DRAG'NDROP

RISC OS **Pi** • Beagle • Armini • Iyonix

April 2014 Volume 5 Issue 3 £3.00

Wallace Goes Walking

Type-in arcade fun

Hardware

GPIO for beginners

SPI bus

Graphics

Scrolling stars

Anatomy of a Drawfile

Sound

User voices

Doing more with text

Inside Drag 'N Drop

Wakefield

Acorn & RISC OS Computer Show

Saturday 26th April

2014

Show organised by

Wakefield RISC-OS Computer Club
Formed 1983

The North's premier RISC OS show

Now in its 19th year

The Cedar Court Hotel
Denby Dale Road, Calder Grove, Wakefield
West Yorkshire WF4 3QZ
(Adjacent to J39 of the M1)

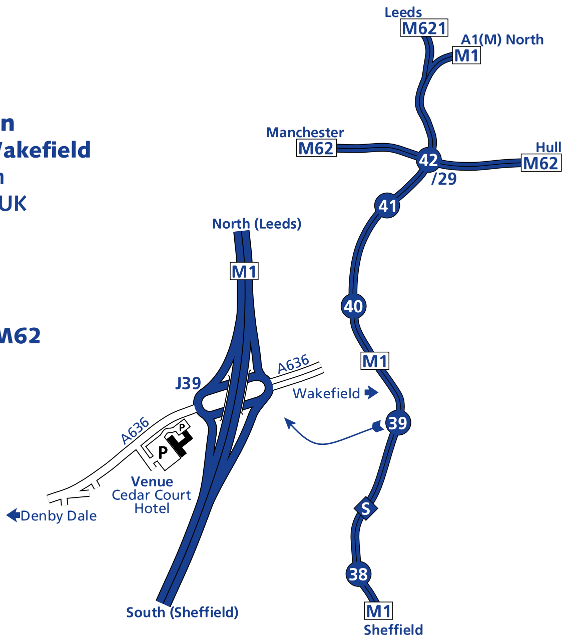
- **Big Names and Small Developers**
- **Raspberry Pi**
- **Show Theatre**
- **On-site Catering**
- **Charity Stall**
  raising funds for
  Wakefield Hospice

Getting there by train

- Free minibus serving **Wakefield Westgate** station, with connections across the UK

Getting there by car

- Easy access from ***M1***, ***M62*** and major routes
- Free on-site parking



• *Date*	**Saturday, 26th April 2014**
• *Opening time*	**10.30am to 4.30pm**
• *Ticket price*	**£5** on the door Entry for children aged 12 or under (accompanied by an adult) is **FREE**
• *Access*	Disabled access to the show venue is via a lift
• *Enquiries*	Wakefield Show 2014, c/o 3 Riverdale Avenue, Stanley, Wakefield, WF3 4LF
• *Email*	show2014@wakefieldshow.org.uk

Wakefield 2014 is sponsored by

M Software W

For further information, visit

www.wakefieldshow.org.uk

Produced on RISC OS computers

This issue has been blessed with contributions from the following people:
Gavin Wraith (Doing More with Text)
Matthew Thompson (Mountain Panic)
Rob Wilks (Experience of the SPI)
Jon Robinson (Bitcoin)
Christopher Dewhurst (everything else)

The views expressed in this magazine are not necessarily those of the editor. Alternative views are always welcome and can be expressed by either writing an article or a short editorial.

All articles and advertisements are published in good faith. No materials in this publication are meant to be offesnsive or misleading. If you come across something you believe is either of the above please contact the editor using the details below.

Contact Information

Editor: Christopher Dewhurst
Email: editor@dragdrop.co.uk
www.dragdrop.co.uk

EDITORIAL

Welcome to the Spring 2014 edition of *Drag 'N Drop*. Thank you for picking up the sample (hopefully you will buy the full copy) and special thanks to the contributors, I really appreciate your articles as it saves me having to think about what to write! To the readers, I hope this issue provides you with a wide range of subjects and you can find something you enjoy reading.

20 years ago when the RiscPC was released I was still (just) at school, Even if I had been working it is doubtful I would have ever been able to afford that computer, as excellent as it was. With the advent of the Raspberry Pi, RISC OS computing has become more accessible to more people than ever. Who knows, this time next year we may be using RISC OS on a tablet. But whatever the computer looks like physically, we should still be enjoying learning about programming – what *Drag 'N Drop* is here for.

Even if you don't want to write an article you can always make suggestions for what you want to appear in the magazine.

Chris.
Christopher Dewhurst

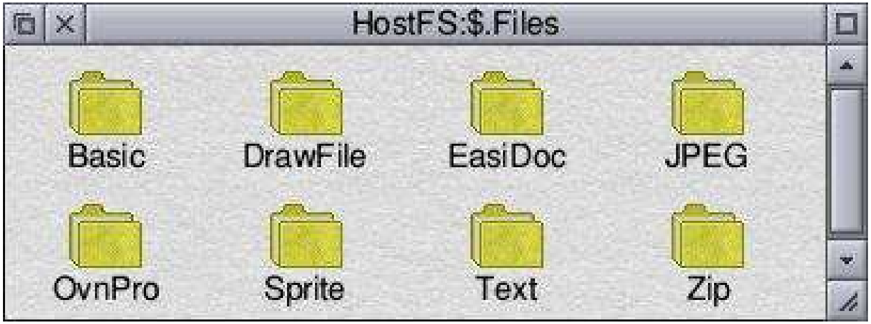
At a glance...

Beginner's Tips: Remembering Filetypes

Some other computer operating systems tack a three letter *file extension* onto a filename to show what the file should do, .txt for a text file, .jpg for a JPEG photo and so on. The dot appears as a slash, or forward stroke, in RISC OS (/txt, /jpg).

RISC OS uses *filetypes*, These filetypes can sometimes get "lost in transmission" (just as file extensions can). For example, somebody sends you a Basic program called "Abacus" whose file type "Data". So save the file to your hard disc, click menu over Abacus and choose File 'Abacus' > Set type > then type in "Basic" so that RISC OS knows what to do with it.

Most file types are obvious and easy to remember (Basic, Text) but Draw files have the file type "DrawFile" whereas Sprite files are"Sprite" and not "SpriteFile". As a memory aid, you can set up a series of directories (folders) whose names themselves are the *filetypes exactly as the computer shows them*. Whenever you need to change a filetype, just look at the folder whose name you want and if you are *really* organised save it inside that folder too.



HostFS:$.Files

(I have changed the colour of my folders from blue to yellow by following the "Themes" tutorial in the October 2013 edition of *Drag 'N Drop.*)

News and Applications

Wakefield 2014 Show

The 19th show organised by the Wakefield RISC OS Computer Club takes place on Saturday 26th April at the Cedar Court Hotel near Wakefield from 10.30am to 4.30pm. Admission £5. Visit www.wakefieldshow.org.uk for details. *Drag 'N Drop* will be there with a supply of 55 BBC Micro Books CD-Rom (£10) plus an updated back issues collection, this time on memory stick.

Drag 'N Drop in !Store

Your favourite magazine is now available in the !Store. To use !Store download from it from www.plingstore.co.uk. If you buy it this way the cost is currently £4.00 per issue but that includes the "type in" programs in their ready-to-run format. You can still buy *Drag 'N Drop* through the website at www.dragdrop.co.uk if you prefer.

