Présentation de la plateforme Pynq-Z2

INF3500
Conception et réalisation de systèmes numériques

POLYTECHNIQUE
MONTRÉAL

3 types de plateformes pour embedded

Raspberry PI

Desktop Linux on Arm Microprocessors

Arduino

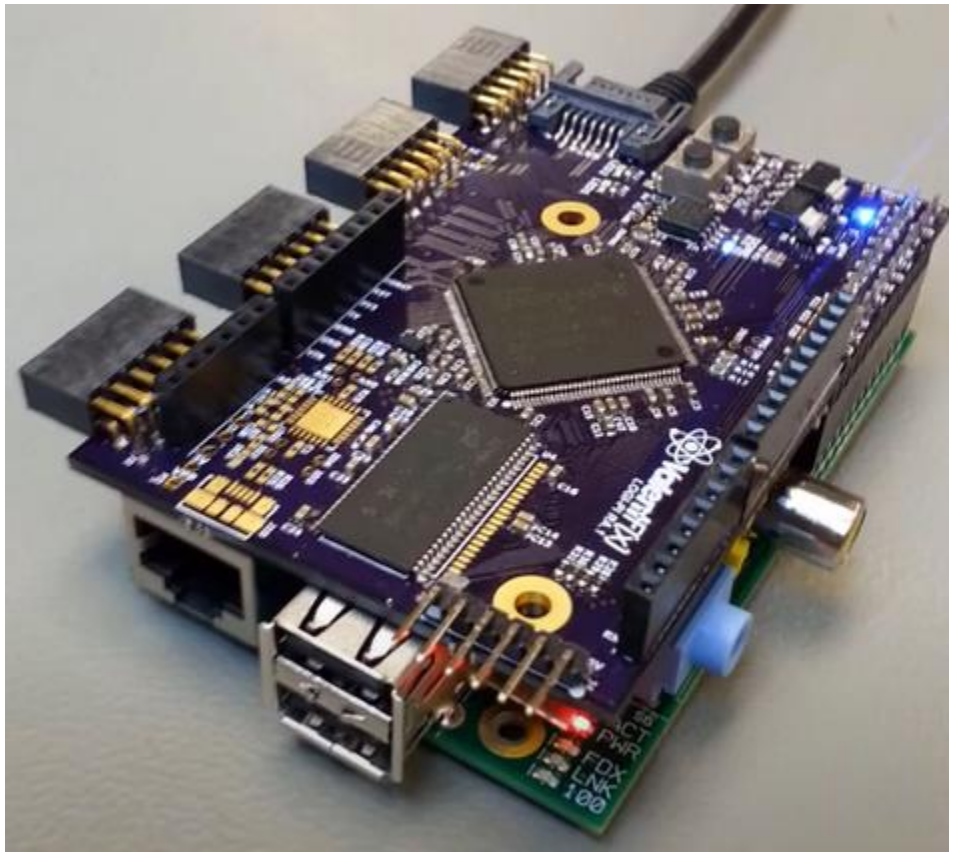
Low-level, ‘bit-banging’ microcontroller

FPGA

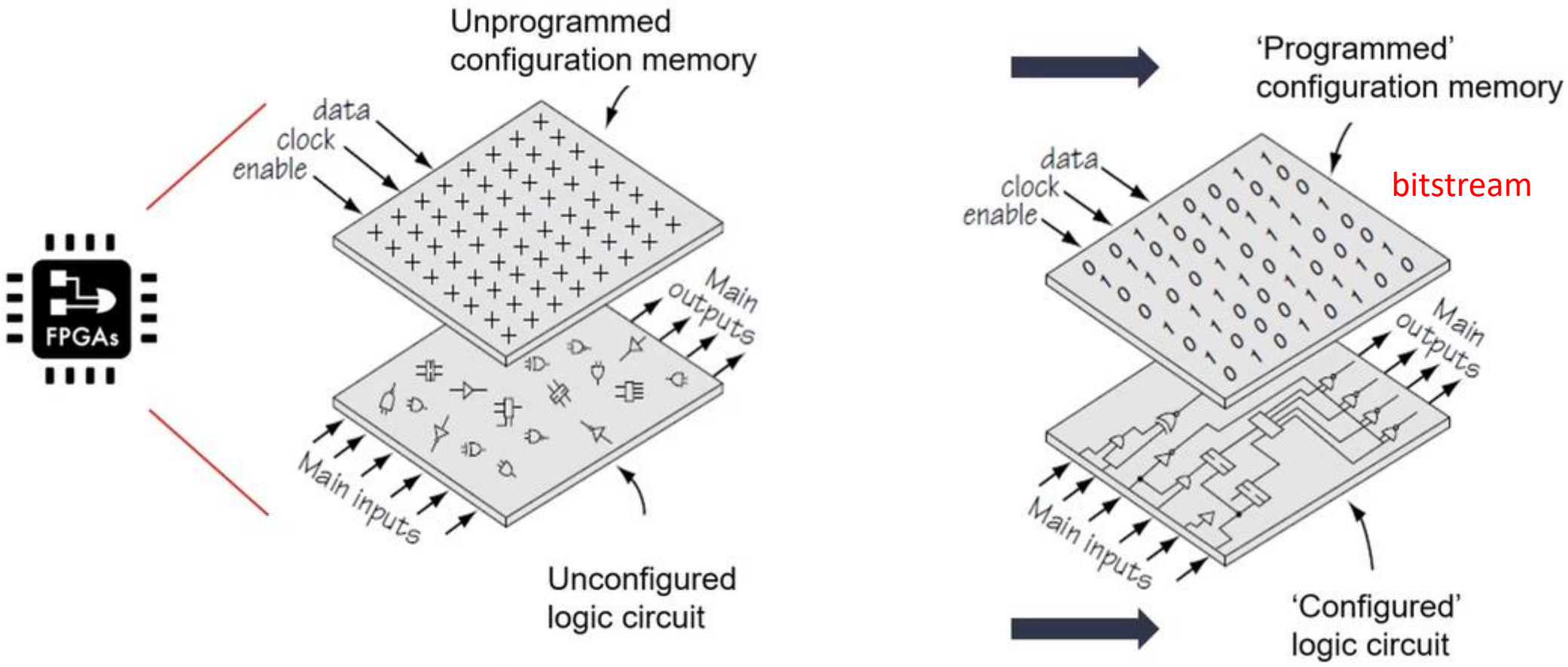
Field Programmable Gate Arrays

Fast, parallel, customizable logic

Ras-pi with FPGA Hat

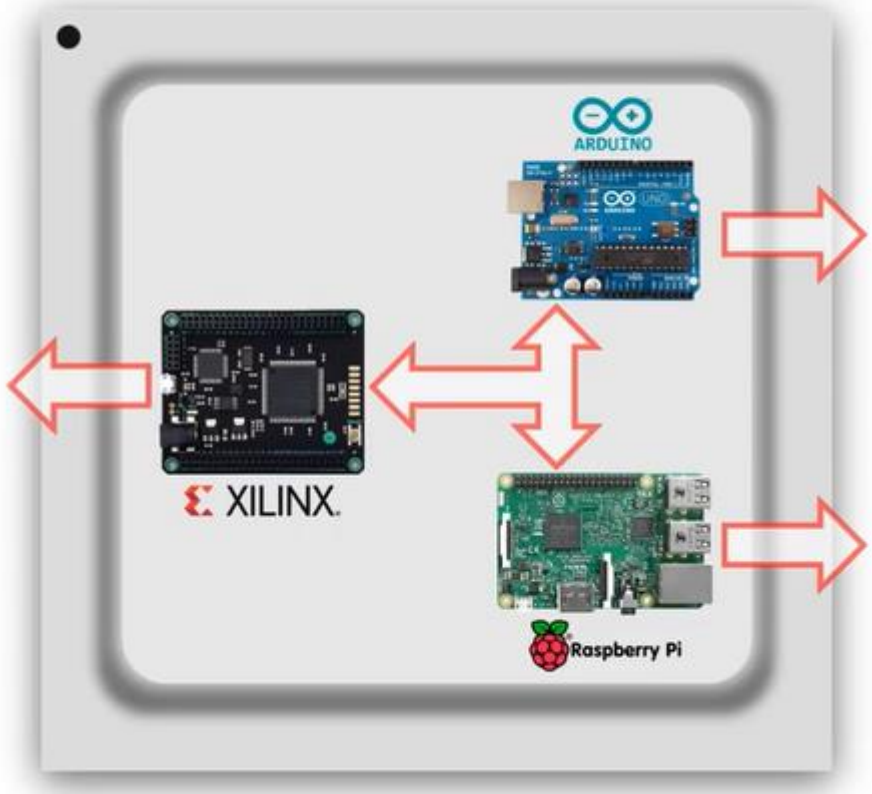


Représentation d'un FPGA



Credit: 'Bebop to the Boolean Boogie: An Unconventional Guide to Electronics'

