Python Productivity for Zynq

XILINX

© Copyright 2018 Xilinx

Outline

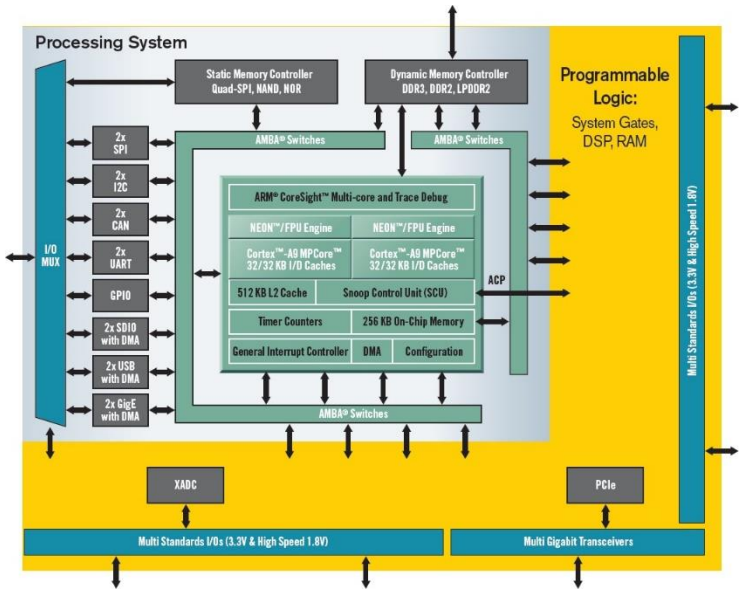
- Zynq & Zynq Ultrascale+
- PYNQ Framework
- Technologies
- Community

© Copyright 2018 Xilinx

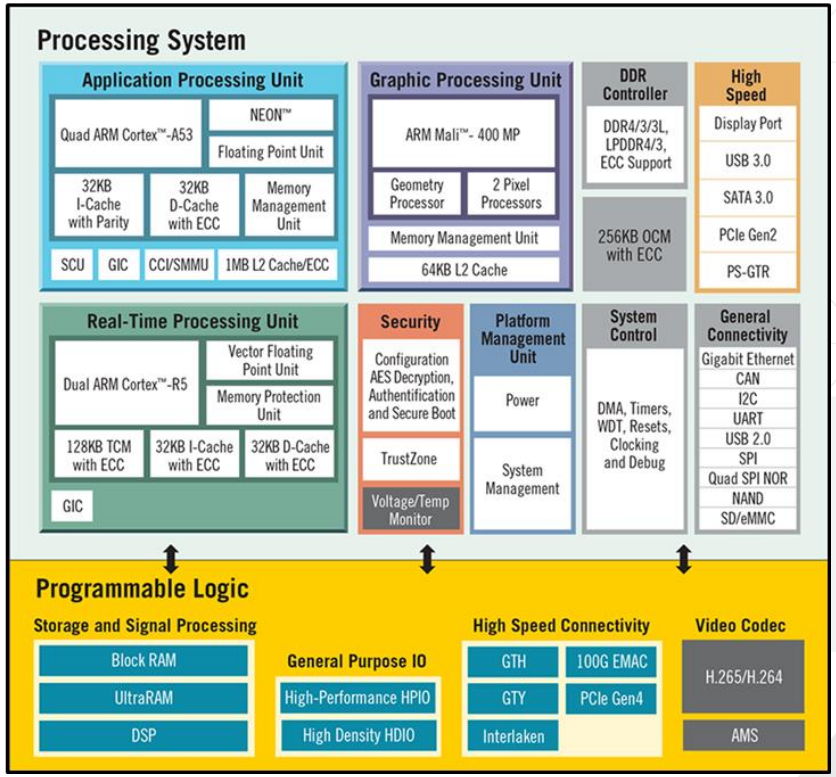
ZYNQ and ZYNQ UltraSCALE+

Best-in-class, All Programmable SoCs

ZYNQ 7000



ZYNQ UltraSCALE+



FPGAs and tightly-integrated CPUs enable entirely new opportunities

© Copyright 2018 Xilinx

XILINX.

Zynq applications

© Copyright 2018 Xilinx

XILINX.