Drag 'N Drop price rise

The price of *Drag 'N Drop* has remained unchanged since its launch in 2009. Any and all proceeds go towards running costs – attending shows, buying memory sticks/CDs, postage, etc. These are creeping up all the time and for the magazine to continue the cover price has to go up slightly. Full details to be decided and annouced in the next issue.

OS Updates

The latest stable version of RISC OS for the Raspberry Pi is Beta RPpi ROM RC12 (the RC stands for "release candidate), RISC OS 5.21. See www.riscosopen.org/content/downlads/other-zipfiles.

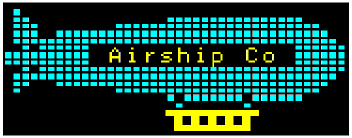
Using SystemDisc (see the News and Applications in the Winter 2014 edition) makes it really easy to keep your Rom up to date, but if you are worried then don't forget that ROOL still sell a pre-packaged SD card. This is always fantastic value for money with proceeds helping the ROOL team's development.

More adventurous Pi owners may like to try the development builds.

R-Comp have released OS Update 4 for subscribers to the ArminiX and Pandaboard scheme. These can be obtained from the subscribers area as described in your welcome pack.

Screen-u-like

Anymode is a brilliant little module by Steve Harrison, only suitable for the Pi at the moment, which allows you to put the computer in any mode including the old ones. Mode 13? no problem· Mode 7?! Yea, even teletext! No more faffing about with the monitor definition file, and you certainly have no excuse for not typing in that games listing from *Drag 'N Drop*. Anymode is free and version 0.06 can be downloaded from www.pi-star.co.uk/anymode.



(Teletext graphic courtesy of 8-bit Magazine issue 34, April 1994, available from www.8bs.com.)

J Rawle webpages

Jonathan Rawle's software is back online − educational, games, utilities and "desktop sillies" which can be downloaded from jonathan.rawle.org/software/ These are not guaranteed to run on modern machines, though we did get the ferret running along the top of the iconbar before he stiffed the machine. You have been warned!

RiscPC 20th Anniversary

April 2014 marks 20 years since the release of RiscPC, Acorn's flagship computer and (some would say) the acme of Acorn technology before its demise several years later. Throughout April R-Comp are offering 20% off certain products when ordered via the !Store, see www.plingstore.co.uk.

WimpBasic resources

After an absence of 18 months John Peachy's website is back online at jpeachy.orpheusweb.co.uk/wb/index.htm

Note this is a resouce area for WimpBasic, the actual program is called WimpBasic2 and allows development of Wimp programs without having to understand the windows environment in depth. Wimp Basic2 can be purchased from www.apdl.co.uk/prog.htm for £39.

DrWimpC

Another popular software package for writing Wimp programs is DrWimpC. It's an application builder using the C programming language and the best bit about it is that unlike WimpBasic it's free. Version 1.21 is available from www.drwimpc.co.uk. Don't forget the regular DrWimp package for application building in BBC Basic, available at www.rayfavre.me.uk/drwimp.html.

PackMan

Packman is the front end for software distributed via the RISC OS Packaging project at www.riscpkg.org. Version 0.8.1 checks for modules, improved documentation,minor bug fixes. It can be downloaded from https://sites.google.com/site/alanriscosstuff/packman.

GraphTask

GraphTask allows many non-multi-tasking programs to run in a multi-tasking desktop window, including ones which change Mode – which you can't do via the ordinary Ctrl+F12 task window. Version is still at 4.03 but the teletext support has been updated to work properly on Arm 7 machines, available from www.armclub.org.uk/free. *[AnyMode described earlier provides a much superior looking Mode 7 though only single tasking - Ed]*

MultiTask

Not to be confused with GraphTask, this small but extremely useful utility sets the CSD (currently selected directory) to the last file or directory opened. So when programming in BBC Basic you don't have to worry about the pathname of any file referenced by the program because it is assumed to always be in the same directory where your Basic program resides. Version 7.20 can be downloaded from www.svrsig.org/software/MultiTask.zip

Director

Director is a desktop customising tool. It allows you to create menus of useful commands and icons and provide a fast directory viewer (for instance, drag a directory onto Director and flick through the subdirectories therein without having to doulbe click to open them all). Version 0.39beta2 can be downloaded from director.sourcefourge.net

ClassicRip

A simple audio CD ripper by Jim LeSurf, ClassicRip extracts several tracks from a music CD and puts it in one long WAV (waveform) file. It also generates a textfile describing where each track began on the original CD, described by Jim as being useful for classical works (the "Classic" in ClassicRip).

McKie cover art on sale

Angus McKie painted several acrylic-on-board illustrations for covers of BBC micro books published by Granada in the 1980s. His works are appearing on eBay.com, this one has a starting bid of $379 (about £227). Offers, anyone?

GPIO for Beginners

So far in this series we've looked at controlling light emitting diodes (LEDs) from the Raspberry Pi's GPIO (General Purpose Input Output) port. We saw that relatively simple Basic programs can do things like generate morse codes by flashing the LEDs.

This time we'll look at wiring up a simple 7-segment LCD module shown in figure 1.

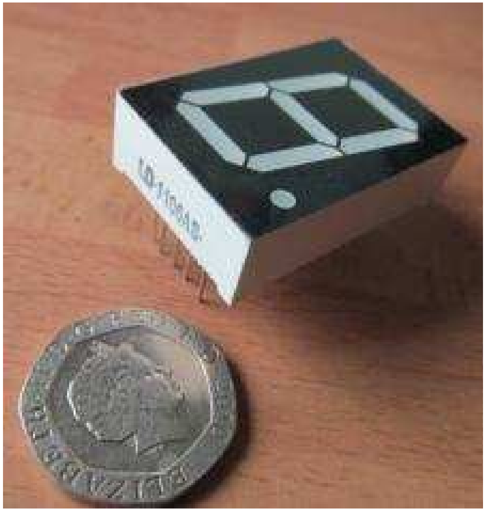


Figure 1. A 7 segment LED module

It can display the digits 0 to 9 and has 10 pins, five at the top and five at the bottom. Two pins (the middle one on the top and bottom row) are connected to the ground (0v) pin of the GPIO. The remaining eight control pins light up segments of the display (7 segments of the digit plus the "decimal point"). The LCD module is rather large to fit on the breadboard so connect it with male-to-female jumper leads (example shown in fig.2).

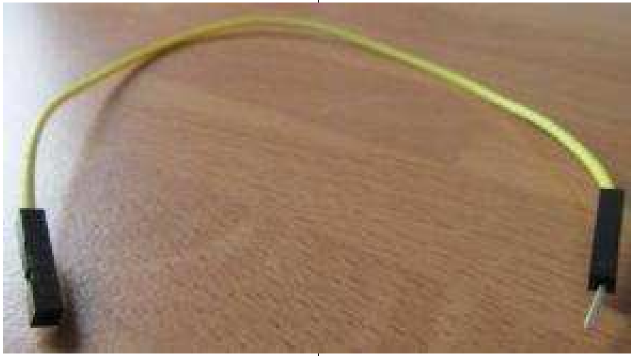


Figure 2. A male-to-female jumper lead

By switching on combinations of the pins we can get the digits we want. We must first find out which pin of the module controls which segment.

You can do this by connecting in turn each of the eight control pins of the module to one of the 5v pins of the GPIO. Figure 3 on the next page shows the wiring diagram and a photo.

Write down which pin lights up which segment. There is a convention on the internet of labelling the segments with letters of the alphabet which is what I have done in figure 4.

