



DRAG'NDROP

RISC OS **Pi** and all RISC OS 5 machines

July 2014

Volume 5 Issue 4

£3.00

RiscOSM



Hardware
LCD Backpack
Sky Pi HDD

Programming
Edit search/replace
Armcode

Graphics
Panel designer
Drawrite



Midlands User Group Midsummer Show

Saturday 12th July 11.00 - 4.00



£1.00 children free

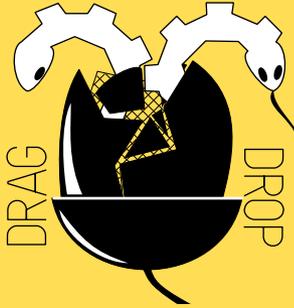
RISCOS and Raspberry Pi

TM

St John's Church Hall, 205 Warwick Road, Kenilworth CV8 1HY
exhibitors, maps, directions, charges and other details on website

<http://mug.riscos.org>

<http://raspberrypi.org>



Copyright © Drag 'N Drop 2014
Produced on RISC OS computers

This issue has been blessed with contributions from the following people:
Gavin Wraith (Doing More with Text)
Jon Robinson (Unicode and RISC OS Programming in Basic)
Paul Dunnington (LCD Backpack)
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 offensive 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

I hope you are having an enjoyable summer. Weatherwise it has been pretty good in Britain for a number of years and I have just been picking some real raspberries from the garden! As far as the Raspberry computer is concerned we have an eclectic mix of RISC OS articles in this edition.

If graphics are your thing then take a look at Drawrite and Games Panel Designer.

On the other hand if you like text manipulation then read about Doing More with Text, Edit Search & Replace.

On the hardware side we're driving one of those LCD panels (seen on radio sets, vending machines, music keyboards etc) with a new relocatable module. Excellent machine code commentary too if you want to learn about programming. If you aren't that brave Mr Frog is back with his irreverant look at Arm code.

Remember to keep backing up your files regularly; we have an inexpensive solution with the Sky Pi Box article. And if you are looking for something specific we have the updated index to everything that has ever appeared in *Drag 'N Drop*.

Thanks to everyone who has contributed articles, I really appreciate your hard work. I'm finding newsgroups and forums nowadays difficult to follow with people going off on tangents. So here is another focussed and polished publication which I hope you enjoy.

Chris.

Christopher Dewhurst

At a glance...

Editorial 2

Beginner's Tips 3

News and Apps 4

Reviews 6

Drawrite 9

Panel Designer 15

User Sprites 18

Doing More with Text 27

Edit Search/Replace 29

RISC OS programming
in BASIC 33

Sky Pi HDD 38

LCD Backpack 40

Unicode 55

Mr Frog 57

Vol 1-5 Index 58

Order Form 63

Getting Started with RISC OS: a refresher



Setting up an SD card for your Pi is a complicated business so we'll have a quick recap on creating a card using another RISC OS machine.

1. (The easiest option.) Buy an SD card pre-loaded with RISC OS for £12 from the ROOL store at www.riscosopen.org.uk/content/sales, OR

2. Buy your own SD card (2GB or larger) and an SD card reader. Buy SystemDisc from piccolosystems.com. Go to github.com/raspberrypi/firmware/tree/master/boot and download **bootcode.bin**, **start.elf**, **fixup.dat**.

If using Netsurf ensure Javascript is disabled and you must click on Raw to actually download. These files are needed regardless of the operating system which is why they are on that website rather than a RISC OS one. Rename them in CAPITALS and add the file extension: BOOTCODE/BIN, START/ELF, FIXUP/DAT.

Download the latest RISC OS rom image from www.riscosopen.org.uk > Downloads > Raspberry Pi. Unzip it and copy out the **riscos** file. Again rename it in caps to RISCOS. Then

CONFIG/TXT

Save in same place as BOOTCODE/BIN etc.

```
fake_vsync_isr=1  
hdmi_drive=2  
gpu_mem=64  
init_emmc_clock=10000000  
kernel=RISCOS
```

Include this line only
only if you are using a
HDMI to VGA adaptor cable.

This is the ROM image name.
It must be in CAPITALS.

use Edit to create a file called CONFIG/TXT (see illustration), Use SystemDisc to copy all five files to the boot partition, select SCSI drive 0. Finally download and install the RISC OS HardDisc4 boot sequence and applications. If in doubt follow the interactive instructions of SystemDisc, they are quite helpful.

Note Drag 'N Drop is not affiliated to Piccolo Systems nor RISC OS Open. Other methods may work on alternative systems but may not be as reliable.

News and Applications

Magazine price increase

As mentioned in last issue's News section a price increase is necessary for Drag 'N Drop. We've maintained the current price of £3 per issue for five years and while we don't have printing costs the rest of the world's prices are going up all the time (transport to shows, postage for sales of memory sticks, CDs etc.) The price from Volume 6 issue 1 onward will be £3.50 per issue.

Subscriptions

We are offering a subscription to Volume 6, open to everyone although designed for users with limited internet access and/or no Paypal account. A year of magazines (4 issues) costs just £14.00. Please fill in the form at the back of this issue, the form can also be downloaded from www.dragdrop.co.uk.

APDL founder dies

David Holden who founded and ran the Archimedes Public Domain Library (APDL) lost a very short battle with cancer and passed away at his home in East London in May 2014. The good news is that David's 25+ year long commitment will not be lost and the library will, in due course, be taken over by others in the RISC OS world.

Midlands RISC OS Show

The Midlands User Group (MUG) show takes place on Saturday 12th July in Kenilworth. Admission just £1. *Drag N Drop* isn't able to attend this year but we have pulled out all the stops to get the magazine issued before the show.

Pico

May 1964 was when one of the first versions of BASIC was released and to mark the 50th anniversary RISC OS Open have released Pico, an OS card to put in your Pi and program in BBC Basic. Pico costs £5 from www.riscosopen.org.uk and reviewed in this issue.

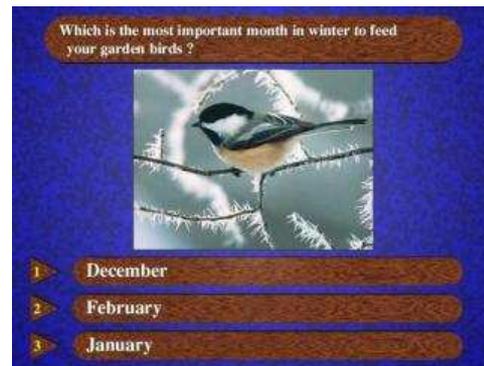
RiscOSM

A delightful release at the Wakefield 2014 show was a new mapping tool for RISC OS using data from the Open Street Map project. RiscOSM costs £20 on CD-Rom from www.sinenomine.co.uk and we have a full review in this issue.

Quizmaster

Jon Robinson has been busy updating Quizmaster, a graphically lavish, multiple choice quiz game with sound effects and the ability to design your own quizzes.

Available for £20 through eBay in download format.



Drawprint 1.46

The latest version offers significant speed improvements. Drawprint allows you overcome the limitations of bog standard A4 printers: it works out what it should print on what sheet so you can tile them all together to make a huge display, for example adverts, large spreadsheets or even family trees. sinenomine.co.uk/software.

Crowdsourced Acorn book

A 132 page full colour book is in the offing. *The Potted History of Acorn Computers* needs £2000 for a print run

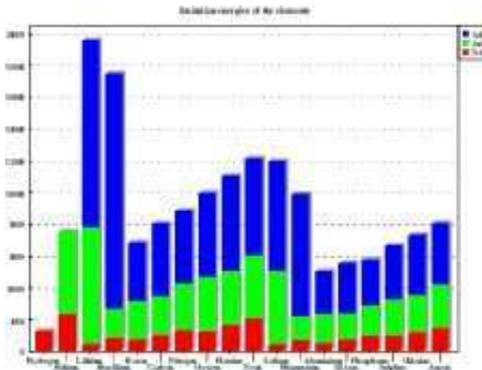
of 100 books. Pledge £5 for a digital download or £20 for a printed copy (other options are available). Just under half the funds have been raised with only 39 days to go so get along to www.indiegogo.com/projects/a-potted-history-of-acorn-computers and offer your support.