PYNQ Vision

> **Make Zynq so easy-to-use that programmers can access the benefits of FPGAs without learning advanced digital design skills**

© Copyright 2018 Xilinx

PYNQ framework

XILINX

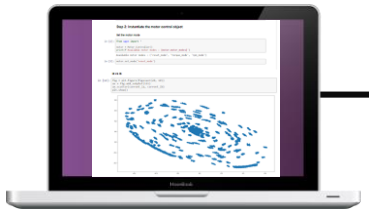
PYNQ

© Copyright 2018 Xilinx

Python Productivity for Zynq

Jupyter notebooks,
browser-based interface

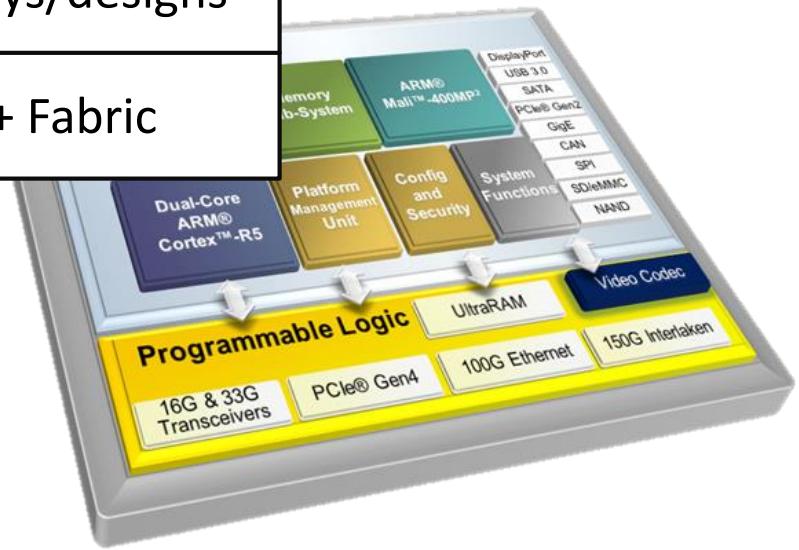
Most competitive, open
and advanced GUI



PYNQ enables JupyterLab
on Zynq and ZU+

Jupyter web server	
IPython kernel	
Ubuntu-based Linux	Overlays/designs
ARM A9 / A53	ZU+ Fabric

Hardware C-drivers wrapped in Python packages



© Copyright 2018 Xilinx

XILINX

Ubuntu-based Linux versus embedded Linux

Ubuntu-based Linux

Optimized for developer productivity

- All the Linux libraries and drivers you expect
- Pre-built SD card image
- Ubuntu/Debian ecosystem & community
  - >>145,000,000 Google hits

3 orders of magnitude difference

Embedded Linux

Optimized for deployment efficiency

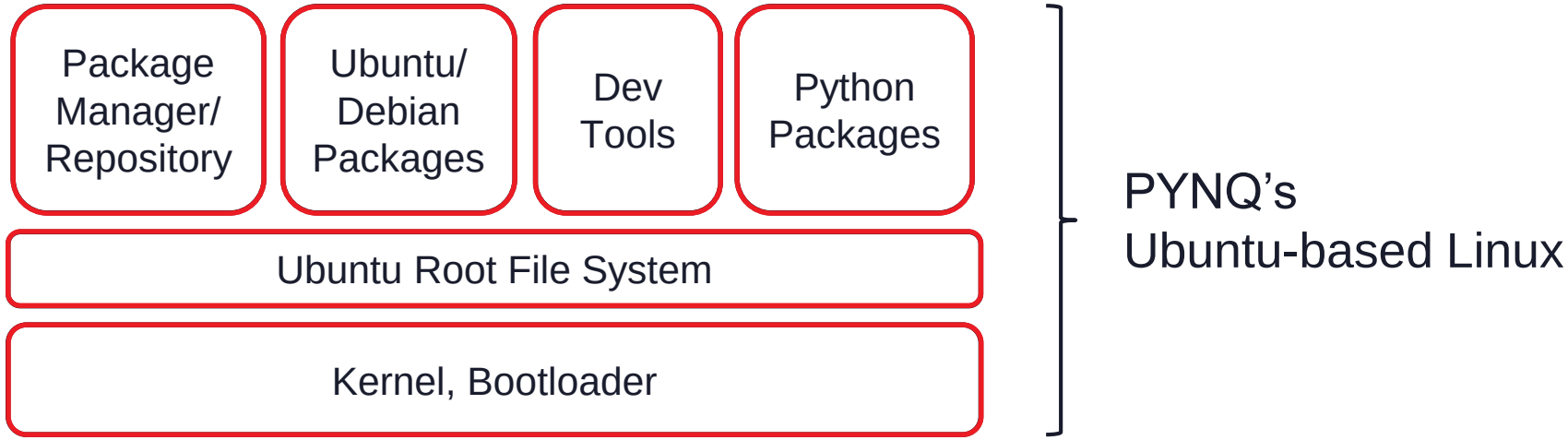
- Selective Linux libraries and drivers
- Commonly delivered in flash memory on board
- PetaLinux ecosystem:
  - >> 143,000 Google hits

© Copyright 2018 Xilinx

PYNQ's Ubuntu-based Linux

PYNQ uses Ubuntu's:
- Root file system (RFS)
- Package manager (*apt-get*)
- Repositories