Zynq : Intégration Microprocessors, Microcontrollers and Programmable Logic



Systems-on-Chip integration

Zynq Programmable Platform integrate

- Arm microprocessors
- Programmable logic (FPGA)
- High-speed, programmable IO
- As many ‘soft’ microcontrollers as needed
- Fast connections between components

Comparaison of plateformes

	Raspberry PI	Arduino	Zynq
Arm Application Microprocessor	YES	NO	YES
Real-time Microcontroller	NO	YES	YES*
Integrated Programmable Logic	NO	NO	YES
Linux	YES	NO	YES
CPython Ecosystem	YES	NO**	YES

* Multiple soft microcontrollers in programmable logic

** More limited MicroPython and CircuitPython options are available

Applications

Typical Application Areas

> Hardware-accelerated algorithms

> Precision robotics

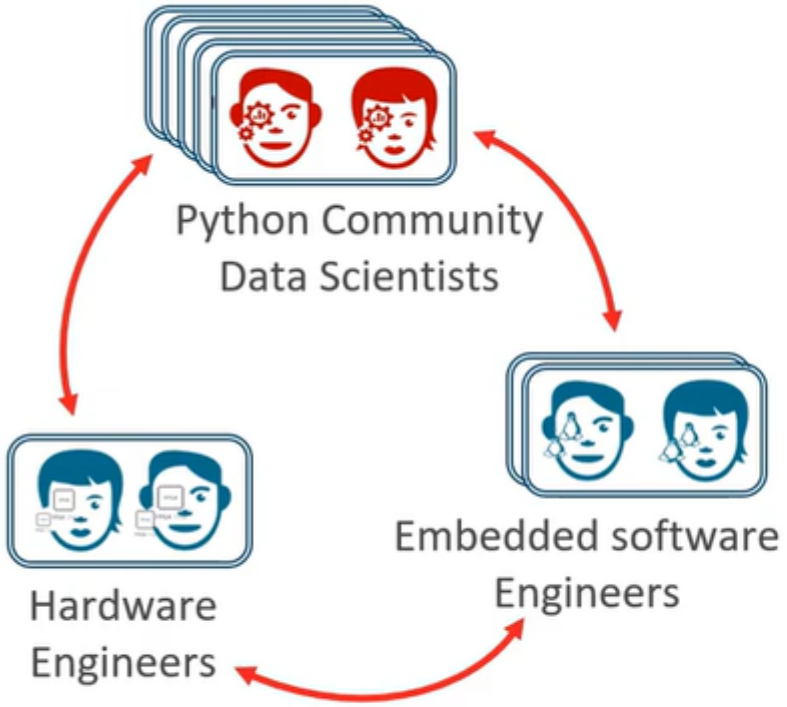
> Real-time, high-resolution video processing

> State-of-the-art instrumentation

> Unique, highly-differentiated designs in research and spin-offs

> Teaching: logic design, computer architecture, digital signal processing, control, projects

Collaboration



New users can be Python programmers, Data Scientists and domain experts of all kinds

PYNQ™

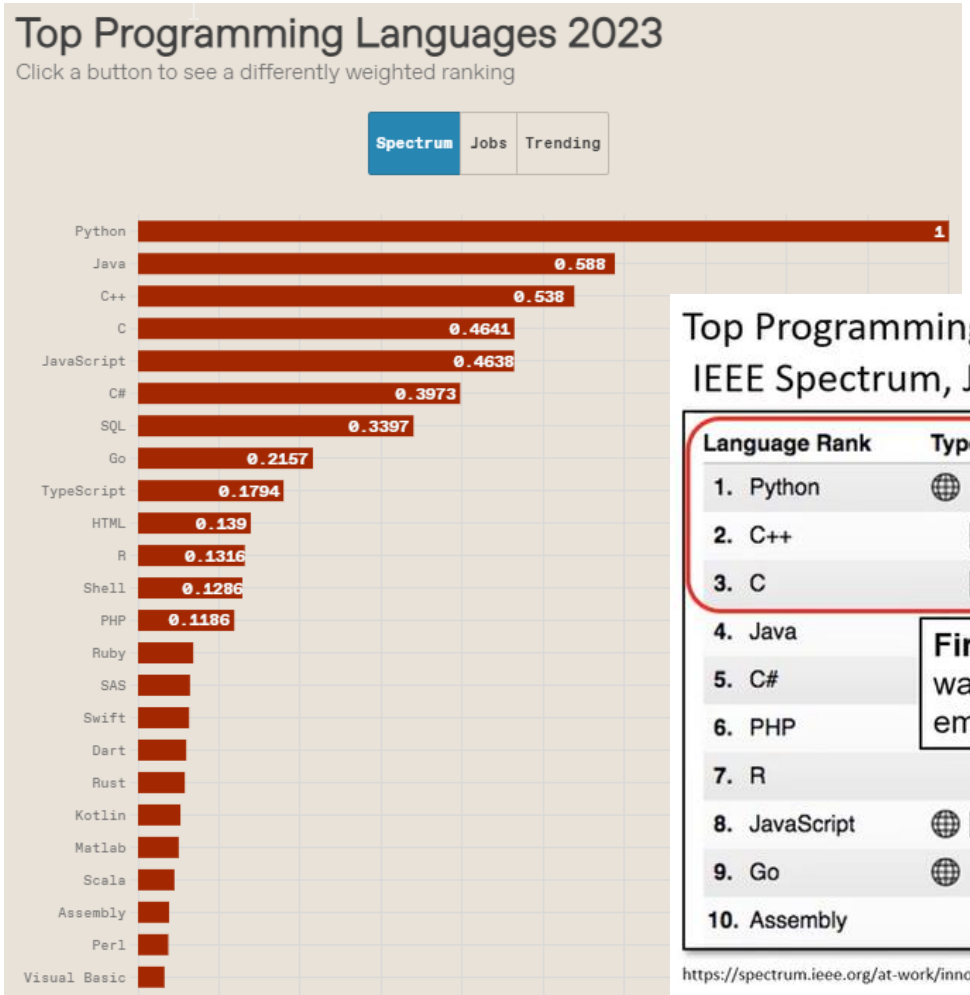
A Pythonic Framework that gives Python developers access to the benefits of programmable platforms

AND

A Pythonic methodology for hardware designers to:

- *make them more productive*
- *make their designs accessible to more people*

Python



Top Programming Languages,
IEEE Spectrum, July'18

Language Rank	Types
1. Python	
2. C++	
3. C	
4. Java	
5. C#	
6. PHP	
7. R	
8. JavaScript	
9. Go	
10. Assembly	

First time that Python was listed as an embedded language

https://spectrum.ieee.org/at-work/innovation/the-2018-top-programming-languages

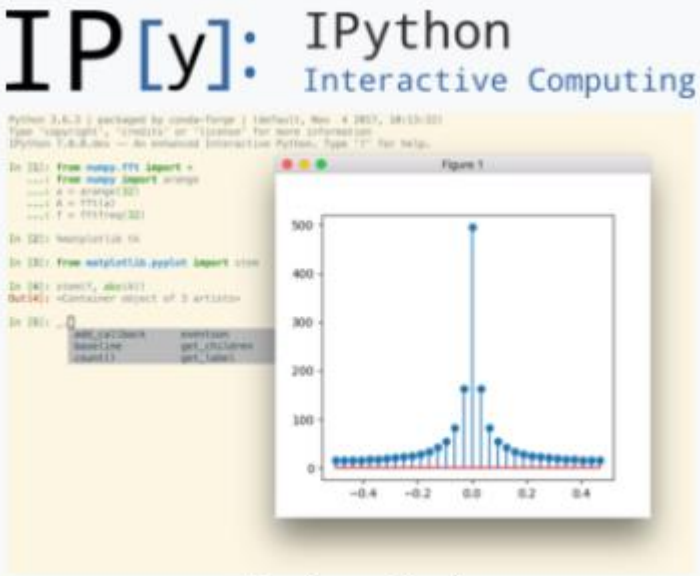
Benefits of Python in Embedded/Edge Apps

- Millions of developers
- On-target development
- Rapid iteration cycles
- Huge ecosystem and community
- Interoperability with C/C++
- Agile hardware & software codesign
- Portable code

IPython (Interactive Python)

IPython (Interactive Python) est un interpréteur de commandes pour l'informatique interactive dans plusieurs langages de programmation, développé à l'origine pour le langage de programmation Python, qui offre l'introspection, les médias riches, la syntaxe du shell, la complétion par tabulation et l'historique. IPython fournit les fonctionnalités suivantes :

- Shells interactifs (terminaux et basés sur Qt).
- Une interface de notebook basée sur un navigateur prenant en charge le code, le texte, les expressions mathématiques, les tracés en ligne et d'autres médias.
- Prise en charge de la visualisation interactive des données et utilisation de boîtes à outils GUI.
- Des interprètes flexibles et intégrables à intégrer dans vos propres projets.
- Outils pour *parallel computing*.



