

# DRAG 'N' DROP

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

April 2016

Volume 7 Issue 3

£3.50

**RISC OS  
Pi-Top**

**PLAY**  
Flappy Fluffy

**LEARN**  
Python, C  
Ghostscript

**COMPOSE**  
with Milkytracker

**The world's only regular RISC OS mag!**

# Wakefield

Acorn & RISC OS Computer Show

Saturday 16th April

# 2016

Show organised by



## The North's premier RISC OS show

Now in its 21st year

The Cedar Court Hotel  
Denby Dale Road, Calder Grove, Wakefield  
West Yorkshire WF4 3QZ

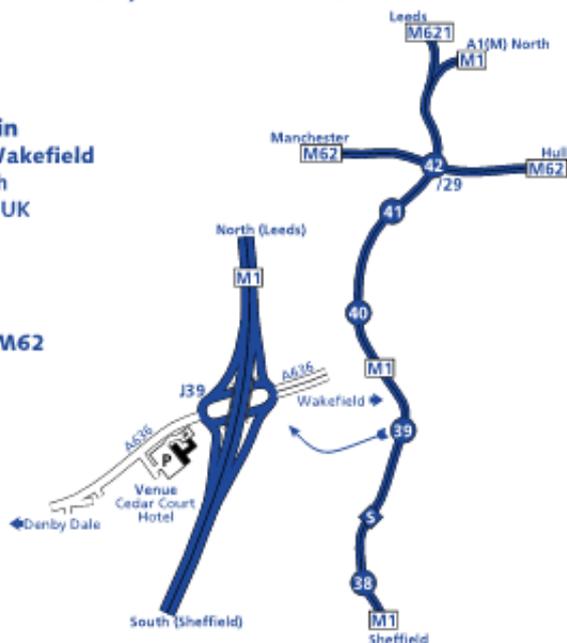
(Adjacent to J39 of the M1)

### 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



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

- *Date*
- *Opening time*
- *Ticket price*

**Saturday, 16th April 2016**

**10.30am to 4.30pm**

**£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*

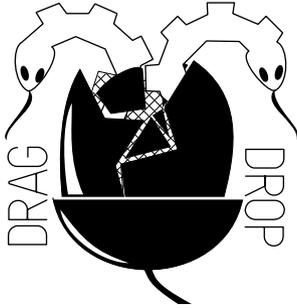
Wakefield Show 2016,  
c/o 3 Riverdale Avenue, Stanley,  
Wakefield, WF3 4LF

- *Email*

show2016@wakefieldshow.org.uk

For further information, visit

[www.wakefieldshow.org.uk](http://www.wakefieldshow.org.uk)



Copyright © Drag 'N Drop 2016  
Produced on RISC OS computers

This issue has been blessed with contributions from the following people:  
Tony Bartram (Flappy Fluffy)  
Paul Dunnington (Python Primary School)  
Gavin Wraith (Blynken)  
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](mailto:editor@dragdrop.co.uk)  
[www.dragdrop.co.uk](http://www.dragdrop.co.uk)

# EDITORIAL

I hope you had a good Easter. Thanks for your continued support of *Drag 'N Drop*. Every penny of the proceeds are put back into the production of the magazine, research, and attendance at shows. Hope to see some of you at the Wakefield exhibition!

There are several new RISC OS machines, of which Pi-Top is one and we take a look in this issue. (Very handy – I don't have to lug so many bits and pieces to computer shows now!)

There are tutorials on using Ghostscript and programming in C and Python. If you are musical you will like the article on composing 'Tracker' tunes. I had a go with Milkytracker, it's great fun.

**Chris.**  
Christopher Dewhurst

## At a glance...

<b>Editorial</b>	<b>2</b>
<b>How do I...?</b>	<b>3</b>
<b>News &amp; Apps</b>	<b>4</b>
<b>Pi-Top RISC OS</b>	<b>5</b>
<b>Using Ghostscript</b>	<b>7</b>
<b>Elesar Keyboard</b>	<b>10</b>
<b>Flappy Fluffy</b>	<b>11</b>
<b>Blynken</b>	<b>14</b>
<b>Python</b>	<b>17</b>
<b>Milkytracker</b>	<b>29</b>
<b>Weblog</b>	<b>33</b>
<b>C-Side</b>	<b>35</b>

# How do I...?

## ...get the BBC Basic prompt?



To get the BBC Basic prompt press F12 and type \*BASIC and press Return. You can change the screen mode with MODE n where n is a number e.g. MODE 7 or MODE 0. Type AUTO for automatic line numbering. Press Escape to stop and type *SAVE "myprog"* followed by Return to store *myprog* on hard disc.

