

FROBNICATE...

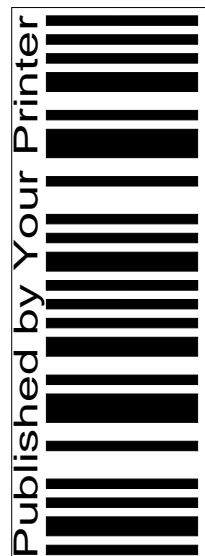
JULY
1995

Issue 2. The ULTIMATE techie magazine for Acorn Enthusiasts.



In this issue:

- ➔ DIY serial port for the A3000.
- ➔ How to use the serial blockdrivers (from BASIC).
- ➔ Internet legislation rumours. Could it affect Fidonet??
- ➔ How to set yourself up as a Fidonet point.
- ➔ Have you heard about Tornado? Read this!
- ➔ A brief look at the latest server - ArmBBS.
- ➔ NMOS, PMOS, VMOS? Huh?



Index:

Page 2	Index
Page 3	Build a DIY serial port for your A3000.
Page 5	How to use the serial block drivers from BASIC.
Page 8	About Internet legislation rumours. Could it affect Fidonet?
Page 10	Easy peeC, 'fuzzy logic' string searching routines.
Page 12	Tornado - the amazing preemptor.
Page 14	How to set yourself up as a Fidonet point.
Page 17	Readers Letters.
Page 18	RiscOSity
Page 19	<1b>[p;i;i;z;i;a - A look at ArmBBS.
Page 20	Froblicate, TLAs and ETLAs.
Page 21	Feetch, feetch! 5 things you didn't know about CastAVote.
Page 22	Advertisements.

Distribution stuff:

Editor	Richard Murray
Contributors . .	Richard Murray, Niall Douglas, John Rex (C routine), Keith Hall (info on ArmBBS), Chris Jackson & Robin Abecasis (for help with the blockdriver bits through the ages!).
Graphics	Richard Murray

This magazine is distributed locally in hardcopy form and widely on Arcade, ArcTic and Digital Databank BBSs as archived files. You may print the files UNALTERED.
Back issues are available from Encina BBS, as are stylesheets, fonts and logos/graphics.

The editor can be contacted by fido netmail as "Richard Murray" at 2:254/86.1.
Feel free to comment or add your own submissions.

Unless otherwise stated, the contents of this magazine including all articles and images are copyright. Copyright and intellectual property rights belong to Richard Murray unless otherwise stated. All copyrights and/or trademarks used are gratefully acknowledged.

All opinions expressed are those of the article author and not necessarily that of 'Froblicate' in general.

All reasonable care is taken in the production of this magazine, but we will not be legally liable for errors, or any loss arising from those errors. As this magazine is of a technical nature, don't do anything you are unsure of.

Reliance is placed in the contents of this magazine at the readers' own risk.



DIY SERIAL PORT FOR THE A3000

This article will detail how you can add a serial port to your A3000 if there isn't one installed already.

WHY?

The A3000 was designed as a cut-down version of the Archimedes system. Built-in keyboard, only one (normal) podule socket, ARM2 soldered in, no serial port by default... It is to the Archimedes range what the Electron was to the BBC micro. Only, the A3000 can be expanded to a specification that is as good as, if not better, than the Archimedes. It's 2Mb memory upgrade limit has been broken. It has provisions for the larger RiscOS3 ROMs and so on.

But in some cost-savings, Acorn in their wisdom decided to omit a few pounds worth of hardware. Now, we shall stick that hardware back in.

STEP ONE:

Go purchase the hardware.

You will need:

6551 serial IC
Costs approx. £3

LT1133CN line driver
Costs approx. £2

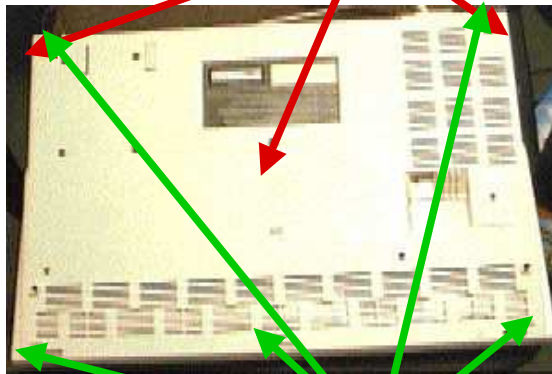
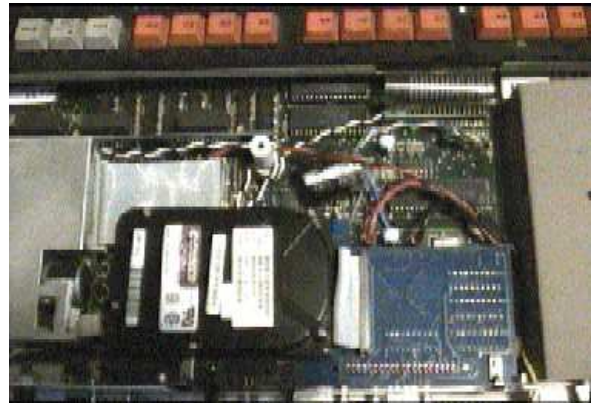
Both of these ICs, however, are rather hard to find. Unfortunately, neither are stocked by Maplin.

Unclipping the three front clips is best done with a thin flathead screwdriver or blunt knife. Slide it down from the top of the machine and gently prise the casing away from the keyboard.

This takes a bit of knack, and in my case - some super glue. :-)

STEP TWO:

Open your A3000...
Unscrew here:

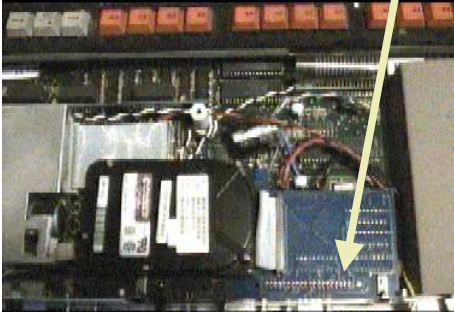


Then unclip here:

Once you have removed the lid and put the screws somewhere safe, rotate the machine so the power supply is to your right, and you are looking at it from the rear. The serial interface area is just beside the power supply, behind the 'Econet' socket. You will have to remove any expansion cards you have

in the internal expansion connector.

The serial port is about:



As you can see from above, I have a harddisc to remove.

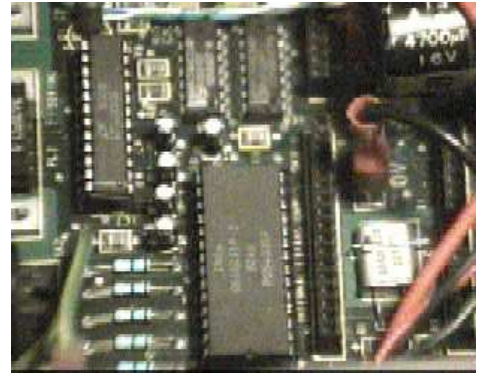
Now, rotating the machine so the power supply is closest to you, the floppy drive furthest and any expansion cards removed, we can begin the installation.

groove facing you. You can't get them the wrong way around as they are physically different sizes.

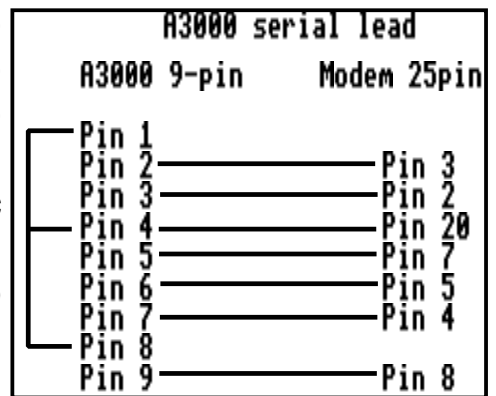
CHECK!

Make your the ICs are inserted correctly (there is a 'notch' printed on the circuit board to help). Make sure the ICs are fully pushed in and none of the legs are in the wrong place or have broken off.