Now wire up the eight control pins to eight IO pins of the GPIO. It doesn't matter which ones, figure 5 and the photo shows one arrangement and figure 6 the wiring diagram.

The example being shown in red is the digit 2 consisting of five segments controlled by pins 7, 14, 18, 24 and 25.

Incidentally there is no rhyme or reason for the coloured jumpers. The set of leads I bought came in ten colours and using the rainbow mnemonic "Richard Of York Gave Battle In Vain" (plus Brown and Grey) in ascending order of IO pins I connected them to.

Experience of the SPI Bus

Having had the Pi for a while, I purchased a Gertboard to experiment with writing programs for flashing LEDs and controlling stepper motors.

My main objective was to resurrect my interest in using Forth as it is an ideal language for interfacing to and controlling peripheral hardware.

Eventually, the stepper motor was whizzing around and lights were flashing as required when using the GPIO lines as outputs.

My next task was to try to control the stepper motor's speed of rotation using an external input i.e. via the Analogue to Digital Convertor (ADC) on the Gertboard. This was when I first became aware of the existence of an SPI bus which stands for **S**erial **P**eripheral **I**nterface bus. This was the method by which the ADC was to be accessed.

There is not yet a driver under RISC OS for the SPI bus although one is being developed by ROOL. This will be quite a sophisticated package to cover all relevant hardware and methods of control (polling, interrupt and possibly DMA).

My own needs were much simpler than this, so I decided to try to develop a simple polling system just for the Pi.

First, a short digression. You can refer to the datasheets on the MCP3002 and the BCM2835 peripherals, where a much fuller explanation is available, but briefly there is a "master" SPI on the Pi, and a "slave" SPI ADC (MCP3002) chip on the Gertboard.

The slave is accessed by pulling the chip-select line down low and a command is sent via a "First-in-First-out" (FIFO) register. The slave then sends back some data, overwriting what was in the FIFO.

So there is no addressing of the chip in the conventional manner, there being only four connections between master and slave, as shown in figure 1:

In addition there are three registers on the SPI Master which require to be accessed and manipulated to make the transfers work. These are:

- SPI Master Control and Status
- SPI Master TX and RX FIFOs and
- the SPI Master Clock Divider.

The first hurdle is to find the actual addresses of these registers, since the physical address on the ARM is mapped onto the peripheral bus address internally to the ARM, and we require the bus addresses.

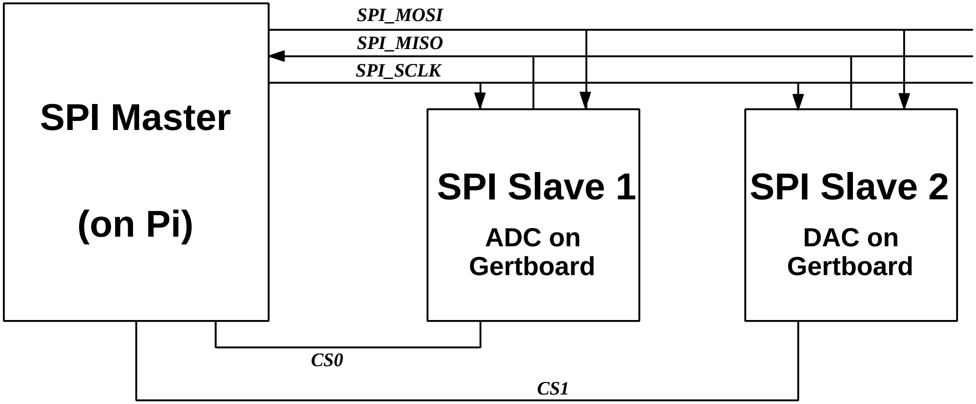


Figure 1 - Basic Layout of SPI Bus. CS0 and CS1 are "Chip Select" lines (active when low). Data Transfer Lines: SPI_MOSI : "Master Out - Slave In" ; SPI_MISO: "Master In - Slave Out"

Wallace Goes Walking

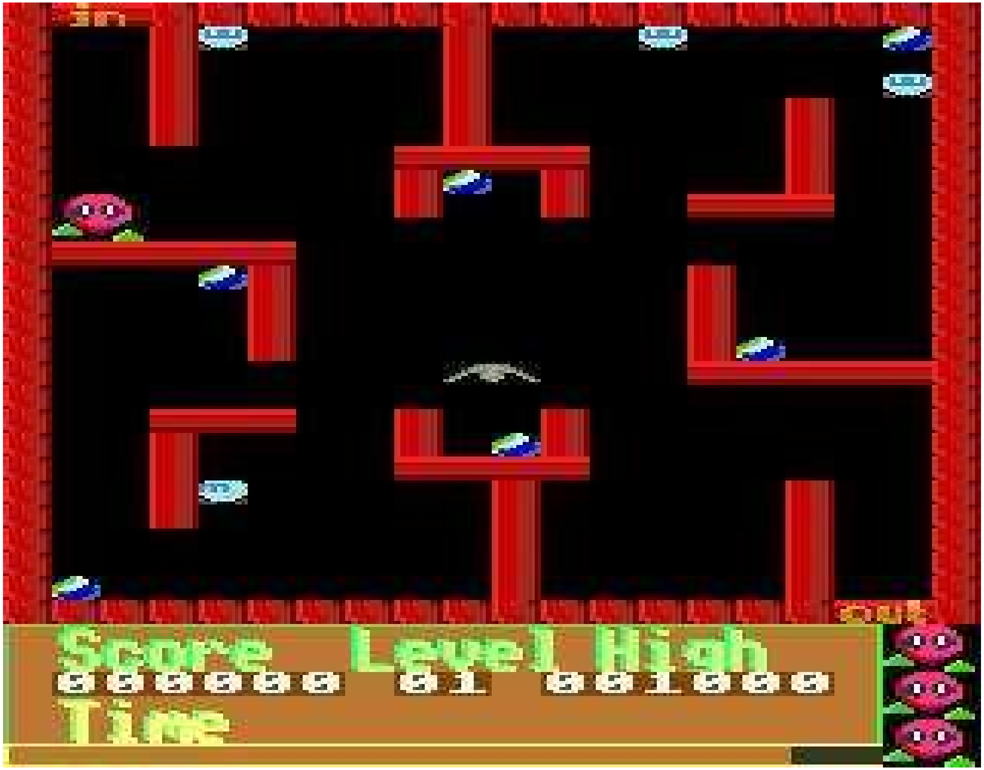
In this game (inspired by a BBC Micro game of a similar name) the evil Gaarts have stolen all the Easter eggs. Wallace must collect them all and make his way to the exit on the bottom right of the screen which will move him to a different and harder maze.

There are four robots and a bird that try to stop you. Contact with them is fatal and after every nine screens another bird appears up to a maximum of three. The game features colourful MODE 10 graphics and a high score table.

!Paint doesn't allow you to create MODE 10 sprites directly, MODE 10 has 22.5 dpi horizontally whereas the minimum Paint lets you choose in the "Create new sprite" window is 45 dpi.

So the first thing you must do is type in Listing 1 which generates a sprite file of "blank canvasses" (see figure 1). Load the resulting BLANK into Paint and colour in the shapes as shown in Figure 2. Save them as "WSpr" then type in listing 2, the main game.

Before running the game, ensure memory is allocated to the system sprites by right clicking on the Raspberry Pi icon and under "System memory allocation" drag the slider a small distance. To get sound effects install the User Voices as described in the separate article in this issue.