To return to the desktop type \*QUIT. Programs listed in *Drag 'N Drop* are assumed to work on all machines with RISC OS 5 e.g. Raspberry Pi, unless otherwise stated.

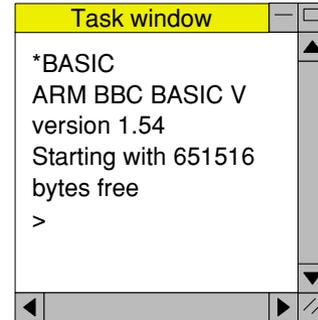
## ...open a Task window?

Menu click over the Raspberry icon on the right side of the iconbar and select click on Task window. Or press Ctrl + F12.



You may need to reserve more memory for the task in which case adjust-click on the Raspberry icon and under Application tasks click and drag the Next slide bar out to the right.

You can also type programs in a task window, hold down Ctrl and press F12. You can't use the cursor editing facility or change MODE, however.



You can also program and run Basic programs from the desktop. Double-clicking on the filer icon runs it, holding down Shift and double clicking loads it into your text editor.



## ...select the currently selected directory?

Articles may tell you to set the CSD (currently selected directory). Just click menu over filer window and choose Set directory ^W or you can use the IEasyCSD application presented in *Drag N Drop* 6i1.

## ...open an Application Directory?

Application directories begin with a ! called 'pling'. Hold down shift and double click select to open the directory.

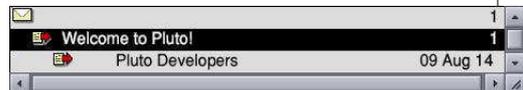
# News and App Updates

## Wakefield Show

The 21st annual show for the north of England takes place at the Cedar Court Hotel, Wakefield, on Saturday 16th April. Admission is £5, more details at [www.wakefieldshow.org.uk/index.php](http://www.wakefieldshow.org.uk/index.php). Drag 'N Drop will be there from 12 noon with the full edition of this issue for only £3 (bring a memory stick or your email address!).

## Book of Draw Stuff

To celebrate the reprinted edition and 1st anniversary of this book, we'll be giving a USB memory stick with digital copy and type-in listings FREE with every copy purchased at the Wakefield show!



## Pluto 3.16

Pluto is a freeware email and news application. The latest version features better HTML-to-

text conversion, larger debatching (up to 256meg) and various bug fixes. Get along to [www.avisoft.f9.co.uk](http://www.avisoft.f9.co.uk).



## Final Doom

Doom is a 3D "first person shooter" type game which first appeared in 1993. The classic has received an update for modern hardware thanks to R-Comp, plus enhanced soundtracks and comes with over 70 levels. Price is £11.99 available from !Store.

## Connector 1.07

Connector is a terminal emulator allowing transfer of files and data between computers, also useful

for gadgets which communicate via plain-text serial data. Connector is in the Bonus Binaries in the downloads section of [www.riscosopen.org.uk](http://www.riscosopen.org.uk). You will also need !SerialDev in the Block Drivers archive at [www.xat.nl/en/riscos/sw/bd/](http://www.xat.nl/en/riscos/sw/bd/).

## ImpEmail 1.12

ImpEmail is a mail-merging email editor compatible with Hermes and POPstar (used in Messenger). You can create a template then merge in data from a database or spreadsheet for a set of personalised emails. A demo version can be downloaded from [sinenomine.co.uk/software/](http://sinenomine.co.uk/software/) and the full version is £10.

## Legends of Magic

The 3D arcade adventure released last October now comes with a second quest (free to existing users). Price just £9.99 from !Store.



# RISC OS Pi-Top

The Pi-Top is mooted as the first true RISC OS laptop since the Acorn A4 way back in the 1990s. We take a look at the Raspberry-Pi-powered computer.

The Pi-Top is available in kit format from CJE Micros [www.cjemicros.co.uk/](http://www.cjemicros.co.uk/). Prices start at £269.00.