IPython Shell

IPython Notebook to to Jupyter Notebook

Julia Python R

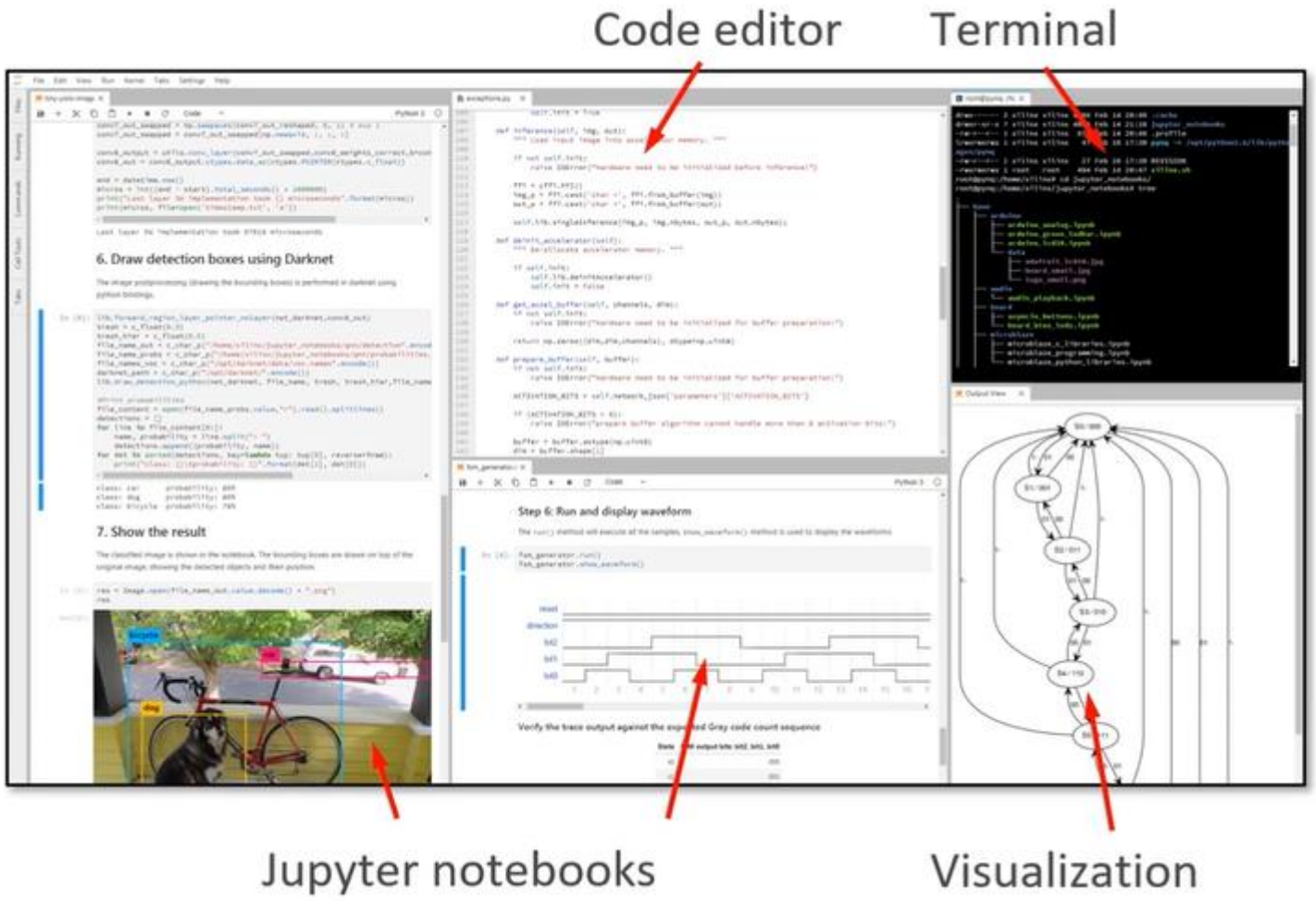
jupyter

Default engine of data science, machine learning and AI

7 million notebooks on GitHub
Usage growing exponentially

Taught to 100,000's of college students every semester

Jupyter Notebook to JupyterLab IDE



Next-generation Integrated Development Environment

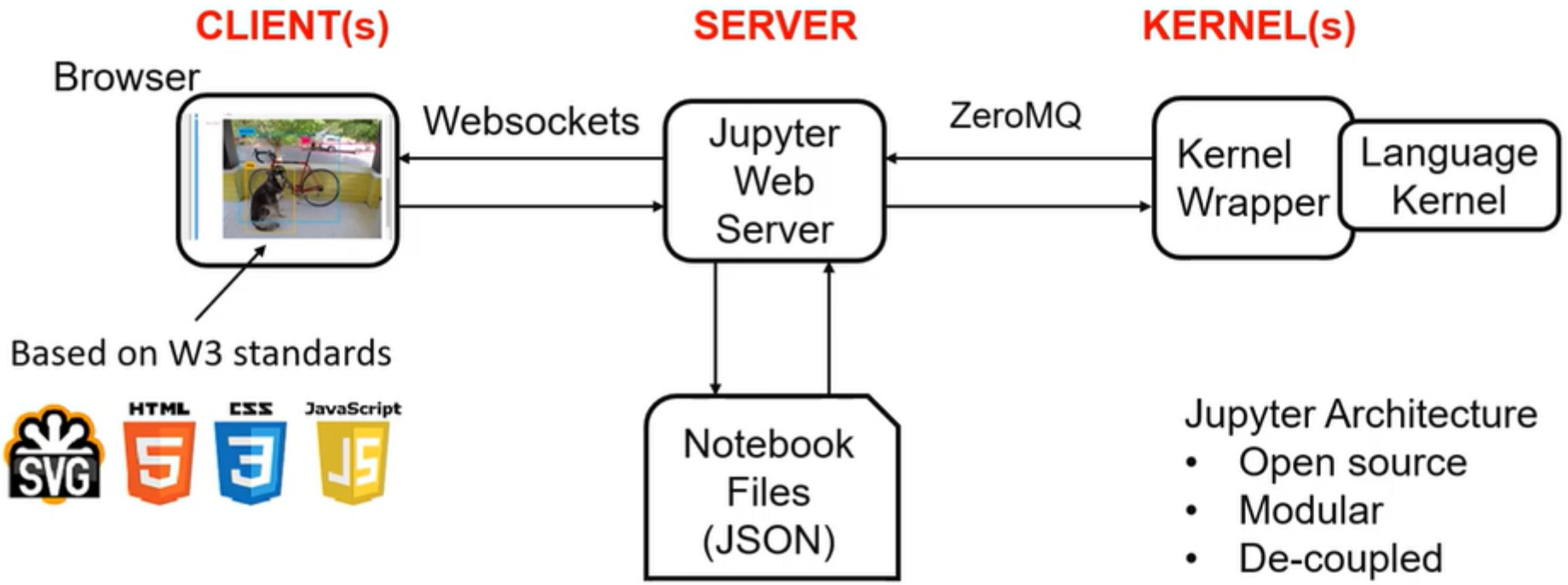
Browser-based GUI

Multiple re-sizable frames in one browser window

Completely extensible

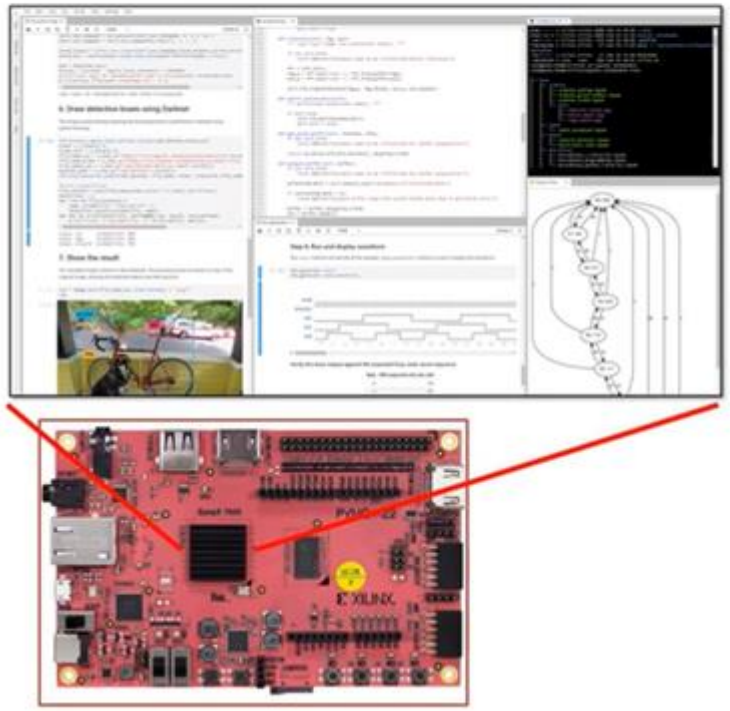
Jupyter Notebooks are one of many plug-ins in JupyterLab