Sprites and full listing begin on page 17

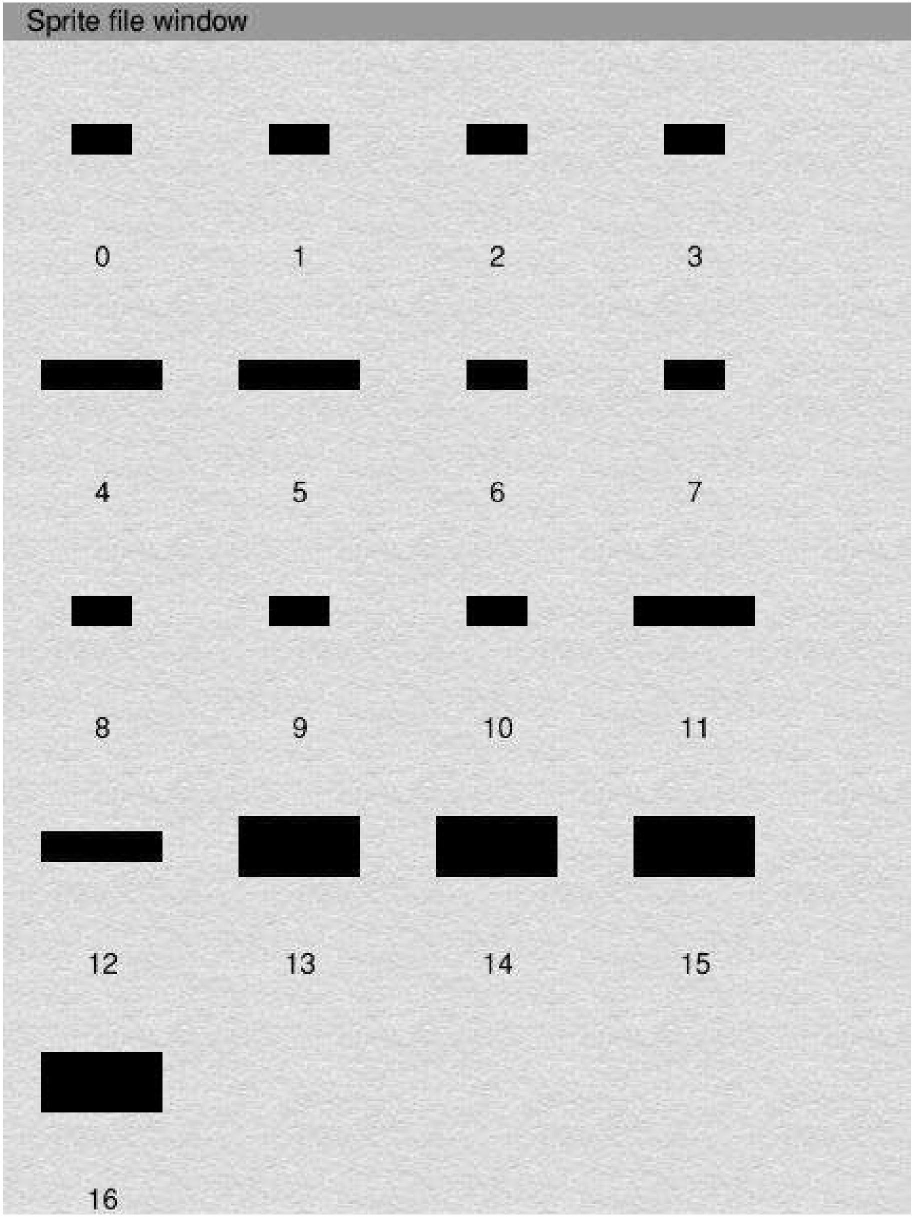


Figure 1: the blank MODE 10 sprites created by Listing 1

Listing 1

```
   10REM Wallace Goes Walking
   20REM Generate blanks
   30REM (c) Drag N Drop 2014
   40DIM sprites% &1400
   50number%=17
   60ptr%=12
   70!sprites%=number%
   80sprites%!4=&10
   90FOR shape%=0 TO number%-1
  100READ w%,d%:size%=w%*d%*2
  110ptr%!sprites%=44+size%:RE
M offset to next sprite
  120$(sprites%+ptr%+4)=STR$sh
ape%+STRING$(12-LENSTR$shape%,
CHR$0)
  130ptr%!(sprites%+16)=w% DIV
2 -1
  140ptr%!(sprites%+20)=d%-1
  150ptr%!(sprites%+24)=0
  160ptr%!(sprites%+28)=31
  170ptr%!(sprites%+32)=&2C
  180ptr%!(sprites%+36)=&2C
  190ptr%!(sprites%+40)=10
  200FOR x%=0 TO size%-1
  210x%?(sprites%+ptr%+44)=0
  220NEXT
  230ptr%=ptr%+size%+44
  240NEXT
  250sprites%!8=ptr%+4
  260SYS "OS_File",0,"Blank",0
,0,sprites%,sprites%+ptr%
  270*settype blank sprite
  280DATA 8,8, 8,8, 8,8, 8,8,
16,8, 16,8, 8,8, 8,8, 8,8, 8,8
, 8,8, 16,8, 16,8, 16,16, 16,1
6, 16,16, 16,16
```

Writing a simple app

Hopefully you have finished your homework which was to design a window with the WinEd template editor.

We covered quite a lot so don't worry as there isn't much to do in class today. We are going to find out how to load in the templates file and display the "About this program" window on the screen.

Firstly, though, we have to tell the computer where to find the window template file. As the human user, we know it's inside !mycalc which may be in a folder called "Programming" on your hard disc perhaps called "16GbPi".

But the computer doesn't know that. We have to tell it by assigning what's called a *system variable*, a string variable holding the pathname of the templates file.

Open the !mycalc.!Boot file in your text editor by holding down shift and double clicking on !Boot with select. Update it so it reads:

```
iconsprites <Obey$Dir>.!sprites
set Calc$Dir <Obey$Dir>
```

As you know, the first line tells the computer where the sprites file is for displaying the icon in the filer directory. <Obey$Dir> means "the directory in which the obey file was obeyed". The second line sets the system variable called Calc$Dir in the same way. The templates file is then referred to as <Calc$Dir>.Templates. The computer always knows what <Obey$Dir> is, it's a "built in" system variable, but will only know what <Calc$Dir> is when you have set it up.

Take a look Listing 1, the updated !Run file for this instalment. It's the same as last time with additional lines and so you can easily identify the extra lines I've indented them by two spaces.

block% the general-purpose workspace has been increased from 256 bytes to 1024.

windows% is the actual block of memory workspace needed to initialise the windows, it is DIMensioned to the size *wsize%* (&1400 bytes).

Three SYStem calls are used to load in the window template. SYS "Wimp_OpenTemplate" opens up the templates file. You can see <Calc$Dir>.Templates as the second item, or *parameter*, given in the SYS call.

Where you see multiple commas (two or more commas next to each other) in SYS calls, it means that parameter isn't used or we don't care about it. In this case parameter zero isn't used and parameter one is the name of the templates file. That's why there are two commas between Wimp_OpenTemplate and <Calc$Dir>.Templates.

SYS "Wimp_LoadTemplate" is a bit complicated. Even then I've simplified it a bit because we only have one window to load in! But all you need to know is that it looks for the window specified (in this case "info") and loads its details into the windows% area of memory, using *block%* as a buffer.