The kit comes in a box the size of a shoe box and comprises a base (housing the battery), keyboard, 13-inch screen, hub, 18 volt/3 amp power supply, fixing screws and five internal cables.



You can supply the Raspberry Pi yourself, which must be a Pi 2, or opt for it at extra cost in the kit.

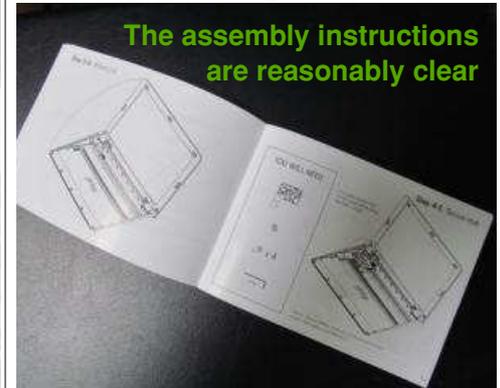


It took me about two hours to assemble following the well-written, 30-page instruction manual. The final computer weighs around 1.5kg and measures 13½ × 8½ × 2 inches; a 17-inch laptop bag on the general market is ample to hold it and several accessories.

The Pi 2 sits inside the Pi-Top

towards the right hand end and the 'hub' on the left with the cables running between the two.

The plugs on the cables tiny and wouldn't stand a great deal of plugging / unplugging so it is important to get them right first time when building the Pi-Top.



There is a hole in the casing to access the Pi2's USB sockets, but it's not easy. One socket is occupied by the keyboard cable and I found the best configuration to be as per the following photo.

The Pi-Top's has a touchpad with two buttons, which works on RISC OS but there's no option (yet) for the menu button.

# Ghostscript on RISC OS

**Have you wanted to easily generate an image from one or more pages of a PDF file? Or extract its text content? Or merge multiple PDFs? Or convert an image to various raster formats?**

You could do this with !PDF using the Save > Page as Sprite option then convert each of the Sprites with ChangeFSI but what happens when you have many different PDFs each of which could have hundreds of pages?

The solution to your problems can be found in using Ghostscript.

## What is Ghostscript?

Ghostscript is written entirely in the C programming language and has been ported to run on a wide variety of operating systems including RISC OS.

If you have purchased the SD card in your Raspberry Pi from RISC OS Open then you will have Ghostscript. It lives inside !Boot.Resources so is automatically installed on bootup.



If not then you will need to follow the instructions at [www.mw-software.com/software/ghostscript/ghostscript.html](http://www.mw-software.com/software/ghostscript/ghostscript.html).

Open a task window by pressing Ctrl+F12 or clicking menu over the Raspberry icon at the right hand end of the icon bar. Type

```
gs --version
```

and press the return key. If Ghostscript is installed the current version will be displayed e.g. 8.54

## How to use Ghostscript

Here are some basic examples of how to use Ghostscript for various tasks.

### Convert PDF Pages to Images

Open a filer window on your computer where you've got a PDF that you want to convert. Note down the full pathname of the folder, displayed in the title bar of the filer window e.g. **SDFS::16GbPi.\$.Docs.example/pdf**. If

you haven't done so already open a task window. If you enter the following command, Ghostscript will convert each PDF page into a PNG image that fits into a pixel dimension of 768x1024:

```
gs -dNOPAUSE -sDEVICE=png16m -sOutputFile=image%i.png -dPDFFitPage -g768x1024 -q SDFS::16GbPi.$.Docs.example/pdf -c quit
```

Be careful to type the command exactly as shown and the command must all be typed on one line followed by the Return key.

Below is an explanation of each of the 'flags' which are the characters after the hyphens.

**-dNOPAUSE**: do not pause and await user input before continuing on to the next page

**-sDEVICE**: set the output file format, using a particular process or 'device'. In this case **png16m** is the device to convert to a PNG with 16-million colours (other formats include **pnggray**,

The text content will be extracted to **example/txt**. Note you do not have to specify the /pdf extension so it is **example** and not **example/pdf**.

If you want to extract text from a certain pages, separate them as described above then apply the pdf2text Obey file on the separated pages:

**pdf2text p22-p36**

assuming the extracted PDF with a sub-range of pages is called **p22-p36/pdf**.