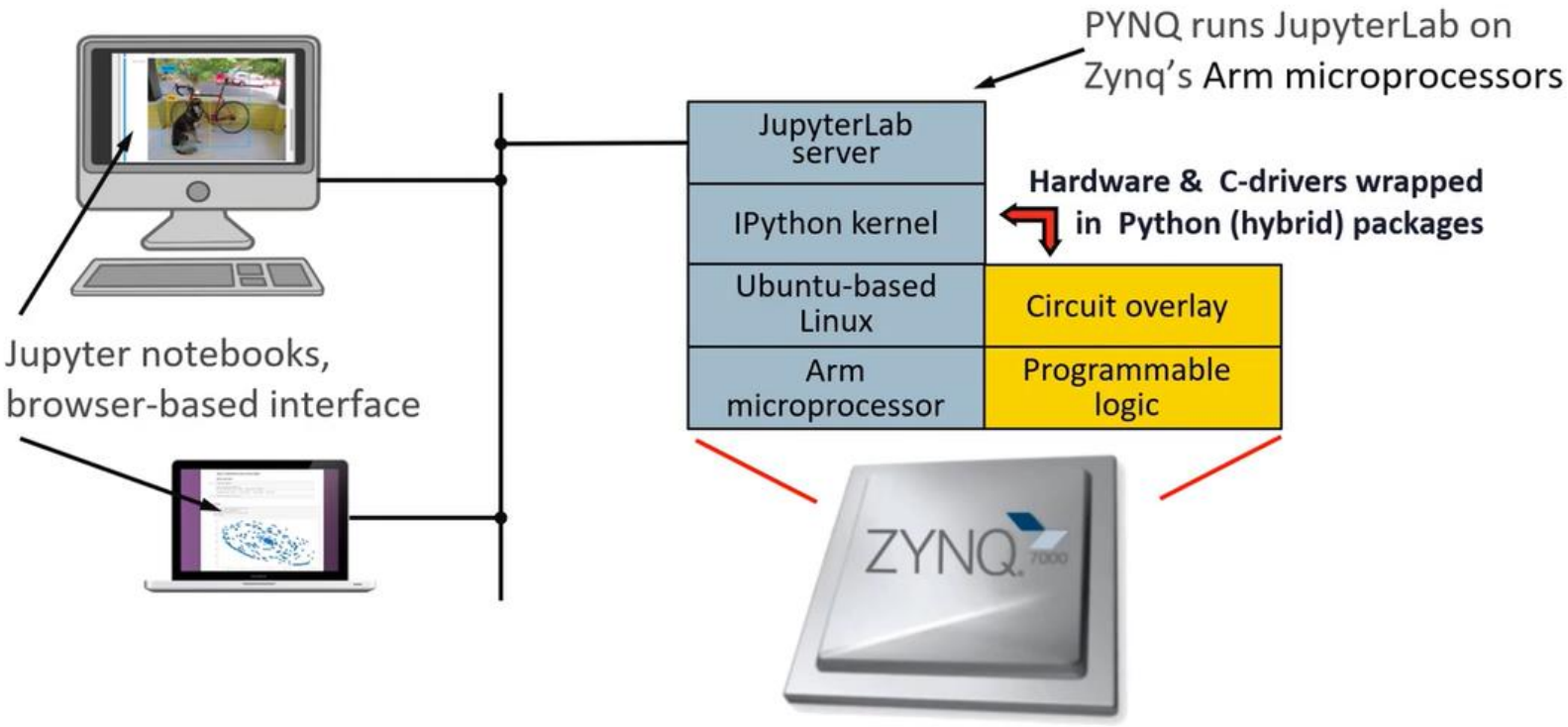
Architecture Jupyter’s client-server-kernel



Pynq Architecture

PYNQ uses Jupyter's server architecture to host a browser interface to an integrated development environment (IDE) on Zynq





Wraps FPGA Circuit & Drivers in Python Package

Vidéo présentant la carte

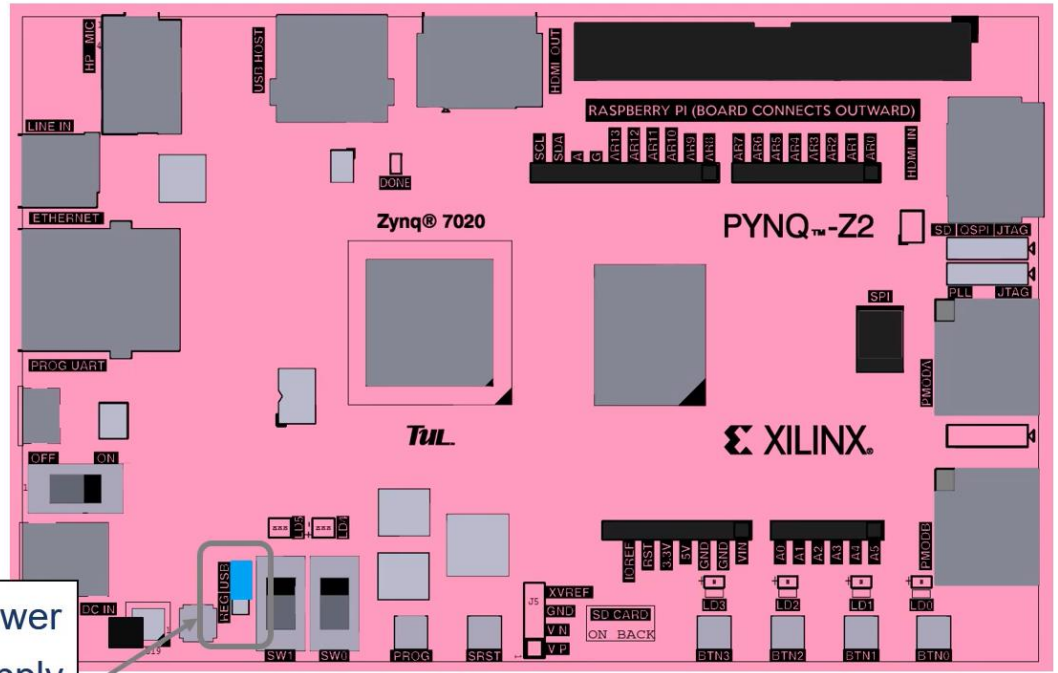
- Lien pour partir la carte la première fois : https://youtu.be/RiFbRf6gaK4?si=i4iXrlpsfk04iEhY