PYNQ bundles :
- Development tools
  - Cross-compilers
- Latest Python packages



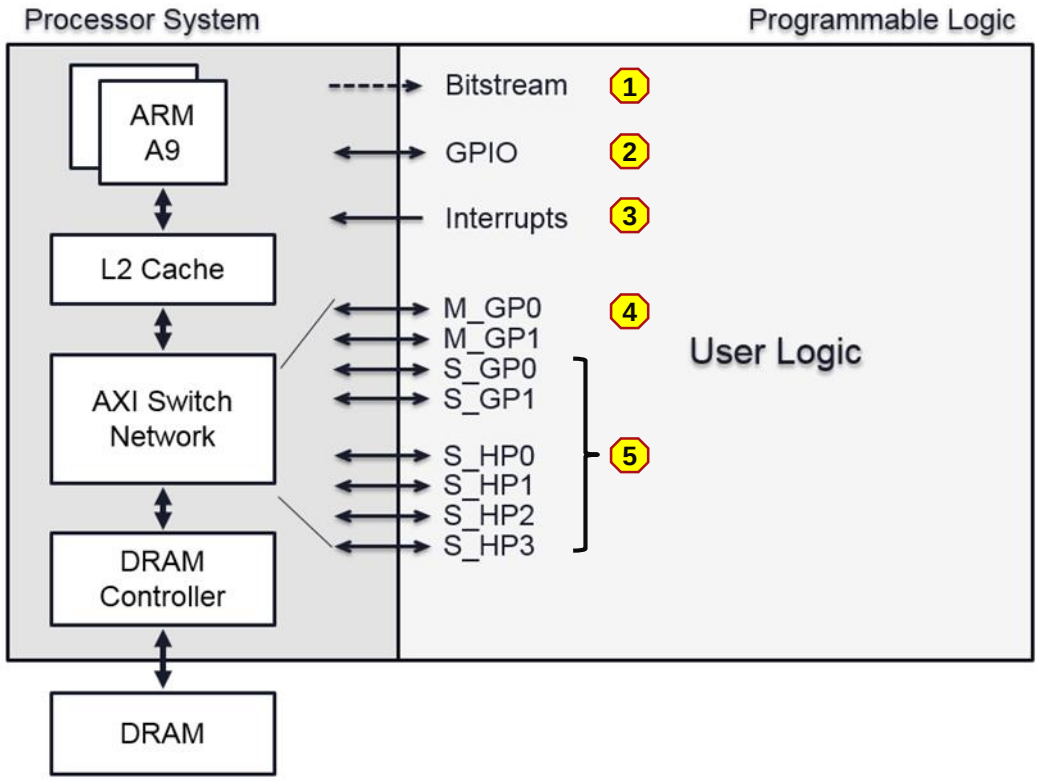
PYNQ uses the PetaLinux build flow and board support package:
- Access to all Xilinx kernel patches
- Works with any Xilinx supported board
- Configured with additional drivers for PS-PL interfaces