● Based with thanks on information at [www.grenadeco.com/tutorial/introduction-to-ghostscript/](http://www.grenadeco.com/tutorial/introduction-to-ghostscript/) and other internet sources.



## Review

Product: 105 key UK layout mechanical keyboard, cog logo

Supplier: Elesar

Website: [shop.elesar.co.uk](http://shop.elesar.co.uk)

Price: £66.00

Although marketed as just a "105-key mechanical keyboard" the most striking feature of this computer keyboard is the BBC-micro style red function keys.

I am typing this review using the keyboard and it is certainly reminiscent of BBC micro, the keys are solid and responsive.

Sadly some of the key caps are loose because the courier managed to damage the keyboard in transit.

This could be avoided by with



better i.e. padded packaging.

Only the right hand Control key appears to work (maybe that's the courier's fault), not a problem unless you have a hardened habit like me of doing a one-handed Ctrl+Q and W zoom in/out in Draw for instance!

There's a key with the RISC OS cog logo instead of the Windows symbol as found on mass-produced keyboards.

It doesn't seem to do anything, though. Perhaps there is a way of mapping it to some useful function. I thought the Enter key might be labelled Return but it isn't.

£66 isn't cheap considering the higher quality keyboards generally market for around £35,

(That's twice as much as a Raspberry Pi.) But I haven't found anything similar and the keyboard feels solidly made. If you have a high-end RISC OS computer like an ArmX6 or perhaps even one of Elesar's Titanium motherboards then it will make an ideal partner. ■

# Flappy Fluffy

**Flappy Fluffy is a short but fun game in BBC Basic. I wrote it in a couple of hours on my Raspberry Pi, as an education exercise for children to experiment with and calibrate.**

A short introductory screen is displayed then you press the Space Bar to start the game. Guide Fluffy through the gaps in the pipe by clicking the mouse to gain height, not clicking causes him to fall back to the earth. If you collide with the pipe or the ground you lose the game.

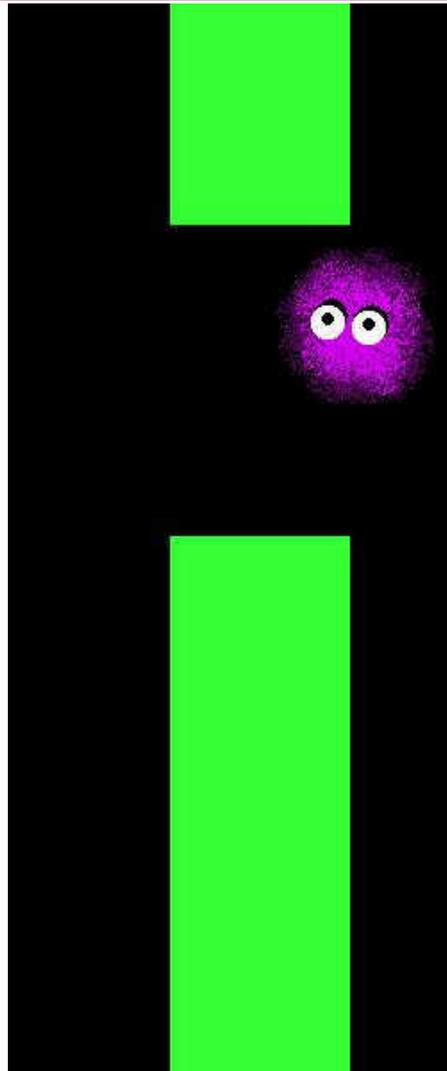
Before typing in the game you need to prepare a couple of files; listing. Load up !Edit and !Paint from the Apps folder.



Create an application directory on your hard disc by clicking menu over a filer window and choosing New directory > type !Fluffy. Hold down Shift and double click !Fluffy.



Click menu over Edit's icon bar icon and choose Create > Obey.