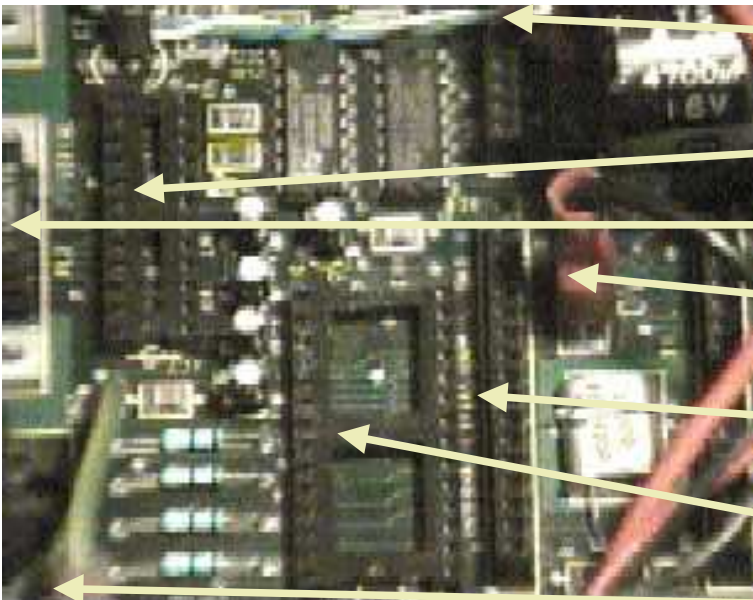
Both ICs are static sensitive, so do not touch them until it is time to install them. Be gentle. Touch something at earth potential (a radiator?) to dissipate any static



You should now be looking at something like that above.



Time to replace the cover and make up a serial lead.



- ← +5V power lead (red)
- ← Line driver socket
- ← Serial port
- ← 0v power lead (black)
- ← Internal expansion connector
- ← Serial IC socket.
- ← Earth wire

Unfortunately, due to various technical cock-ups, you won't have the joy of paying £15 for a PC serial lead. Instead, go buy £5 worth of components and build your own lead. Or buy one from one of the

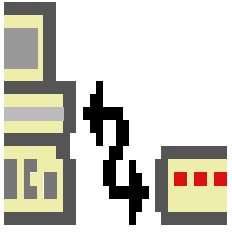
Above, what you should be looking at.

electricity that may have built up in your body.

advertisers in Acorn User.

Insert the ICs. Both are inserted with the notch or

Happy bauding!



Using the serial blockdrivers.

This little article will detail how *you* can use the serial block drivers from BASIC.

WHAT IS A BLOCKDRIVER?

Over to Hugo Fiennes:
To provide a standard way for applications to access serial devices. Any application supporting block drivers can be used with many serial devices, with no modification required to the application code - and it's easier to use than the serial SWIs anyway!

Right. There are four major concepts that you must understand and perform:

- 1 - Initialise block driver.
- 2 - Poll block driver.
- 3 - Receive data.
- 4 - Transmit data.

This is, all in all, quite simple and involves the minimum of fancy programming.

So let's get on with it...

Okay. Now a few things are going to be assumed here:

1. You know a bit of BASIC. You don't need to be a whiz...
2. You have written a basic comms terminal.

This can be as simple as a 'receive & show and get key & send'. Doesn't need to be fancy. Look at MTerm!

Your program is awaiting four routines -

PROCinitdriver, PROCpolldriver, PROCrxdriver and finally, PROCTxdata. For good measure I'll add a few more useful routines.

DEFPROCinitdriver(blockdr\$)

This procedure will load the indicated driver... This first part looks to see if 'no blockdriver' is selected, and if it is - it will see if any blockdriver is loaded, and if that is - it will kill it off.

```
IF blockdr$="None" THEN
  IF driverinstalled%=TRUE THEN
    A%=18
    B%=port%
    C%=0
    D%=0
    CALL driver%
    driverinstalled%=FALSE
    !(driver%+&100)=0:!(driver%+&100+4)=0
```

Here it is advised that you disable your DTE rate/parity menus etc.

```
ENDIF
ELSE
```

At the beginning of your program, you should have defined 'driver%' to be 4096 bytes long (DIM driver% 4096). Another requirement is that 'port%' be set to a valid value. You can safely default to '0'.

```
OSCLI("LOAD SerialDev:modules."+blockdr$+".driver "+STR$"driver%")
```

Reason code 17 is your 'initialise' code.

```
A%=17
B%=port%
C%=0
D%=0
CALL driver%
```

Here is a good place to set up your DTE rate menus.

```
driverinstalled%=TRUE
```

Reason codes 6 and 7 flush the Tx and Rx buffers (respectively).

```
A%=6
B%=port%
CALL driver%
A%=7
```

```
B%=port%
CALL driver%
```

```
ENDIF
ENDPROC
```

DEFPROCpolldriver

This procedure will, after making sure a driver is loaded, poll the driver... Reason code 19 polls.

```
IF driverinstalled%=TRUE THEN
  A%=19
  B%=port%
  CALL driver%
ENDIF
ENDPROC
```

DEFPROCrxdriver

This procedure will receive the data from the serial port. 'a%' is the ASCII code of the character received. If 'a%' is -1, then no data is available.

```
IF driverinstalled%=FALSE THEN ENDPROC
REPEAT
  A%=1
  B%=port%
  a%=USR(driver%)
  In the line below, the procedure PROCdrawchar is
  called. You take the ASCII value passed and draw
  the character however you want. You could branch
  off if the character is ASCII 27 (Hex &1B) or
  whatever.
  IF a%<>-1 THEN PROCdrawchar(a%)
UNTIL a%=-1
```

Note that there is no Wimp polling call in this procedure. That is of lesser importance. You can safely add Wimp_Polls if you are using an sp_dual serial port, but Internal port users will soon feel the effects of too many Wimp_Polls - mainly lost characters etc.

```
ENDPROC
```

DEFPROCtxdriver(byte%)

This procedure will get the blockdriver to output the ASCII code you pass as 'byte%'.

```
IF driverinstalled%=FALSE THEN ENDPROC
A%=0
B%=port%
C%=byte%
CALL driver%
ENDPROC
```

And that is that. The basis of a really simple terminal. Here then is a simple comms program (monotasking BASIC) that will demonstrate this all in action. Yes, I know I've copped out by making it monotasking, but to fit in a multitasking application - I'd need to reset this

text at about 4 point (this is 12 point Homerton... This is 6 point...).

```
REM >Blockdriver demo
REM By Richard Murray, for Froblicate.
REM
REM June 1995. Feel free to plagiarise this. :-)
:
MODE 12
:
ON ERROR PRINT REPORT$+" at "+STR$(ERL/10):END
:
DIM driver% 4096
driverinstalled%=FALSE
REM Alter the '1' below to be your chosen port.
port%=1
:
REM Alter the 'sp_dualPC' below to be your
REM chosen driver...
PROCinitdriver("sp_dualpc")
:
REPEAT
  PROCpolldriver
  PROCrxdriver
  byte%=INKEY(1)
  IF byte%>0 THEN PROCtxdriver(byte%)
UNTIL FALSE
END
:
DEFPROCdrawchar(a%)
  PRINT CHR$(a%);
ENDPROC
:
:
REM Add the library routines detailed in the
article into here...
:
DEFPROCinitdriver(blockdr%)
ENDPROC
:
DEFPROCpolldriver
ENDPROC
:
DEFPROCrxdriver
ENDPROC
:
DEFPROCtxdriver(byte%)
ENDPROC
```

Simply add in the routines given in this article, load it up and type ATDT 141 0181 903 1308 and take a tour of ArcTic. Smart people might think of loading an 'ANSI' system font first. :-)

Okay. That's it. Or is it? Where's the fancy stuff? Different DTE rates, looking for available drivers, sending breaks etc etc... Well, PTO!

