First boot with the Star64 board (RISC-V processor)

Stéphane Bortzmeyer
<stephane+blog@bortzmeyer.org>

First publication of this article on 30 April 2023. Last update on of 3 May 2023

https://www.bortzmeyer.org/star64-first-boot.html

I now have a Star64 board <https://wiki.pine64.org/wiki/STAR64>, a board which uses a RISC-V processor, a processor whose instruction set is free (free as in free speech, not free as in free beer). Let's see the first boot. (*"Désolé pour mes lecteurices francophones mais il y a peu de retours d'expérience sur le Star64 publiés donc je pense qu'il vaut mieux commencer par l'anglais pour toucher davantage de monde."*)

First, a bit of context about RISC-V since it was my main motivation to buy this board. A processor is first defined by its instruction set since it is what the software will have to know : a compiler, for instance, will generate machine code for a given instruction set. The two best known instruction sets today are the Intel one ("x86") and the ARM one. They are not free, in any sense of the word. Even if you have the knowledge and the money, you are not allowed to create a processor using this set (unless you enter in an agreement with Intel or ARM), and even if you can, you have no say about its future evolutions. This is a serious issue since, in our widely digitalized world, the company controlling the processors has a lot of power.

This is why the RISC-V was developed : its instruction set is free (as I said above, quoting Stallman, "free as in free speech, not free as in free beer") and anyone can create a processor implementing it. Of course, since there is a big difference between free hardware and free software, this will not be gratis. Unlike the software, the hardware has marginal costs which are not zero; each unit produced will cost money. So, it won't be like the free software world where a student in his home can produce brilliant programs that you just have to distribute over the Internet. Actual production of the physical devices will still depend on corporations. Also, not all the components of the board are free (and, for the processor, the fact that the instruction set is free does not mean the actual processor's schematics are). But it is still a big step towards freedom. (Today, RISC-V seems mostly used in the embedded world, not in general-purpose devices.)

So, I wanted to play a bit with a RISC-V processor. Not owning a factory able to produce them from the schematics, I had to buy the processor. And not just the processor, because I'm not a hardware guy, I'm not going to connect all the necessary stuff to the SoC. I went to Pine64 <https://www.pine64.org/>, which is mostly known for its various devices using an ARM processor, but which also produces a board with a RISC-V processor, the Star64 <https://pine64.com/product-category/

star64/> (the processor is a StarFive <https://www.starfivetech.com/> JH7110). I ordered one <https://pine64.com/product-category/star64/> (70 US dollars, for the 4 GB of memory model) and it arrived :

(If you want to try RISC-V, there are also several other boards <http://krimsky.net/articles/riscvsbc.html>.)

If you want better pictures of this board, there is Linux Lounge's video <https://www.youtube.com/watch?v=CndQJb83WG8>.

Warning if you plan to do the same : the current status of the Star64, and of software which can run on it, is **experimental**. Do not expect a fully finished and smooth product. If you don't know about computers, then it is not for you.

You will probably also need some accessories (the board is sold alone, without even a power supply, accessories can be found on the Pine64 store <https://pine64.com/product-category/star64-accessorie
> or at your regular dealer, do not forget to check the technical specifications) :

- Power supply,
- microSD card,
- Probably also an adapter from serial to USB, for the serial console of the board.

Here is a picture of such adapter (the DSD TECH SH-U099 USB-to-TTL model) :

Speaking of the console, the Star64 has a HDMI port but it is not activated by the firmware, you cannot use it as a console. Either you boot blindly, or you connect the serial ports, by soldering or by using a pre-made cable like the one above, for instance converting to USB. As the documentation says <https://wiki.pine64.org/wiki/STAR64> (see also the schematics <https://files.pine64.org/doc/star64/Star64_Schematic_V1.0_20220721.pdf>, section "40pin GPIO Bus"), transmission is on pin 8 and reception on pin 10. You therefore connect the RXD on 8 and TXD on 8 :

I then used Minicom to connect from an Ubuntu machine, with this configuration file :

```
# Connection to the serial port, for managing the Star64 card
pu port             /dev/ttyUSB0
pu baudrate         115200
pu bits             8
pu parity           N
pu stopbits         1
pu minit
pu mreset
pu mhangup
pu rtscts           No
```

The way I booted my card was the following :

- Get an image for microSD card (I used the ones by Fishwaldo <https://github.com/Fishwaldo/meta-pine64>, where he uses his own tree of the kernel <https://github.com/Fishwaldo/Star64_linux>) and uncompress it (after decompression, they are in the - currently quite undocumented - WIC format),
- Copy it on a microSD card (I used dd on an Ubuntu laptop, `dd if=star64-image-weston-star64.wic of=/dev/mmcblk0 bs=1M status=progress`, you can ignore "GPT :Primary header thinks Alt. header is not at the end of the disk" / "GPT :Alternate GPT header not at the end of the disk", it will be fixed later),

- Insert it in the slot under the Star64,
- Set the switches on the board to boot on the SD card. Warning, they are wrongly labeled ; if the board tells you that they have to be Off-On to boot on SD, it is actually the opposite (this is documented on Pine64 Wiki <https://wiki.pine64.org/wiki/STAR64>),
- Power on and **be patient**, specifically the first time if the operating system has to resize the filesystem partitions. The operating system, made with Yocto <https://www.yoctoproject.org/> using `.deb` (the format of Debian packages), will request an IP address with DHCP, and you will connect either through your serial console, or with SSH (warning, one of the images does not have a SSH server <https://github.com/Fishwaldo/meta-pine64/issues/1>). Login and password are documented with the images <https://github.com/Fishwaldo/meta-pine64>.