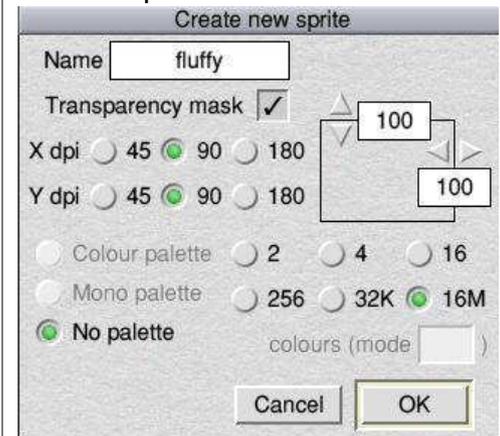


In the Edit window type the following two lines:

```
Set Fluffy$Dir <Obey$Dir>  
Run <Fluffy$Dir>!RunImage
```

Save the file inside !Fluffy as !Run (press F3, type !Run in the save box and drag to the filer window)

Next select click on !Paint iconbar icon to design a Fluffy sprite. The details in the Create new sprite dialogue box should be as in the picture.



If you're confident using Paint skip the next bit. Double click select on the Sprite file window. If the tools and colours windows are not showing click menu over the

# Linking Basic Programs

## AppBasic origins

One of Joe Taylor's ideas for AppBasic – see [appbasic.jettons.co.uk/](http://appbasic.jettons.co.uk/) – was that it made more sense for the !Runimage of an application to be a directory than a single file.

Its structure should reflect the relationships between the Toolbox objects that the application uses. AppBasic itself would supply libraries of generic handlers for the toolbox objects, which the user could select and modify.

The final stage, to produce an application, involved linking the libraries together into a single !Runimage file containing just the functions and procedures needed, leaving out all those that were not.

What Joe wanted to use for this stage was Mohsen Alshayef's StrongBS tool; but it had been written many years ago, was not 32-bit clean and would not run on the Lyonix. Efforts to contact the author were unsuccessful, so Joe

asked me to run up a quick and dirty alternative. The result I called Blynken.

Eventually Mohsen Alshayef rejoined the RISC OS scene and produced an updated StrongBS, which rendered Blynken redundant. So I thought I would write this article, which considers what is involved in writing a program to link Basic programs.

It must take a Basic program as input (or rather its pathname) and produce another Basic program as output, which does the same job when run but contains no LIBRARY commands.

Evidently it has to scan all the libraries called by the original program (and the libraries that they call, and so on) and augment the original program by just those function and procedure definitions in them that are needed.

## Blynken's restrictions

The BBC Basic guide states on page 235 that the argument to the LIBRARY command can be an

expression, evaluating to a string giving the pathname of the library file to be loaded.

If the linker program is to avoid parsing and evaluating Basic expressions (and so having to run the program which is to be linked) then we must impose a restriction:

- Only literal strings can be allowed as arguments to LIBRARY.

The linking program can recognise where function and procedure definitions start from the occurrence of the keyword DEF, followed by FN or PROC. How can it tell when they end? It is not enough, in the case of procedures, to say that they are terminated by the keyword ENDPROC. Consider lines like

```
IF condition THEN ENDPROC
```

which show that lots of ENDPROCs can occur inside a

# Python Primary School

**This term we start with the SWI module basics and where things can go wrong if we aren't careful, before using some SWIs in anger.**

Inside **Python-2/4-additions-2005-09-30/zip** there is a StrongHelp **Docs** file and a **HTMLDoc** directory containing help files for RISC OS. They document some SWI functions and methods.



Start the Python interpreter and enter these two lines:

```
import swi
r=swi.swi("OS_WriteC","i;i",65)
```

SWI names are case sensitive. A capital A appears before the prompt on the next line. Now type the letter r followed by return and you should get 65 on the next line and the prompt on the line after that.

**OS\_WriteC** is documented in

the Programmer's Reference Manuals (PRMs), page 517 of **Prm1/pdf** in **Documents.Books.PRM**s directory on your SD card. It writes the character whose Ascii code is in register 0 to the screen, and returns with all registers preserved. The variable **r** gets set to the value returned in R0 by the SWI function in the SWI module.

**"OS\_WriteC"** is SWI we're calling, **"i;i"** is the list of values passed to and returned from the SWI separated by the semi colon, and the comma separated values are passed to the SWI. In this example one integer is passed in R0 and one integer is returned also in R0 which is 65 in both cases.

Enter:

```
swi.swi(0,"i",65)
```

We again get:

```
A>>>
```

Here we have called **OS\_WriteC**

by its number, passing the integer 65 in R0 as before but not returning any value. So no semi colon in the values list and no **r=** at the start of the call.