Pour flasher la carte SD

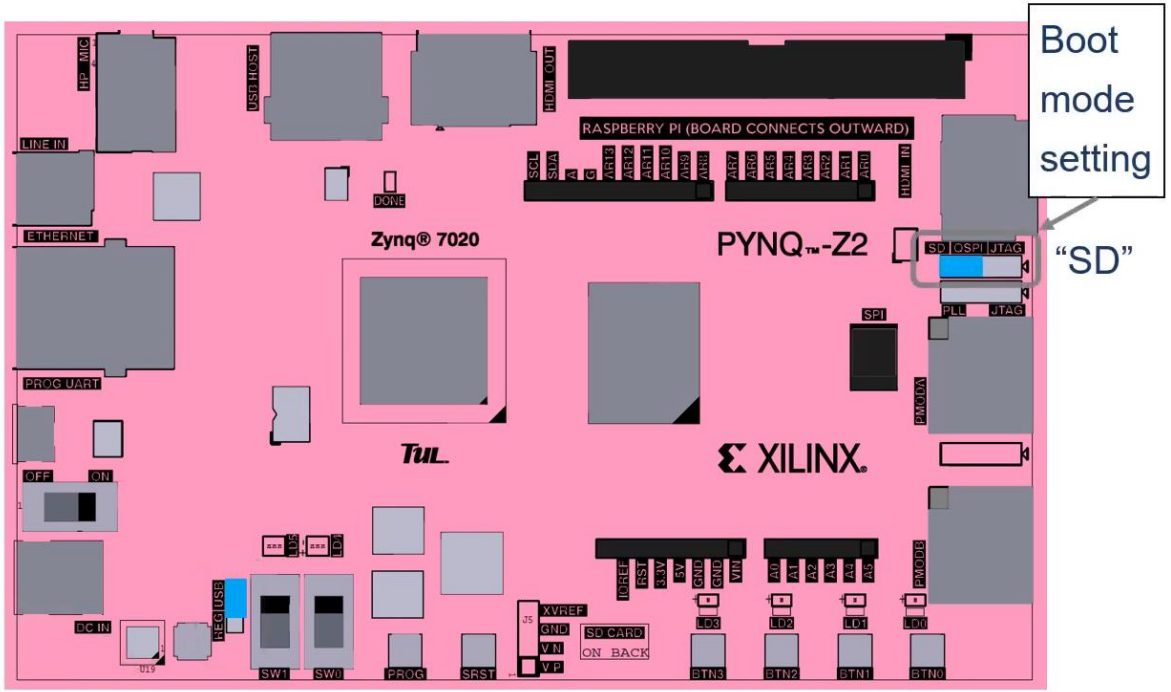
- Télécharger l'image dans : https://bit.ly/pynqz2_v3_0_1
- Brancher votre lecteur de carte SD (avec la carte à l'intérieur)
- Utiliser logiciel balenaEtcher ou Win32DiskImager pour télécharger l'image sur la carte.

Pour premier allumage : configuration des cavaliers

Utiliser le câble USB à micro-usb

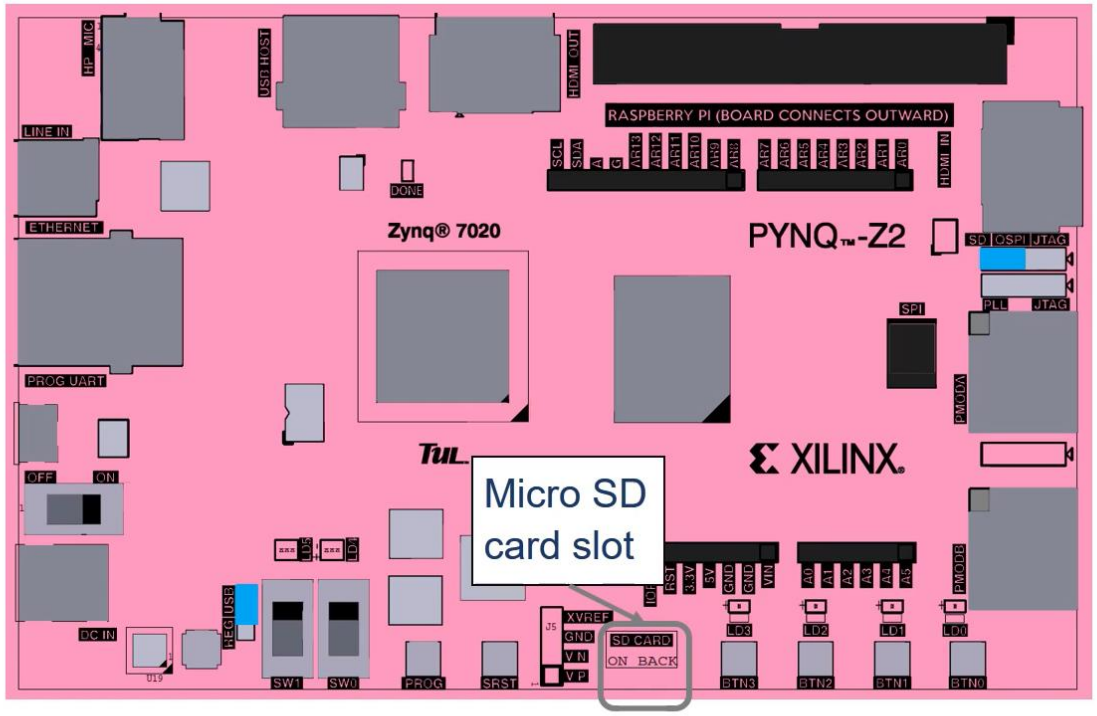


Utiliser la carte SD



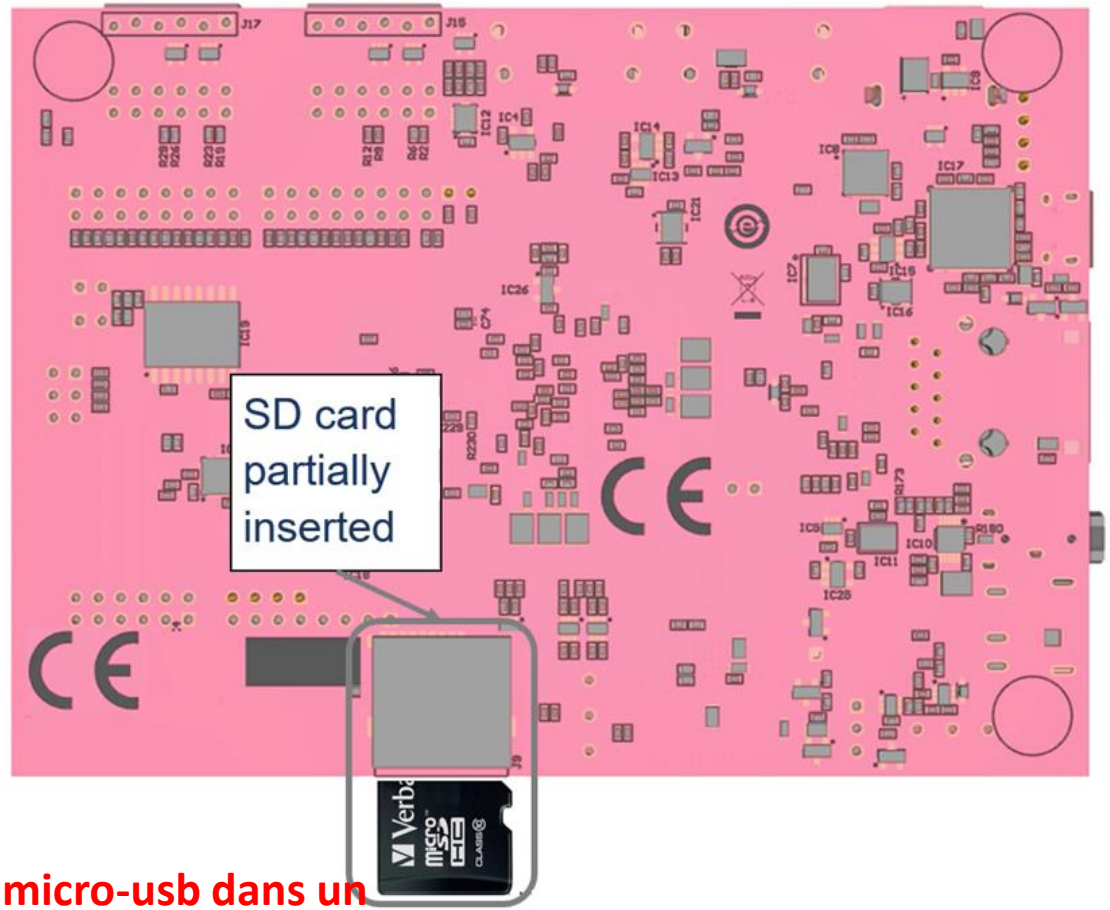
Pour premier allumage : prépare et brancher la carte SD