USB Audio

You may have printers and cameras connected to your computer by USB leads but did you know you can play sound files through USB too? Assuming you've got suitable hardware attached a suite of programs designed for use with a USB are in continuous development by its author Jim Lesurf: USBPlayer plays a WAV file, USBRecord records and a couple of other apps for analyzing the sound files. Download from www.audiomisc.co.uk/software/index.html (the RISC OS section is further down)

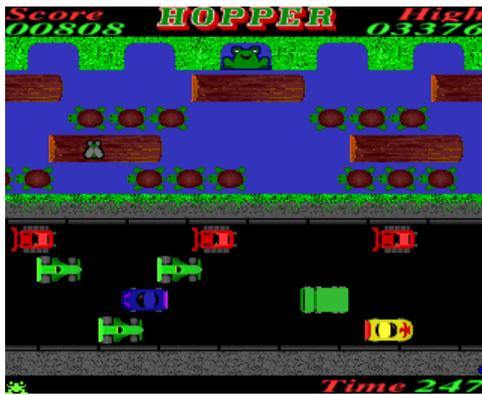
ChartDraw

This simple but elegant table/chart editor allows production of a variety of charts. Import your own comma separated files (CSV), edit and insert data then export the resulting bar or line charts, pie charts etc. as a Drawfile. ChartDraw 3.52 can be downloaded from www.chrisjohnson.plus.com/software/chartdraw.html



Hop on to a Meteor

Two classic BBC Micro games have received a RISC OS makeover: *Meteors* and *Hopper* now run on the Pi with souped up graphics and music. They are included in the HardDisc image from www.riscosopen.org.uk > downloads. Note you need the "Nightly Beta Harddisc4" as the games aren't in yet in standard, stable Harddisc4 archive.



ProCAD

Whilst the built-in Draw app is good for most diagrammatical needs, ProCAD is well worth the investment if you want to produce advanced drawings such as engineering diagrams, PCB designs, maps and the like. Version 1.08 of the CAD package now imports DrawPlus files and the manual has been made into HTML format with its own search engine. ProCAD costs £60 for a new version (upgrade options available) from www.dsnell.zynet.co.uk/

Pipedream 4.52

Version 4.52/04 of the three-in-one spreadsheet, database word processing package is available free from croftnuisk.co.uk/coltsoft/pipedream. Pipedream is similar in some ways to Fireworkz but with advanced features such as command (macro) recording. We'll be starting a series in the next issue of *Drag N Drop*.



Reviews

Product: **RiscOSM**

Version: **1.03**

Supplier: **Sine Nomine Software**

www.sinenomine.co.uk

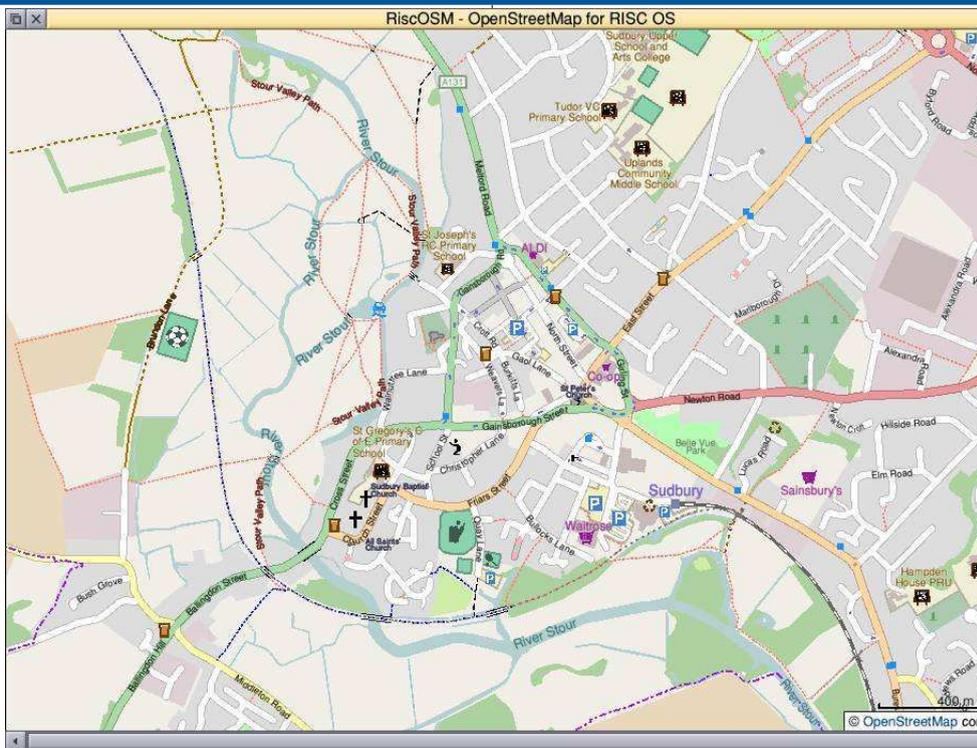
Price: **£20.00**

The Open Street Map (OSM) project was started in 2004 in Britain. Anyone can get free geographical information from its growing database to create maps such as town plans (www.osmfoundation.org). Most importantly the database has been built up by volunteers with local knowledge and is free from commercial restrictions you find in e.g. Google maps.

(Editor's note: apologies to residents of Risca in south-east Wales. Searching for a suitable illustration for the front cover of the mag I typed in RISC and your town came up!)

RiscOSM from Sine Nomine software is the RISC OS 'front end' to the database. It is essentially a mapping tool which creates high quality vector maps in Draw. Draw is the free vector editing application which comes with every version of RISC OS. Top quality maps for printing or web publishing can be created for virtually anywhere in Great Britain.

The application, plus a cut down database for Great Britain, comes on CD-Rom. You install the RiscOSM app from CD onto your hard disc.



The database (!OSMData) can also be installed or left on CD.

The recommendation is to install it for speed of access but at over 560 megabytes I found the data part took a very long time to copy so I left it on CD. !RiscOSM has to "see" !OSMData before it works.

Once installed, double clicking on !RiscOSM installs itself on the icon bar and select click brings up the New map

dialogue box. Start typing a town or locality name and the interactive search box brings up matching names. This is faster if you have !OSMData on your hard disc. You can choose one of the 14 preset scales (1:1250 very close up maps to 1:1,000,000) or type in a custom scale. You can also choose the degrees of longitude and latitude the map is centred on, though I didn't really find that useful.

Drawrite

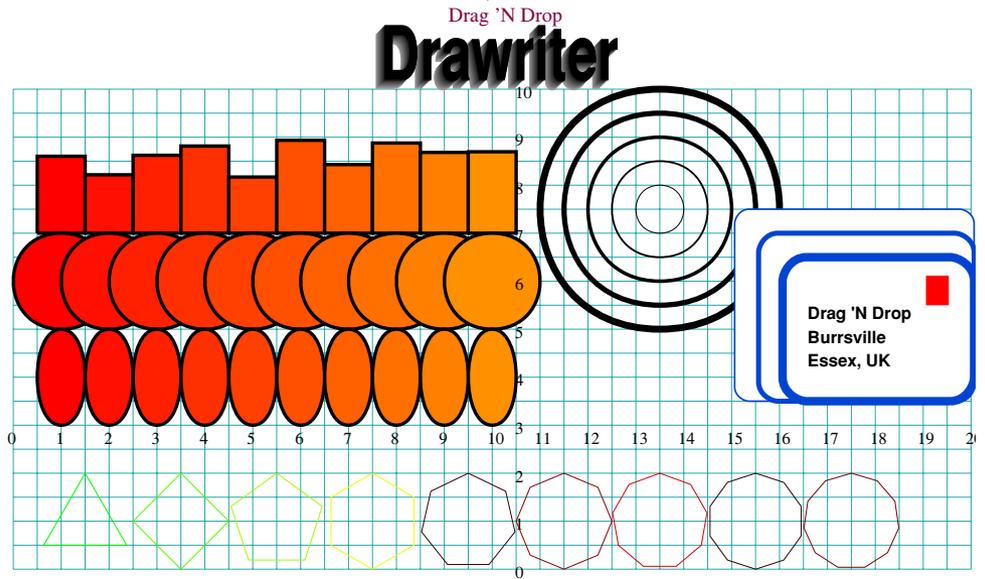
Over the past year we've been delving into the file structure of Draw files. We have been using our discoveries to write Basic programs to generate drawings of patterns more easily and accurately than you could just by pointing and clicking with the Draw application alone.