We can pass:

- an integer with **i**
- a pointer to a string with **s**
- a pointer to the start of a block with **b**
- a pointer to the end of a block with **e**
- insert a 0 to 9 into a register or skip a register with **.** (fullstop)
- insert the negative of the following number into a register with **-** (minus)

We can return:

- an integer with **i**
- a pointer to a string with **s**
- skip a register with a **.** (fullstop)
- return the carry flag as an integer with **\*** (asterisk)

Blocks of memory are used extensively in RISC OS to pass information to and from SWI calls. A block object has been added to the SWI module with functions to extract the values from a block.

The functions are: integer, integers, string, tuple and tuples.

There is a block function to create a block, a resize method to resize an existing block, a register function to register an existing block as a SWI block and a SWI function to call the SWI with.

Other methods of **swi.block** are **bitset**, **ctrlstring**, **nullstring**, **padstring**, **tostring** and **tofile**.

There are also three data items of a block: start, end and length - giving the start address, the end address and the block length in bytes respectively.



Back in the Python interpreter try the following lines:

```
import time
```

```
imports the time module
```

```
time.time()
```

```
Prints the seconds elapsed since the Epoch ("Epoch" in Python is
```

Jan 1 1970 but in RISC OS it is Jan 1 1900)

```
time.ctime()
```

```
Prints the date and time
```

```
time.ctime(65)
```

```
Prints a date 65 seconds after Epoch.
```

```
swi.swi("OS_WriteN", "si", time.ctime(), 24)
```

```
Uses OS_WriteN to write the ctime() string of given length followed
```

```
swi.swi("OS_Write0", "s", time.ctime())
```

```
Uses OS_Write0 to write the zero terminated ctime() string.
```

```
help()
```

```
time
```

```
Lists the help for the time module.
```

The **OS\_WriteN** and **OS\_Write0** are redundant in Python since Python's **print** statement can do the job. But they are easy to set up and instructive for our first attempt at the SWI function.

You may care to read up on the following in the PRMs:

- PRM3 page 817:

Territory\_ConvertStandardDateAndTime

- PRM2 pages 36,38 and 46: OS\_File 4, 5, 17 and 23

- PRM2 pages 102, 108 and 121: OS\_FSControl 18, 24 and 37

We'll be using **os.stat()** which we met last term and **time.ctime()** whose help we have listed in Python's interactive help.

Our first program this term reads the catalogue information of a file called **star** with OS\_File 17, converts the time stamp to a string with **Territory\_ConvertStandardDateAndTime** and prints the string with **swi.string()**. Next we get the Python time stamp for the same file with **os.stat()**. **st\_ctime** and print it with **time.ctime()** then add a second to it before printing again.

We then convert the Python time to Risc OS time and print it again with **Territory\_ConvertStandardDateAndTime** and **swi.string()**. Remember line numbers are given for reference and should not be typed in.

#### *TimeStamp/py listing*

```
1# TimeStamp tester
```

```
2# by Paul Dunnington
```

```
3# (c) Drag N Drop April 2016
```

```
5import os, swi, time
```

# Composing with Milkytracker

**Milkytracker is a cross-platform, desktop application which lets you compose 'Tracker' tunes, also known as 'Chiptunes' owing to their predominance on micros as background music to games, e.g. Lemmings.**

Tracker tunes are still a good way to write games tunes and the RISC OS port of 'Milky' can be downloaded for free at

[www.riscos.info/packages/AudioDetails.html](http://www.riscos.info/packages/AudioDetails.html).

In this article I'll show how to compose short tunes from a musician's viewpoint. Milky isn't like writing notes on a stave and the way you input tunes takes some getting used to.

After installing and running Milky you'll notice the distinctly un-RISC OS-like screen. There is no drag and drop like we are used to on RISC OS because the application was originally a DOS (later Windows) program. This is one aspect of Windows which makes Microsoft difficult to use.



You'll also be thinking that the window and buttons are hard to see because they are so tiny. You can change this by clicking on the Config button (group of grey buttons near the top left, third row on the right). Click Layout and in the Resolutions box on the right there are a dozen or so sizes to choose from, so pick say 800x600 and click Apply. A dialogue box pops up stating you have to re-start Milky. Do this just by clicking on the close icon and double-

clicking !MilkyTracker in wherever you have chosen to install it (mine's in 16GbPi\$.Sound). It restarts with a bigger window.