Préparer la carte SD avec une image de base



Go to pynq.io for instructions on how to prepare a Micro SD card

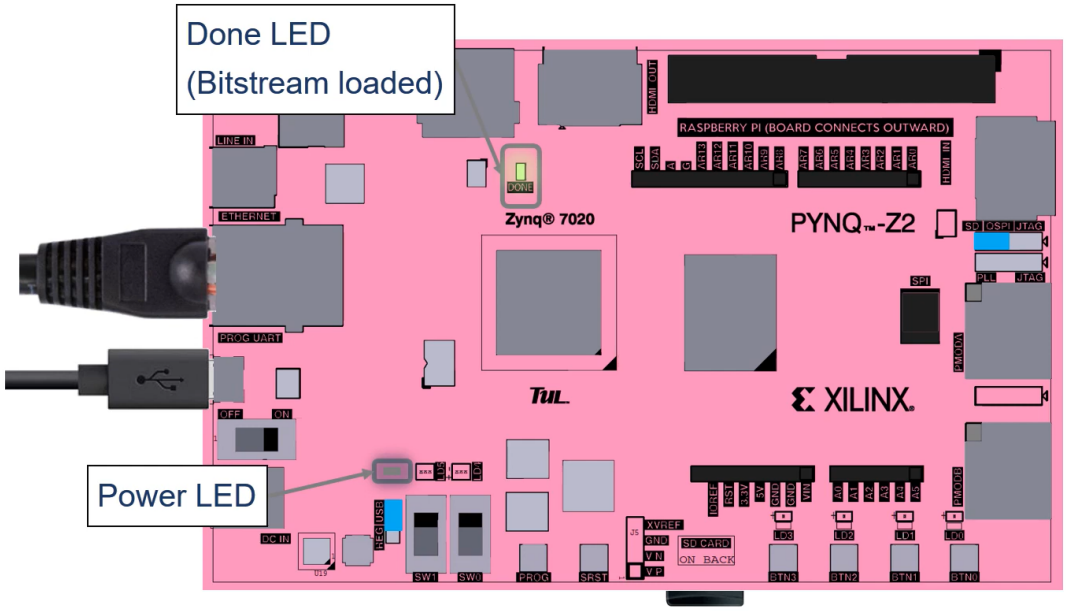
Brancher la carte SD

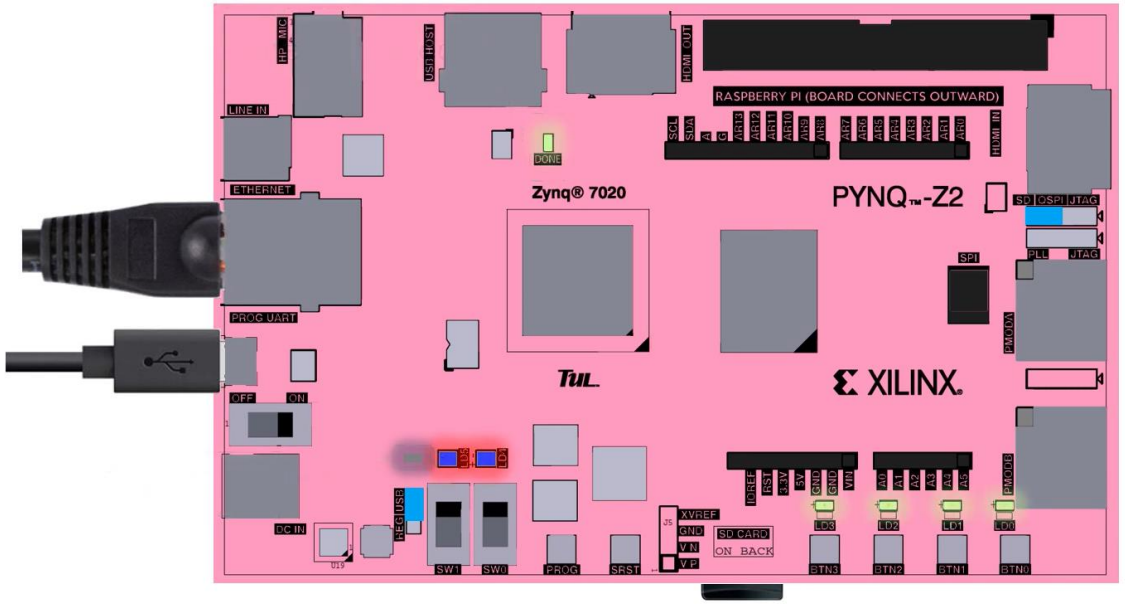


Vous pouvez maintenant connecter la câble micro-usb dans un prise USB pour allumer la carte

Pour premier allumage : séquence de lumières

Attention, le processus de Boot peut être long.

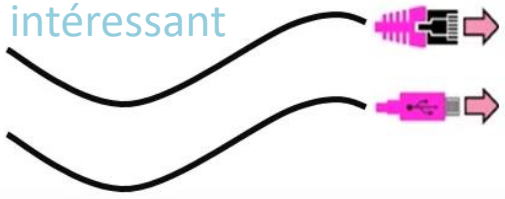




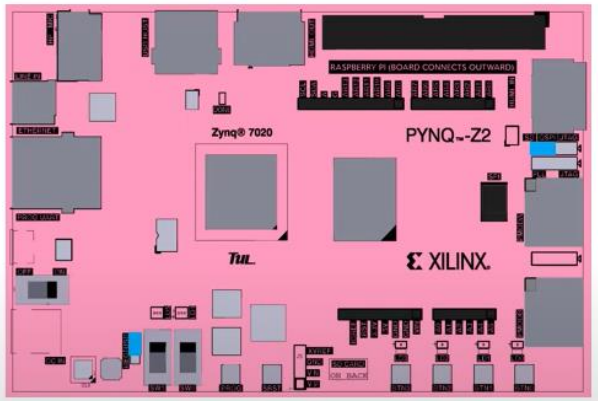
Connexion à Jupyter par http

1

Utilisation d'un câble croisé peut être intéressant

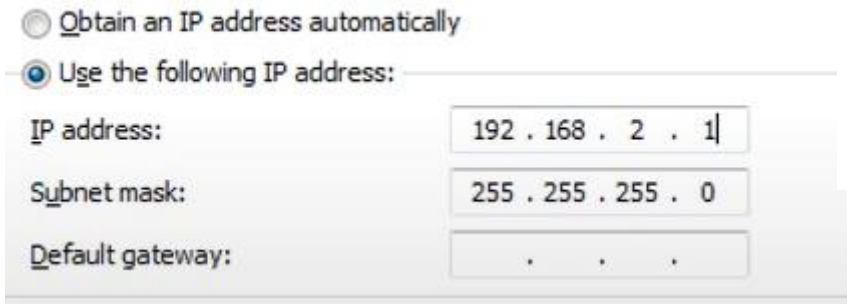


L'IP statique par défaut de la Pynq est : 192.168.2.99 (port 9090)



2

Configurer la carte réseau de l'ordinateur comme suit

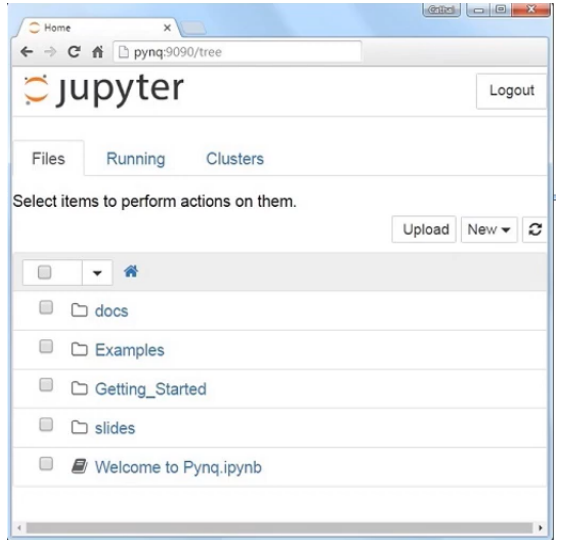


3

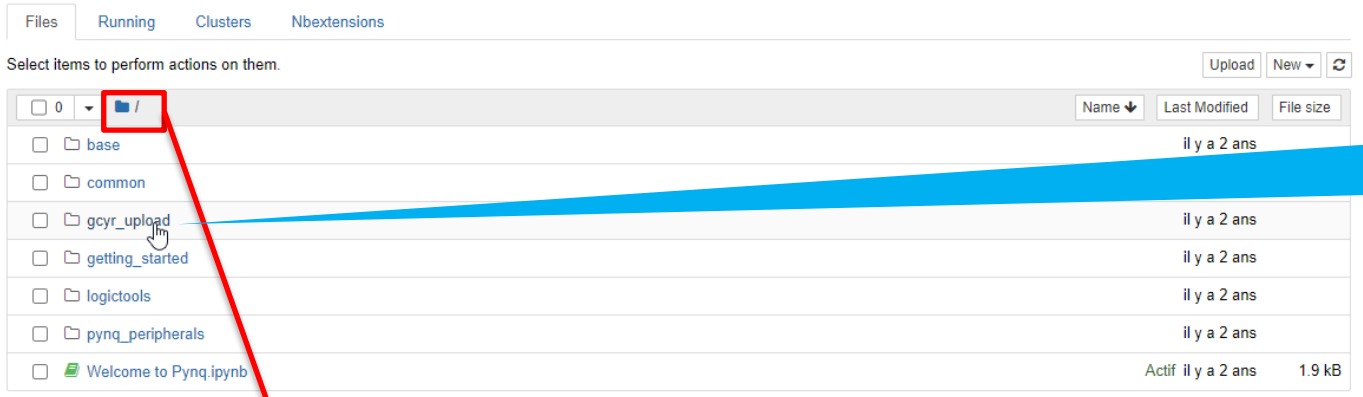
Sur un navigateur WEB :