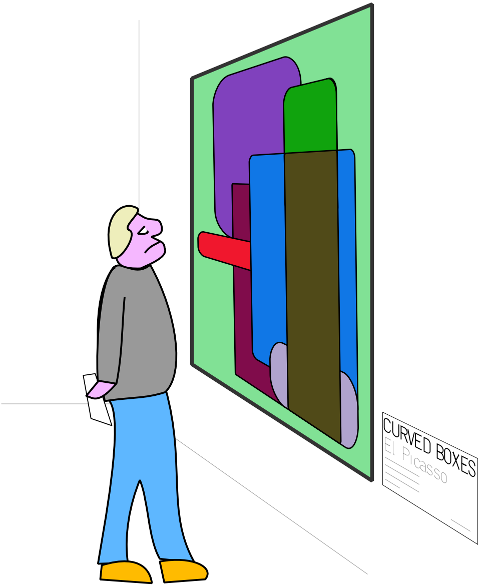
Now the window file is in memory, SYS "Wimp_Create" window creates the window. The TO means "return information TO the user" and the computer is returning the window handle or window ID, this is *info%* so we can refer to it in future.

I've reduced the number of menu options to three and instead of a simple list of menu items in the DATA line (About,Option,Quit), there is an extra piece of information which describes whether a *sub menu* is to open when the user moves the pointer to the right.

-1 means there is no sub menu. Other values mean either open a submenu of options or open a window. In this case we give the window handle *info%* of the "About this program" window to be opened.

Because there are three menu options, the size of the menu is smaller and I've altered PROCclick so that the y coordinate of where the menu appears is 240 instead of 300 because the menu smaller because it has less items on it.

Anatomy of a Drawfile



So far in this series on looking at the innards of a Drawfile we have found out how path objects and text objects are stored.

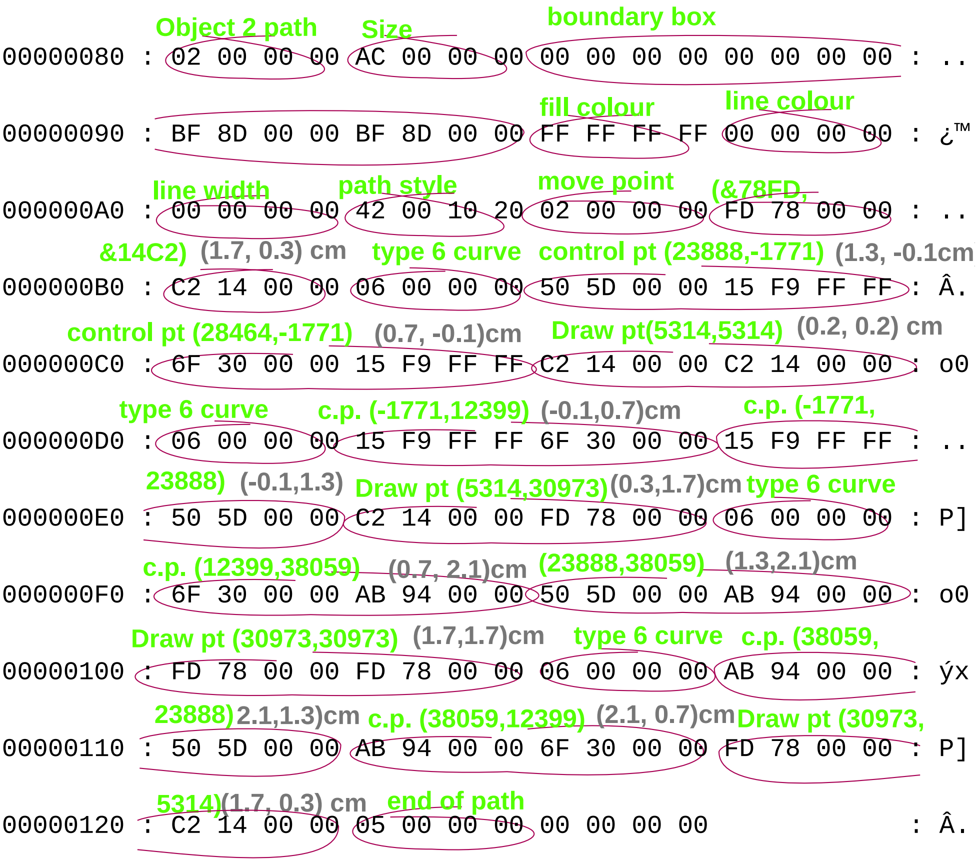
With path objects we know that simple straight lines consist of *identifiers*. Identifier number two tells the computer to move to the coordinates specified in the following two words of memory.

Identifier number eight tells it to draw a line to the coordinates in following two words. A word is four bytes of memory, the same as a four byte integer in BBC Basic.

We'll now look at identifier six which means "draw a curve". Figure 1 is a memory dump of a Drawfile with a circle drawn using Draw's circle drawing tool.

I've started the dump at offset &80 because the preceding bytes are the same as last time. Refer back to part 1 if you're unsure of what they mean.

Figure 1.


```
00000080 : 02 00 00 00 AC 00 00 00 00 00 00 00 00 00 00 00 : ..
00000090 : BF 8D 00 00 BF 8D 00 00 FF FF FF FF 00 00 00 00 : ¿™
000000A0 : 00 00 00 00 42 00 10 20 02 00 00 00 FD 78 00 00 : ..
000000B0 : C2 14 00 00 06 00 00 00 50 5D 00 00 15 F9 FF FF : Â.
000000C0 : 6F 30 00 00 15 F9 FF FF C2 14 00 00 C2 14 00 00 : o0
000000D0 : 06 00 00 00 15 F9 FF FF 6F 30 00 00 15 F9 FF FF : ..
000000E0 : 50 5D 00 00 C2 14 00 00 FD 78 00 00 06 00 00 00 : P]
000000F0 : 6F 30 00 00 AB 94 00 00 50 5D 00 00 AB 94 00 00 : o0
00000100 : FD 78 00 00 FD 78 00 00 06 00 00 00 AB 94 00 00 : ýx
00000110 : 50 5D 00 00 AB 94 00 00 6F 30 00 00 FD 78 00 00 : P]
00000120 : C2 14 00 00 05 00 00 00 00 00 00 00             : Â.
```

Scrolling Starfield

This is a simple but visually stunning program which displays a backdrop of scrolling stars in Mode 13. It is written entirely in Basic and can be incorporated into your own programs e.g. as a backdrop to a space invaders game.

Input the number of stars (for example 50) and sit back and watch the display.

The program first uses a SYStem call, OS_ReadVduVariables, which returns information about the current screen mode. On entry the parameter block must contain 149 for the first word and minus one for the second. On exit, the third word contains the address in memory where the screen Ram begins. This is assigned to *start%* in line 110.

Two tables of coordinates are then set up, *xc%* and *yc%,* then filled with random coordinates within the screen boundaries (320 pixels across and 256 down).

PROCstars plots the stars by looking up the stars coordinates and working out the correct screen address. The loop counter, *i%*, is the actual value EORed into the screen memory. It saves having a separate table of the stars' colours.

PROCmovestars calls PROCstars to erase the stars and increments each star's y co-ordinate by one. As there are 256 pixels down the screen, the Y coordinate is ANDed with 255 so that if it goes off the bottom of the display it appears back at the top. Then PROCstars is called again to plot the stars in their new position.

The *FX19 in line 220 waits for the ‘frame flyback’ to give a smoother effect.