Tracker tunes consist of sound samples (WAV format) and information on when and how to

play the samples (the events) all rolled into one file.

These events make up the tune (piece of music) and the tune is subdivided into patterns, which can be thought of as phrases or bars. Patterns consist of events (notes). A pattern/phrase/bar can be used more than once like a musical repeat.

More of that in a minute but first you need to get yourself some WAV files. In the 'Files of the World' article in last issue of *Drag 'N Drop* there were some

# Weblog: Label Templates

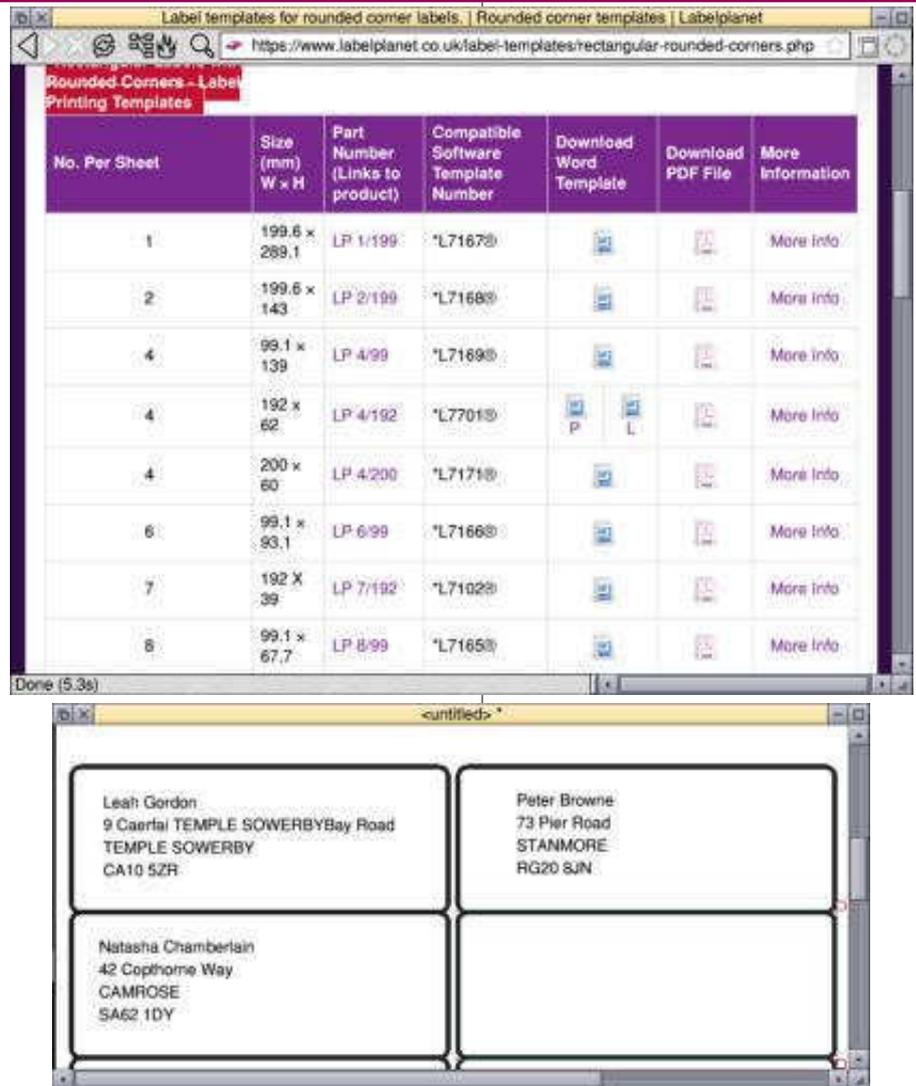
One of the attractive features of RISC OS is the ease with which PDF files can be exported using !PDF to Draw and edited.

You can find numerous websites from which to download templates in PDF for self-adhesive labels. For example [www.labelplanet.co.uk](http://www.labelplanet.co.uk) offers templates for lots of popular labels including Avery format.

