ElkPi Electron

INTRODUCTION

The ElkPi Electron is based on a Raspberry Pi running RISCOS pico inside an original Electron case, using the original keyboard and power supply. It was designed to provide a retro programming experience and to be a showcase for original BBC micro and Electron educational software running natively on the machine. Storage is on the internal 2Gb SD card and a single USB socket allows external USB devices and storage to be connected.

A non-working Electron computer was obtained very cheaply. The motherboard had a fault and the computer would not boot. There was also no power supply with the machine. In addition a large number of key switches on the keyboard did not work.

It was decided that a Raspberry Pi computer would be used to give the Electron a new lease of life and produce an educational programmer's computer. The motherboard was removed to await later repair.

The power supply was functional and only needed a 19v power source of sufficient amperage with the correct connector. A surplus laptop supply was used and it was found that the Electron power supply provided 4.95v on load, an ideal voltage to run a Raspberry Pi.

The key switches which were not working were carefully de-soldered from the keyboard PCB. The pins on these switches unscrew easily using thin-nosed pliers. One at a time the pins were unscrewed and cleaned with 1200 grade wet and dry paper and then screwed back into the switch body. In all over 20 key switches were cleaned in this way.

It was decided to use one of the original Raspberry Pi computers which was surplus to requirements. The one chosen was a Pi 1, model B, with 256Mb of RAM. The ElkPi computer would need only two USB sockets and 256Mb would be enough for the memory needs of the project.

In order to connect the existing Electron keyboard an Electron keyboard USB interface, produced by Tynemouth Software, was used. This originally was designed to allow the Electron keyboard to act as a retro USB keyboard for a PC computer but working with Tynemouth Software allowed a custom interface to be produced which faithfully copied the Electron key mapping for use with RISCOS on the Raspberry Pi.

The version of RISCOS used was RISCOS pico, a console version of RISCOS. This operating system has a very small footprint and boots very quickly, ideal for the project. Not only that but it includes faithful emulation of the BBC micro MODE7 teletext display and the Electron MODEs 0, 1, 2, 3, 4 and 5. There are also a number of RISCOS only MODEs giving 63 colours.

A custom system boot and menu system for the educational software titles was produced along with utilities to read Acorn DFS SSD images and extract files from them and to copy directories from the SD card to external USB storage.

SPECIFICATION

Power Input	110 - 250v 50/60Hz power 'brick' providing 19v DC at 4.2A
Connections	HDMI USB (low power devices only)
Input Device	Standard Electron keyboard
Output Device	HDMI monitor HDMI to DVI HDMI to VGA
Operating System	RISCOS pico
SD Card	Standard 2Gb
Available RAM	96Mb

CONSTRUCTION

The first stage was to take the Electron to pieces and remove all the component boards. The Raspberry Pi and the keyboard interface board were positioned so that cut outs could be made at the rear of the case for the HDMI socket.

Figure (i) shows the case after this process had been done.

Figure (i)

Extra grooves were made in the plastic wall between the PSU board and the main area of the case so that the various cables could be looped neatly in the case. Originally it was intended to use a chassis mounting HDMI socket connected to the Pi. When the setup was tested it was found that this arrangement resulted in an unreliable signal from the Pi so the Pi was mounted directly against the back panel and a cut out made for the HDMI socket there. This is shown in close up in figure (ii).