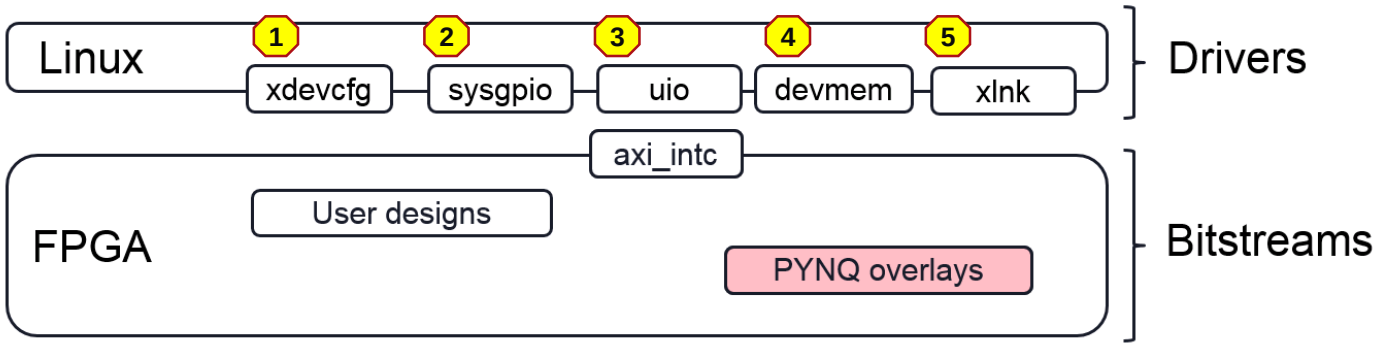
© Copyright 2018 Xilinx

PYNQ provides Linux Drivers for PS-PL Interfaces …

wrapped in Python Libraries

Zynq

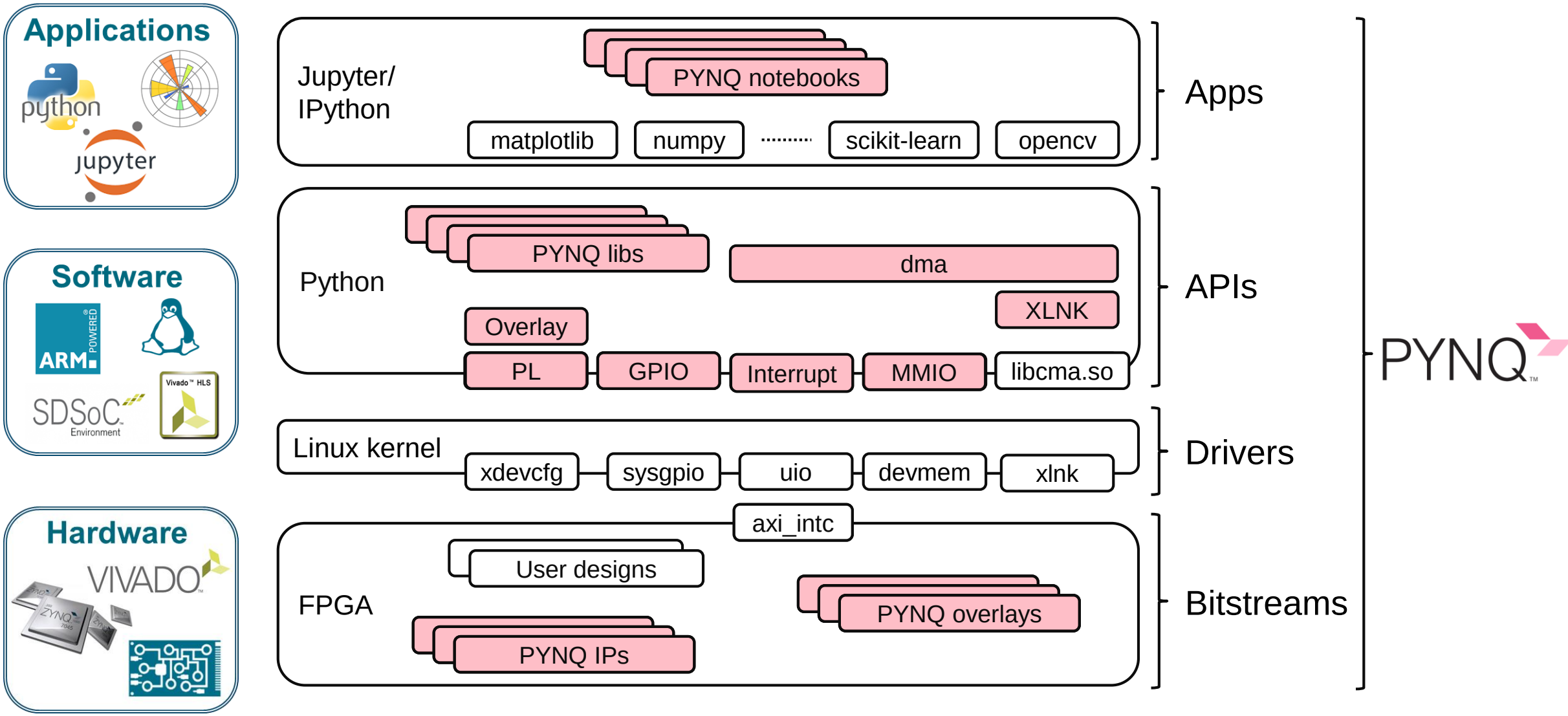




© Copyright 2018 Xilinx

XILINX.

PYNQ is a Framework



© Copyright 2018 Xilinx

XILINX

Vivado Design Metadata available from Python

- **PYNQ passes the Vivado metadata file to the target platform**
  - Initially Vivado TCL file
  - Now moving to newer DSA file (which replaces the TCL file)

- **It then parses the Vivado metadata file to:**
  - Set Zynq clock frequencies automatically
  - Assign memory-map attributes for every IP
  - Assign default MMIO drivers to IP, when no drivers are specified

- **Creates a Python dictionary for the IP in the bitstream from the metadata file**
  - Enables bitstream metadata to be queried and modified in Python at runtime