Firstly, we shall make our list of serial drivers. The list will be read into an array DIMed as **blockdriver\$(26)**. It's not a great method, it will fail with more than 25 devices... But that's not likely yet. We are only up to 9 official devices (about 12 in all).

```
DEFPROCserialdevicelist
  OSCLI("EnumDir SerialDev:Modules <Obey$Dir>,List")
  devlist%=OPENIN("<Obey$Dir>.Resources.List")
  counter=1
  blockdrivers$(0)="None"
  WHILE NOT EOF#devlist%
    blockdrivers$(counter)=GET$#devlist%
    counter+=1
  ENDWHILE
  CLOSE#devlist%
  OSCLI("Remove <Obey$Dir>.Resources.BlockDrv")
ENDPROC
```

Next, we shall create a list of available DTE speeds. Again, we predefine an array - **driverspeed\$(35)** - which should be adequate.

```
DEFPROCdtespeedlist
  counter=0
  offset%=driver%+&100
  REPEAT
    driverspeed$(counter)=STR$(!offset%)
    offset%+=4
    counter+=1
  UNTIL !offset%=0
ENDPROC
```

Now, we shall set the chosen speed. Two points to note; firstly the speed parameter passed is a STRING (this is done because the speed array is a string array), secondly the speed must be set separately for Tx and Rx.

```
DEFPROCdriversetspeed(speed$)
  A%=13
  B%=port%
  C%=VAL(speed$)
  CALL driver%
  A%=14
  B%=port%
  C%=VAL(speed$)
  CALL driver%
ENDPROC
```

How to read the number of ports available...

```
ports%=?(driver%+199)
```

Now for the fancy stuff you'll probably *never* use:

Firstly, this procedure will send a 'break' to the modem.

```
DEFPROCsendbreak
  A%=11
  B%=port%
  C%=20
  CALL driver%
ENDPROC
```

See if there is a byte in the buffer... This is like PROCrxdriver, except it leaves the byte in the buffer. Useful to see if the next byte is ASCII 27 to see whether or not you need to branch off for ANSI parsing.

```
DEFPROClookindriver
  A%=12
  B%=port%
  a%=USR(driver%)
ENDPROC
```

Finally, a daft routine to flash lights on your modem, so you can tell which modem you are using - if you are lucky enough to have that many!

```
DEFPROCindicateline
  IF driverinstalled%=FALSE THEN ENDPROC
  A%=8 : B%=port% : C%=-1
  CALL driver%
  FOR x=1 TO 10
    A%=8 : B%=port% : C%=0
    CALL driver%
    t=TIME
    REPEAT UNTIL t+20<TIME
      A%=8 : B%=port% : C%=3
      CALL driver%
      t=TIME
    REPEAT UNTIL t+20<TIME
  NEXT
ENDPROC
```

That's all folks... I hope this has been of some use to you.

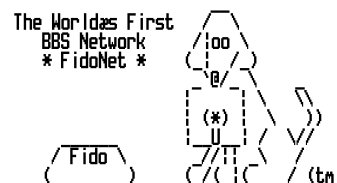
Useful reference materials (supplied with some versions of Froblicate, and available from Encina):

- The terminal program, fully working.
- MTerm_RX version 1.00 alphatest (cut down).
- Hugo Fiennes's blockdriver documentation.

Questions?

"Richard Murray", 2:254/86.1 at FidoNet.

Internet legislation rumours - could it affect



Well, it is about time an article like this was written. Something to pull out *all* the stops. Something to promote the idea that **Perfect Paranoia is Perfect Awareness...**

You know what I mean. Every phone call you make is taped by the NSA. Every email you send is read by some 'official' of some dubious organisation that they all deny knowing about. Every step you take is tracked by satellites powerful enough to examine every bit of loose stitching in your shoe - not the claimed "1 foot per pixel resolution".

Okay. Back to reality. :-)

But we have to face one thing. In the mind of a politician, a computer network is a computer network. Who cares about the subtle (read 'immense') differences between the Internet and FidoNet? They're all full of pervs and weirdos.

With that attitude in mind, the S.314 was born.

I know a number of you will have heard about the S.314, but not know exactly what it is.

S.314

The "Communications Decency Act of 1995" (S. 314), introduced in early February by Senators Exon (D-NE) and Gorton (R-WA), would place substantial

criminal liability on telecommunications carriers (including traditional telephone networks, Internet service providers, commercial online services such as America Online and Compuserve, and independent BBS's) whenever their networks are used to transmit any material which is deemed indecent or harassing. In order to avoid these penalties, carriers would be forced to restrict the activities of their subscribers and censor all public and private communications.

Thanks to the Voters Telecommunications Watch bulletin.

Are you starting to get the picture? In a nutshell then:

S.314 will force communications service providers to limit/remove access to file archives, newsgroups and email (as what you do/say will suddenly affect them a lot more).

S.314 would kill the free flow of information, mess up the entire concept of free speech and make privacy a thing of the past.

This is entirely not a good thing. Sure, some form of censorship is needed, but not like this.

You'll no doubt be asking: What has this to do with England, and what has this to do with FidoNet? It's Internet junk.

I'll answer both of those in order. Firstly, it has a lot to do with England. The Americans are 'up there' with electronic communications, where our pitiful mess of a government (no hard feelings John, but...) are not really into it. The 'electronic frontier' is the same place, paying no attention to political, cultural or race boundaries. Therefore it makes sense that we shall look to the Americans for guidance. For sure, it won't become law over here, but it is a big pointer. Look at the Church of Scientology vs a SysOp (due to be cased soon). That won't really affect us over here, but if a SysOp is found to be responsible for *every* aspect of his/her BBS - then we shall wisely assume the same general result would happen here.

Secondly, this has nothing to do with the 'Internet'. This is about electronic communications, which encompasses BBSs, Fido, Internet, Janet, FreedomNet and just about anything else with a modem attached. Looking at the O.J.Simpson trial, this could either be slung out as undemocratic (the Americans have a solid constitution, unlike us) or it could become immensely silly, like the need to censor in-house email and other totally daft ideas.

But now you are saying, "Don't be stupid... The Americans are like that. Not us Brits!"

Well, you can say that until the cows come home - I hope it lets you sleep better at night.

Let's consider the point that the government thinks most of us are subversives, and doesn't want us to talk to each other.

Proof?

OFTEL come up with stupid excuses to prevent BT from making local rate phone calls free. If the company is pulling a profit of a thousand a second - then they've plenty spare to do this.

Back in the early 80's. Remember the 'CB wars'? The kind government of ours tried to kill CB? They failed to stop it, but they did successfully allocate a band that was compatible with nobody else in the world. And a limit was imposed on antenna sizes. Since then, we are slowly reverting to CEPT (European) CB frequencies (I use both) and the restrictions on 1.65m antenna lifted - as the excuses were totally crap. If a CB using an 11 metre antenna at 4W output can risk crashing airplanes, that means no airplanes can fly within 500 miles of Italy. Hook up your CEPT CB or scanner to a half-wave antenna and tune in at night. Hear those Italians. What was this about CB's range being about 10 miles? You'll hear lots of Italians. Their rigs make our rigs look positively sad. 1kW, 2kW, huge antennas... Wouldn't the DTI love that.

Back now to the concept of 'all networks are alike'. You can scream the differences until you

are blue in the face, just like you can scream most other things. A riot didn't kill the poll tax. Unemployment is rising, and they make stupid statistics like "We have no unemployment figures for Surrey as most people commute out of the county to work". What a pile of crap. It's the people that *don't* work they should be counting, not those that do.

Well. I'm sure that in amongst my political insults, you can see the picture. What happens there will happen here in time. What happens to Internet will attempt to be applied to FidoNet (and other networks).

A little birdy informed me that a message from London to Paris is routed via the US. In a roundabout way that makes sense as it is the US's network and designed to re-route around places in the US (in other words, it's probably hard-coded into it to do that). Where that affects you is for the brief time your message is in America, you must be aware of their ways.

PGP, Pretty Good Privacy is an up and coming system developed by Philip Zimmerman (who has apparently stopped developing it after a few hundred criminal charges were brought against him for arms distribution. Yup, you read that right. PGP is a weapon! As far as I know, all charges failed because Philip hasn't actually sold PGP. It's free). It incorporates Public Key Cryptography - a slightly complex system (easy once you understand it) that can lock up

your email so well it would take a supercomputer a year to unlock. This is a delicate situation as the governments (obviously) loathe it and will do anything to stop it - never mind your 'right' to privacy.