For a parallalax effect change line 400 to:

```
400pos%!yc%=(pos%!yc%+1+(i% AND 1)) AND 255
```

This just advances each second star's Y coordinate by two. To scroll the stars up the screen just decrement instead of increment the Y coordinate, or even scroll across by changing the X coordinates instead.

```
 10REM Scrolling Stars
 20REM (c) Drag N Drop April 2014
 30:
 40MODE 13
 50INPUT "Number of stars",nstars%
 60:
 70DIM block% 8
 80!block%=149
 90block%!4=-1
100SYS "OS_ReadVduVariables",block%,block%+8
110start%=block%!8
120
130DIM xc% nstars%*4
140DIM yc% nstars%*4
150
160FOR i%=0 TO nstars%*4-4 STEP 4
170i%!xc%=RND(320)
180i%!yc%=RND(255)
190NEXT
200PROCstars
210REPEAT
220*FX19
230PROCstars
240PROCmovestars
250PROCstars
260UNTIL 0
270END
280
290DEF PROCstars
300FOR i%=0 TO nstars%-1
310pos%=i%*4
320addr%=start% + pos%!yc%*320 + pos%!xc%
330?addr% = ?addr% EOR i%
340NEXT
350ENDPROC
360
370DEF PROCmovestars
380FOR i%=0 TO nstars%-1
390pos%=i%*4
400pos%!yc%=(pos%!yc%+1) AND 255
410NEXT
420ENDPROC
```

RISC OS Font System

The Font Manager module under RISC OS provides everything necessary for "proportionally spaced" fonts. The space occupied by a character depends on the character itself, for example an "i" occupies more space than an "m" or a "w".

One problem with inferior operating systems is that the resolution of the screen is much lower In terms of dots-per-inch (dpi) than can be obtained on paper using a laser printer. Fonts which look smooth when printed on paper tend to have a very ragged appearance on the screen.

RISC OS however can display each individual pixel in any one of a whole range of colours. This can be used to good effect to make shapes appear smoother on the screen using a technique called **anti-aliasing** because it reduces **aliasing**, where the raggedness of the outline of a low resolution shape makes it look like something else.

Essentially, each pixel is plotted with an intensity level representing the proportion of the pixel which is inside the shape. A pixel which lies entirely within a shape is plotted at full intensity and one which is entirely outside the shape is not plotted at all. The interesting case is when the precise edge of the shape runs through the middle of a pixel.

For example, where a line should really be running between opposite corners of a pixel, instead of either leaving the pixel white or totally filling it in black, a mid grey is used instead

This anti-aliasing method is used by the RISC OS Font Manager to smooth the edges of the characters and give them a more rounded appearence as if they were drawn at a higher resolution than they really are.

The RISC OS Font Manager module stores the shapes of the characters in terms of the lines and curves which make up their outlines, which is why they are sometimes called **outline fonts**.

These line and curve outlines can then be scaled up and down mathematically to any desired size before being used to draw and fill the shape to the screen.

This has the obvious benefit of reducing the amount of space used to store the fonts and the even greater advantage of producing very precise character shapes throughout the range of sizes. The results on both the screen and the printer are noticeably more pleasing to the eye than with old fashioned, bitmapped systems.

Options are provided to switch anti-alisaing on and off. It is even possible to set a threshold size below which anti-aliasing is used and above which the characters are rendered directly. It is a matter of taste but on a standard monitor I tend to use the anti-aliasing only on very small lettering.

The definitions of characters are loaded into memory in an area called the **font-cache**.

This speeds up the process of drawing the fonts onto the screen since they need not be fetched from harddisc each time. It is impractical to hold definitions of all the fonts since even using the outline-based system the amount of memory would be huge.

Instead the font manager keeps the definitions of the characters which have been used here in groups of 32 by ASCII value.

The number of groups which can be held depends on the size of the cache which can be configured. For example:

```
*CONFIGURE FONTSIZE 1020K
```

sets the cache size to 1020K bytes.

The size of the font cache can be seen and altered directly from the task manager window under the RISC OS desktop. There is an upper limit of 32Mbytes in case you are interested!

The Font Manager treats fonts in a way similar to the manner in which the Operating System treats files. You have to "open" a font, read things from it and then "close" it again. This process is performed using the Font Manager

Inside Drag 'N Drop

Drag 'N Drop is produced entirely on RISC OS machines. By popular request we sneak inside the *Drag 'N Drop* production room to see how it's done, courtsey of your editor.

A lot of the work is done with commercial software packages e.g. Artworks but there is also a great deal which can be achieved with the free apps e.g. Draw which comes with every copy of RISC OS. So I hope to provide an interesting insight and you can pick up some hints and tips along the way.

I use are a Raspberry Pi with RISC OS 5.20 as my main machine supplemented by RPCEmu on a Windows computer when I am away from home (having a spare few minutes in the day job for instance!).

For those who bought the back issues collection, printing of the CD-Rom labels takes place on an Epson Stylus RX585 inkjet printer, with a Gutenprint driver. (Availability and quality of the CD stickers varies and I found local shops just as good as online retailers.)

I have owned Epson printers for most of my computing life and generally found them the most reliable. The individual ink tanks can be replaced. I had a Canon inkjet once with individual tanks but I see now even they are opting for the all-in-one cartridge. The RX series (at least models up until my 585) has a direct scan-to-memory-card feature, meaning you don't have to rely on computers for e.g. scanning text for OCRing. The scan-to-card aspect of the RX585 was heavily used during the production of the 55 BBC Micro Books CD-Rom.

Printing of show leaflets takes place on a Brother HL5050 mono laser jet, for which there a native RISC OS driver (the HL5040, found in PrintDefs:Brother if you select "Install new printer" from your printer icon bar icon).

The page size of *Drag 'N Drop* is 21 by 18cm, the same width as an A4 sheet but about 70% of its height. This is deliberate because historically TVs and computer monitors have a 4:3 aspect ratio. Modern widescreen monitors are 16:9 but in both cases a page can fit comfortably on the screen, assuming it is viewed at 100%. Also 21 and 18 are easy whole numbers to remember!

The magazine covers are put together in Artworks. The main reason for this is the graduated fill tool (used in the *Drag 'N Drop* logo for instance) though the same can be achieved using the Interpolate/Grade option in Draw albeit to a limited degree. But actually I usually start with sketches in Draw because I find the line editing tools in Draw easier to use than Artworks. Though that could be because I've been using Draw for a lot longer!

The 'washing powder' on the cover of 5i2 started life as a 'flat pack' and 'assembled' using the object distortion tool in DrawWorks XL (fig.1), a commercial appliation (but very good value for money) then imported into Artworks to apply light and shade to the sides of the box.