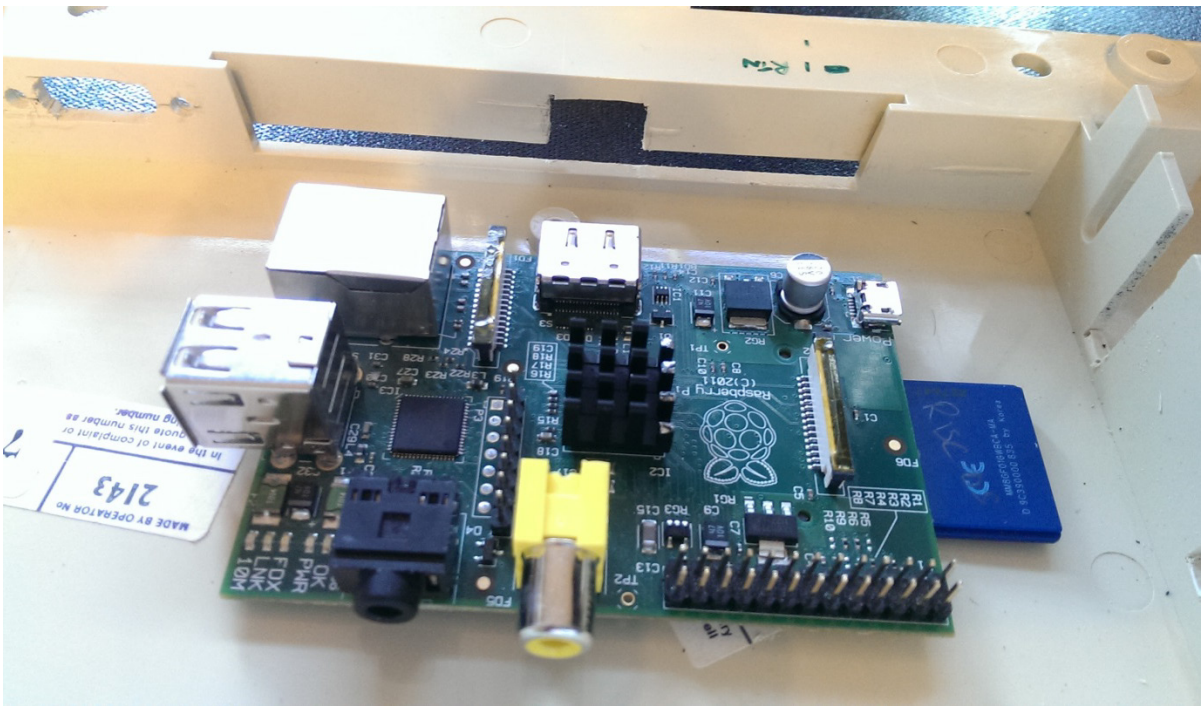

Figure (ii)

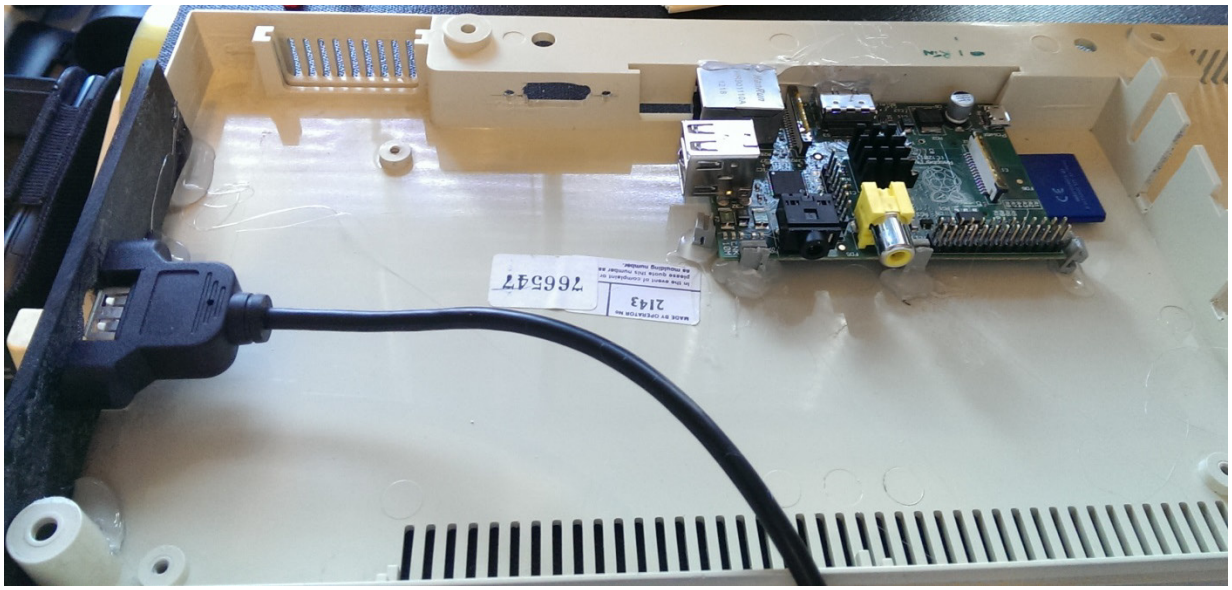

Figure (iii)