© Copyright 2018 Xilinx

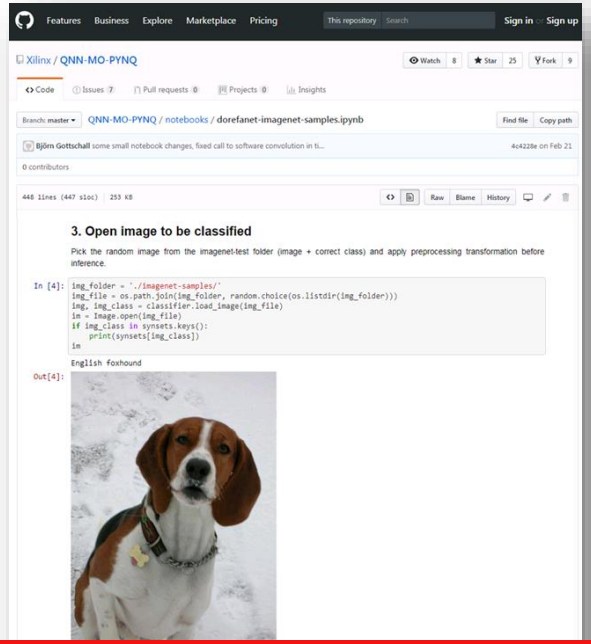
Hybrid Packages

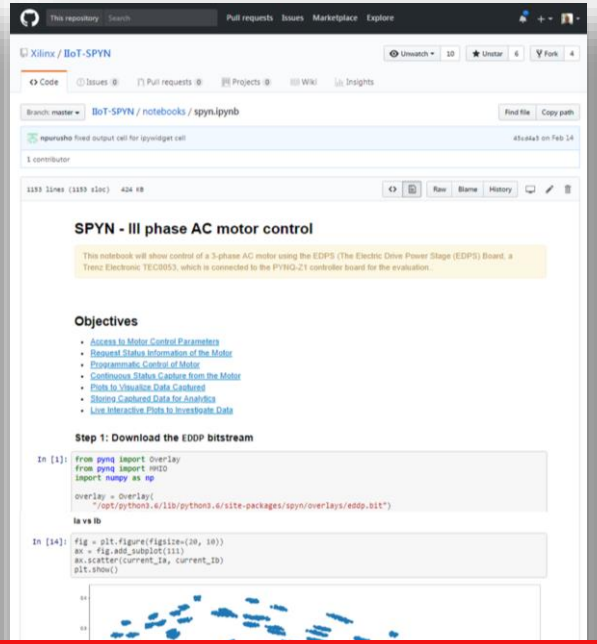
- **New *hybrid packages* are created by extending Python packages with additional files:**
  - Design Bitstream
  - Design metadata file
  - C drivers
  - Jupyter notebooks
- **Hybrid packages enable software-style packaging and distribution of designs**
- **Use the Python package installer, `PIP` to install a hybrid package just like any regular Python (software only) package**
  - Delivers package's files to target board
- **Uses Python standard `setup.py` script for installation**

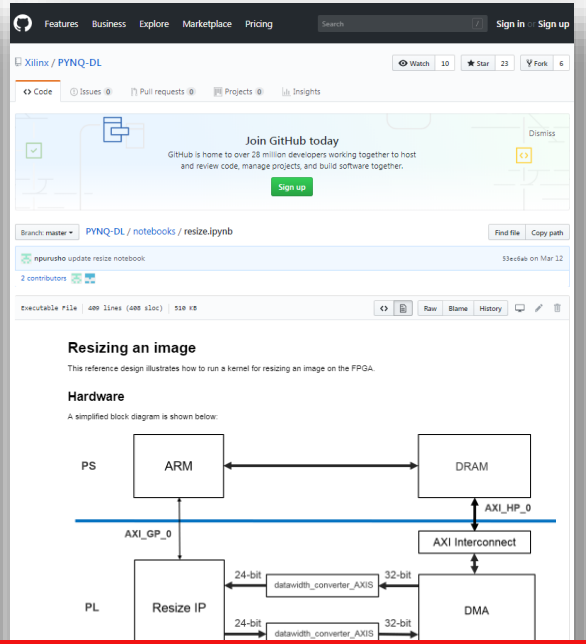
© Copyright 2018 Xilinx

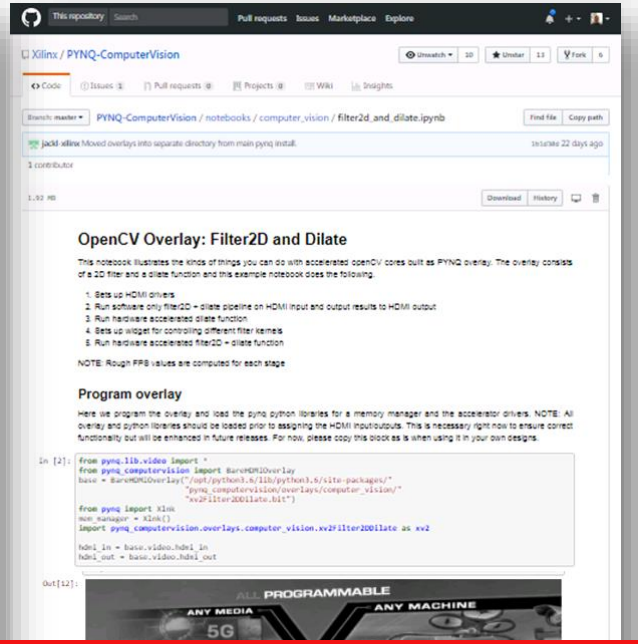
Software-style Packaging & Distribution of Designs

Enabled by new *hybrid packages*










```
Download a design from GitHub with a single Python command:

pip3 install git+https://github.com/Xilinx/pynqDL.git
```


© Copyright 2018 Xilinx

XILINX.