- **For board connected to computer**
  - http://192.168.2.99:9090
- **Password: xilinx**

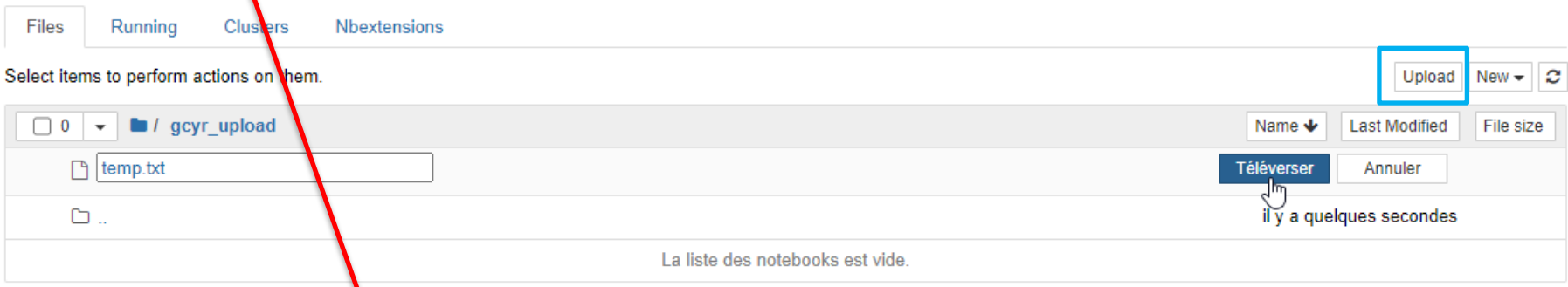
4



Transfert fichiers par http

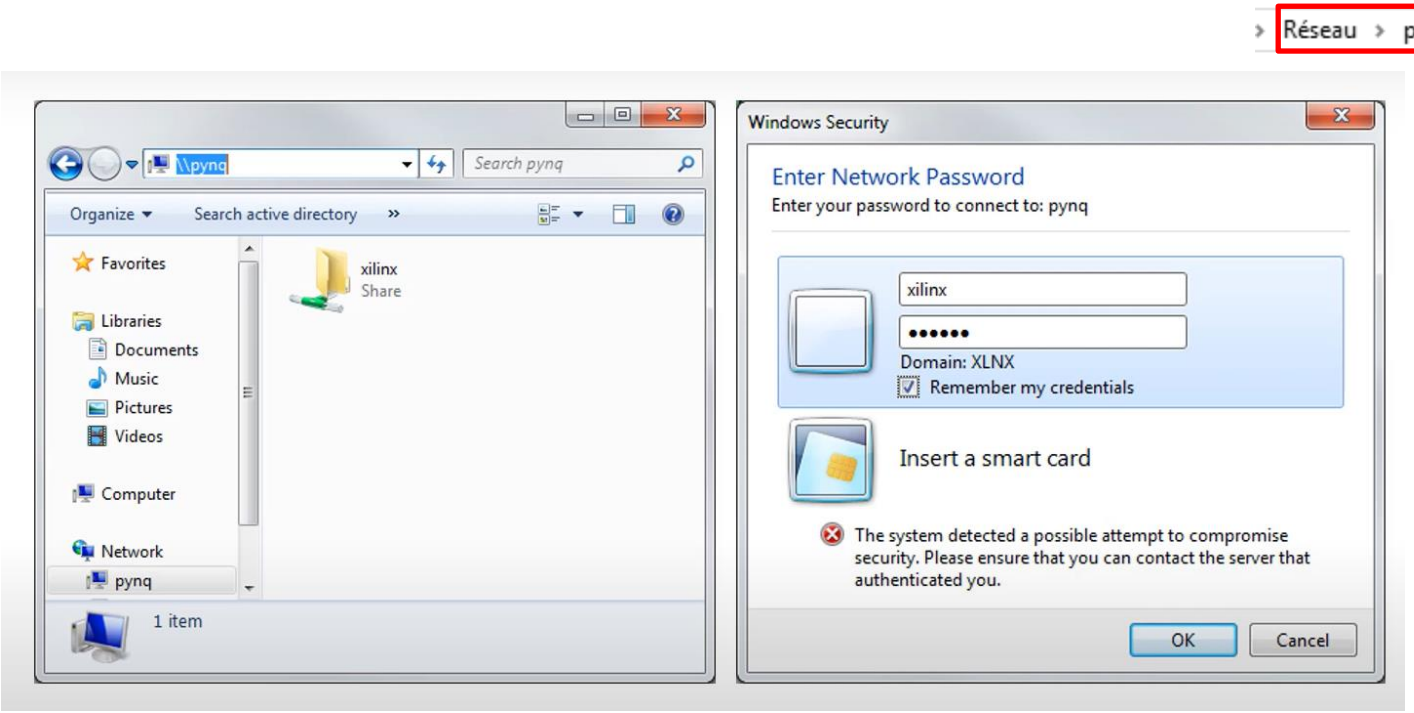


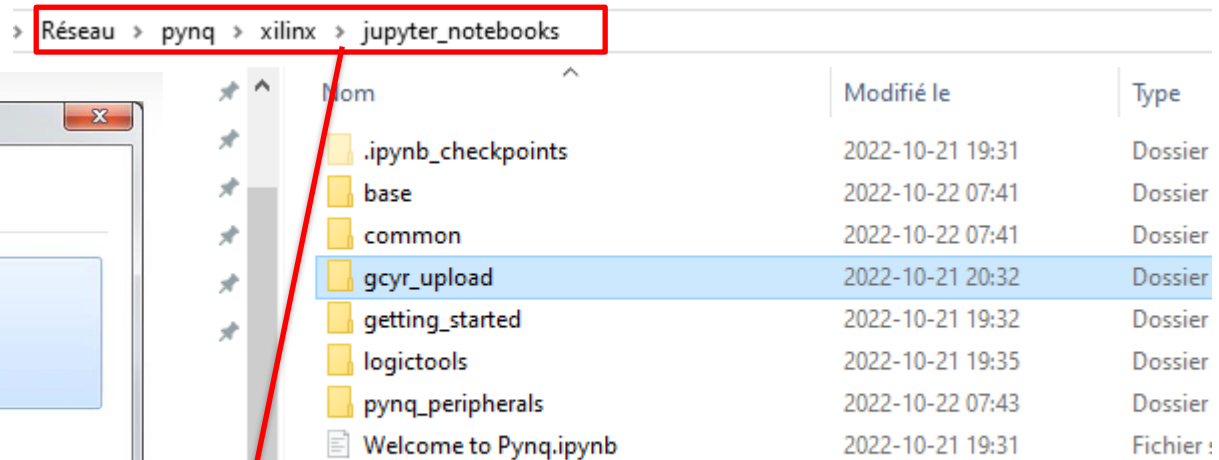
Répertoire crée pour mettre des scripts par exemple




```
root@pynq:/home/xilinx/jupyter_notebooks/gcyr_upload# ls
temp.txt
```

