LINUX VOICE TUTORIAL

RASPBERRY PI MODEL B: VOID YOUR WARRANTY

BEN EVERARD

Now we have a shiny new B+, it's time to try some dangerous experiments on our old Raspberry Pi model B.

WHY DO THIS?

- Learn the limits of your Raspberry Pi.
- Let an old model B go out in a blaze of glory.
- Add new features and personalise your Pi.

Now the Raspberry Pi B+ has come out, we've found ourselves with some of the original model B's that we're not going to use any more. These are still fully functional computers, so it seems a waste to let them rot in a drawer, or worse, throw them out. Instead, we decided to use one as a test bed for some riskier experiments.

We didn't break a Pi while researching this article, but we certainly could have done. We accept no responsibility should you slip and fry your Pi, but what better way is there to get to know a device than to push it to its limits?

Overclocking

Raspbian comes with **raspi-config**, a tool that lets you set various configuration options for your Raspberry Pi. One of which is the overclocking level. It has a series of safe levels that can give you a bit of a speed boost without damaging your Pi (though not all Pis will work at the highest speeds). This is useful for getting a bit more oomph, but it obviously raises the question of just how fast you can push your Pi.

To take things further than **raspi-config**'s menu will allow, you'll need to edit the **config.txt** file on the boot partition of the SD card. It's easiest to do this after you've set the Pi to Turbo overclocking (one of the options in the config tool) since this makes most of the options visible. You can edit this file either by putting the SD card in another computer, or from within the Pi with:

```
sudo nano /boot/config.txt
```

The de-soldering pump we used. The orange button triggers the suction and pulls the molten solder off the board.

Before going any further, we should say that there's a chance that following this tutorial will void your Pi's warranty, and there's a small chance that it'll explode in a shower of sparks (and a slightly larger – but still small – chance that it'll break in a less spectacular way). In other words, don't try this if you're not prepared to accept the risk that your Pi will stop working permanently.

If you're already in Turbo mode, you should find the following options set:

```
arm_freq=1000
core_freq=500
sdram_freq=600
over_voltage=6
```

You can mess with these to boost the performance. The three frequency settings are all in MHz, so this configuration has the main ARM processor running at 1GHz, the GPU running at 500MHz, and the SD RAM running at 600MHz. We found that we couldn't squeeze any more speed out of the GPU or the SD RAM. However, there does tend to be a little headroom in the ARM frequency.

In order to take advantage of this, though, you'll need to increase the voltage. The voltage for the core defaults to 1.2V, and each increase in the **over_voltage** setting sends an extra 0.025V. With a setting of 6, the core is running at 1.35V. Increasing the voltage enables you to increase the speed, but it can also decrease the life expectancy of the chip. Since we're seeing how much speed we can get, we whacked this up to its maximum setting of 8 (1.4V).

There's another option that you'll need to set if you want to take it beyond the normal overclocking levels:

```
force_turbo = 1
```

Just add this line to the **config.txt** file, and it'll let you push the performance up.

There are a couple of things you need to be aware of as you increase the clock speed. The most obvious is that it will become more prone to crashing, so don't use a heavily overclocked machine for any important work. The second important thing is that it will tend to run hotter than at slower speed, so you need to keep an eye on the temperature to make sure it doesn't get too hot.

You can check the temperature at any time with the command:

```
cat /sys/class/thermal/thermal_zone0/temp
```

This gives the temperature in 1000ths of a degree Celsius, so 45000 is 45°C. As a general rule of thumb, you want to keep the temperature below 70000, but again this depends on how much you're willing to risk breaking your Pi. We gradually increased the clock speed in 50MHz increments, and performed a simple benchmark of unzipping an archive.

We found that we could run our Pi at 1.2GHz, though it wasn't very stable. At this speed, our benchmark ran about 40% faster than at non-overclocked speeds, and about 20% faster than Turbo overclocking, with a core temperature of around 60°C. This was, however, quite a simple benchmark. A more complex task may well have proved too much for the SoC at this speed. However, we did find that our Pi was reasonably stable at around 1.1GHz.

Modifying the board

You may think that, unlike desktop PCs, you can't change much on single-board computers like the Raspberry Pi. You may think that the Raspberry Pi Foundation choose what goes on the board and you just have to go with it. This isn't necessarily true. They're certainly not as flexible as desktop PCs, but with a little soldering, you can certainly tweak them to your needs. We stripped off a component we didn't need, and added one we did.