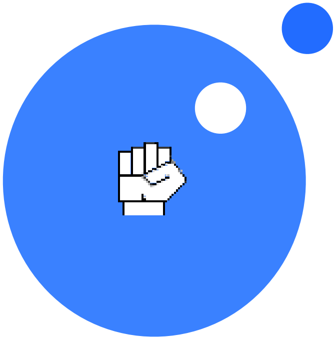
Doing More With Text

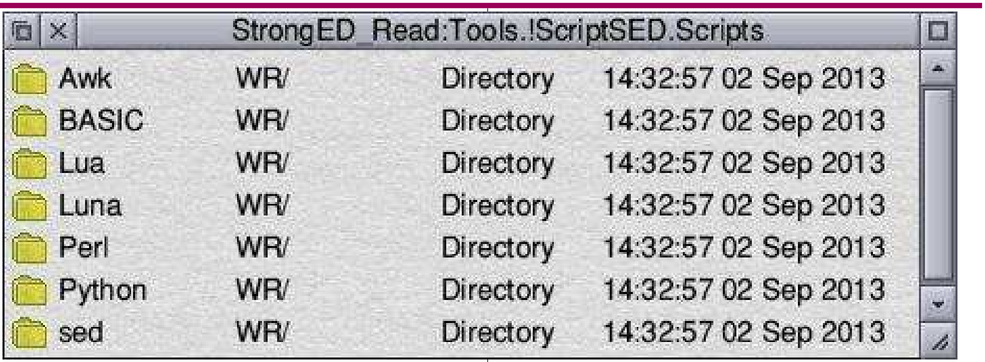
If you use StrongEd, and it's a recent version, you may have noticed this icon on the smart iconbar along the top of a StrongED window. I refer to it as the *apply icon*. Its purpose is to let you run programs to alter the text in the window. These programs are called *scripts* and in this article I will show you how to create your own scripts.

If you drag a script onto the apply icon with the Shift key held down, the text in the window will be overwritten by the output of the program

If you drag with the Ctrl key held down the output will be displayed in a new window, from where you can save it.

If both Shift and Ctrl keys are held down the script will be applied to *all* the currently open StrongED windows simultaneously. These defaults can be altered by editing StrongED's BaseMode, should you wish to.





StrongED_Read:Tools.!ScriptSED.Scripts

Awk	WR/	Directory	14:32:57 02 Sep 2013
BASIC	WR/	Directory	14:32:57 02 Sep 2013
Lua	WR/	Directory	14:32:57 02 Sep 2013
Luna	WR/	Directory	14:32:57 02 Sep 2013
Perl	WR/	Directory	14:32:57 02 Sep 2013
Python	WR/	Directory	14:32:57 02 Sep 2013
sed	WR/	Directory	14:32:57 02 Sep 2013

StrongED has many facilities for editing text already. Scripts can be used for doing things for which StrongED was not originally designed for. Some scripts are already provided by StrongED. Click the apply icon with Select and a filerwindow will open showing various directories, labelled by scripting language, which may hold scripts in that language.

Unfortunately some of these already provided scripts may be out of date or not functioning correctly. Your own scripts can be kept wherever you like; you do not have to use the directories that StrongED provides.

We will create scripts in Lua, or rather RiscLua, an implementation for RISC OS. You can obtain the latest version of RiscLua from PlingStore or from http://lua.riscos.org.uk/ . Unzip the zipfile and put !lua in $.!Boot.Resources. Then StrongED can use Lua scripts as soon as your machine has been booted.

Beside extending the powers of StrongED, scripting is also a good way of learning how to program by experiment. You get immediate feedback; your script and the text it is to act on can both be displayed alongside each other in StrongED windows.

In short, StrongED provides a convenient development environment.

To avoid overload the examples presented in this article are not completely polished.

StrongED will recognize a file dragged to the apply icon as a Lua script if it has filetype Lua (& 18C) or if it is a textfile whose first line is:

```
#! lua
```

Note that the # must be the first character on the line. This can catch you

Bitcoin

Bitcoin is a form of electronic currency. There are a lot of benefits to using bitcoin, and the benefits will only grow as the system gains wider acceptance.

But the problem is that, unlike the pound in your pocket, you have to know quite a lot about the underlying technology before you can start to use it safely. So it is well worth spending some time to find out more about it.

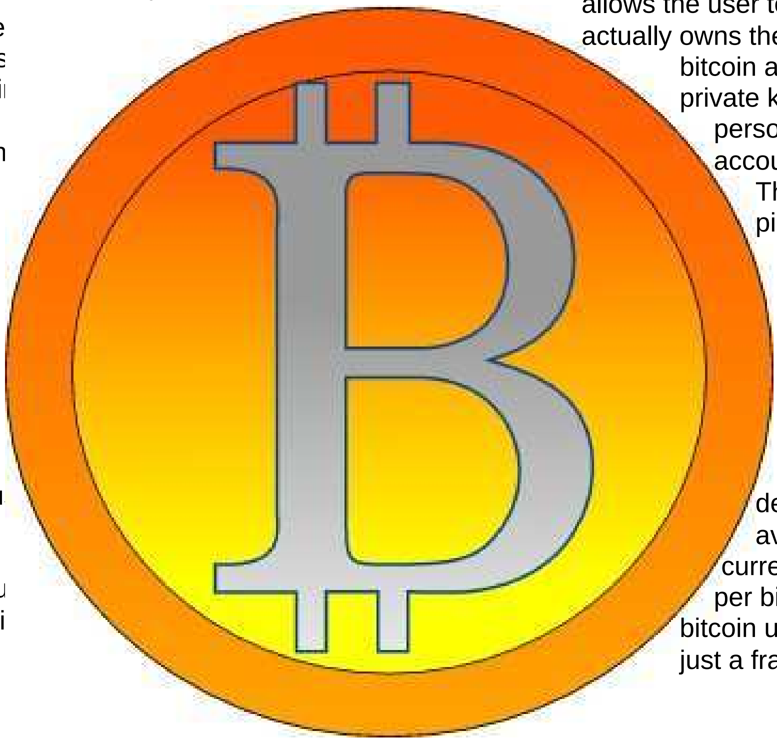
A single bitcoin is just the answe complex mathematical problem. It is effectively, a very rare number. At fi glance, it may seem strange that something that is nothing more than number can be a form of currency.

But all currencies, like the euro, the dollar and the yen, have no inherent value. They only have value because people let you swap them for things. In the same way, bitcoin became a form of currency as soon as people started being prepared to accept it as payment.

Its value is determined by variou factors, including its scarcity, the number of people who want it, the number of people willing to let you u as payment, and the risks involved i using it.

Bitcoin Wallets

In order to become a bitcoin user, you need to install a bitcoin wallet on your computer or mobile phone. The wallet software will then generate your first bitcoin address, a long string of numbers and letters which you can use to spend and receive bitcoins. It is similar to your email address, but used for handling money.



However, you can also buy physical bitcoins that you can jingle about in your pocket rather than just existing as numbers stored on a big computer network.

Private Keys

A private key is a secret code which allows the user to prove that he or she actually owns their bitcoins. Every bitcoin address has a matching private key, which is kept by the person who owns the account.

This is the most critical piece of information which just like your bank card PIN number you have to keep secret. How much is one bitcoin worth? The price is constantly fluctuating according to supply and demand. But the average price is currently about £300-400 per bitcoin. However, many bitcoin users have balances of just a fraction of one bitcoin.

User Voice Generator II

This program originally appeared in Risc User and has been updated for the Raspberry Pi. Games programs can be enhanced greatly with the addition of sound effects e.g. Wallace Goes Walking which appears elsewhere in this issue.

The accompanying listing assembles and saves a module file called UserVoices. Simply double click it in a filer window, or include a command

```
*Run UserVoices
```

in your programs.