We round off the series with a powerful 'draw writer' program called Drawrite. This brings together all the routines (Basic procedures to create grids, circles etc.) in the series plus several more. The 'sampler' shown on the right is generated by the program.

Type in the listing – there aren't any line numbers as none are needed. Save it in a directory somewhere and make sure the CSD (currently selected directory) is set to it by clicking menu over the filer window and choosing Set directory.

The first part of the program is the demonstration and the second section contains the procedures. When you come to write your own writer program then substitute the first part with your own code but keep the procedures it calls on, which are described in the reference section below.

PROCinitialise should begin every writer program, it sets up the default line width, fill colour and so on.



It also sets up the standard Draw header in PROCheader. PROCheader in turn calls PROCfonttable which sets up a table of fonts.

Font ID number 0 is always the System Font and the program sets up ID number 1 as a sans serif (Homerton Bold) and ID number 2 as a serif (Trinity Medium) typeface. The text procedures (described below) require the font's ID. If more fonts are required add them to the DATA line after the PROCfonttable (terminating with ""). See the Winter 2014 edition of *Drag 'N Drop* for details of font tables and text objects.

PROCcircle(r) Draw a circle of radius r . The centre is assumed to be the current coordinates as set by PROCmove(x,y)

PROCcurvedbox(l,w,r) Draws a box with rounded corners of length l and width w with the curvature of the corners r . The higher the r value the more rounded the box appears.

PROCcms and **PROCins** set the units to centimetres and inches respectively. From then on all coordinates are measured in centimetres or inches as

Games panel designer

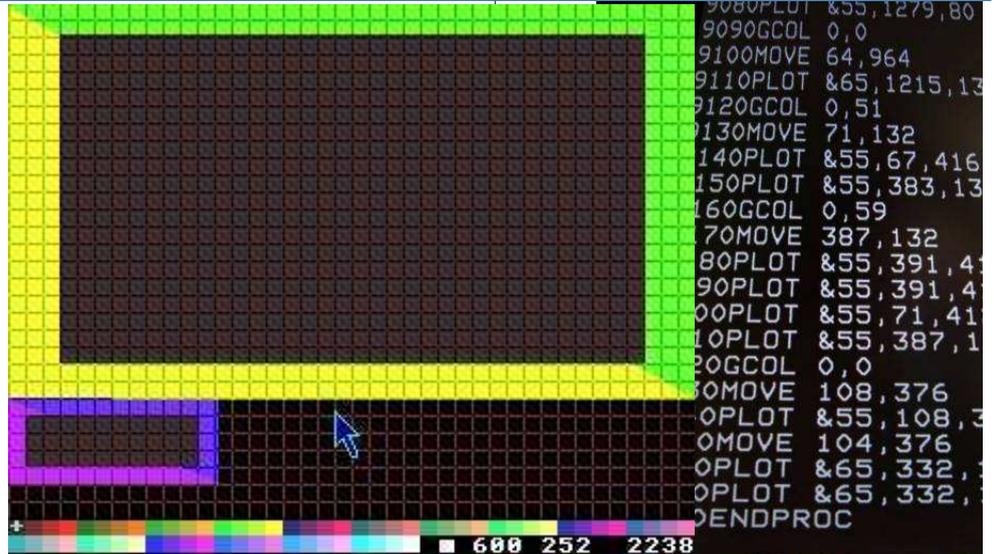
When writing games you usually need some sort of background with borders and boxes for the score and lives etc. Working out the necessary Basic MOVE and PLOT commands by hand is tedious so I wrote this program to assist.

You point and click with the mouse to draw lines, triangles, and rectangles in colours of your choice. Other shapes can be easily incorporated. When you have finished the program outputs a Basic procedure called PROCpanel to draw the screen which can be appended (*EXECed) onto your own program.

Type in the listing ensuring there are no errors and run it. The call to PROCdelay in line 460 is to slow down the mouse otherwise one click results in trying to process about 50 repetitions. Increase the number if you still find it too fast on your RISC machine.

At the bottom of the screen is a colour palette. Click on a square to set the colour. Select (left) click on the line symbol to cycle through line, triangle and square tools. Next to the tool icon is the coordinates of the pointer.

To start drawing click select anywhere on the screen and a line will follow the pointer. Click select to fix the next point or adjust (right button) to finish the path. The delete key deletes the last line segment or shape.



Rectangles need one previous point which could be part of a line or simply a click if you already have the rectangle tool selected. Triangles are exactly the same except two previous points are needed.

Keys 1 and 2 on the keyboard draw two types of grid, a square (intended for games running in a 'square pixel' mode like Mode 13) and a rectangular grid (for games in 'wide pixel' mode like MODE 10). Each line of the grid marks out text character positions to assist accurate positioning of boxes which will have text like the score PRINTed in them at the text cursor.

The other two numbers at the bottom

of the screen show the serial number of the shape being drawn and the total coordinates stored so far. Don't worry too much about these except the program has a limit on 200 shapes and 400 coordinates. This should be enough for drawing most backdrops but increase the value of *max%* in line 190 if desired.

Be aware that changing colour records the colour in memory so when deleting a shape you might need to press delete once more than you may think.

When you have finished press Return and type in a filename. The necessary DRAW and PLOT commands are saved in the currently selected directory (CSD).

User Sprite Areas

Before you start using sprites on RISC OS you have a choice to make. There are two areas where your sprites can live. One option is to place them in the area which the operating system reserves for sprites. The alternative is to define your own user sprite area within application or module workspace (depending on the type of program you are producing).

The advantage of using the system sprite area is that you can manipulate the sprites using simple star commands. The disadvantage is that you don't have full control over the area.

The Task Manager (adjust click over the Raspberry Pi icon in the bottom right corner of the desktop) has full control over the sizes of all the different memory areas and so memory can be reallocated from one area to another as the need arises.

The problem now is the fact that your program would have to share this area with any other sprites which happen to be there at the time. This means that potential name clashes could occur leading to the user's existing sprites being accidentally overwritten. In addition, your sprites would be vulnerable to editing by the user so you could not rely on them remaining a constant shape or size.

Defining your own area is the recommended method. It gives you full

control over the size of the area used – you can make it as large as the memory permits without the user being aware of what you are doing. However, to manipulate the sprites you have to call operating system routines either via the SWI mnemonic in assembler or SYS in BASIC. The routines available are explained below.

SWI OS_SpriteOp

This SWI can be used to perform all sprite operations. The action taken depends on the value passed in register zero (R0) which is known as the "reason code".

If the reason code is in the range 0 - 255 (&00 to &FF) then the SWI acts on sprites in the system sprite area – very often duplicating the action of the * commands mentioned above.

The range of reason codes we are interested in are those in the range 256 to 511 (&100 to &1FF) which act on the user sprite area. Hexadecimal is easier to remember because the reason codes are the same as for the system sprite area but with a one tagged onto the beginning.

An area of space can be claimed and initialised for use as a user sprite area by calling SWI OS_SpriteOp with the reason code 257 (&101) and the following register values:

CLAIM SPACE FOR SPRITES

R1 = 0

R2 = number of bytes of space required

This will return a pointer to the initialised area in R1.



Offset	Content
0	Size of area
4	Number of sprites in area
8	Position of first sprite
12	Position of first free word
16	Extension words

Figure 1
Format of the User Sprite Area

Doing More With Text

In the last issue of *Drag 'N Drop* we looked at RiscLua scripts for manipulating text files in StrongEd. You can obtain the latest version of RiscLua from Plingstore or from <http://lua.riscos.org.uk/>. Download StrongEd from <http://stronged.iconbar.com>.

Because RiscLua scripts have no line numbers, the examples in this article show you when to start a new line with `▶` (but do not try to type the `▶` character).