So netizens, it is up to you to either sit back and watch the electronic communications systems be legislated into the farcical nothingness that is our current CB system - or get up there and protect your rights.

If anybody came up with the idea of employing censorship in the post office and reading all your love letters, I'm sure people would give a damn.

This article was written by Richard Murray, and uses a healthy mix of paranoia, hearsay, circumstance and rumour. Before you religiously believe a word of it, bear that in mind.

Still, it makes better reading than "Cannibal's cookbook found on the Internet. This file details how to prepare and cook humans" (article on front page of recent newspaper. Daily Star???)



Due to the size of this program, there are no operational comments. The program itself is commented though.

Enjoy!

```

/*
 * Fuzzy string searching subroutines
 *
 * Author   : John Rex
 * Date    : August, 1988
 * Converted : Richard Murray, April 1995 for Froblicate.
 *
 * Compile time preprocessor switches:
 *   DEBUG - if defined, include test driver
 *
 * Usage:
 *   char *pattern, *text; - search for pattern in text
 *   int degree;          - degree of allowed mismatch
 *   char *start, *end;
 *   int howclose;
 *
 *   void App_init(pattern, text, degree); - setup routine
 *   void App_next(&start, &end, &howclose); - find next match
 *
 * - searching is done when App_next() returns start==NULL
 */

#define DEBUG 1

#include <stdio.h>
#include <stdlib.h>
#include <string.h>

/* local, static data */
static char *Text, *Pattern; /* pointers to search strings */
static int Textloc;          /* current search position in Text */
static int Plen;            /* length of Pattern */
static int Degree;          /* max degree of allowed mismatch */
static int *Ldiff, *Rdiff;  /* dynamic difference arrays */
static int *Loff, *Roff;   /* used to calculate start of match */

void App_init(char *pattern, char *text, int degree)
{
    int i;

    /* save parameters */
    Text = text;
    Pattern = pattern;
    Degree = degree;

    /* initialize */
    Plen = strlen(pattern);
    Ldiff = (int *) malloc(sizeof(int) * (Plen + 1) * 4);
    Rdiff = Ldiff + Plen + 1;
    Loff = Rdiff + Plen + 1;
    Roff = Loff + Plen + 1;
    for (i = 0; i <= Plen; i++)
    {
        Rdiff[i] = i; /* initial values for right-hand column */
        Roff[i] = 1;
    }
    Textloc = -1; /* current offset into Text */
}

void App_next(char **start, char **end, int *howclose)
{
    int *temp, a, b, c, i;
    *start = NULL;
    while (*start == NULL) /* start computing columns */
    {

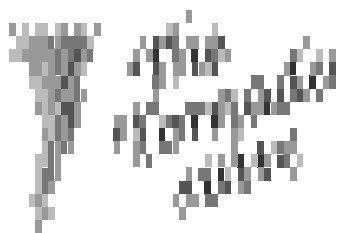
```

```

if (Text[++Textloc] == '\0') /* out of text to search! */
    break;
temp = Rdiff; /* move right-hand column to left ... */
Rdiff = Ldiff; /* ... so that we can compute new ... */
Ldiff = temp; /* ... right-hand column */
Rdiff[0] = 0; /* top (boundary) row */
temp = Roff; /* and swap offset arrays, too */
Roff = Loff;
Loff = temp;
Roff[1] = 0;
for (i = 0; i < Plen; i++) /* run through pattern */
{
    /* compute a, b, & c as the three adjacent cells ... */
    if (Pattern[i] == Text[Textloc])
        a = Ldiff[i];
    else a = Ldiff[i] + 1;
    b = Ldiff[i+1] + 1;
    c = Rdiff[i] + 1;
    /* ... now pick minimum ... */
    if (b < a)
        a = b;
    if (c < a)
        a = c;
    /* ... and store */
    Rdiff[i+1] = a;
}
/* now update offset array */
/* the values in the offset arrays are added to the
current location to determine the beginning of the
mismatched substring. (see text for details) */
if (Plen > 1) for (i=2; i<=Plen; i++)
{
    if (Ldiff[i-1] < Rdiff[i])
        Roff[i] = Loff[i-1] - 1;
    else if (Rdiff[i-1] < Rdiff[i])
        Roff[i] = Roff[i-1];
    else if (Ldiff[i] < Rdiff[i])
        Roff[i] = Loff[i] - 1;
    else /* Ldiff[i-1] == Rdiff[i] */
        Roff[i] = Loff[i-1] - 1;
}
/* now, do we have an approximate match? */
if (Rdiff[Plen] <= Degree) /* indeed so! */
{
    *end = Text + Textloc;
    *start = *end + Roff[Plen];
    *howclose = Rdiff[Plen];
}
}
if (start == NULL) /* all done - free dynamic arrays */
    free(Ldiff);
}

#ifdef DEBUG
void main(int argc, char **argv)
{
    char *begin, *end;
    int howclose;
    if (argc != 4)
    {
        puts("Usage is: approx pattern text degree\n");
        exit(0);
    }
    App_init(argv[1], argv[2], atoi(argv[3]));
    App_next(&begin, &end, &howclose);
    while (begin != NULL)
    {
        printf("Degree %d: %.*s\n", howclose, end-begin+1, begin);
        App_next(&begin, &end, &howclose);
    }
}
#endif

```



Tornado...

I, Richard Murray, Editor extraordinaire (hehe!) begged Niall Douglas to write about Tornado. I didn't want your "This is Tornado, this is how to program in it"... I wanted a different angle. How it came about and the trials of getting it to work.

Richard asked me to write a bit about preemption and where it came from. So, here it is!

Well, let's see, where did it all start?

I suppose I had been trying for years, but to no avail. As part of the Tornado project, preemption was required so I had had the idea queued for a long while, but never really got round to it. But one night (31st April) I went to a rugby dinner, drank altogether way too much and instead of sleeping the night (like the sensible thing would be) in a bloke's house (his name is "Hacka" BTW, and rugby fans will know where that came from :-)) we decided to try driving home. We made it, and I was home for 3am. It was May 1st. And, being in the industrious spirit of May day, I tried the old preemption chestnut yet again.

So, there we go. That's how this came about: a drunken stupor!

AFAIK, there are three people who have done this: the bloke in Acorn, a Russian bloke, and me. I ran my way through the Russian and Acorn debugged sources, and attempted to figure it out.

Here I'll digress: for some odd reason the two SWI's OS_AddCallback and OS_SetCallback after using OS_ChangeEnvironment are both labelled by the PRM's as being transient callbacks, and to me both seem to do the same thing ie; the routine is called upon the system

being threaded out of, that is the processor switches to user mode with interrupts disabled.

Don't ask me how RISC-OS knows this, but it does. Either way, both seemed very similar, with OS_AddCallback being the easier as you don't have to worry about the environment etc.

Ah, but as I was to learn, this isn't quite true. In fact, if you look very closely at the PRM wording, OS_AddCallback-type transients are also called upon system idling, like in I/O routines (again, don't ask me how RO knows). Which in other words also means that these type *can be called from IRQ mode!!!*

In other words, calling Wimp_Poll, which itself uses the OS_SetCallback method of callbacks (see PRM's), causes address exceptions when called from a USR mode threaded out of an interrupt, due to corrupted workspace. Why Wimp_Poll should need to use callbacks is beyond me (and I've perused the WindowManager code a lot), so I've emailed Acorn customer services to ask them. If they know, or reply, I'll stick the reply in ARCPROG.

So, the obvious method is to simply use the longer method of using the callback handler. This is only called on system thread-out, and if the processor is not in baselevel USR in say five seconds, then neither will the callback handler be called. This can be a problem, as most SWI's run entirely from privileged modes (SVC or above), as is evident if you ever try doing a long running OS_File (eg; loading 200k from floppy) from within a task window. This can't really be avoided, as the designers of RO2 wanted to avoid preemption rather much. I would agree.

Ok, that's the mystery laid out. Once you've

grasped that it's plain sailing from then on. I only realised this after much trial and error, so now that I've gone to all this trouble you all can rest easy. I won't bother breaking down the code as I assume you all are reasonably competent as reading what is rather easy assembler. What I will do, is to break down how RO works internally, as this information may not be readily available.