Transfert de fichiers par SAMBA

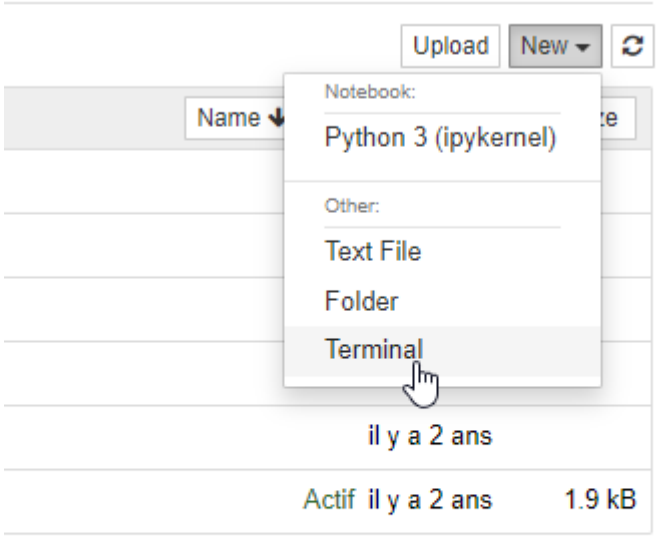


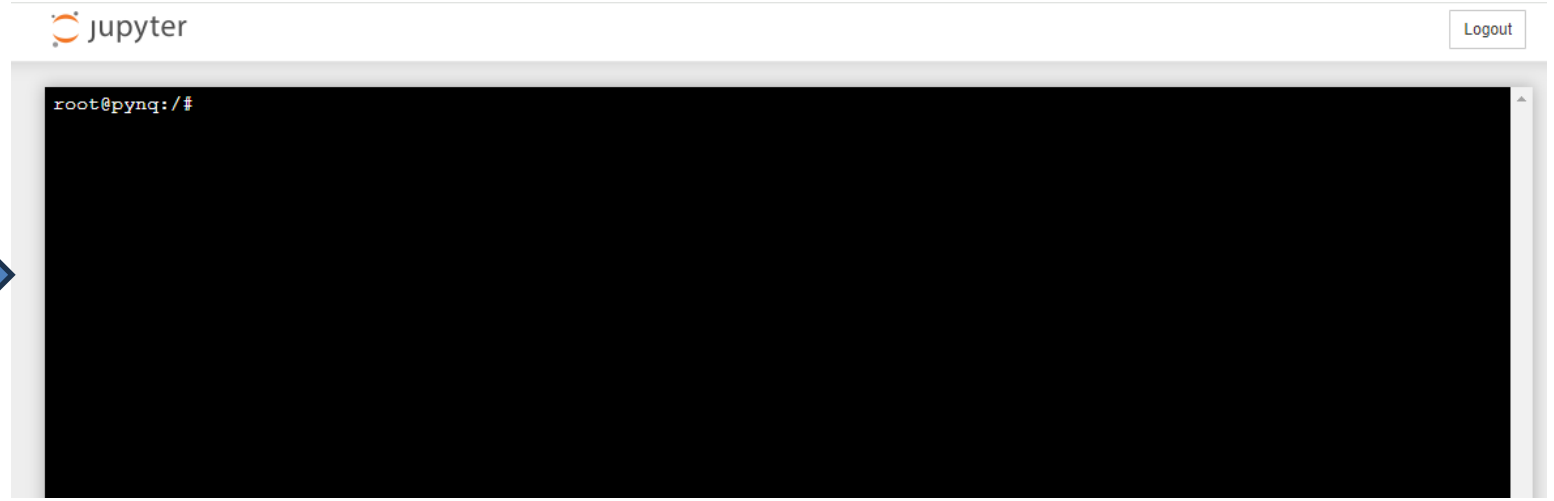



```
root@pynq:/home/xilinx/jupyter_notebooks/gcyr_upload# ls -l
total 12
-rw-r--r-- 1 root root 3845 Oct 22 00:36 compteur_gray.py
-rw-r--r-- 1 root root  112 Oct 22 00:36 eth0_bk
-rw-r--r-- 1 root root 2088 Oct 22 00:36 led_flash.py
-rw-r--r-- 1 root root    0 Oct 22 00:32 temp.txt
```

Ouverture d’un Terminal : par http

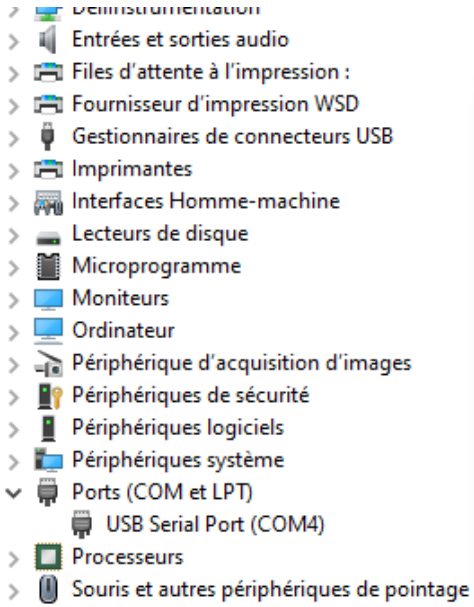
Avec un câble Ethernet

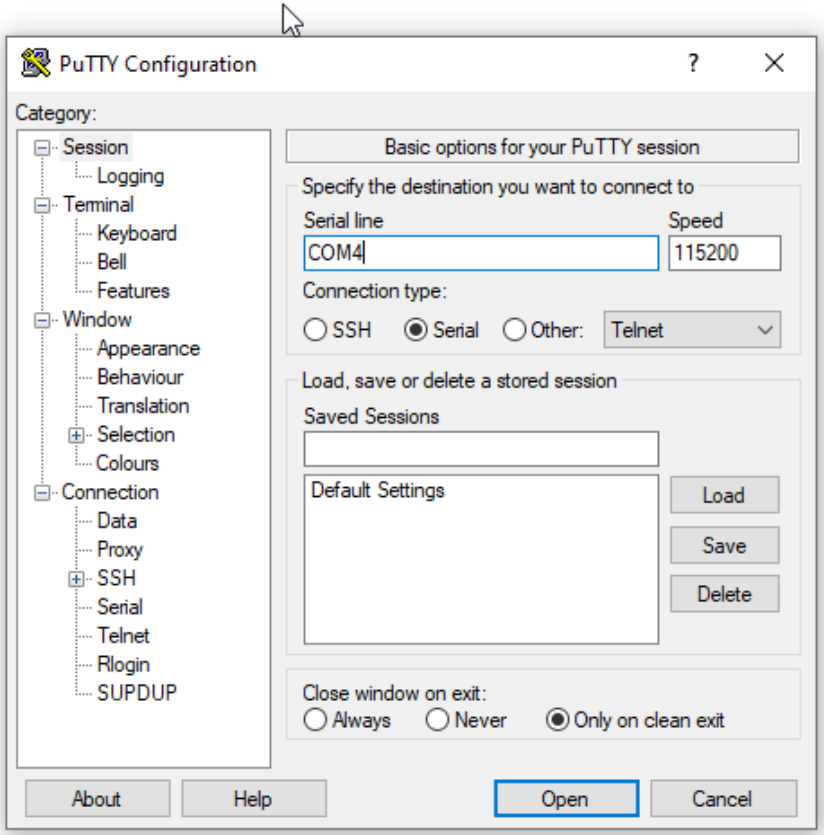


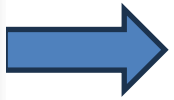


Ouverture d’un Terminal : par Putty

Avec le câble micro-usb à USB

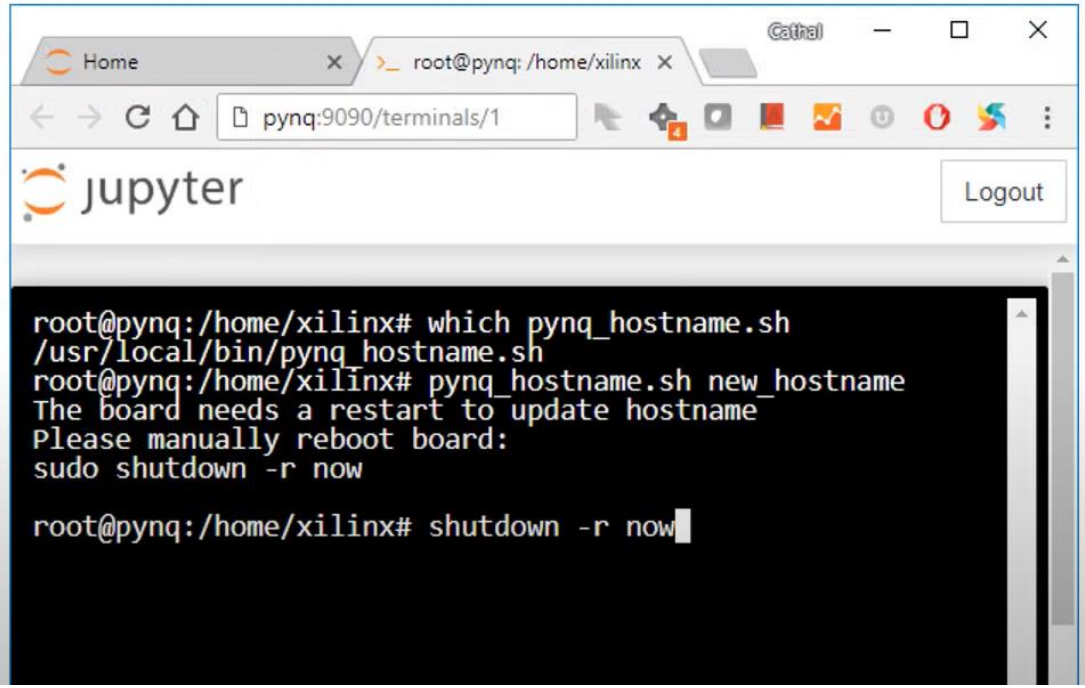






Pour changer le nom d’hôte

>hostname.sh new_hostname
>shutdown –r now



Pour voir les configurations IP