The analogue video output may be useful to some people, but not for us. It just takes up space and makes the board look cluttered. This wasn't enough for us to risk removing it before, but now we've had enough, and decided to take it off. It's only attached by three soldered points that are quite large and easy to access, so it's easy to remove.

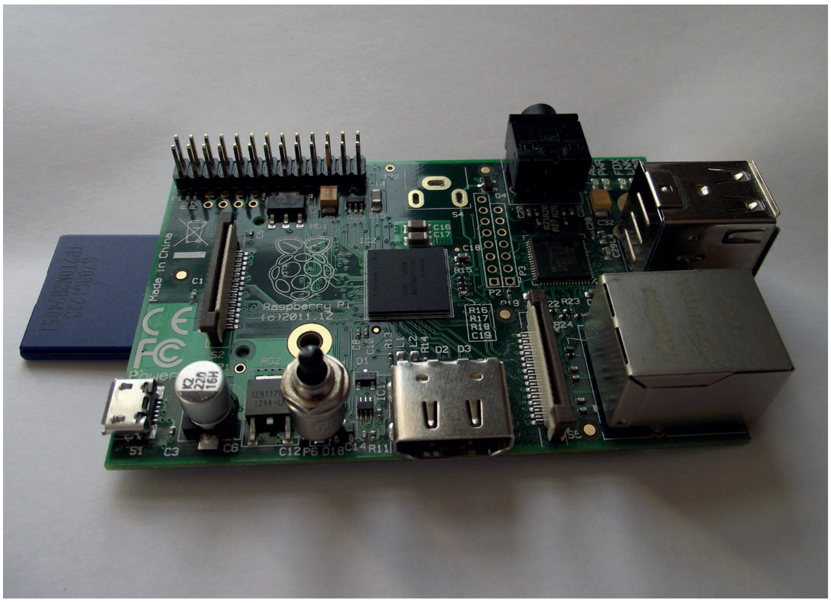
You will need either a desoldering pump or wick – we used a pump. This is a device that looks a bit like a synringe from a sci-fi film. It has a spring-loaded plunger that you press down, then a button that releases the plunger. As the plunger shoots up, it sucks air in through the nozzle. If the nozzle is placed near molten solder, it'll suck the solder off the board.

The trick of desoldering is to heat the solder up until it's molten, then use the pump or wick to remove it before it cools. If you get enough off, you should find that the component slides out. Be careful not to heat up the board too much, or pull too hard on the component, or you could damage the board. It's made of several layers stuck together, so there could be wiring you can't see.

This might seem a bit pointless, and on its own it is a bit, but for some projects where size or weight is important, removing unnecessary components can be useful.

Extra features

You may have noticed that there are loads of extra holes in model B Pis that don't seem to be used. The two larger ones are mounting holes. Some of the others are used for manufacturing and testing, but some of them allow access to extra features. One mildly annoying feature of the Raspberry Pi is the fact that there's no power button. You can halt it from software, but to turn it on you have to unplug the power cable, then plug it back in. That's not the most user-friendly way of doing things, and it could be awkward in embedded settings. Fortunately, there's an alternative. Between the HDMI and power connectors, you should see two holes labelled P6. These are mounting points for a reset switch.


The board after our modifications. If nothing else, it now feels like it's truly our board and not just another Raspberry Pi model B that's rolled off the production line.

All you have to do is make a connection between these two points and the Pi will reset. You can test this out using a flat-head screwdriver. This will reboot your Pi whether it's switched on or off.

In order to be able to reset your Pi, you simply need to add a normally-open push switch between these two points. The best way to do this will depend on how your Pi is set up. We use ours without a case, so we simply soldered the switch straight onto the board. However, if your Pi is inside a case or some other enclosure, you may find it easier to solder wires onto the board and attach those wires to a switch in a more convenient location. On the B+, the connections for the reset button are labelled Run, and are located next to the micro SD card slot.

Not many of the holes in the model B are useful. There are a couple of extra GPIOs on the P5 header that you could find functions for. If you really want to modify your Pi, you could replace components on there with better ones. For example, it is possible to take off the linear regulator from a model B and replace it with a switching reg. In doing this, you'll gain one of the best features of the B+. It's not a particularly simple process, but there's some guidance on Dave Akerman's excellent blog at **www.daveakerman.com/?page_id=1294**.

Despite being superseded by the B+, there's still plenty of fun left to be had with model B's, so don't let yours rot in a drawer. Get it out and void its warranty – and learn more about it in the process.