What happens when a SWI instruction is encountered:

- i. The processor is forced into SVC with IRQ's disabled and the current PC (and the processor flags before the changes) is written into r14svc. For preemption we'll assume this originating mode is USR, with IRQ's enabled.
- ii. The PC is loaded with 08. The computer branches to address 08 in logical RAM and executes the instruction it finds there. This instruction can be replaced if you really feel like it with another routine, which may/may not call the ROM based routine. Anyway, normally the processor now branches into ROM.
- iii. The ROM SWI handler fetches the original SWI instruction by using the contents of r14svc. Interrupts are set to that in r14svc (ie; enabled again if the originating mode had them enabled)
- iv. The vbit in r14svc is cleared, unless the SWI being called is OS_BreakPt or OS_CallAVector. The 24bit SWI no. is extracted and the loaded modules examined for that number. If found, the SWI handler within the module code is called, otherwise the unknown SWI service call is issued. If that isn't claimed, UKSWIV is threaded in and this, by the ROM handler, issues a call to the Unused SWI handler (as set up by OS_ChangeEnvironment), which by the ROM handler prints an error message.
- v. Okay, we're assuming that a module's SWI handler is now in effect. This may now call a vector, which then does the SWI, or usually the SWI is executed

straight. Either way, the r14svc passed to the SWI handler is NOT the same as the r14svc as used internally by the ROM SWI handler. In fact, it is a branch back into ROM, to do the following:

- vi. The SWI code ends, and control returns to the ROM SWI handler. Now checks are made for errors, and OS_AddCallBack type transient callbacks.
- vii. The r14svc is checked to see if it is a baseline USR mode. If there wasn't any other transient callbacks made yet, a check is made on the internal callback flag, and if set the callback handler is called.
- viii. Finally, the PC is loaded using a MOVS PC,r14svc. It is not clear exactly what happens here, as somehow the SWI can indicate or rather not indicate a mode change or flag change upon exit from the SWI back into the originating code.

Blast. I hope you can make some sense of that.

Anyway, cheers!
Niall

The **Tornado** module is *out there* and it can be put to a wide variety of uses...

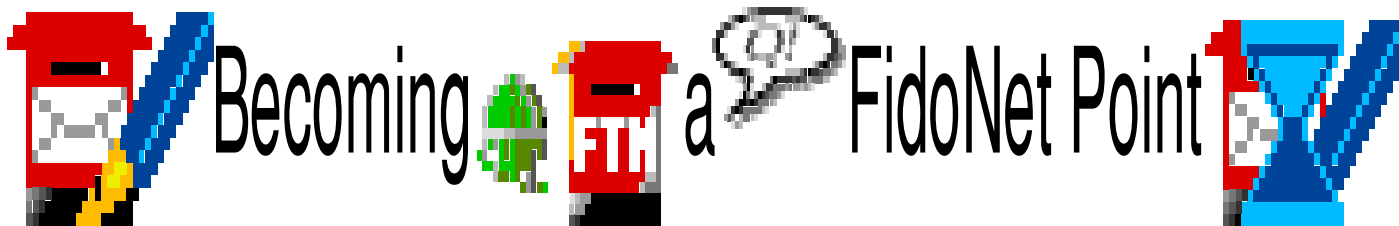
Add to **FidoMail** to allow multitasking during tossing/sorting/scanning.

Add to **ChangeFSI** so it won't hold up your computer whilst converting that JPEG...

Add to **StdFinder**, **VoteEdit** and many other programs...

Use it as a quick hack to make non-multitasking 'proglets' you 'knocked up' once, now multitask.

The possibilities are endless!



Ingredients:

- 1 FidoMail (v1.87 used)
- 1 Binkley (v2.19/Tst16 used)
- 1 harddisc with about 10Mb free.
- 1 ArcQuoter
- 1 modem (obviously)
- 1 SysOp who will set you up as a point.

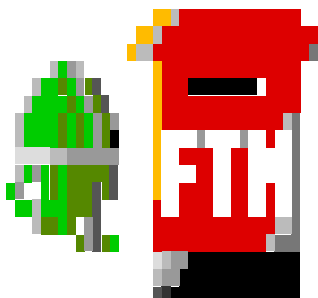
My system is set up to do its stuff at 38400bps, lockbaud. We shall assume those settings. Look in your modem's handbook for details of how to set this up.

I poll manually before and after browsing each mail packet. Likewise, you shall too. It is possible to automate all of this - but that would make this a very long article indeed. :-)

Note to USR users: The USR modem has no distinct way to select a locked baud rate. Instead, you just enter the AT&W (write new settings to NVRAM) command with the modem operating at the speed you want selected. Smart eh? :-)

We shall look at each program in order of importance. You might think the mailer is of prime importance, but it is not. Binkley is. Fidomail packets cannot be simply Zmodem'ed like a normal file. It takes a special program to handle this. You have *all* seen Binkley, most will know it best as

the "PRESS ESCAPE TO ENTER BBS" contraption. You will now start to see it from the other side.



Install Binkley on your harddisc.

Open up the !Run file and comment out the line:

```
If "<Binkley$NodeList>...
```

as you are a point, and it is easier for you to use DEFNODEs. NodeListMgr *can* be used, but it's a waste of RMA in this context.

Next open the 'Config' directory, and set up the DEFNODEs.

Your first (and only?) DEFNODE will be that of your boss node.

```
DEFNODE <address>@Fidonet #<phone number>
$<sysopname> @<BBS name> !<password>
```

The password field is optional. ArcTic's DEFNODE looks like:

```
DEFNODE 2:254/86.0@Fidonet #14101819031309 $Steve @ArcTic
```

Next we move on to the other config files:

EVENTS:

```
EVENT ALL 00:00 23:59 R
```

This is a 'default' event, lasting 24 hours. The 'R' sets Binkley to RECEIVE ONLY, otherwise it might try to send your mail for you.

MAIN:

You shouldn't need to alter anything in here.

MISC:

You shouldn't need to alter anything in here. If you are running Binkley as a front-end to a BBS (beyond this article's coverage), you will need to alter the messages Banner, BBSNote, DoingMail and EnterBBS.

MODEM:

Port @<blockdriver> <port>

For example:

Port @sp_dualPC 1

Port @internal 0

Baud <baudrate>

For example:

Baud 19200

Baud 38400

My initial Modem configuration settings are:

Port @sp_dualPC 1

Baud 38400

LockBaud

;AutoBaud

The semicolon before AutoBaud disables it. Alter the blockdriver as your hardware dictates.

The next few settings, Init, Prefix etc depend upon your modem. If you are stuck, leave them as they are. If that doesn't work...

Init ||ATZ~~

Prefix ATDT

Answer ATA|

Busy AT|

I have removed PreInit, PreDial and Suffix.

SYSTEM:

This is an all important one.

System Worrying Heights [Encina]

Here you put the name of your system. Make it something witty and original. Worrying Heights (an egotistical parody of Wuthering Heights) is what my point is called.

SysOp Richard Murray

It doesn't matter if you aren't a SysOp, Binkley still exchanges this information, so put your name here.

Address Fidonet#2:254/86.1

This is where you put your fidomail address. If you are a point, your address is unique to you, and will have been given to you by the SysOp. Point addresses are usually the host address with a different number at the end (for example, I am 2:254/86.1 - my boss node (who I poll mail from) is 2:254/86).

NLbaud 14400

NLflags V32B, V42B

NLphone 44-252-#####

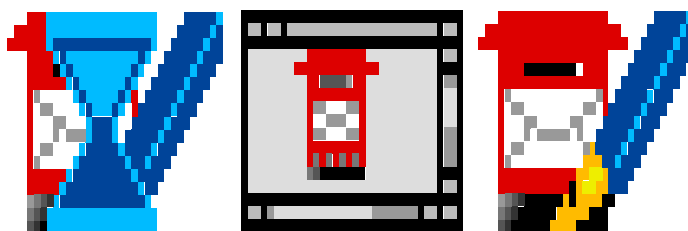
NLcity Aldershot, Hampshire, UK, Europe

These lines give more detailed information on you. Set NLbaud to the maximum connect speed your modem handles. Leave NLflags as they are (I'm not quite sure what they all do, but Fidomail doesn't need them anyway!). Set NLphone to be your telephone number in 'nodelist' format

(<countrycode>-<areacode>-<number>).