PYNQ enabling technologies

XILINX

PYNQ

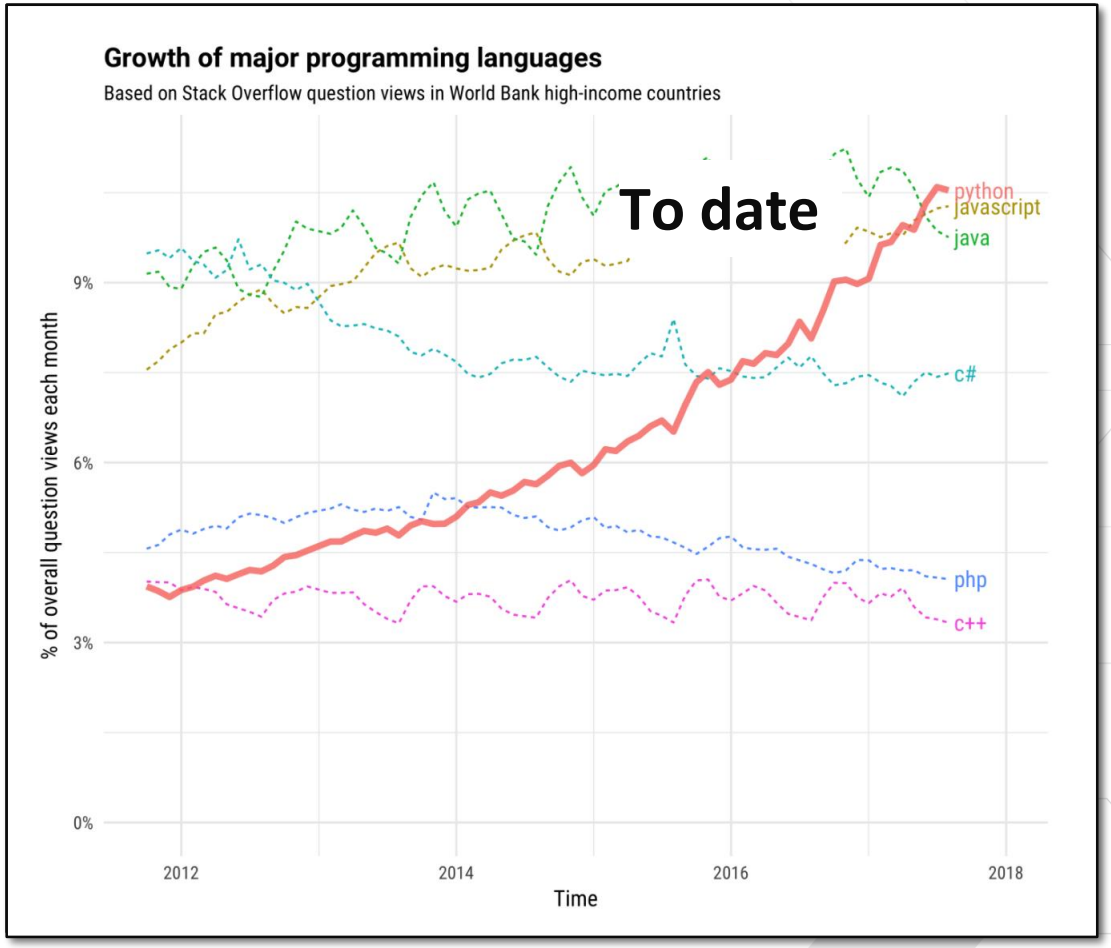
© Copyright 2018 Xilinx

Python is increasingly the language of choice

Top Programming Languages,
IEEE Spectrum, July'17

Language Rank	Types	Spectrum Ranking
1. Python		100.0
2. C		99.7
3. Java		99.4
4. C++		97.2
5. C#		88.6
6. R		88.1
7. JavaScript		85.5
8. PHP		81.4
9. Go		76.1
10. Swift		75.3