Figure (iii) shows the Pi hot glued into position against the back panel. Plastic spacers from a plastic project box were used to support the Pi. Issue 1 of the Raspberry Pi did not come with mounting holes as do the later models, hence the need for hot glue. The Pi was positioned so that the SD card could be removed should it need to be.

The left of figure (iii) shows the chassis mounted USB socket fastened to a piece of black plastic which was hot glued in position to cover the holes for the RGB, Video and Tape sockets in the original Electron. Space is available for a 3.5mm chassis socket to carry audio out should it be felt necessary to add it at a later date.

No provision for access to the GPIO pins was made since it was not the purpose of the computer to use these.

Figure (iv) shows the keyboard interface stuck in position.

Figure (iv)

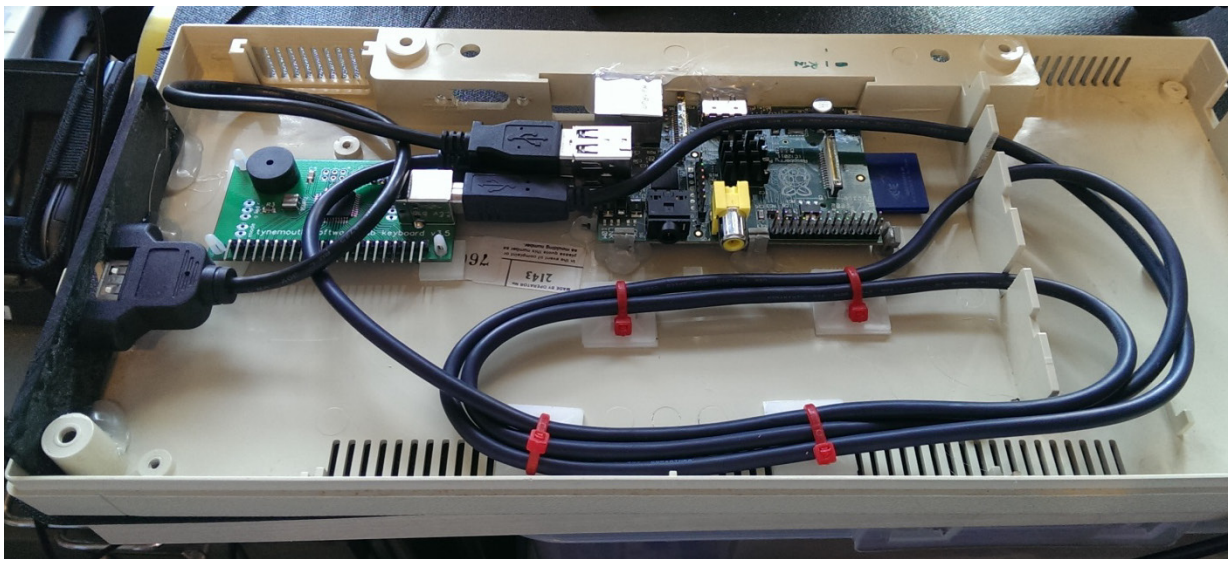

Figure (v)

Figure (v) shows the USB cables connected. The long cable connecting the keyboard interface was supplied with the interface. A shorter cable was initially used and, although neater, it was found to be unreliable.

The PSU was modified. The original cable and white power plug (see figure (i) above) was unsoldered and replaced by a micro USB cable soldered in place to carry +5v to the Pi. This is shown in figure (vi) below, which also shows the keyboard in position and connected to the interface board.

