

Parallella: a real starter kit

1°) Introduction

2°) Goal

- a) Intended audience
- b) General constraints

3°) In practice

- a) Hardware
- b) Downloads
- c) Extra manipulation ? The serial port
- d) SSL, putty... Parallella boots !

1°) Introduction

Parallella is a fascinating coding platform.

Its potential is huge, some of its properties are really unique.

As a hobbyist, I am an early supporter of the Kickstarter campaign.

Only recently I made up my mind, tried to code on this platform.

Since I succeeded,

since the learning curve was quite heavy for a real beginner like me,

since the documentation (does exist, is open but) is on rare occasions scattered and ambiguous, this «real starter kit» is my gift to thank the forumers or people that indirectly helped me, beginning with Andreas Olofsson for this wonderful visionary initiative,

and to help real starters that want to **test Parallella in a day**.

In a short series of nut-in-a-shell documents you'll hopefully find everything, from hardware to software, to start from scratch, if like me you share the same **restricted goal**.

2°) Goal

a) Intended audience

Parallella may be used for many, many things.

My primary goal is to check if Parallella may be a good **coprocessor**: crunching integer numbers at low cost. Yes indeed, **integers, not floats**.

The intended audience is definitely **real starters** first,

but maybe students or programmers interested in **SPMD**.

SPMD = Single Program Multiple Data.

A problem may be subdivided into parallel and independant subproblems.

Being parallel and independant drastically reduces complexity, moreover with no CPU overhead. Even with a basic 16-core Parallella we can explore this way.

b) General constraints

- the less time, the best
- the less money, the best
- the less hardware, the best
- the less coding stuff, the best
- the less coding complexity, the best

In a word: the KISS principle, Keep It Stupid Simple

3°) In practice

a) Hardware

So. What to buy:

- 100 € ; a 16-core Parallella, of course.
 - 15 € ; a fan
 - 15 € ; a 5V power
 - 9 € ; a micro SD card reader for your computer
 - 6 € ; a 8 GB micro SD card
 - 5 € ; a standard Ethernet cable
- => 150 € for French people like me, approximately 150 \$ for American people I guess.

About the fan

The early Kickstarter models absolutely need it, or you'll burn your Parallella.
Avoid my naive mistake: cheap USB fans may be deadly noisy.

Picture 1 under

<https://www.parallella.org/quick-start/>

«Before you begin the tutorial, please follow these instructions to properly install the heatsink for the board. Do NOT attempt to use the board without the heatsink!»
Credits: Suzanne MATTHEWS. Her tutorial on the forum helped me a lot.

Basically if you have an early Kickstarter kit you have to stick the heatsink to the central black IC ZYNQ. It takes 30 s, however do NOT put your fingers anywhere on the board, remember that the static electricity may damage or kill some IC (Integrated Circuit).

About the 5V power

This is really, really essential for stability, moreover a time-consuming task to do first.

Do not rely on your USB computer or any cheap power.

Refer to the Parallella manual (REV 14.09.09): you need a « high quality 2000mA rated 5V DC power supply with 5.5mm OD / 2.1mm ID center positive polarity plug.»

This hardware is exotic and hard to find ; I got mine from this provider: VHBW.

www.vhbw.fr ; SKU OS4-C-61 ; only 10 € ; fully functional, zero problem after 10 days.

Other examples for your country are on the forum, I suggest searching « polarity plug» or «5.5mm» on the forum:

<https://parallella.org/forums/>

<https://parallella.org/forums/viewtopic.php?f=12&t=1319&p=8222&hilit=polarity+plug#p8222>

About the micro SD card reader

This format is not common either. Make sure your multi card reader can read micro SD cards. It is about 10 millimeters wide, near the µ-HDMI connector ; please refer to above Picture 1 link.

About the 8GB micro SD card

This is so cheap you can afford 16 or 32GB if you like.

As we use a «headless» configuration, even 4 GB would be enough - just enough indeed.