Speaking of the microSD slot, it is on the other side of the Star64 :

At this stage, you are now connected to a Unix machine, and you can use regular Unix commands, and Debian package management commands :

```
% uname -a
Linux grace 5.15.0 #1 SMP Thu Apr 13 06:38:09 UTC 2023 riscv64 riscv64 riscv64 GNU/Linux
```

You can now continue with a regular Unix.

Among the hardware of the board, SD card reader, Ethernet and USB work fine (I was able to mount a USB key). I did not yet test the rest. For instance, there is not yet a package to play sound files so I did not test the audio output.

Now, let's try to compile. I installed the required packages with `apt install automake libtool dpkg-dev gcc binutils libgcc-s-dev`. We can now write a C program :

```
% cat plus.c
void main()
{
  int a;
  a = 3;
  a++;
}
```

It compiles and runs. Great. Now, let's see some RISC-V assembly language :

```
% gcc -O0 -S -o plus.s  plus.c

% cat plus.s
.file "plus.c"
.option pic
.text
.align 1
.globl main
.type main, @function
main:
addi sp,sp,-32
sd s0,24(sp)
addi s0,sp,32
li a5,3
sw a5,-20(s0)
```

```
lw a5,-20(s0)
addiw a5,a5,1
sw a5,-20(s0)
nop
ld s0,24(sp)
addi sp,sp,32
jr ra
.size main, .-main
.ident "GCC: (GNU) 11.3.0"
.section .note.GNU-stack,"",@progbits
```

And you can now study RISC-V, from the code produced by the compiler.

A few more technical details, now. First, you can see the boot messages of the card (en ligne sur https://www.bortzmeyer.org/files/star64-boot-serial.txt), then the boot messages of the kernel (en ligne sur https://www.bortzmeyer.org/files/star64-boot-dmesg.txt). Then, the processor :

```
% cat /proc/cpuinfo
processor : 0
hart : 1
isa : rv64imafdc
mmu : sv39
isa-ext :
uarch : sifive,u74-mc
```

(There are also three other cores.) The partition table of the SD card is :

```
% fdisk -l /dev/mmcblk1
Disk /dev/mmcblk1: 29.73 GiB, 31927042048 bytes, 62357504 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disklabel type: gpt
Disk identifier: B7092089-5983-4D4B-B075-78EABD7A7AFB

Device          Start      End  Sectors   Size Type
/dev/mmcblk1p1   4096     8191     4096     2M HiFive Unleashed BBL
/dev/mmcblk1p2   8192    16383     8192     4M HiFive Unleashed FSBL
/dev/mmcblk1p3  16384   793803   777420 379.6M Microsoft basic data
/dev/mmcblk1p4 794624 62357470 61562847  29.4G Linux filesystem
```

As you can see, /dev/mmcblk1p4 is where the root filesystem resides.

Among the things that I plan to try (one day !) :
— Performance tests,
— Trying DNS software (my day job),
— Build my own image, at least the Unix partition on the SD card (/dev/mmcblk0p4). May be populate it with a ”real” Debian ? There is documentation <https://github.com/janvrany/riscv-debian> but apparently outdated. There are also instructions for another RISC-V processor <https://wiki.debian.org/InstallingDebianOn/StarFive/VisionFiveV1>, which might be useful. There is no official port of Debian on RISC-V <https://www.debian.org/ports/>, and the Wiki <https://wiki.debian.org/RISC-V> seems quite abandoned but the work goes on <https://wiki.debian.org/Ports/riscv64>.

———————————-

- Compile Go programs (I'm not sure there is yet a Go compiler for RISC-V),
- Compile Rust programs (same remark),
- Try LLVM (there is no ready-made package <https://github.com/Fishwaldo/meta-pine64/issues/4>),
- Try to build and run a Gemini server,
- (Much more ambitious.) Try to compile Erlang/OTP.
- Enable the HDMI interface so I could try the large screen.

If you want more information, you can go to :

- About the hardware, on the official Wiki <https://wiki.pine64.org/wiki/STAR64>.
- About the boot process of the RISC-V processor of the Star64, all the gory details <https://doc-en.rvspace.org/JH7100/PDF/JH7100_Boot_User_Guide.pdf>. For uboot, see this documentation <https://u-boot.readthedocs.io/>.
- Another good account of running Linux on a Star64 <https://lupyuen.codeberg.page/articles/linux.html>, with a lot of details, and the discussion on Ycombinator <https://news.ycombinator.com/item?id=36579963>.
- Booting from the network <https://lupyuen.codeberg.page/articles/tftp.html> (and not from the SD card as I did), a detailed documentation.
- Inspecting existing images <https://lupyuen.codeberg.page/articles/star64.html>.
- Starting Armbian, an experience <https://forum.armbian.com/topic/28912-star64-jh7110/>.
- A comparison of the various SBC with RISC-V <http://krimsky.net/articles/riscvsbc.html>.
- If you like videos, here is a good introduction to RISC-V <https://www.youtube.com/watch?v=Ps0JFsyX2fU>, technical, economical and political aspects.

To get help or simply to discuss with other fans, you can :

- Use Reddit, specially the RISC-V channel <https://www.reddit.com/r/RISCV/> and the official Pine64 channel <https://www.reddit.com/r/PINE64official/>,
- Use Pine64 Web forum <https://forum.pine64.org/>,
- Use Discord or Matrix, see Pine64 Wiki for details <https://wiki.pine64.org/wiki/Main_Page#Chat_Platforms>.