Seven new voices are installed. To hear them in action in BBC Basic, first of open a task window by pressing CTRL+ F12. Type:

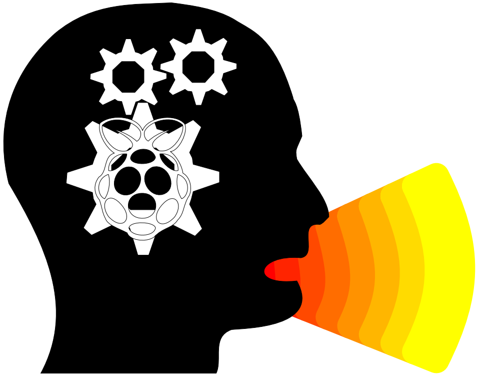
```
*basic
```

followed by the Return key. Next type:

```
*voices
```

A list of current voices is shown, similar to the below:

```
         Voice      Name
1          1   WaveSynth-Beep
           2   StringLib-Soft
           3   StringLib-Pluck
           4   StringLib-Steel
           5   StringLib-Hard
           6   Percussion-Soft
           7   Percussion-Medi
um
           8   Percussion-Snar
e
           9   Percussion-Nois
e
          10   Ping
          11   Warble
          12   Bash
          13   Pow
          14   Trem
          15   Trill
          16   Rise
^^^^^^^^ Channel allocation ma
p
```



The first nine voices the default ones. Voices 10 to 16 are the new ones. The "1" to the left of voice 1 means channel 1 is assigned to WaveSynth-Beep, the tone your computer makes when it boots up.

You can change this with the *CHANNELVOICE command. For example:

```
*CHANNELVOICE 1 10
```

or

```
*CHANNELVOICE 1 Ping
```

makes the computer play the "Ping" sound instead of "WaveSynth-Beep".

The CHANNELVOICE command like other star commands isn't case sensitive but the name of the voice is. You can either use the name or the numeric voice number (1-32, up to 32 voices can be installed).

It is better to use the voice name, though, because you might not know exactly which number the new set of voices start at (voice 10 might be the Ping, on the other hand it could be voice 11 if a previous program had installed a voice at number 10).

You must type it in exactly otherwise the computer will complain.

```
*channelVoice 4 Rise
```

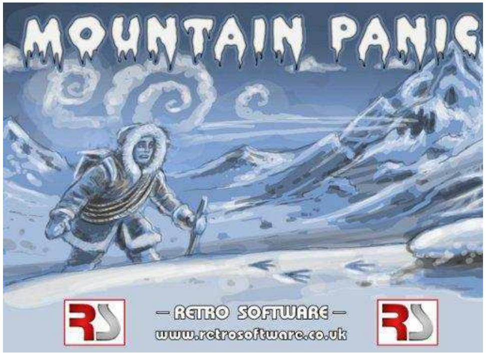
is acceptable (install Rise on channel 4) but

```
*Channelvoice 4 trill
```

isn't because Trill has a capital T.

If you want to use more than one of

Review

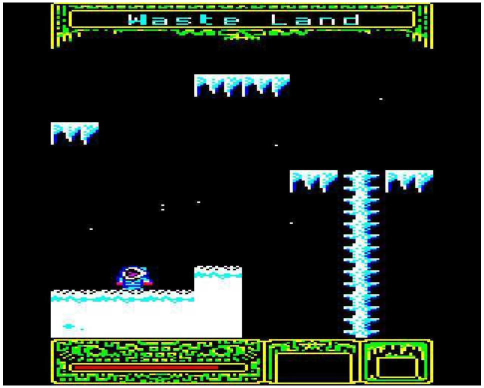


Mounain Panic

It is quite amazing that in 2014 brand new games are still being released for the good old BBC Micro. *Mountain Panic* is the latest release from Retro Software, it has been in development for a while, but thankfully it has been finished and is here for us to play. It can be played on a 'real' BBC as well as emulators such as !Beebit on RISC OS.

Set in a snowy wintry scenario, you take control of an explorer who goes by the name Bill, who whilst on an expedition wakes up one morning to discover the rest of his team have vanished, now alone, Bill will have to find his way through the caverns to make it out alive.

The game loads in the traditional Shift/Break way, after the introduction screens, you get the option of defining the keys and there is a more in-depth description of the background story for the game, press fire to start and you appear from out of your tent and the adventure begins.



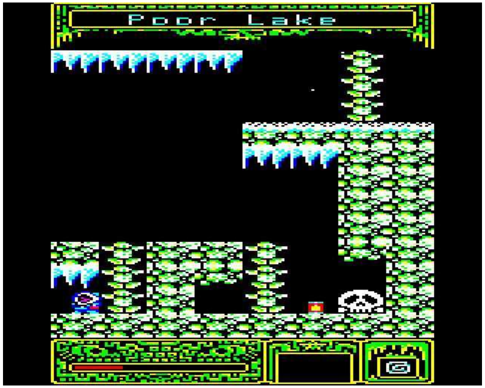
Mountain Panic is a standard arcade adventure very much in the same mould of the likes of *Citadel*, *Thunderstruck* and *Quest*, but there is a difference to this game and that is the rope. It doesn't sound much but it is the key to the game.

You need to collect the rope as soon as possible because it allows access to different areas of the adventure. There is no jump option in this game which does take a bit of getting used to, I have to admit, but it adds something different to the game.

The further you go into the game the more ropes you can collect to extend it and make it longer, so those high ledges which were previously out of reach can now be accessed. There are other items which need to be collected and used to solve various puzzles throughout the game.

It is a 'flip' screen game set across many screens – to find out how many it would be worth making a map as you go. Luckily the rooms are named so it will help to remember places. Watch out for the darkened catacombs – you'll need something to help you see down there !

So far I have managed to get quite a way into the game. I won't spoil any of the puzzles or give anything away as part of the fun is solving them yourself! From what I have seen they don't seem to be too difficult to solve and the items look like what they are supposed to be.



You also get a description of the item so that helps. There are a number of stars dotted about which need to be collected also.

Community Contacts

Developers & Publishers (RISC OS 5)

Archive www.archivemag.co.uk

CJE Micros www.cjemicros.co.uk

Datawave www.datawave.nl

David Pilling Software
www.davidpilling.net/riscos.html

Electronic Font Foundry thefonts.com

MW Software www.mw-software.com

Orpheus Internet
www.orpheusinternet.co.uk

Organizer www.organizerpim.co.uk

Piccolo Systems
piccolosystems.com/risc-os

PiLearn www.pilearn.com

R Comp www.rcomp.co.uk

RISC OS Code www.riscoscode.com

RISC OS Open Ltd
www.riscosopen.org

Webwonder/ProCad
www.zynet.co.uk/dsnell/Welcome.html

User Groups

HHRUG (Hemel Hempstead RISC OS user Group)
Area: Hemel Hempstead
Meets every 3rd Wednesday of the month
£3.50 per evening (£3.00 if paid 3 months in advance)
www.hhrug.org/

ICENI
Area: Ipswich
Meets every 1st Wednesday of the month
Visitors free for the first time
www.icenicomputerclub.org.uk

LAUG (Liverpool Acorn User Group)
Area: Liverpool
Meets every second Tuesday of the month
Free entry
www.orpheusweb.co.uk/bob.williams/laug/index.htm

ROUGOL (RISC OS User Group of London)
Area: London
Meets every third Monday of the month
Free entry
http://rougol.jellybaby.net

RONWUG (RISC OS North West User Group)
Area: North West
Meets 3rd Weds every month
Free admission
www.ronwug.org

SASAUG (Surrey and Sussex Acorn Users Group
Area: Surrey and Sussex
Meets every second Monday of the month
£1.50 per meeting for members
www.sasaug.org.uk

WROCC (Wakefield RISCOS Computer Club)
Area: Wakefield
Meets First Wednesday of the month
£2 for non members
www.wrocc.org.uk/

Ronwug
Laug
Wrocc
HHrug
Iceni
Sasuag

Are your RISC OS club details up to date? If not please let us know!