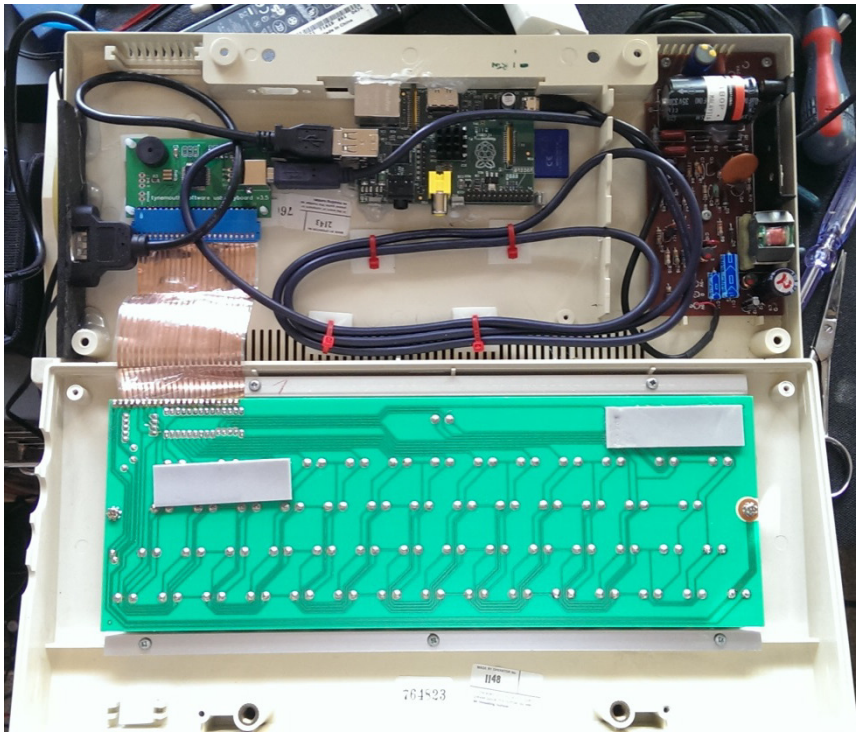

Figure (vi)

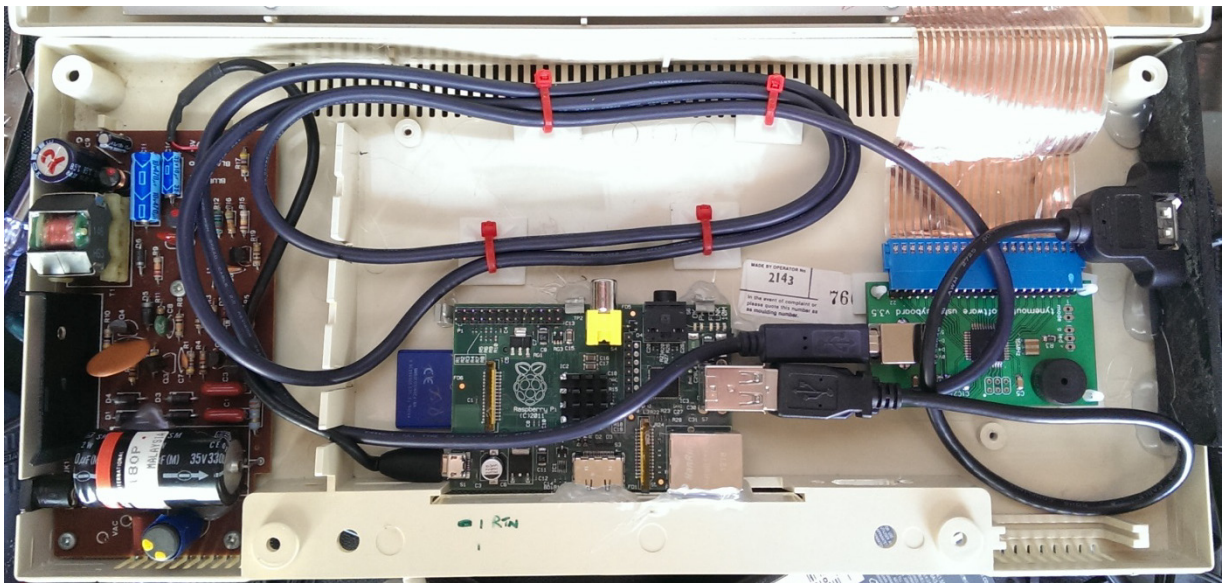

Figure (vii)

Figure (vii) shows the complete wiring in close up before the case was fastened together.