«**Headless**» is **lightweight** on the contrary to the more user-friendly «HDMI» config, that now (April 2016) requires a 16 GB micro SD card:

- no X client running on the host Linux
- no dedicated ZYNQ FPGA resources for the HDMI

Besides, this is more up to date, dedicated to performance, not the same audience as the

«HDMI» also called «Desktop» version.

With our «headless» configuration we are definitely NOT blind since we'll use SSH from our favourite Linux computer.

About the standard Ethernet cable

Some people might say you need a crossover cable.

It is true... if you work with really old hardware.

But for a few years now, any basic Ethernet card can safely manage both connections: crossover or standard Ethernet cable. A computer technician told me that.

So I took a crossover cable, tested it... fine.

I pinged the connection... 160 ms not a very good latency but enough for our purpose.

So I took a standard Ethernet cable, tested it... fine.

I pinged the connection... 160 ms again. No difference in any kind.

IMO you can buy what you want.

CAT5/5E/6/7 are speed norms ; CAT5E tends to replace CAT5, better transfer rates.

CAT6 and 7 families are even better.

But for transferring a few C files while debugging, CAT5 should be enough ; I personally use a CAT6 Ethernet cable.

b) Downloads

My current favourite flavour of Linux is Fedora Core 23.

Allow me a brief digression:

- Ubuntu rocks, that was good for me, coming from a Windows environment
- the Debian Jessie 8.1 is a very good distro ; very stable
- Fedora is a good surprise to me. No bloatware, well-organized, bleeding-edge tools... this is why I prefer it to the Debian, a more «conservative» choice.

So I refer to my Fedora experience from now on ; the syntax might differ slightly if you have another Linux.

The SD card image

Follow the «quick steps» here:

<https://www.parallella.org/quick-start/>
to burn an SD image, this is all good.

Find a «headless» image for Kickstarter... so for Xilinx 7020.

I personally used ubuntu-14.04-headless-z7020-20150130.1.img.gz

Things evolve quickly, I see there's a 2016.3 release now:

<http://www.parallella.org/create-sdcard/>

Here are my **quick notes** during installation:

```
#flash SD CARD 2016/03/25
```

```
sudo su...
```

```
gunzip -d ubuntu-14.04-headless-z7020-20150130.1.img.gz
```

```
df -h
```

```
Sys. de fichiers
```

```
Taille Utilisé Dispo Uti% Monté sur
```

```
(...)
```

```
/dev/mapper/fedora-home
```

```
94G 23G 67G 26% /home
```

```
(...)
```

```
/dev/sdc1
```

```
7,5G 32K 7,5G 1% /run/media/ylav/6927-9626
```

```
#unmount the SD card
umount /run/media/ylav/6927-9626
#burn the image to the SD card
dd bs=4M if=ubuntu-14.04-headless-z7020-20150130.1.img of=/dev/sdc1
#it took 6 minutes to complete ; it's a cheap class 10 µSD card by the way
931+1 enregistrements lus
931+1 enregistrements écrits
3905945600 octets (3,9 GB) copiés, 359,75 s, 10,9 MB/s
sync
exit
```

Extra downloads

Make sure you have **SSL**.

And download **putty 0.67**: we'll use its utility psftp for file transfers.

c) Extra manipulation ? The serial port

Some say the serial port is mandatory with the headless config.

There is an explanation on the forum under

<https://parallella.org/forums/viewtopic.php?f=49&t=3289&p=16131&hilit=serial+console+how+to+turn+on#p16131>

Is it useful or not, as a beginner I cannot answer ; probably for FPGA though.

But it is good to see one way of modifying the initial image from your computer :)

So let's do it.