A common size of address label is 99 × 38.1 (14 per sheet) on Planet's *Rectangular Rounded Corners* page. Load up Draw, click on the PDF icon on the webpage to download it. Double click to open in !PDF and choose Menu > Save > Page as drawfile. Drag-drop it on Draw's iconbar icon.

You could type names, addresses etc. directly using Draw's text tool. What I do is to type my addresses in a Fireworkz document and export them to Draw as I need.

Load up Fireworkz, select click on its iconbar icon and choose



The top screenshot shows a web browser window displaying a table of label templates. The table has the following columns: No. Per Sheet, Size (mm) W x H, Part Number (Links to product), Compatible Software Template Number, Download Word Template, Download PDF File, and More Information. The data rows are as follows:

No. Per Sheet	Size (mm) W x H	Part Number (Links to product)	Compatible Software Template Number	Download Word Template	Download PDF File	More Information
1	199.6 x 289.1	LP 1/199	*L7167@			More info
2	199.6 x 143	LP 2/199	*L7168@			More info
4	99.1 x 139	LP 4/99	*L7169@			More info
4	192 x 62	LP 4/192	*L7701@			More info
4	200 x 60	LP 4/200	*L7171@			More info
6	99.1 x 93.1	LP 6/99	*L7166@			More info
7	192 X 39	LP 7/192	*L7102@			More info
8	99.1 x 67.7	LP 8/99	*L7165@			More info

The bottom screenshot shows a Draw application window with a label template. The template is a 2x2 grid of rounded rectangles. The top-left cell contains the address: Leah Gordon, 9 Caerial TEMPLE SOWERBY Bay Road, TEMPLE SOWERBY, CA10 5ZR. The top-right cell contains the address: Peter Browne, 73 Pier Road, STANMORE, RG20 8JN. The bottom-left cell contains the address: Natasha Chamberlain, 42 Cophorne Way, CAMROSE, SA62 1DY. The bottom-right cell is empty.



Previously at the C-Side we've been using libraries of standard functions imported at the top of the C program:

```
#include <stdio.h>
```

You can use other libraries (be they downloaded from the internet or "home brewed" and this time we'll be look at a library called DeskLib with graphics in mind.

DeskLib is free and can be downloaded from

[www.riscos.info/index.php/DeskLib](http://www.riscos.info/index.php/DeskLib).

At this point I'd like to thank the RISC OS open forum help on getting started.

Download the core/zip file and extract !DeskLib to where you do your C programming (mine's in 2GBPi\$.Programming.C)

Double click !DeskLib. A window of files open on the right of the desktop. These are the libraries we talked of earlier. We are interested in the one called GFX ("graphics").

Before we set to work on writing any C code, we have to be aware that the way we compile the program (to object code which we double click on to run) is different from last time and also varies depending on whether you are using Acorn C or GCC.

### Acorn C

```
cc -c -IDeskLib: foo.c -o foo.o
```

```
link foo.o DeskLib:DeskLib.o C:stubs.o
```

### GCC

```
gcc -c -IDeskLib: foo.c -o foo.o
```

```
gcc foo.o DeskLib:DeskLib.o
```

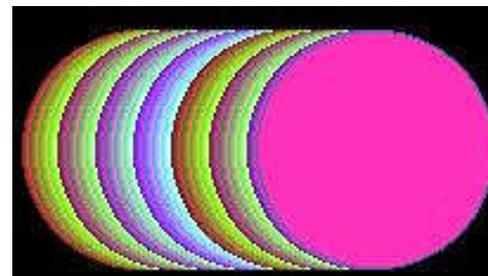
"foo" means "some file or other, you will find frequent reference to it in programming textbooks. It's just the name of the program, whatever you've called it.

The first line generates some intermediate code from the C

program **foo.c**. The name of the new library (DeskLib) appears by the -I switch.

The second line links (joins up) the intermediate object code with the stubs object code, which in C language is the built-in code to call the main() routine of our program.

The result is a file called !RunImage in the C directory, which you double click to run.



Rather than type in commands every time it's easier to save them in an Obey file, called *Compile*. As you develop your C program all you need do is click *Compile* then double click !RunImage. Remember the CSD should be your C work directory (in my case 2GbPi\$.Programming.C).

The program this month, listed below, draws several circles on the screen. Save it in **2GbPi.\$**.