https://spectrum.ieee.org/computing/software/the-2017-top-programming-languages



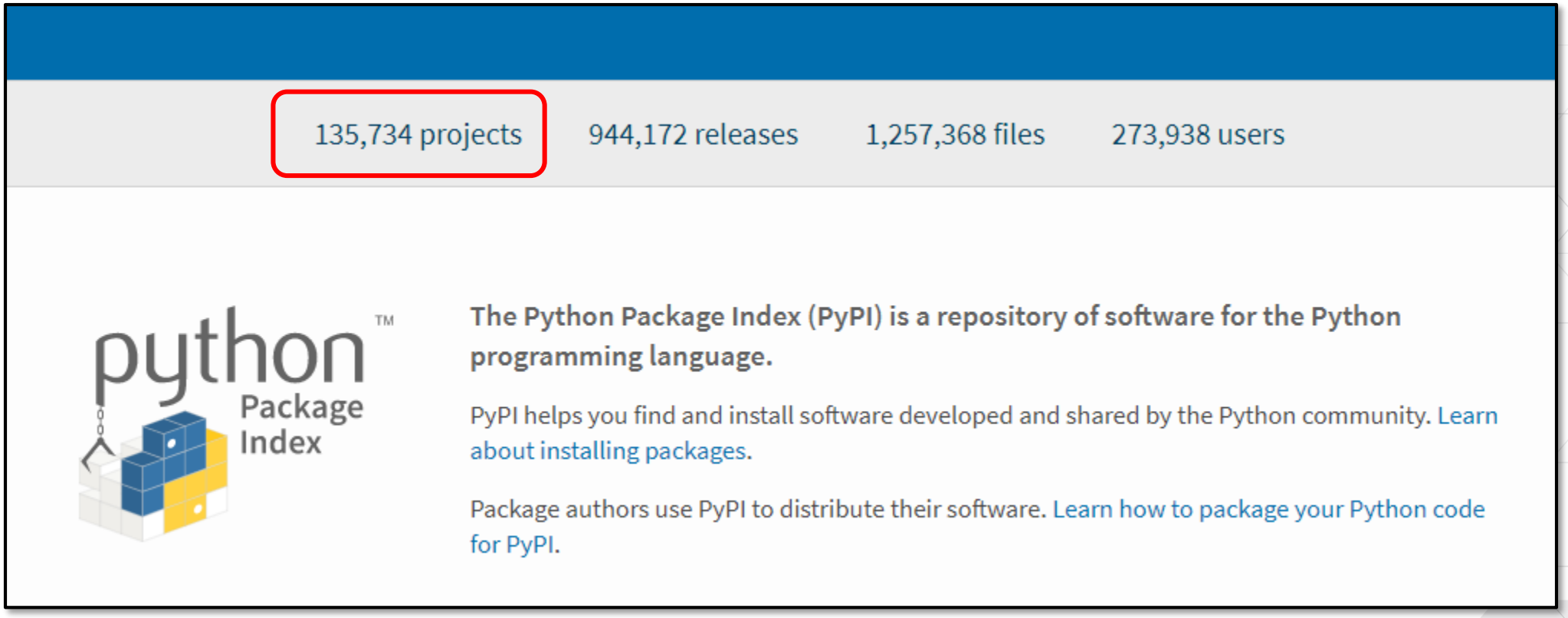
https://stackoverflow.blog/2017/09/06/incredible-growth-python/

Python is the fastest growing language: driven by data science, AI, ML and academia

© Copyright 2018 Xilinx

XILINX.

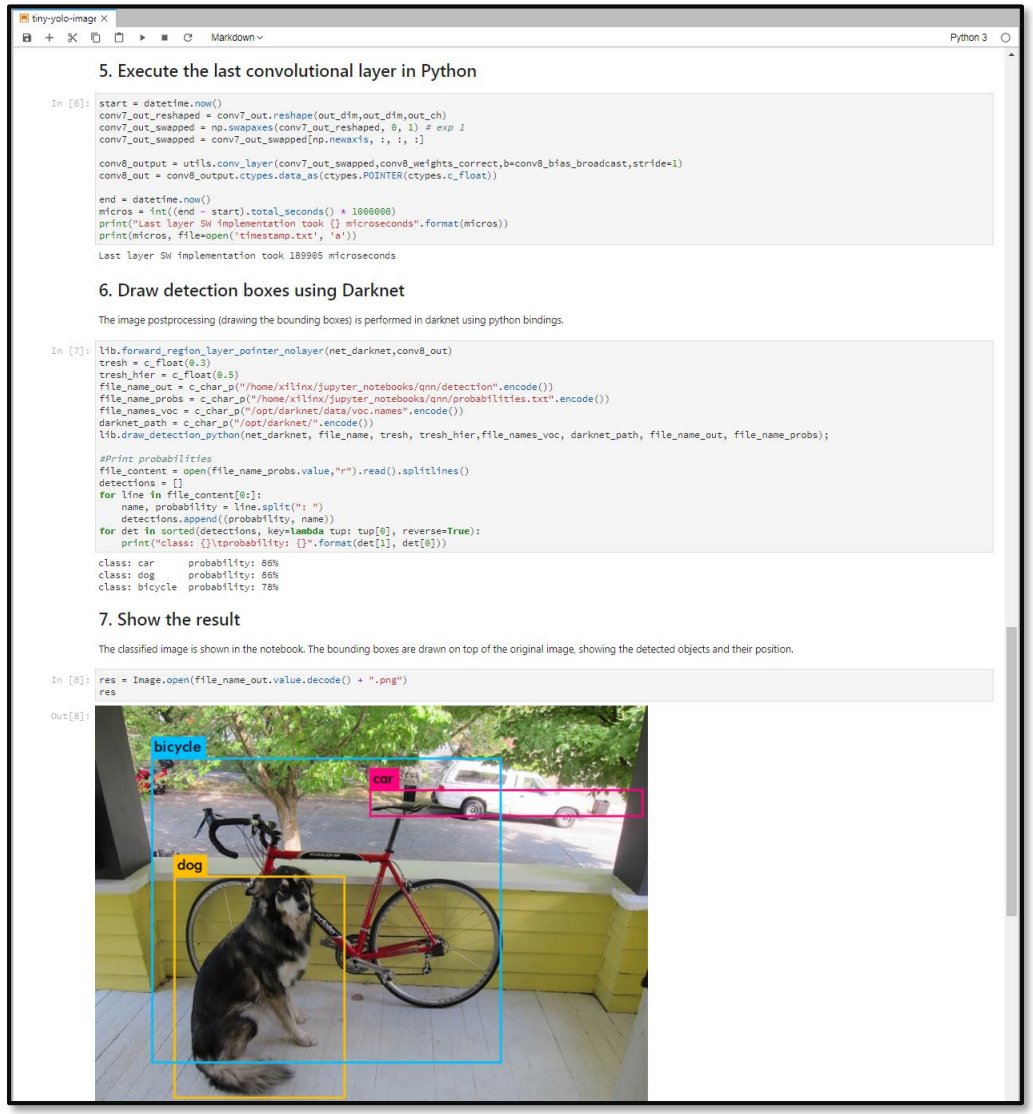
Ecosystem advantage: there's a Python library for that…