Sorting

Suppose we have a file that lists, say, names of persons and their ages, as in figure 1. The names form a column to the left and the ages a column to the right.

We could sort the lines in various ways: alphabetically by name, numerically by decreasing age, etc.

Walter Quatsch	37
The MacGregor of the MacGregors	81
Thierry de Villehardouin	20
...	..

Figure 1

Lua provides the function `table.sort`, which takes as its first argument a table

and as optional second argument a Boolean-valued comparison function of two values to determine the ordering. To sort the lines alphabetically this script will do:

```
▶#! lua
▶- Ex4_1
▶t = { }
▶for line in io.lines (arg[1])
do
▶ t[1 + #t] = line
▶end -- for
▶table.sort (t)
▶for _,line in ipairs (t) do
print (line) end -- for
```

The function `ipairs` generates in order the key/value pairs of the listlike part of its argument. We could equally well have written

```
▶for i = 1, #t do print (t[i])
end -- for
```

In place of the last line.

To sort the file by decreasing age we would use the following script. We change to a slightly higher gear here by using local variables wherever possible, and by using the dot syntax whereby `t.age` stands for `t["age"]` and in table constructors the

expression `["age"]` can be simplified to `age`. These considerations apply whenever a key is a string that can be used as a variable name.

```
▶#! lua
▶- Ex4_2
▶local t, print = { }, print
▶local pat = "^%s*(%D+)%s*(%d+)%s*$"
▶for line in io.lines (arg[1])
do
▶ local name, age = line:match
(pat)
▶ if name and age then
▶ t[1 + #t] = {
▶ line = line,
▶ name = name,
▶ age = age,
▶ }
▶ end -- if
▶end -- for
▶table.sort (t, \ (x,y) =>
x.age > y.age end)
▶for _,v in ipairs (t) do
print (v.line) end -- for
```

Note that the `print` on the right hand side of the multiassignment is a built-in global variable whereas that on the left is a local variable. This is a commonly found trope. The point is that local variables are stored as stack offsets and are quick to find, whereas global variables have to be looked up in the global-variable table. I suppose that when there are millions of

Edit Search & Replace

Edit comes free with every version of RISC OS and is a capable text editor offering more than the text editors on other platforms e.g. Notepad on Windows.

Other advanced text editors like StrongED exist for RISC OS, we'll look at those in a future edition of *Drag 'N Drop*.

In Edit, pressing F4 to calls up the search dialogue box. Alternatively click Menu > Edit > Find.

option if you want to restrict the search to the same case as which you typed the text to be searched.

Some people type two spaces after a full stop, as taught on traditional typewriting courses.

These days it's more usual to leave just one space. You search for the occurrence of two spaces and replace them with one space.

Tap F4, press space bar twice in the Find box, press the down arrow key to move to the Replace box, press space bar once, then select click on Go.

The Found Text dialogue box appears with seven options, click Replace to replace the first occurrence of the search string. Clicking End of File will replace all occurrences in the file. This is useful if you are sure that you want to do a blanket replace without checking each and every occurrence.

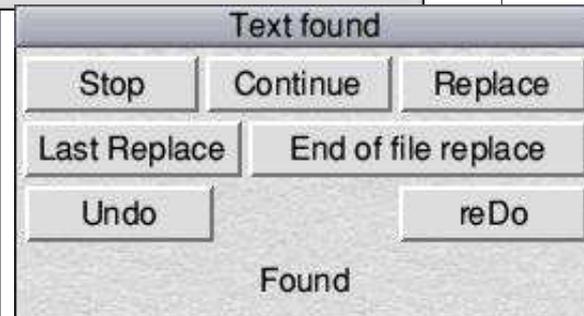
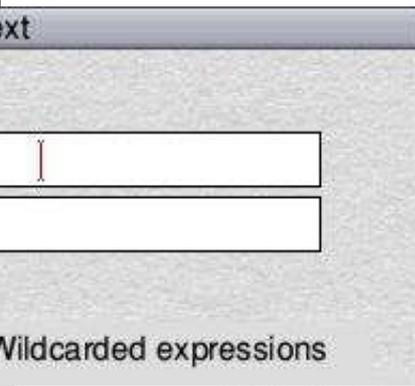
'Special' characters can be entered with the Alt and numeric keypad. For example, say you had some text where someone had used the capital X character for multiplication and you wanted to replace it with the multiplication symbol which on RISC OS is number 215 in the standard character set.

Press F4 again. Type X in the search box, press the down arrow, hold down the Alt key and then 2, 1, and 5 on the numeric keypad (right hand side of your keyboard).

The letter X could also form part of a word and we don't want to replace it with the multiplication character in that case. In this case check each occurrence by clicking Replace or Continue in the Found text dialogue box -- don't use the "End of File" option.

In this article we will summarise the basic search function before concentrating on to the mysterious "Magic Characters" and "Wildcarded Expressions" options. They are in fact no mysteries when you know how to use them.

In the basic search mode, simply type in what you want to find in the Find/Replace box. Click the Case sensitive





RISC OS Programming

BASIC Programming Tutorial 4 - Detecting Incoming Files

In the last instalment, we used WinEd to create the Info window, got its Close icon to work, and got the program to quit from its icon bar menu. This was way back in the Summer 2012 issue of *Drag 'N Drop*, so in case your memory is hazy the panel gives a quick summary of how the program looks so far.

In this instalment, we are going to get some more practice at using WinEd, to create the main program window, and get the program to display this screen when a file is dragged onto its icon.

WinEd can be obtained from

www.riscos.info/index.php/WinEd.

Create the Main Program Screen as follows:

1) Open the !PDFText application (hold down Shift and double click on it), then drag the Templates file onto !WinEd, to open it.

2) Menu-click inside the Templates window, and do Create > Save, to create the main screen called Save.

3) Double click on the Save icon, to open a new screen area, currently labelled Untitled, as shown in Figure 1.

Turn to page 34

If you have followed the series so far you should have an application directory called !PDFText as created in the previous 3 instalments in the Winter to Summer 2012 issues of *Drag 'N Drop*.

Your application directory should contain the following files (the arrow symbol indicates where you start a new line but don't type in the arrow symbol):

!PDFText.!Boot

Obey file (type &feb)

```
▶Iconsprites <obey$dir>
. !Sprites
```

!PDFText.!Run

Obey file (type &feb)

```
▶Wimpslot -min 2000K -max 2000K
```

```
▶Iconsprites <obey$dir>
. !Sprites
```

```
▶Set PDFText$Dir <Obey$Dir>
```

```
▶Run <Obey$Dir>.Stage3
```

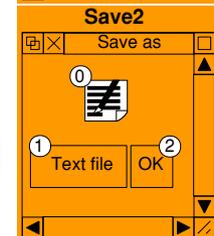
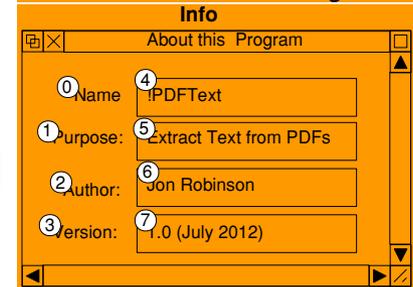
!PDFText.!Sprites22

Sprite file (&ff9) containing a 26x30 sprite called !pdftext in 16M colours, a 17x17 pixel sprite sm!pdftext in 16M colours. You can design your own from scratch or cheat by copying file_adf and file_fff from a themes file (e.g. !Boot.Resources.!ThemeDefs.Themes.Iyonix.Sprites) to the editing window, choosing Use sprite as brush, Name=file_adf or

file_fff. sm!pdftext is a scaled down version.

!PDFText.Templates

Templates file (&fec) containing two windows as shown in the diagram. Numbers in circles refer to the icon numbers as shown with the Monitor function of WinEd. A comprehensive tutorial on using WinEd was given in the Winter 2014 edition of *Drag 'N Drop*.



!PDFText.StageX

Basic program, the application itself, where X=number in the series (e.g. Stage3 for the Summer 2012 instalment).

Sky Pi HDD

As a Sky viewer (but not a subscriber) I wondered if I could make use of my Sky Plus box's Hard Disc Drive (HDD) with RISC OS. It turned out I can and it now serves as a useful backup store in case of SD card failure.