These are my words, adapted for Fedora:

1/ put your µSD card into your multi card reader

2/ run Gigolo or df or equivalent ; you retrieve the right name for your SD card:

```
/run/media/<user>/boot
```

3/ edit the device tree to add the serial port:

- if you don't have dtc, install it (sudo apt-get install device-tree-compiler or similar)

- mkdir test && cp /run/media/<user>/boot/devicetree.dtb test/devicetree.dtb

(keep the name 'devicetree': dtc uses it explicitly)

- cd test

- dtc -I dtb -O dts -o devicetree.dts devicetree.dtb

- edit devicetree.dts ; insert 'console=ttyPS0,115200 ' before the keyword 'root' inside the 'chosen' tag, or just cut/paste this:

```
chosen {
```

```
bootargs = "console=ttyPS0,115200 root=/dev/mmcblk0p2 rw earlyprintk
```

```
rootfstype=ext4 rootwait";
```

```
linux,stdout-path = "/amba@0/serial@e0001000";
```

```
};
```

- now go back:

```
dtc -I dts -O dtb -o devicetree.dtb devicetree.dts
```

```
cp devicetree.dtb /run/media/<user>/boot
```

There it is ! Serial port activated !

d) SSL, putty... Parallella boots !

It's probably not the best way (some people use DHCP for example), but it works for me.

Here is my context: I have no internet box, no switch... no nothing at home.

So I guarantee nothing at all - just be brave enough, or resign now.

You'll certainly lose the internet connection while you «talk» to Parallella.

Make a backup of your internet connection ; create a dedicated connection for Parallella... we'll see how.

And when you're done with Parallella, unplug/plug the ethernet cable, select the internet connection again... that's all.

We edit the SD image, /run/media/<user>/rootfs/etc/network/interfaces.d/eth0 :

```
#save it -- auto eth0
```

```
#save it -- iface eth0 inet dhcp
```

```
allow-hotplug eth0
```

```
iface eth0 inet static
```

```
address 10.0.2.15
```

```
netmask 255.255.255.0
```

and we create a specifig «network connection» through the Fedora menu, Preferences /

Network connection:

```
10.0.2.1
```

```
255.255.255.0
```

There's another file to modify, on your computer this time: /etc/hosts

Add this line:

```
10.0.2.15 parallella
```

FYI hostname is 'parallella'

Power the fan, plug the Ethernet cable, plug the SD card, last plug the 5V/2A power.

Open a terminal ; the initial boot is 1 or 2 minutes, other boots take about 15 s to answer the ping command :)

```
ping 10.0.2.15
```

For ssh, remember we use the '**parallella**' user and password, and this has **no root privilege** for security purpose.

You can automate many things if you like - I do ! Linux is great.

```
ssh 10.0.2.15 -l parallella
```

```
ssh parallella@10.0.2.15
```

You can now browse the Linux distro run by the ARM processor, give commands WITH your regular keyboard and your monitor to control what is done :)