Figure (viii)

The completed computer is shown working in figure (viii) above. The monitor is a retro VGA monitor connected through an HDMI to VGA adapter, the white component in the centre between the monitor and computer.

USING THE ELKPI ELECTRON

The computer uses the original Electron keyboard and this behaves almost exactly as in the original Electron. The Shift, Ctrl and Func modifier keys work exactly as expected. There is a difference from the Electron in that RISCOS gives uses function keys F1 to F12 rather than the Electron's F0 to F9.

Figure (ix)

These keys have been mapped as follows;

F1 to F9	Func + F1 to F9
F10	Func + F0
F11	Func + =
F12	Func + Break

The Caps Lock function has been mapped as Escape + Caps Lock.

Break by itself is equivalent to Escape. This is a RISCOS feature rather than a keyboard interface one.

Ctrl-Break and Shift-Break both reset the Raspberry Pi and initialise RISCOS Pico. This is standard RISCOS pico behaviour.

There is no keyboard equivalent for Page Up and Page Down.

SYSTEM BOOT

A simple boot file, !BOOT, was written to run a BASIC program to display a custom boot screen. !BOOT has its type set as Obey and contains the command;

```
BASIC Elk
```

The BASIC program "Elk" defines the Acorn character, displays the screen message, sets $ as the currently selected directory (CSD), initialises the keyboard to have Caps Lock set, deletes itself and returns to BASIC. The command LIST shown in figure (x) was to test that the Elk program had self-deleted.

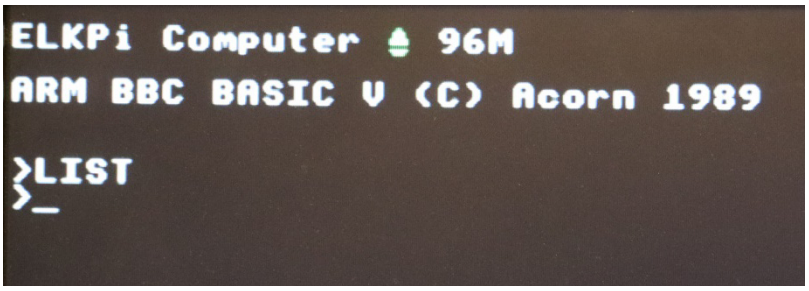

Figure (x)

```
 10REM> Elk
 20MODE28
 30VDU23,235,24,60,60,126,126,0,126,60
 40PRINT'"ElkPi Electron ";
 50COLOUR12:PRINTCHR$255;
 60COLOUR63:PRINT" 96M"'
 70PRINT"ARM BBC BASIC V (C)Acorn 1989"'
 80*DIR $
 90VDU21:*KEY1NEW|F|M
100*FX138,0,129
110*FX202,32
120*FX118
```

When used at Computer Shows, this Elk program is modified to CHAIN the main educational software menu after Shift-Break or Ctrl-Break.

CONCLUSION

This has proved to be a very reliable, compact and quirky computer. It fulfils its purpose for demonstrating BBC educational software running natively on a modern RISCOS computer. The programming experience is similar to that of a BBC micro and an Acorn Electron but with the power of RISCOS and BASIC V.

There is room to add BBC micro style user ports and a stereo 3.5mm jack socket for audio if it is required.

The construction is compatible with versions of the Raspberry Pi below the Pi 3. There is no real benefit to be obtained by using a more recent Raspberry Pi unless more USB sockets are required to allow the full RISCOS with GUI to be used. A Pi Zero would need an internal USB hub and would not afford much space saving but it could realistically allow the ElkPi to be battery operated - an Electron laptop?