You need a USB-to-IDE converter as shown in Photo 1. These can be purchased quite cheaply (mine cost about £3.00 online). In this article I describe how to modify your box.



Photo 1,

Please note that if you are a Sky subscriber this is not suitable and you will invalidate your warranty if you tamper with your equipment!

Sky appears to only allow recording of programmes if you subscribe; if you just watch the free channels then the HDD is redundant and may as well be put to good use.

Additionally if, like me, your television and satellite connection is near your

computer workstation you can continue using your box as normal to watch the free channels whilst it's connected to your Pi.

Second-hand Sky Plus boxes can be bought inexpensively – I purchased mine off a car boot sale for the princely sum of ten pounds.

The HDD is 80GB which is small by today's standards but plenty big enough for my needs. The model described here is a Thomson but there are various manufacturers of Sky boxes and the principle should work the same because as far as I am aware all Sky Plus boxes use a standard 3.5" IDE hard disk.



Photo 2.

Disconnect the Sky box from the mains and remove the three screws from the back of the box and slide off the top cover (Photo 2).

Now remove the ribbon cable from the back of the HDD. Fold the cable away somewhere e.g. under the HDD caddy. In its place, connect the IDE end of the USB-to-IDE cable (Photo 3) ensuring it's the right way up – the square 'hump' in the centre of the plug is upper most and the USB cable will go toward the back of the box.



Photo 3.

Unscrew the panel on the rear of the box. This leaves a convenient hole through which to feed the USB end of

LCD Backpack

The Hobbytronics Backpack (Photo 1) can be coupled to the Raspberry Pi via the AB Electronics Real Time Clock (RTC) module (photo 2) which has a five volt IIC connection on board.

The IIC (Inter-Integrated Circuit) also known as the I²C is in your Pi's hardware and is used to connect a variety of low speed peripherals.



Photo 1: the backpack (Hobbytronics)



Photo 2: the RTC (AB Electronics)

AB Electronics can be reached at www.abelectronics.co.uk (product code RTC Pi costing £13.00) and Hobbytronics at www.hobbytronics.co.uk

(LCD Backpack V2 - I2C and Serial RX costing £6.00)

The LCD is a standard Hitachi HD44780 compatible display and they can be bought for a few pounds through numerous websites (photo 3).



Photo 3 - LCD Module

Connect the three components to your Pi with the power off (don't be tempted to connect the LCD directly to the Raspberry Pi GPIO pins.) The RTC is plug 'n play on the Pi - just set the time and you're away. I should mention that I have no connection with either AB Electronics or Hobbytronics other than being a satisfied customer!

The accompanying listing assembles and saves a relocatable module called **HtBackPack** which provides a series of extra SWIs for RISC OS (or SYS commands if using Basic) to drive the LCD via the backpack.

Double click HtBackPack to install and it becomes part of the operating system. Typing

```
*help HtBackpack
```

in a task window brings up:

```
==> Help on keyword HtBackpack
Module is: HtBackpack 1.00
(16 Jun 2014)
Provides SWI's for the
Hobbytronics
LCD IIC/Serial backpack
for Hitachi
HD44780 compatible LCD
displays
```

The module makes use of another, built-in module in RISC OS called IIC. Type

```
*Modules
```

and you should see it listed in among a long raft of other modules installed on your machine.

I started with a shell of a module that I'd had worked on previously and added my SWI names in the table and the corresponding branch instructions in the SWI branch table. Below I describe the assembly code in detail

The header (lines 230 to 400) is the standard module header for every relocatable module on RISC OS.

Lines 50 to 120 set up variables and can be easily altered if needed. Line 140

is the Basic error routine. Lines 160 to 220 set up the Basic assembler to compile our code. Line 160 reserves some memory to store the code while it is being compiled. Line 170 to 210 set the assembler for two pass offset assembly.

It's called *two pass* because in the *first* pass the assembler may come across a forward reference. It doesn't know what memory address this reference is at so it puts in a dummy value to be filled in on the *second* pass.

And the *offset* comes from the fact we are compiling code stored at one location to be run at another location.

The statement `O% = code%` sets the location the code is stored to at assembly time, and `P% = 0` sets the location the code is to run at, which is zero for modules.

Line 200 enters the assembler and line 210 sets the `OPT` to 4 which suppresses screen output on first pass and 7 on the second pass which does give screen output.

The `ALIGN` in line 220 makes sure we assemble the code to word boundaries. Each ARM instruction takes four bytes or one word of memory to store it and must be *word aligned* which means it has to begin on a four byte boundary.

`ALIGN` pads the code with extra bytes if needed. You will get very strange bugs if your code is not word aligned!

There are thirteen words of memory that define the module header, lines 250

to 370. Every module that runs on the Pi has a module header like this and is how RISC OS knows which module to call for every SWI or star command.

The first word (word zero) is the offset into the module where the code would be if this module was to be run as an application. Here it's zero to indicate it isn't used. Twelve of the thirteen words are offsets like this and may be set to zero if not used.

The next two words point to the INITIALisation code and FINAlisation code. `EQU`D reserves four bytes of memory and sets them to the contents of the variable or immediate value following the `EQU`D statement. Line 260 for example sets the address of `INIT` but on the first pass through the assembler `INIT` has not been defined yet which is why we need two passes. On the second pass `init` will contain the address of our initialisation code and all is well.

The semicolon character `;` is the equivalent of the `REM` statement in Basic. Anything after are comments is ignored, unless you type a colon character which is a command separator the same as in Basic.

When the assembler finds a full stop it creates a variable with the name that follows the full stop and sets it to the value of `P%`. These are *labels* and are used to mark places in our code we may need to refer to again.

`P%` is the program counter of our code and points to the next free byte in the code as it is being assembled.

Word three of the module header is

the offset for a service call. We have no service call so it is set to zero in line 280.

Line 290 and 300 point to the title string and the help string. The title string is printed to the screen when the `*MODULES` command is executed and the help string when `*HELP` (title string) is executed as we saw earlier.

Incidentally the use of capitals or small case letters makes no difference in Star commands but must be correct for SWIs.

Word six (line 310) is zero again because we have no star commands. Line 320 is the SWI Chunk number for this module and is registered with RISC OS Open Limited. This is the only word of the module header code that is not an offset and is how RISC OS knows which module contains which SWI's.

The next line points to our SWI handler code, discussed below.

Then comes the offset to the subroutine which decodes the SWI . Here it's zero and the pointer to the Message Translator file name, also zero as we aren't using them.

The last word is the module flags pointer and is how RISC OS knows this is a 32 bit compatible module. It points to `FLAGS`, a word that has the bottom bit set to indicate 32 bit code.

That was the module header, now onto the code.

`TITLE` at line 420 is the module's title. `EQU`S stores the string contained in the quotes in the following memory locations and adds the length of the string to `P%`. `EQU`B stores one byte and increases

Working with Unicode 2

In part 1 (in the Winter 2014 edition of *Drag 'N Drop*) we learnt how to extract Unicode text from web pages and PDF documents, and save it using Edit.

In this part, we are going to learn how to enter Unicode text into our own web pages, or word processor files.

Web Pages

If you only want to use a small number of Unicode characters, you can find the characters online, then look up their code points at rishida.net/tools/conversion/ for example. (Use Firefox for this webpage because Netsurf can't yet handle it. Ed.)

These code points can be specified either in hex, or decimal. For instance, the code point `¥` or `¥` specifies the symbol for the Japanese Yen, or the Chinese Yuan. (You can also get it with `¥` but you can't do that with the vast majority of Asian, or 'CJK' characters).

The two code points `玉` & `兔` produce the characters for the name of the robotic rover, Jade Rabbit. And `嫦` & `娥` produces *Chang'e* – the name of the Chinese landing craft, that recently delivered it to the Moon's surface.

嫦娥

number - therefore you should instead click on that number as well as the other escapes in this

To continue with this example, click on Hex

You should now see the conversion results. You the results. For more information see the notes in

Mixed input [1] Convert Convert numbers
The decomposition mapping is <?, ?, U+1
<p>The title says W3C ,???????? ?????.
abc‫????????

