TeX on the Raspberry Pi

Hans Hagen

This is a short status report on Pi, not the famous version number of TeX (among other things), but the small machine, meant for education but nowadays also used for Internet Of Things projects, process control and toy projects. While the majority of TeX installations run on an Intel processor, the Raspberry Pi has an ARM central processing unit. In fact, its main chip has the same foundation as those found in settop boxes all around the world. It's made for entertainment, not for number crunching.

At the ConTeXt meetings, it has become tradition to play with electronic gadgets. Every year we are curious what Harald König might bring this time. The last couple of meetings we also had talks about using TeX and MetaPost for designing (home-scale, automated) railroad systems, using LuaTeX for running domotica applications, using MetaPost for rendering high quality graphics from data from appliances, presenting TeX at computer and electronics bootcamps, and more. This year Frans Goddijn also brought back memories of low speed modem sounds, from the early days of TeX support. It is these things that make the meetings fun.

This year the meeting was in Belgium, close to the border of the Netherlands, and on the way there Mojca Miklavec traveled via my home, where the contextgarden compile farm runs on a server with plenty of cores, lots of memory and big disks. But the farm also has an old Mac connected as well as a tiny underpowered Raspberry Pi 2 for ARM binaries that we had to fix: the small micro SSD card in it had finally given up. This is no surprise if you realize that it does a daily compilation of the whole TeX Live setup and also compiles LuaTeX, LuaMetaTeX and pplib when changes occur. Replacing the card worked out but nevertheless we decided to take the small machine with us to the meeting. We also took an external (2.5 inch) SSD box with us. The idea was to order a Raspberry Pi 4 on location, the much praised successor of the older models, the one with 4 GB of memory, real USB 3 ports and proper Ethernet.

At the meeting Harald showed us that he had version 1, 3 and 4 machines with him because he was looking into an energy control setup based on Zigbee devices. So we had the full range of Pi's there to play with.

This is a long introduction but the message is that we are dealing with a small but popular device with up to now four generations, using an architecture supported in TeX distributions. So how does that relate to ConTeXt? One of the reasons for LuaMetaTeX going lean and mean is that computers are no longer getting much faster and 'multiple small' energy-wise has more appeal than 'one large'. So then the question is: how can we make TeX run fast on small instead of gambling on big becoming even bigger (which does not seem to be happening anyway).

At the meeting Harald gave a talk "Which Raspberry Pi is the best for ConTeXt?" and I will use his data to give an overview: see Table 1.

After some discussion at the presentation we decided to discard the (absurd) bogomips value for the tiny Pi 1 computing board and not take the values for the others too seriously. But it will be clear that, especially when we consider the external drive that things have improved. The table doesn't mention Ethernet speed but because the 4 now has real support for it (instead of sharing the USB bus) we get close to 1 GB/s there.

The real performance test is of course processing a TeX document and what better to test than *The TeXbook*. The processing time in seconds, after initial caching of files and fonts is:

Pi model:	1	2	3	4
The TeXbook	13.649	7.023	4.553	1.694
`context --make`		19.949	11.796	6.034
`context --make` TL	89.454	46.578	29.256	14.146

Table 1: Capacities for Raspberry Pi models.

Pi model:	1	2	3	4
chipset	BCM2835	BCM2835	BCM2835	BCM2835
CPU core	v6l rev 7	v7l rev 5	v7l rev 4	v7l rev 3
cores	1	4	4	4
free mem	443080	948308	948304	3999784
idlemips	997.08	38.40	38.40	108.00
bogomips	997.08	57.60	76.80	270.00
read SD	23.0 MB/s	23.2 MB/s	23.2 MB/s	45.1 MB/s
read USB		30.0 MB/s	30.0 MB/s	320.0 MB/s

The test of making the ConTEXt format using LuaTEX gives an indication of how well the I/O performs: it loads the file database, some 460 Lua modules and 355 TEX source files. On my laptop with Intel i7-3840QM with 16GB memory and decent SSD it takes 3.5 seconds (and 1 second less for LuaMetaTEX because there we don't compress the format file). Somehow a regular TEX Live installation performs much worse than the one from the contextgarden.

We didn't test real ConTEXt documents at the meeting but when I came home the Pi 4 was bound again to the compile farm. Harald and Mojca had prepared the machine to boot from the internal micro SSD and use the external disk for the rest. So, when we could compile LuaMetaTEX again, I made an ARM installer for LMTX, and after that could not resist doing a simple test. First of course came generating the format. It took 6.3 seconds to make one, which is a bit more than Harald measured. I see a hiccup at the end so I guess that it has to do with the (external) disk or maybe there is some throttling going on because the machine sits on top of a (warm) server.

More interesting was testing a real document: the upcoming LuaMetaTEX manual. It has 226 pages, uses 21 font files, processes 225 MetaPost graphics, and in order to get it LuaMetaTEX does more than 50% of the work in Lua, including all font and backend-related operations. On my laptop it needs 9.5 seconds and on the Pi 4 it uses 33 seconds. Of course, if I take a more modern machine than this 8-year-old workhorse, I probably need half the time, but still the performance of the Raspberry Pi 4 is quite impressive. It uses hardly any energy and can probably compete rather well with a virtual machine on a heavily loaded machine. It means that when we ever have to upgrade the server, I can consider replacement by an Ethernet switch, with power over Ethernet, connected to a bunch of small Raspberries, also because normally one would connect to some shared storage medium.

Because I was curious how the dedicated small Fitlet that I use for controlling my lights and heating performs I also processed the manual there. After making the format, which takes 6 seconds, processing the manual took a little less than 30 seconds. In that respect it performs the same as a Raspberry Pi 4. But, inside that small (way more expensive) computer is a dual core AMD A10 Micro-6700T APU (with AMD Radeon R6 Graphics), running a recent 64-bit Ubuntu. It does some 2400 bogomips (compare that to the values of the Pi). I was a bit surprised that it didn't outperform the Raspberry because the (fast SSD) disk is connected to the main board and it has more memory and horsepower. It might be that in the end an ARM processor is simply better suited for the kind of byte juggling that TEX does, where special CPU features and multiple cores don't contribute much. It definitely demonstrates that we cannot neglect this platform.

◇ Hans Hagen
http://pragma-ade.com