For example, inside a SSH session you can run the hello example and you look at the results on your own screen. Your good old monitor.

```
parallella@parallella:~/epiphany-examples/apps/hello-world$ ll
total 52
drwxrwxr-x 4 parallella parallella 4096 Jan 30 01:39 ./
drwxrwxr-x 18 parallella parallella 4096 Jan 16 16:24 ../
-rw-rw-r-- 1 parallella parallella 13483 Jan 16 16:24 .cproject
-rw-rw-r-- 1 parallella parallella 2436 Jan 16 16:24 .project
drwxrwxr-x 3 parallella parallella 4096 Jan 31 12:33 Debug/
-rwxrwxr-x 1 parallella parallella 799 Jan 31 12:21 build.sh*
-rwxrwxr-x 1 parallella parallella 775 Jan 30 01:39 build.sh.sav*
-rwxrwxr-x 1 parallella parallella 51 Jan 16 16:24 run.sh*
-rwxrwxr-x 1 parallella parallella 51 Jan 30 01:39 run.sh.sav*
drwxrwxr-x 2 parallella parallella 4096 Jan 31 14:26 src/
parallella@parallella:~/epiphany-examples/apps/hello-world$
parallella@parallella:~/epiphany-examples/apps/hello-world$ ./build.sh
parallella@parallella:~/epiphany-examples/apps/hello-world$ ./run.sh
0: Message from eCore 0x8ca ( 3, 2): "Hello World from core 0x8ca, my_row 0x00, my_col 0x00 !"
1: Message from eCore 0x84b ( 1, 3): "Hello World from core 0x84b, my_row 0x00, my_col 0x00 !"
2: Message from eCore 0x84b ( 1, 3): "Hello World from core 0x84b, my_row 0x00, my_col 0x00 !"
3: Message from eCore 0x888 ( 2, 0): "Hello World from core 0x888, my_row 0x00, my_col 0x00 !"
4: Message from eCore 0x849 ( 1, 1): "Hello World from core 0x849, my_row 0x00, my_col 0x00 !"
```

```

5: Message from eCore 0x88b ( 2, 3): "Hello World from core 0x88b, my_row 0x00,
my_col 0x00 !"
6: Message from eCore 0x88b ( 2, 3): "Hello World from core 0x88b, my_row 0x00,
my_col 0x00 !"
7: Message from eCore 0x8ca ( 3, 2): "Hello World from core 0x8ca, my_row 0x00, my_col
0x00 !"
8: Message from eCore 0x80a ( 0, 2): "Hello World from core 0x80a, my_row 0x00, my_col
0x00 !"
9: Message from eCore 0x808 ( 0, 0): "Hello World from core 0x808, my_row 0x00,
my_col 0x00 !"
10: Message from eCore 0x8c8 ( 3, 0): "Hello World from core 0x8c8, my_row 0x00,
my_col 0x00 !"
11: Message from eCore 0x8c9 ( 3, 1): "Hello World from core 0x8c9, my_row 0x00,
my_col 0x00 !"
12: Message from eCore 0x88a ( 2, 2): "Hello World from core 0x88a, my_row 0x00,
my_col 0x00 !"
13: Message from eCore 0x88b ( 2, 3): "Hello World from core 0x88b, my_row 0x00,
my_col 0x00 !"
14: Message from eCore 0x8cb ( 3, 3): "Hello World from core 0x8cb, my_row 0x00,
my_col 0x00 !"
15: Message from eCore 0x84a ( 1, 2): "Hello World from core 0x84a, my_row 0x00,
my_col 0x00 !"
16: Message from eCore 0x88a ( 2, 2): "Hello World from core 0x88a, my_row 0x00,
my_col 0x00 !"
17: Message from eCore 0x84b ( 1, 3): "Hello World from core 0x84b, my_row 0x00,
my_col 0x00 !"
18: Message from eCore 0x848 ( 1, 0): "Hello World from core 0x848, my_row 0x00,
my_col 0x00 !"
19: Message from eCore 0x8ca ( 3, 2): "Hello World from core 0x8ca, my_row 0x00,
my_col 0x00 !"
parallella@parallella:~/epiphany-examples/apps/hello-world$

```

What's next ?

Well, you're the oil king now ! in French «le roi du pétrole !» ;)

You are **powerful**:

- you edit the batch or even your source files directly -- nano ./build.sh

- or you work with your standard editor from your computer,

automatically transfer your source files to a folder under the Parallella :

psftp parallella@10.0.2.15 -pw parallella -b psftp.txt

(psftp --help explains everything, the psftp.txt is an optional batch of commands to automate what you want ; you can put any Putty command inside ; yes... psftp and Putty are the same software)

There is a lot of **open information** on almost anything on this exciting Parallella platform.

It's Andreas Olofsson's will:

«For up to date board information, please refer to:

<http://github.com/parallella/parallella-hw>

All support for this board is provided via community support at

<http://forums.parallella.org>»

OK. You learned the hardware/software basics for booting your Parallella.

In tutorial #2 we'll begin with a more advanced «Hello world» example.

I'll share what is a useful (*) **canvas for beginners** like me, interested in

SPMD programming.

The source code for this second tutorial is available under

<https://github.com/DonQuichotteComputers/paralle2>

(*) At least for me :P

Don Quichotte, 2016/04/29