Characters [1] Convert View names [1]
The decomposition mapping is <?, ?,
<p>The title says W3C ,???????? ?????.
?a‫????????

HTML/XML [1] Convert Escape invisible
The decomposition mapping is <?,‫
<p>The title says ‫?????.
?a‫????????

Percent encoding for URIs [1] Convert
The%20decomposition%20mapping%20is%20

Hexadecimal NCRs [1] Convert Show as
The decomposition mapping is <ೢ
and not <ᄎ, ᅳ, ᆸ>
<p>The title says

Unicode U+hex notation [1] Convert Show
The decomposition mapping is <U+CE20
not <U+110E, U+1173, U+11B8>.
<p>The title says U+202B U+05E4 U+05E
U+05D5 U+05EA U+05D4 U+05D1 U+05D9 U+

Hexadecimal code points [1] Convert Shc
0054 0068 0065 0020 0064 0065 0063 006
006F 0073 0069 0074 0069 006F 006E 002
0070 0070 0069 006E 0067 0020 0069 007
0020 0020 0020 11B8 0020 0020 0020 0020

Part of a conversion at rishida.net

As explained in the previous article, you can also incorporate blocks of Unicode into your own text files and web pages, by 'lifting' them out of other people's HTML files, using Edit, or the latest version of Zap (1.47) – currently the only text editors to support Unicode, even partially, on RISC OS.

Remember, if you want to use Unicode in you web pages, you need to include the tag:

```
<META CHARSET="UTF-8">
```

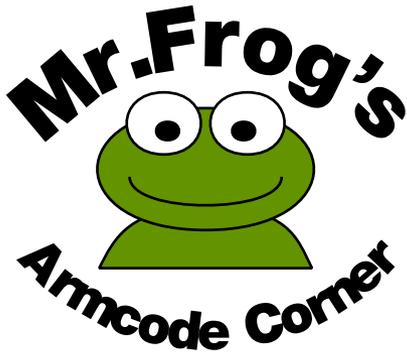
between `<HEAD>` and `</HEAD>`, at the top your web pages. You also need to use NetSurf on a Unicode-enabled system to view them.

DTP documents

Currently, none of the RISC OS word processors – EasiWriter, Style or Ovation – will even display Unicode text properly, let alone allow you to enter it.

But if you want to use a more adventurous layout than can be achieved in Edit, there are a couple of kludgy ways of getting small amounts of Unicode content, into Style and EasiWriter.

The first is to simply snapshot part of the Edit, or NetSurf window, using Paint, and drop the resulting sprite file into an empty frame in Style.



Last time in Armcode Corner we played with a pointy clicky program which showed how Armcode instructions are made up. Mr F was buying a car from a nearby pond which is why he was absent in the last issue of *Drag 'N Drop*.

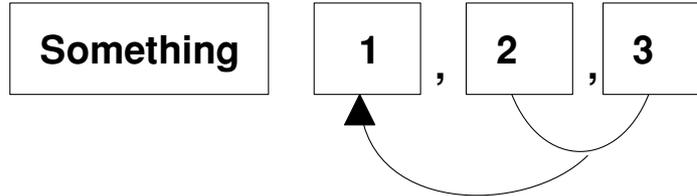
Anyway, I said that both the instruction and the data are packed into 32 bits. This is unlike machine code of yesteryear (where you had one byte for an instruction and one or two bytes of data). Armcode all fits neatly into little packages of 32 bits (four bytes or one word).

Very tidy but that means not all of the 4.3 billion odd numbers can be assembled. Because some of the bits are used to represent the instruction. Once actually *inside* the computer's memory it's not a problem, just the assembling of them that is. In fact there are only 12 bits for numbers and you have to resort to some jiggery pokery which we'll leave for another time.

If you were paying attention you will have seen that the four right most bits

(squares on the display) signify which of the ARM's 16 regs the instruction operates on.

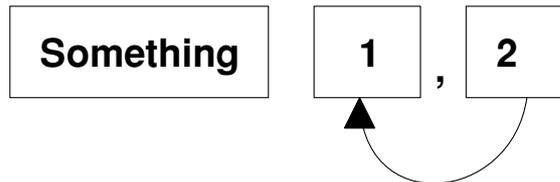
Do SOMETHING with 2 and 3 and put the result in 1



They're called R0, R1 through to R15, the R standing for Reg or Grandpa Reg as I like to fondly remember him. The way the *MEMORY! command disassembles things, it calls reg 15 PC because it's where RISC OS keeps a record of the memory location it's at, the Program Counter or PC.

Now, if you kept clicking merrily away you will have seen that all ARM instructions begin with some a bizarre mess of letters on left – computer boffins call them mnemonics – and several R numbers (the regs) separated by commas.

Do SOMETHING with 2 and put the result in 1



You already saw one when you ran the pointy clicky program: ANDEQ

R0,R0,R0. The three letters AND are followed by condition codes. For now just remember the three letters and three items separated by commas and ignore condition codes.

Exceedingly good programs can be written without them.

Just remember "do something with the second and third regs on the list, the

'something' being what the three letter instruction is, and put the result in the first reg". I'll give you an example. ADD R0,R1,R0 means "add together reg 1 and reg 0 and put the result in reg 0".

There is a simpler version of the "do something with 2 and 3" which is just "do something with 2"

If you wanted to put a number into a reg the something is "move" and 1 is reg 1 and 2 is a number.

An example is MOV R0,#3 which means MOVE the number three into reg zero.

The # symbol is called a hash, and for #3 we say "hash three". Mrs Frog says it's because us programmers are always making a hash of things but Mr F always has three hash browns for breakfast.

Speaking of which, Mr F's stomach is rumbling so is going along to www.whatdofrogseat.info to find something that will fit in his mouth. See you next time.

Drag 'N Drop Volumes 1-5 Index

This index is organised with a reference (e.g. application name) or article/listing/series title followed by the volume – issue number – page.

21st Century Computing (article) 4i2-5
24-bit colour 5i2-47
256 colours 4i2-10, 4i3-9, 5i2-47

55 BBC Micro Books 4i3-21, 4i4-17
76489 chip 1i2-22
7-segment LCD 5i3-6
8-bit roundup – New Releases from Retro Software (article) 3i3-19
8-bit roundup (article) 1i1-29
8-bit roundup (article) 1i2-22
8-bit roundup (article) 2i1-19
8-bit Roundup: Retro Software Roundup 3i4-15
8-bit: BBC ARM7TDI Upgrade 1i1-29
8-bit: BBC GoMMC (article) 2i1-23
8-bit: BBC RAM Disc (article) 2i1-19
8-bit: BBC Repairing your power supply 1i1-30
8-bit: BBC Retro Software Update (article) 2i1-25
8-bit: Game On (article) 2i4-17
8-bit: Game On 2 (article) 3i1-9
8-bit: Look Who?s Calling 2i2-19
8-bit: Repton – The Lost Realms 2i3-19
8-bit: Speak your mind (article) 1i2-22

A Trip Round the Chocolate Factory (article) 1i3-20
Acorn Electron World (article) 4i2-27
Acorn speech system 1i2-22
Acorn World in Pictures 1i1-20
Adding document information to PDF files 3i4-6
All Sorted 1 (article) 1i1-9
All Sorted 2 (article) 1i1-22
All Sorted 3 (article) 1i2-7
All Sorted 4 (article) 1i3-10
All Sorted 5 (article) 1i4-11
All Sorted 6 – the Merge Sort (article) 2i1-4
All Sorted 7 – Heaps of Data (article) 2i2-7
All Sorted 8 – Quicker Sorting (article) 2i3-12
All Sorted 9 – Strands & the Merge sort revisited (article) 2i4-6
All Sorted 10 – Hacks & Tricks (article) 3i2-10
All Sorted 11 – Hacks & Tricks (article) 3i3-9
All Sorted 12 – Mashing it Together (article) 3i4-10
All Sorted 13 – Timsort 4i4-33
AltTab 0.42 1i1-3
Anatomy of a Drawfile (article) 4i4-29, 5i2-36, 5i3-27
APDL 1i3-2
ArcSimp 1i1-3
ARM7TDMI Coprocessor 1i1-29
ARMini computer 2i3-2
ARMini computer 3i3-3

