primary benefit of digital television, from the consumer point of view, is the quality of the picture. While sometimes you might see JPEG artefacts, in general the image will be clear and exactly as you'd like it to be. No ghosting, no snow, no sparklies, and none of the other image disturbances that affect the older analogue systems.

It comes at a price. While an analogue picture can 'fade out' as the quality of the signal deteriorates, the digital picture will only last as long as its error correction systems can cope. When the signal is too weak, the error correction will fail and so will the picture.

Note, also, that what constitutes a 'failure' in digital terms may be a reasonably watchable picture in analogue terms.

Resolution

The proper PAL transmission system utilises 625 lines horizontally, however for historical reasons a chunk of these are not used for the picture information – teletext and sync signals live in the unseen part.

The digital satellite and terrestrial systems generate an image that is 576 lines in height, and 720 pixels wide, for a normal 4:3 broadcast. Widescreen is handled slightly differently, as is HDTV offering up to 1080 lines resolution.

YUV

In analogue systems, the signal is broken into two parts. These are the chrominance (called '**C**') and the luminance (called '**Y**'). Basically, this means the 'brightness' is separated from the 'colour' and they are transmitted separately.

The colour signals, in order to consume less bandwidth, are not the Red, Green, and Blue that you might expect. Instead they are the difference between the luminance and Red (called '**U**') and the difference between the luminance and Blue (called '**V**'). By knowing the overall luminance, and the differences between it and two of the colours, it is pretty simple to figure out what the value for the Green should be. It is the difference of all the other differences! **YUV** is luminance+reddiff+bluediff.

By knowing the overall brightness, and the amount of red and the amount of blue, it is not difficult to calculate the amount of green...

So your colour television will receive the YUV signals (brightness, red difference, blue difference) and from this create the RGB (red, green, blue) signals necessary to drive the three cathode emitters in the television tube.

Data rate

We all know the television frames are received twenty five times per second (actually, it is fifty half-frames per second, interlaced; doing it like this reduces flicker). If we do a little bit of multiplication, we can gasp in amazement at the data rate required to keep a raw digital signal intact second by second...

We have 720 pixels across the screen, 576 lines down the screen, 25 frames per second, three components of data, and eight bits per component.

`720 × 576 × 25 × 3 × 8 = 248832000 bits`

One *second* of image data requires just over 237 megabits, divide by eight to turn this into something we can understand. The result is 31104000 bytes, 30375 kilobytes, or just short of *30 megabytes* per second.

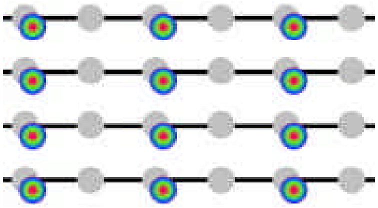
This, very obviously, isn't a workable arrangement!

4:2:2

I first came across this strange marking on a digital Beta camera. Here's what it means...

For each 'frame', we have 720 pixels and 576 lines. The frames are actually 288 lines in height as there is an odd frame (lines 1,3,5,7...) and an even frame (lines 2,4,6,8...) and these are sent in alternating sequence.

In the 4:2:2 system, the luminance (Y) part is sent out for each pixel (an effective resolution of 720 by 576). The chrominance (U and V) is only sent out for every *other* pixel (an effective resolution of 360 x 576). This works because the eye is much more sensitive to changes in luminance than to changes in chrominance, you can double up every other row without affecting the picture too greatly.



Using 4:2:2 coding, we can reduce the data rate to around 20Mb/sec.

Next time...

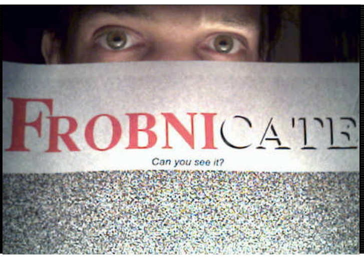
We hit a problem in that the maximum data rate from a transponder is around 34 megabytes. We can fit in a raw video stream, but only one. This provides us with no benefits over analogue. And just where do we think we'd put all those interactive thingies?