ArcTic would be 44-181-9031309 Note the missing 0 in the London area code. This is correct for international format numbers.

You should be able to leave the rest as-is. Hopefully your SysOp will have been kind enough to join you to a few echos - otherwise you'll have to tackle AreaFix. As AreaFix programs differ, either ask your SysOp about it or send a blank message to "AreaFix" at your host SysOp's system (most AreaFix programs reply blank messages with a help file).



Next up is FidoMail, the mailer.

When you first run FidoMail, it will pop up a 'configuration' window. Where it says YOUR NAME, type in your name exactly as you want it recognised (if I typed in R Murray, messages to Richard Murray would not show up for me).

Where it says YOUR ADDRESS, type in your Fidonet address exactly as you just did for Binkley.

Where it says YOUR UPLOAD, type in your Host SysOp's Fidonet address.

Finally, where it says YOUR AREAS, type in a few message areas that you are taking. You can add message areas later. There is help available to explain each option.

Running FidoMail, in the Misc menu (on the icon bar), you have the option "Prefs...". Clicking this will enable you to alter yet more features of FidoMail:

Quote

This appears automatically when you begin to quote somebody.

Footer

This is one of those lines "...Windows, from the makers of EDLIN" type of things. You can leave this blank and let ArcQuoter do it for you.

TearLine

This, by definition, details the program used to create your messages. You could change this, as people overly interested in your software will look at the PID line (that's something you don't see, it means Program IDentification).

Origin

As for Binkley, Origin is what you call your system.

The rest are personal settings. I recommend:

Maximise, Yes ; Open summary, No ; Open Status , Yes ; Allow edit all, No ; Quick tossing, Yes ; Keep heap safe, Yes ; Scrap outgoing without fuss, No ; Keep log open, No
Automatic addressbook, Confirm ; Automatic weeding, Off.

The !FidoMail.Config.Main file is the one used here.

Adding extra message areas:

Message areas are defined in the file "!FidoMail.Domains.FidoNet.DomainCfg". The format is:

```
SetArea "<echoname>" ECHO <realname> <scan to>
```

Where:

<echoname> is the name that you will see in the status window.

<realname> is what the echo is really called.

<scan to> is where to route scanned messages in this area too, usually your boss SysOp's address.

For example:

```
SetArea "SysOp echo" ECHO ACORN-SYSOP 2:254/86.0
SetArea "AcornGeneral" ECHO ARCHIMEDES 2:254/86.0
```

The real echo name is case sensitive, but most (if not all) echos are in upper-case.



Finally, ArcQuoter...

This program will generate random quotes (click SELECT on its icon), insert any file dragged to its icon to the caret, search your quote files for a particular quote, generate a signature (click ADJUST on its icon) and lots more. All thanks to Robin Abecasis (Rob The Slob).

For more detailed instructions, refer to the documents supplied with each application.

How to do your mail:

1. Get binkley to poll your host SysOp. Select the 'poll' option from the menu.
2. If a window opened and a packet was transferred, then load FidoMail and select TOSS from the menu (inbound).
3. Once your mail has been tossed, click SELECT on FidoMail. That will open a status window showing how many messages there are, just been tossed, unread, for you etc etc.
4. Click on one on the RED unread boxes to commence reading your messages.
5. To reply, choose REPLY (in echomail) or PRIVATE REPLY (in netmail).
6. To send a new message, click on the echo name, or 'Netmail' if it is to be private.
7. Select SCAN from the menu.
8. Load up Binkley. The messages you just scanned out will appear in the 'outbound' window. Double click on them to send.

Repeat as desired. :-)



Reader's Letters

Send your letters to:

"Richard Murray" at 2:254/86.1 (FidoNet) or "rmurray@digibank.demon.co.uk".

I have recently acquired a copy of your on-line magazine, 'Froblicate', and I was intrigued by one of your comments:

```
>It also allows you to
>remove bloody stupid
>persistent 'register
>me' windows from
>programs like the one
>in ToolKit >(AW June
>1995)
```

Are you suggesting that registration windows are always a Bad Thing? Or are they only a bad idea when attached to little BASIC utilities? If you dislike them, period, what alternative do you suggest for Shareware or Careware programs?

Chris (c.taylor@ucl.ac.uk)

No, registration windows are not a bad thing, but constant reminders can be more bad than good for a package.

It depends what the utility is... A program like !FidoMail deserves to have something like that, it's a powerful program. My UUcode/decode however, running in at about 1K, doesn't. :-)

Don't get me wrong, I support /proper/ ShareWare - but when at college I had a dose of any-old-crap-calling-itself-shareware-and-saying-send-me-\$10.

Thank goodness the Acorn scene isn't like this.

Froblicate reserves the right to edit letters submitted. Froblicate will not be held liable for your messages or their content. You have the option to remain anonymous if you so desire...

I have neither Ovation nor many of the fonts you used, so I've only seen the text-only version of Froblicate.

I found several of the articles interesting, but the overall style seemed a little unpolished. The title doesn't help, IMHO.

If you don't mind some unsolicited advice, I would suggest that if you want to use non-standard fonts, stick to PD versions and include them in the archive with the magazine. And Ovation is something of a minority product, ATM. This may change with the release of Ovation Pro. Do keep writing, I think there's a demand for a technically-oriented magazine like yours. I'll certainly read part two.

Chris (c.taylor@ucl.ac.uk)

Yes, I've seen a number of mentions of 'Froblicate' around. Wow! Even though you don't think much of the name - it's not one you will forget in a hurry. :-) To answer your points then - IMHO, Impression isn't as 'nice' as Ovation. You see, things like *niceness* come into this as I'm doing all of this because I *want* to. If I it was my job to do this, I'd probably not be using Ovation of Impression. I am glad you found some parts interesting. That's the whole idea - but I can't please all of the people all of the time. I'm trying (I'm very trying, just ask

mum :-)). As for it's unpolishedness, I didn't get a top mark in English. I've never done this before and released it to an unknown audience. I can't draw to save my life. Again, I try - but without plenty of mistakes to learn from, I'll make the mistakes I shall learn from. Catch 22.

Feel free to send all the advice you want, solicited/unsolicited or otherwise. The fonts I use come from three main sources: Ovation (Paladin/Vogue/SwissB/SymbolB), AcornDTP from the Learning Curve (Selwyn), RiscOS3 (Homerton, Corpus, Trinity) and Arcade BBS (MDA1, ANSI, System.Fixed etc etc). Fairly easy to get hold of. The only problem font is Selwyn. The others can be reformatted for Trinity/Homerton/Corpus. This is Trinity, and this is Paladin. Much of a muchness really. Other fonts (issue 1) were Keys1000 and Sassoon. Both were found on-line. I'm not going to even consider adding the fonts to the archive as that would easily add another 500K (uncompressed) to the size of it - and people that already have those fonts wouldn't really appreciate this.

I hope you do read part two. You'll see yourself in print. :-)

See guys, I will print your questions and do my best to write a meaningful reply.

So -- who wants
to be in issue 3?

:-)

RiscOSity



In this regular article, we shall look at the more *peculiar* parts of RiscOS and things computer related... By the way, the title of this article is pronounced *risk-oss-it-ee* (not like *risk-oh-ess-it-ee*).

We shall start this first RiscOSity with a look at the various logic types through the ages, the various *different* types of logic, such as TTL, CMOS, VMOS etc... We've all heard of those, but what about RTL? And I don't mean that stuff you can get on your satellite dish!

1) Primitive Switch Logic (PSL)

Early attempts at logic gates were made with analogue switches. Both AND and OR gates were easily created with SPST switches.

2) Switching Mode Logic (SML)

Eventually, all gates were created from bipolar transistors. There are three types of bipolar switching mode logic gates: current sourcing, current sinking and current mode logic - all to be discussed in a future article.

3) Resistor-Transistor Logic (RTL)

RTL works by placing a resistor or the gate's transistor collector AS WELL AS a current limiting resistor on the transistor's base.