Armcode bit pattern demo (listing) 5i2-32
ArmSort 4.08 3i4-4
Artworks Viewer 2.14b 1i4-3
awk 2i3-25

Backdrops (Desktop Heroes article) 1i3-18
BarFree 3i3-5
Basalt 1.50 3i4-4
BBC Micro Games Retrospective (articles) 4i3-26, 4i4-11, 5i2-61
BBC Playground 4i4-10
BBFlash 0.03 3i4-4
Beagleboard 1i1-15
Beagleboard-XM 1i4-2
Beeblt 0.61 2i4-4
Beginner's Tips 5i1-2, 5i2-2
BgrndCtrl 1i2-5
Birmingham 2009 1i2-20
Bitcoin (article) 5i3-44
Bookmaker 2.15 3i2-3
Bookmarking with PrintPDF (article) 1i4-35
bubble sort 1i1-9

Calendars with Yahoo & Organizer (article) 2i1-32
Calibra 3.23 3i2-3
Change your Perspective (article) 1i1-35
Christmas Show in Pictures 1i2-20
Cineworks 4i4-46
Cogs 1.22 1i3-4
Col256 (listing) 5i2-47
Colour Coordinated (article) 5i2-47
Drag'N Drop Summer 2014 | www.dragdrop.co.uk | Page 59

Combining Documents with PrintPDF (article) 2i1-36
 Contact Sharing with Organizer 1i4-31
 contact us page (Web Log) 1i1-4
 Converting the PMS Manual (article) 4i4-49

 DayCalc 1.06 2i1-3
 DDE 22 3i1-5
 Demos 1i1-16
 Drawrite (listing) 5i4-10

 Desktop Heroes 1i1-18, 1i2-16, 1i3-18, 5i1-6
 Desktop makeover with Themes (article) 5i1-33
 DirSync 1.10 3i1-5
 DiscKnight 2i3-4
 DIY Draw (article) 4i4-26
 D-Link DNS-323 (article) 1i3-29
 Doing more with text (series) 5i3-40, 5i4-26
 Doktor – the no nonsense toolbar (article) 3i2-6
 Draggit (app) 5i2-25
 Drawing with Draw Part 1 (article) 1i1-38
 DrawPrint 1i3-4, 2i2-3, 2i4-4, 3i4-4, 5i2-3, 5i4-5
 DrawWorks Millennium 2i3-3
 Drobe 1i1-4

 Easiwriter 9.12 5i1-04
 Eastor (listing) 4i3-4
 Eastor 8-bit (listing) 4i4-5
 Edit Search/Replace (article) 5i4-30

 Fade – presentation software (article) 3i2-16
 Fireworks 1.35 5i2-3
 FAT32FS 1.39 3i2-3
 FATFS32 1.27 1i2-5
 FFmpeg 1.04 1i2-5
 FFmpeg 1.10 2i1-3
 FFmpeg 1.11 2i2-3
 FFmpeg 1.21 3i3-5
 Firefox 2.0.0.21 1i1-3
 Fireworkz Manuals 3i4-3
 Flicker (article) 3i1-6
 FormFiller 1.14 3i2-3
 Form Filler (Desktop Heroes article) 5i1-6
 Frontender 5i1-28
 Fun Time (article) 2i2-22
 Fuse (Free Unix Spectrum Emulator) 3i3-5

 GCC 4.1.2 (test) 1i4-3
 Games Panel Designer (listing) 5i4-16
 Gem Drop 2i3-23
 Genealogy Records 3i3-3
 Get Archiving with ZipFE (article) 5i1-28
 Get Set For Digital – Updating your RiscTV card for the digital age (article) 2i4-29
 Getting started with RISC OS on the Raspberry Pi (review) 4i2-16
 Getting started with RISC OS: a refresher (Beginner's tips) 5i4-4
 Gnome Sort All Sorted 3 (article) 1i2-7
 Gofer Sort (article) 2i4-15
 GPIO for Beginners (series) 5i1-26, 5i2-4, 5i3-6
 GPIO pinout 5i2-4
 Graded text (listing) 5i2-38

 Hexen II: Hammer of Thyrión 3i4-3
 HongKong 160 3i4-4
 Hopper 5i4-6
 House of Cards 1.06 2i2-3
 Huddersfield 2009 1i1-20

 Iconv 0.11 2i2-3
 IMAP (article) 3i1-16
 Impact 3.35 1i2-5
 InfoZip 3.07 3i4-4
 Inside Drag 'N Drop (article) 5i3-37
 InterGIF 3i1-5
 Internet Message Access Protocol 3i1-16
 Introduction to Unicode (article) 5i1-20

 Joining PDF files 3i4-5
 Jpegtran 3i2-3

 KevMon 1.01 3i2-3

 LEDs (GPIO) 5i1-26, 5i2-4
 Listing Programs (Beginners tips) 5i2-2
 London Show 2009 1i1-5
 Lua 2i3-25
 Luafox 1.21 2i3-4
 LuaFox 1.30 3i1-5

 Mae2PMS (listing) 5i2-43
 Maestro to PMS converter (article) 5i2-41
 Magic Mushrooms 1.12 3i4-4

MakeDraw 2.10 1i1-3
 MakeDraw 2.30 1i2-5
 Male-to-female jumper lead 5i3-6
 Manic Mole (listing) 4i1-16
 Mashing it Together – All Sorted 12
 (article) 3i4-10
 Match Points (article) 2i2-23
 Meteors 5i4-5
 MBBBack 2.01 2i1-3
 MBBBack release 8 1i2-5
 MBOX Alert 2i3-3
 MTP 0.21 5i1-4
 Messenger Pro 6.03 3i1-5
 Messenger Pro 6.05 3i3-3
 Messenger Pro 6.06 3i4-3
 Microsoft Office File Block (article)
 3i3-36
 MoreDesk 2i3-4
 MoreDesk (Desktop Heroes article)
 1i2-16
 Morse Code (program) 5i2-6
 Mountain Panic 4i4-10
 Mountain Panic (review) 5i3-50
 Mr Frog's Armcode Corner 5i2-32,
 5i4-58
 MsgMon 1.01 1i4-3
 Mug Show (advert) 5i4-1
 Murnong 1.31 1i1-3
 Murnong 1.50 2i1-3
 Murnong 2.10 2i2-3
 Music Man (article) 3i4-27
 Music Ripper's Toolkit 3 1i3-4
 Music Writing with PMS (series) 4i2-
 18, 4i3-11, 4i4-18, 5i1-35
 Mycalc (app) 5i2-26

 Netfetch 3.65 3i3-3
 Netsurf 2.6 2i1-3

 Netsurf 2.8 3i1-5
 Netsurf 3.0 4i4-4
 New Breed Software 2i3-23
 Newshound 1.52 2i2-3
 NewsUK 5i1-4

 Odd-Even sort 1i3-10
 Openvector, Opengrid, Drawplus
 3.30 1i4-3
 Openvector, Opengrid, Drawplus
 3.32 2i2-3
 Openvector, Opengrid, Drawplus
 3.37 3i2-3

 PANL (listing) 5i4-16
 Packman 0.3 2i1-3
 Packman Packge Manager 0.05 2i4-
 4
 Pal256 (listing) 5i2-48
 Parallel Lines Pattern 3i2-18
 Passman 1.00 5i2-03
 PasteKeys 0.6 1i1-3
 PDF 3.00.1.20 1i4-3
 PDF 3.00.1.21 2i2-3
 PDF 3.02.1.23 2i3-4
 PDF Maker (article) 1i2-33
 PDF Suite (article) 1i2-33
 PDFTools 3i3-5
 Perl 2i3-24
 Photodesk 3.10 2i3-4
 PhotoDesk 3.10 (32 bit) 3i1-5
 PICprog 1.54 1i1-3
 Piball (game) 5i1-8
 Piball (listing) 5i1-13
 Pico (review) 5i4-9
 Pieces of String (article) 3i4-18
 Pipedream 5i4-6
 PlayIt 3i3-5, 3i4-4