Next time I'll continue with digital television... •

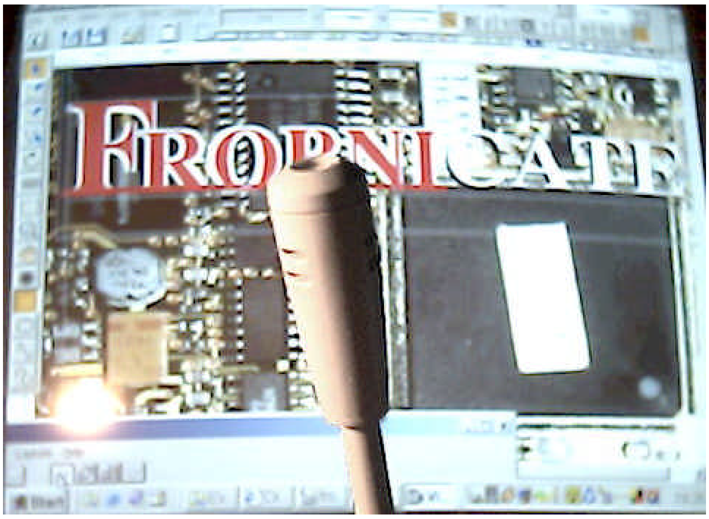
Feedback!

You can write for *Frobnicate*. And if you don't fancy that, ***you*** can write **to** *Frobnicate*!
`heyrick` at `merseymail` dot `com`

The Wrap Party



So there we have another issue of *Frobnicate* in the bag, however an issue with a slight difference. The difference being that selected articles are available as a “podcast”. Or, rather, a series of short MP3 audio files.



Creating the podcasts was simplicity itself. I connected a microphone up to the computer’s soundcard and then ran a program to record from “line in” (and using the audio control to turn off Line In and turn on Mic). The articles were then recorded one by one as WAV audio.

Once the recordings had been made, I used a WAV editor to patch up the audio. The main thing that had to be removed was background hiss. I suspect this may have been due to the close proximity of the microphone and a number of EMF devices. If you listen carefully, you’ll hear the harddisc (high pitched whistle) and the fan mounted on the Digibox (lower pitched whistle), but I think (hope) much of that has been removed. It appears to be more prominent in lower quality earphones.

I also played with the sound, make it a little more interesting.

Lame was then used to convert the audio to MP3, and several different options were tested to get the best compromise between sound quality and file size. I used *Lame* from the command line for more control than many software products provide. For example, as it is vocal I have downmixed to mono. Not only does my software not offer to downmix, if you give it a mono WAV, the resultant MP3 is mono running at double-speed – so it’s buggy too!

Sorry, you can’t beat the command line for the ultimate flexibility. On the other hand, I used WinAmp’s ID3 tag editor as it was easier. :-)

In keeping with much of the Frobnicate ethos, there is no fancy theme tune. Just straight into the articles. Which is pretty much how it ought to be!

As far as this experiment goes, it will be interesting to see how the podcast version is received. Please *please* let me know what you think of it!

Another small change is the use of the bullet marker to specify the end of an article, like a “real” magazine. I would have liked a little square blob, but there wasn’t anything I could find that made it through the PDF conversion. *OvationPro* and *Adobe Acrobat* are Unicode compatible, *Office One PDF Manager* is not.

Shame, because if it was, I could have signed off by saying しよなら, but instead I’ll just have to write it as *sayonara!*

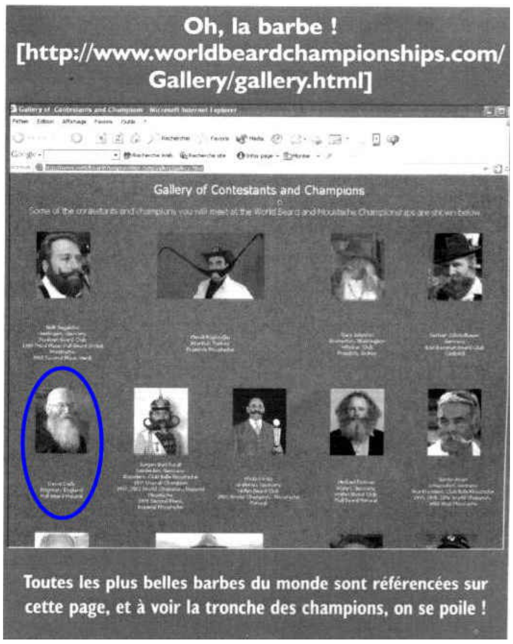
Finally, on the right here is something I saw in a French computer mag. I wonder who I’ve highlighted!

Thanks for reading!

Rick, 2007/06/26

Oh, la barbe !

[http://www.worldbeardchampionships.com/Gallery/gallery.html]



Toutes les plus belles barbes du monde sont référencées sur cette page, et à voir la tronche des champions, on se poile !