4) Resistor-Capacitor-Transistor Logic (RCTL)

The same as RTL, except a capacitor has been added across the current limiting base resistor. This gives a higher operating frequency compared to RTL.

5) Diode-Transistor Logic (DTL)

Now, some bright spark decided to improve on previous efforts (like RTL) by placing diodes on the transistor's base, the control the voltage drop.

6) Transistor-Transistor Logic (TTL)

One of the largest logic families. TTL gates are basically current-sinking switching mode logic transistors with multiple emitters. This gives the gate high speed, low power consumption and a degree of signal noise immunity.

7) Schottky TTL

By adding a Schottky diode between the base and collector of every TTL transistor, an increase in operating speed and a decrease in power consumption is possible. Schottky TTL gates operate at a higher frequency by using the Schottky diode as a conductor of bias current, and therefore, reduce the transistor's need for saturation.

8) Emitter-Coupled Logic (ECL)

By using current switching and emitter amplifiers, the ECL gate

is able to avoid the transistor saturation requirement. Unfortunately, the ECL gates put a constant drain on the circuit power supply.

9) MOS logic

Other than bipolar transistors, metal-oxide-semiconductor field-effect transistors (what a mouthful - it's shortened to MOSFET) can be used in gate construction. Basically, a MOSFET has a metal gate that is insulated from a silicon semiconductor substrate by a silicon dioxide shield. This is the MOS part. There are four usual arrangements : N-channel depletion, P-channel depletion, N-channel enhancement and P-channel enhancement. The major difference between a depletion MOSFET and an enhancement MOSFET is the presence of a conducting channel running between the source and the gate in depletion mode. Enhancement mode MOSFETs don't have this channel.

10) NMOS logic

N-channel MOSFET or NMOS gates have an N-type source and drain in a P-type substrate. Applying voltage to the metal gate forms a current between source and drain.

11) PMOS logic

Like NMOS in reverse, PMOS gates have a P-type source and drain in an N-type substrate. Like the difference between NPN and PNP, PMOS operates just like an NMOS gate, except it has an opposite polarity.

12) VMOS logic

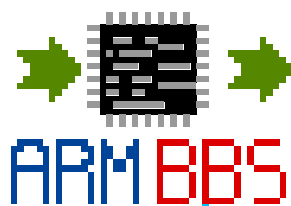
VMOS or vertical MOSFET gates use a unique construction technique that provides the 'vertical' portion of its name. VMOS gates have an increased power capability over NMOS or PMOS gates. VMOS is frequently used in high-density RAM chips.

13) CMOS logic

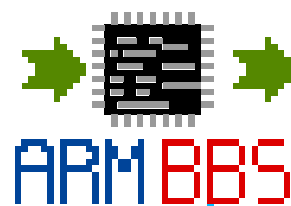
Combining both N-channel and P-channel MOSFETs into the same gate is complimentary MOS, or CMOS logic. CMOS gates have a very low power consumption with a wide range in voltage tolerance. Accompanying this reduced power consumption, however, there is a reduction in speed - compared to an equivalent TTL gate. Another drawback is that a CMOS gate has a high input impedance, which makes CMOS susceptible to static electricity.

14) QMOS logic

High-speed CMOS or QMOS logic is both functionally identical and pin compatible with the 74XXX TTL ICs. QMOS gates require a slightly higher current, but it is still less than required by Schottky TTL. On the other hand, QMOS operates almost 10 times faster than the same CMOS gate - in fact the QMOS gate's speed is close to that of the Schottky TTL gate with less power drain.



A look at ArmBBS



ArmBBS is the new BBS server on the block, and offers much as a replacement to RiscBBS. It plans to be released as ShareWare, although there has been no official release as yet.

FidoNet isn't properly supported, though there is an add-on utility to do this. There is no user guide, though Robin Abecasis is working on it as we speak. It supports ArmBBS doors, ArcBBS doors and RiscBBS doors. It also emulates the ARCbbsFiler_GetStatusPointer command.

In its current stage it beats the pants off of RiscBBS. When it's finished, I can envisage it providing strong competition to the current leader - ArcBBS. If the price is right, it might even topple ArcBBS.

It is feature laden, with more ideas and enhancements being added all the time. I loathed RiscBBS's way of doing things - and as ArmBBS is rather similar to RiscBBS, I wasn't expecting much. Boy was I wrong. I was very impressed with what I saw - even features like crash recovery.

The emphasis is placed on the user, not the application, which is debatable from the SysOp's point of view, but brilliant from the users end. No more odd hang ups or NO CARRIER messages...

The menu system is different to ArcBBS. It is all script based. The menu script starts by defining the options, and what to do with them, then it defines the screen. There are various quirks, like an apparent lack of a BOLD command - but as usual, Keith Hall (of Binkley fame) is addressing these and other points.

So all in all, I can't wait for a proper release, to see how it all is - but from what I've seen - I'm impressed. Here are three ArmBBS systems to call:

The Plasma Sphere (Keith Hall)	+44 1925 757920 and +44 1925 757921
Northern ARM BBS (Chris Jackson)	+44 1274 530831
Renegades (Robin Abecasis)	+44 1224 621956

Whatever does *Froblicate* mean?

Ever wondered what 'Froblicate' means? It's not rude or anything, it's actually a 'hacker' word (not crackers, hackers):

froblicate /frob'ni-kayt/ vt. To manipulate or adjust, to tweak.

Thus: "Please frob the light switch." (That is, flip the light switch.), but also "Stop frobbing that clasp; you'll break it." Usually abbreviated to <frob>, but <froblicate> is recognized as the official full form. Thus one has the saying "to frob a frob". See <tweak> and <twiddle>.

Usage:

frob, *twiddle*, and *tweak* sometimes connote points along a continuum. *frob* connotes aimless manipulation; *twiddle* connotes gross manipulation, often a coarse search for a proper setting; *tweak* connotes fine-tuning. If someone is turning a knob on an oscilloscope, then if he's carefully adjusting it he is probably tweaking it; if he is just turning it but looking at the screen he is probably twiddling it; but if he's just doing it because turning a knob is fun, he's frobbing it. The variant "froblicate" has been recently reported.

frob /frob/

- n. [MIT] The official Tech Model Railroad Club definition was "FROB = protruding arm or trunnion", and by metaphoric extension any somewhat small thing; an object that you can comfortably hold in one hand; something you can frob.
- v. Abbreviated form of *froblicate*.

And in email/echomail you see odd character sequences. Here is a laymans guide:

BCNU	Be seeing you
AFAIK	As far as I know
IMHO	In my honest opinion / In my humble opinion
IIRC	If I remember correctly
ISTR	I seem to recall / I seem to remember
FO	Foxtrot Oscar (= get lost)
FOAD	Foxtrot Oscar And Die (= really really get lost)
H&K	Hugs and kisses (not often used)
ROTFL	Rolling on the floor laughing
ROFL	Rolling on floor laughing
ROFLAO	Rolling on floor laughing a** off
TLA	Three letter acronym (any acronym with three letters; IBM, ARM, IRS, NHS etc)
ETLA	Extended three letter acronym (any acronym with more than three letters; PAYE, RISC, MEMC etc)

Smilies; a way to express yourself. US netizens call these 'emoticons'. They can be with a nose ':-)' or without ':)'...

:~)	Happy smilie, grin, don't take me seriously
:~}	As for :~)
:-(Unhappy, sad, upset
:~)	Winking
;^)	Nerr-nerr nerr nerr-nerr. Used to add sarcasm to a cutting remark ("I think you are a waste of space ;^)")
:~)	Annoyed, miffed
>:~)	Really really miffed, seriously annoyed
:~	Blank expression
:~{	Crying
:~##	User has a beard
8~)	User wears glasses
*~)	User has a nice hairdo (or a big brain)
?~)	User is a scatterbrain
:~*	User is sick (technical rain type sick)
=~)	User is a punk
%~:~)	User is a 'cyberpunk'
:~[]	Wow!
:~()	Wow!
:~)	User wears braces.

The 'user is/does/has' can be combined: 8'~* = User wears glasses, is crying and has a kinda upset stomach.