 Plingstore 5i1-4
 Pluto 3.10 5i2-3
 Potted History of Acorn 5i4-5
 Powerbase 9.70 1i2-5
 PrintPDF 0.80 1i4-3
 PrintPDF Queue Process in Pictures
 2i1-37
 ProCAD+ 1.08 5i4-5
 Pursuit Curves 3i3-14
 Python 2i3-25

 Quizmaster 5i4-5

 Raspberry Pi Assembly Language for
 Beginners (review) 4i2-17
 Raspberry Pi RISC OS System
 Revealed (review) 5i2-60
 Raspberry Pi User guide (review)
 4i2-16
 R-Comp Tablet 3i3-3
 RDPClient 1i3-4
 Remembering filetypes (Beginner's
 tips) 5i3-2
 Reporter 1.66 2i4-4
 Retro Software 1i1-33
 Retro Software Revealed (article)
 1i1-33
 RISC OS 5.2 Rom chips 5i1-3
 RISC OS 6 Pinboard – A quick tour
 (article) 1i2-27
 RISC OS 6.2 (Select 6i1) What's New
 (article) 1i2-35
 RISC OS Classic ROMs Collection
 CD 2i3-3
 RISC OS Font System (article) 5i3-
 35
 RISC OS In Education – Does it have
 A Future? (article) 2i3-31

RISC OS London Show 2011 guide 3i1-27
RISC OS Programming in BASIC (series) 3i2-20, 3i3-39, 3i4-23, 5i4-34
ROOL Roundup & Wakefield 2011 Guide 2i3-11
Rotated Areas in Draw (article) 3i2-26
RiscOSM (review) 5i4-7
Rotated graded text (listing) 5i2-40

SPI Bus (article) 5i3-10
Safestore 3i3-3
Safestore (Desktop Heroes article) 1i1-18
ScreenGrabber 2.21 1i1-3
Scripting with and without StrongEd (article) 2i3-24
Scriolling stars (listing) 5i3-34
Scrolling starfield (article) 5i3-34
Selection and shell sorts (article) 1i4-11
Selling on eBay with Netsurf 4i2-14
Simple Video Editing using Cineworks (article) 4i4-46
Single tasking (Beginner's tips) 5i1-2
SiteMatch 240f 3i4-4
Sky Pi HDD (article) 5i4-39
SmartMenu 0.21a 1i1-3
Snapper 2i1-3
Snapper 1.14 3i3-5
Snapper 1.70 3i4-4
Snes9x 3i3-5
SokoBas 2i4-4
Sound (article) 4i1-23
Source Code Management Tool 1i4-24

Sourcery 1 (article) 1i3-24
Sourcery Source Code Management 2 (article) 1i4-24
Sourcery Source Code Management 3 (article) 2i1-26
South West Show 2010 in pictures 1i3-19
Sprites (series) 3i3-48, 3i4-30, 4i1-6, 4i2-22, 4i3-16, 4i4-39
Sprite Pool 5i2-28
stable sorts 1i1-26
Strands & The Merge Sort Revisited 2i4-6
Stick and Drag (article) 5i2-23
Sticky programming 5i2-23
Stripping email headers with Lua (article) 2i3-27
Super Pack 6 4i4-4
SuperDoku 1.23 1i2-5
SyncDiscs 1.22 2i1-3
SystemDisc 1.01 5i2-3

TaskW 5i2-23
TBA Software 2i4-13
TBX C++ Library 0.4 3i1-5
Techwriter 9.12 5i1-4
Teetfort 2i3-23
The Archives (article) 1i1-4
The Beagle Has Landed (article) 1i1-15
The Big Book of the Raspberry Pi (review) 5i2-60
The Difference Words can Make (article) 3i3-45
The Merge Sort (article) 2i1-8
The Rise of Retro Software (article) 1i2-26
Themes 5i1-33

ThemeSetup 5i1-33
Ticker 0.87 3i2-3
Timsort 4i4-33
Traffic Signs (Weblog article) 5i1-7
Type from PDF (article) 4i1-6

Unicode 5i1-20, 5i4-56
UnitConv 2.32 3i2-3
Upgrading to Messenger Pro 6 (article) 2i4-23
URLtoIE 1.40 3i4-4
User Sprite Areas (article) 5i4-19
User Voices II (article) 5i3-46
Using Cineworks (series) 4i4-46, 5i1-23
Using Fireworkz (series) 4i3-21, 4i4-23, 5i1-30
Using IDE Compact Flash Cards with a RiscPC (article) 2i4-27
Using IMAP in Messenger Pro 6 (article) 3i1-18

Varieties of Numerical Experience (article) 4i2-30
Vectoroids 2i3-23
Vintage Computing Festival (article) 1i4-22
Vpod Review – Change your Perspective (article) 1i1-35
VRML Viewer 1.01 1i1-3

Wallace Goes Walking (game) 5i3-16
Wallace Goes Walking (listing) 5i3-21
Waltz Alice 5i1-35
Wakefield Show 2012 guide 3i3-43
Wakefield Show 2010 guide 1i3-9

Wakefield Show 2011 in pictures
2i4-32

Web Change 2.95.21 1i4-3

Web Log – A Free Online Diary
(article) 2i2-3

Web Log – Blogging on RISC OS
2i3-5

Web Log – British Newspaper
Archive 3i3-6

Web Log – BT image history 3i2-8

Web Log – Dart (article) 3i1-8

Web Log – Google Alerts (article)
2i1-4

Web Log – Google mail (article) 1i4-
3

Web Log – iconbar.com (article) 1i2-
6

Web Log – ReachBy.com (article)
1i1-4

Web Log – riscos.info (article) 1i2-6

Web Log – Google Buzz (article)
1i3-5 – Traffic Signs (article) 5i1-
7

WebChange 2.95.26 3i3-5

WebGen 2.52 5i1-5

Wimp events 5i2-29

WinEd tutorial (article) 5i2-51

Winglebith (game) 5i2-10

Winglebith (listing) 5i2-15

Working with Non-Latin Languages
on RISC OS (article) 2i2-28

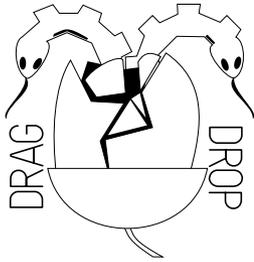
Working with PDFs (article) 3i4-5

Working with Unicode (series) 5i1-
20, 5i4-56

Wrangler 1.07 3i4-4

Writing a simple RISC OS app
(series) 5i2-26, 5i3-25

ZipFE 5i1-28
Zmac 2i2-3



DRAG'NDROP

ORDER FORM

Step 1: Please fill in this form. Print it and write on it or email it (address details opposite).

-
- Drag 'N Drop Vol.3 Iss.1 e £1.00
- Drag 'N Drop Vol.3 Iss.2 e £1.00
- Drag 'N Drop Vol.3 Iss.3 e £1.00
- Drag 'N Drop Vol.3 Iss.4 e £1.00
- Drag 'N Drop Vol.4 Iss.1 e £2.00
- Drag 'N Drop Vol.4 Iss.2 e £2.00
- Drag 'N Drop Vol.4 Iss.3 e £2.00
- Drag 'N Drop Vol.4 Iss.4 e £2.00

- Drag 'N Drop Back issues memory stick (20 issues) (p) £14.00

- 55 BBC Micro Books CD-Rom (p) £13.00

- Subscription to Vol. 6 (e) £14.00

- Drag 'N Drop Programs Archive (e) £5.00

Step 3: Enter your order total below:

ORDER TOTAL £ _____

Step 3: please fill in your address details opposite.

Step 4: either (a) make cheque payable to DRAG N DROP and post with this form to:

Drag N Drop
118 The Avenue
Great Clacton
Essex CO15 4NE

or (b) pay through online banking:

Sort Code: 60-05-33
Account 23377968

Please email this form to
sales @ dragdrop.co.uk
(or post to above address)

NAME:

ADDRESS

EMAIL:

Thanks for your order!

Items marked 'e' will be emailed. Items marked 'p' will be posted (P&P included in price). Please enter your email so we can let you know when items are in the post. If you are just ordering items on email you only need to fill in your email address