There are other ones you will come across; about 100 or so in all. You can download help files from various BBSs that list these acronyms and smilies and other netizen speech.

Feetch, feetch!

A LOOK AT CASTAVOTE

By the time you get to read this, CastAVote will be a year old - and I (yes, I wrote it!) and celebrating by releasing a nifty little 'intelligent' installation utility. No more copying CastAVote and wiping your data or configs. Now this program will add some intelligence and add what you want - where you want. A bit like Windows? No... I didn't say that. :-)

I am not going to detail the history of CastAVote - look in the user guide if you are interested. This article is to show five not-commonly known facts about CastAVote, both from a users point of view and from a SysOps point of view.

FACT 1 : There is a hidden 'debug' feature in CastAVote. Press ALT-D to see. This will show you when CastAVote was initialised and also information on the program/path etc. There is no 'really secret stuff' there - but you may see other options come and go as CastAVote is developed. One version held the names of the top five voters (for working out statistics). You never know, it might come up again. Press any key to leave. You can't alter anything here.

FACT 2 : Despite what you all think, the scrolltext was a quick hack. Andrew Lobel seems to have perfected ANSI scrolltexts (if any of you saw the InterComm notices page). My idea just reprints and offset string and doesn't loop. If you wish to use this feature, you must set up the scrolltext (this can be done in VoteConfig). Then loading the Configure file into !Edit, add the lines:

```
OPT_SCROLL1
OPT_SCROLL2
OPT_SCROLL3
OPT_SCROLL4
```

with the text you want to be show after the relevant OPT_SCROLL. The scrolltext only happens in the main menu. The number after OPT_SCROLL corresponds to the menu option (1 = Add a vote; 2 = Vote; 3 = List votes and 4 = Results).

You must add a SPACE character after your string. If you don't, you'll soon find out why. The maximum length of the string is 220 characters, but aim for a maximum of about 160.

FACT 3 : It may not have been immediately obvious, but CastAVote is highly portable. The software uses the absolute minimum of ArcBBS door commands, and looks only for your username. It should run happily on ANY BBS server supporting ArcBBS doors (like ArmBBS). But what about the displays? Well, not every server understands ArcBBS codes does it? No probs. CastAVote can export ArcBBS, ArmBBS, RiscBBS, Archiboard, plain ASCII and even ANSI!!!

Note: If your BBS server doesn't support ArcBBS doors, a RiscBBS version can be created by very very special order. :-)

FACT 4 : For a long time, people have moaned about the difficulties in exporting CastAVote's data. You can edit the data with VoteEdit, you can output displays for a myriad of screen formats - but what if you just want to sling the data into FoxPro or a spreadsheet? No worries. Now, in the utilities directory, is a whole heap of useful stuff. One program creates CSV files from the CastAVote data (that's Comma Separated Variable; any database worth it's salt will accept CSV. You can even feed CSV to Ovation for it's mailmerge, though I'm not sure if it would like CastAVote's data :-). Another program creates TSV files (like CSV, but separated with TABs). More programs convert CastAVote's data to/from the old formats. Finally a program to run simple diagnostics on the data.

FACT 5 : CastAVote is well supported. Here's a list of planned extensions:

Tidy up the incoming message displays (from Parlez...). Maybe make a Window overlay like SysOpChat?

More little utilities. How about outputting DIF, DFX, dBase and other such file types.

Speed up VoteEdit. Maybe use Niall's Tornado pre-emptor so it puts less load on the DeskTop - of the ability to scroll down screenfuls, rather than a vote at a time?

Add in a feature to allow a user to upload/download data to/from CastAVote (for remote maintenance by people like Ben Jeffreys ("Vote Master" on Digital Databank), this may require me to create some kind of X-modem system).

Whatever else I think is cool, or any other neat little ideas you - the users - think of.

FACT 6 : Okay, so I'm no good at maths... Did you know CastAVote has an obscenity filter? Yup - when a user creates a question, it will look for about 15 obscenities (the list is slowly growing) and throw the user off if such a word is found. The SysOp can add a further 10 choice words.

I have had a few complaints from people looking in the !RunImage and finding the words list. The compressor removed the explanatory REMs, and some people didn't read the user guide well enough, and wondered why those words were there. Well, worry no longer. The words are now compound CHR\$ commands - if you can speak fluent ASCII I'm beaten - if you can't - then it'll be better. At least there is now no change of somebody finding it in the application code and moaning. :-)

The latest CastAVote (version 2.48) will be available soon - and you can all enjoy it. Feel free to ask me questions, make comments or whatever - but PLEASE read the user guide. I get so many questions to which the answer is 'RTFM' (techie slang for "Read the ***** manual" :-).

One final note - if CastAVote crashes on shutdown, it is quite likely that you have forgotten to change the path of the <CastVote\$Result> variable. This is set up to point into !ArcServer.Text but it may need to be changed. You'll find it in CastAVotes !Run file.

Advertisements

Files for the Acorn,
PC and other
computers...

Local messages,
fido echomail and
fido netmail.

Doors, on-line
utilities, games, all
the latest such as
Card Sharks and
ONLINE and Top
10s of over 500
different subjects.

Local rate within
outer London.

Why wait? Come
call ArcTic BBS on:

+44 181 903 1309
[28800bps modem]

or on the second line:

+44 181 903 1308
[14400bps modem]

SysOp: Steve Pursey

CastAVote,
HappyHak,
LastUsers,
Linker
ONLINE
Parlez
ReadTasks
SysOpChat
ReadUsers
InfoSys

All these programs and more for
your BBS needs - from
BUDGIESOFT.

You'll find the latest versions on
Arcade, Digital Databank and
ArcTic BBS.

If you
would like
to
advertise
in
Froblicate,

Simply send us your
advertisement in
DrawFile or ArtWorks
format, and we'll do
our best to add it.

Assume about A7 size
(quarter of A4 page).

From the archives, I found
this... So all you A3000
owners - here is a guided tour
of your machine!

The first thing you notice, before even lifting off the cover, is a few 'Not fitted' stickers on the serial port and the Econet port. Inside, just under the Econet port is an IC carrier. This is looking for a comms chip (an MC6850P I think...). Under the serial port is a smaller IC carrier looking for a serial line driver chip.

Just above the power connectors (can't miss them) is a row of pcb header sockets. These are to allow expansion through the A3000's own minipodule setup.

Looking along the back behind the disc drive, notice the AC96way Eurocard connector. This is the allow 1 normal Archie expansion card to be fitted. There is no podule select address line. But, there is a way to fit up to 4 podules (simultaneously!).

The other side of the disc drive, a row of pins. Those of you with a magnifying glass will already know that this is the RAM expansion connector. Look at the ICs to the lower left of the RAM connector. They're marked IC20, 21... to IC27. These 8 ICs provide you with 1Mb of memory. It's amazing when you think that just a quarter of that little IC can hold as much data as the middle right-hand side of the BBC micro!

Just below the memory is a chip called ARM. That is your CPU! To the left of your RAM are four ICs. These are the DeskTop and OS. Beyond that is a silver thing that says VARTA on it. It's the battery for your CMOS RAM. Next you'll notice the flexicircuit keyboard connector. A warning here, you can take your keyboard apart, but NEVER try to separate the membrane layers. Yes... It's a membrane keyboard!

Just above the extreme left of the keyboard connector is an area marked ALT KEYBRD. I am trying to see if you can connect an Archie keyboard to this... Move up a little. You'll see the Econet socket link. This link goes to the Econet socket and nowhere else (our IIC port uses this...). Up more, an IC marked ACORN, INTEL & PHILIPS (phew!). This is the keyboard encoder. It takes in info. from the A3000's keyboard and converts it to serial format. To the right is a big IC called ALBION. This is your IOC, handling the serial port, the econet, the printer, the floppy disc... Among other things it converts the serial keyboard data to a parallel format (there is a reason for these conversions). Above the keyboard IC is the connector for the Econet board. Under your floppy disc cable is the disc drive controller. It is a 1772 (as in BBC micro DFS!).

Well, that's a brief rundown of the 'bits' in the A3000. Next time, the ICs and their functions.

If you think your circuit board is a little empty, you should check out the new A3010!!!