CPython is written in C … and most popular C/C++ frameworks have Python libraries

© Copyright 2018 Xilinx

Jupyter Notebooks … the engine of data science



jupyter

Open source browser-based, executable documents

Live code, text, multimedia, graphics, equations, widgets …

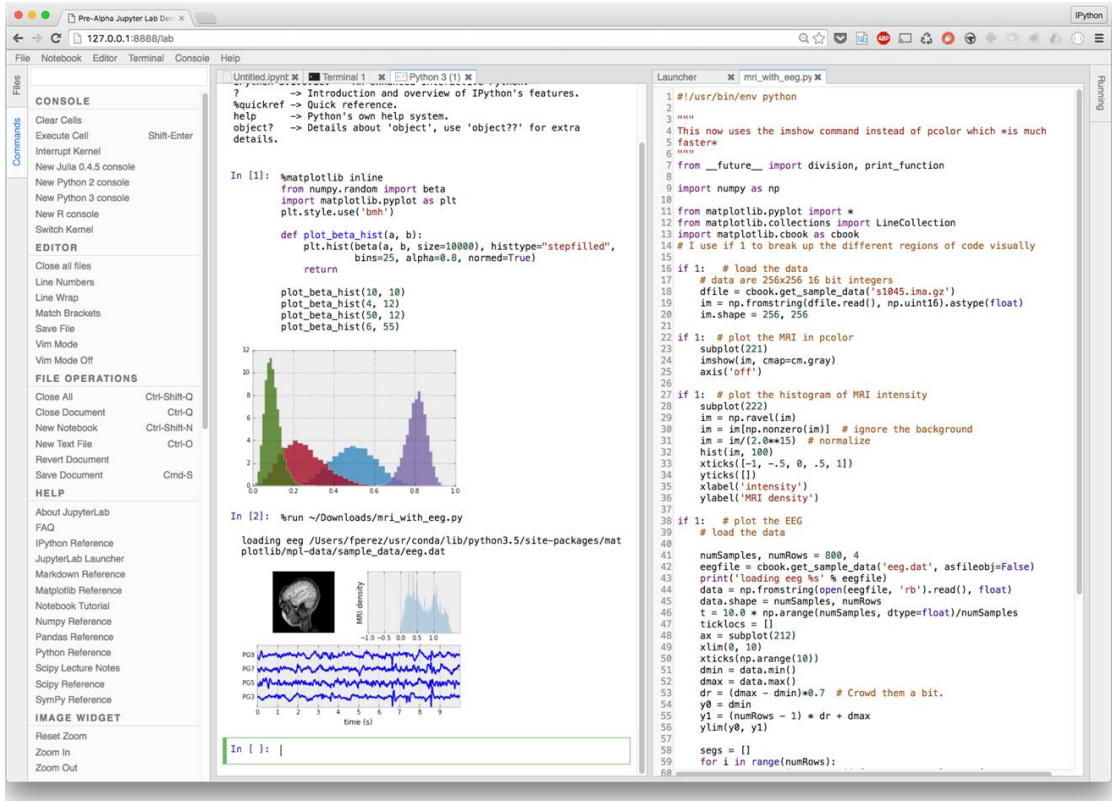
1.7 million notebooks on GitHub

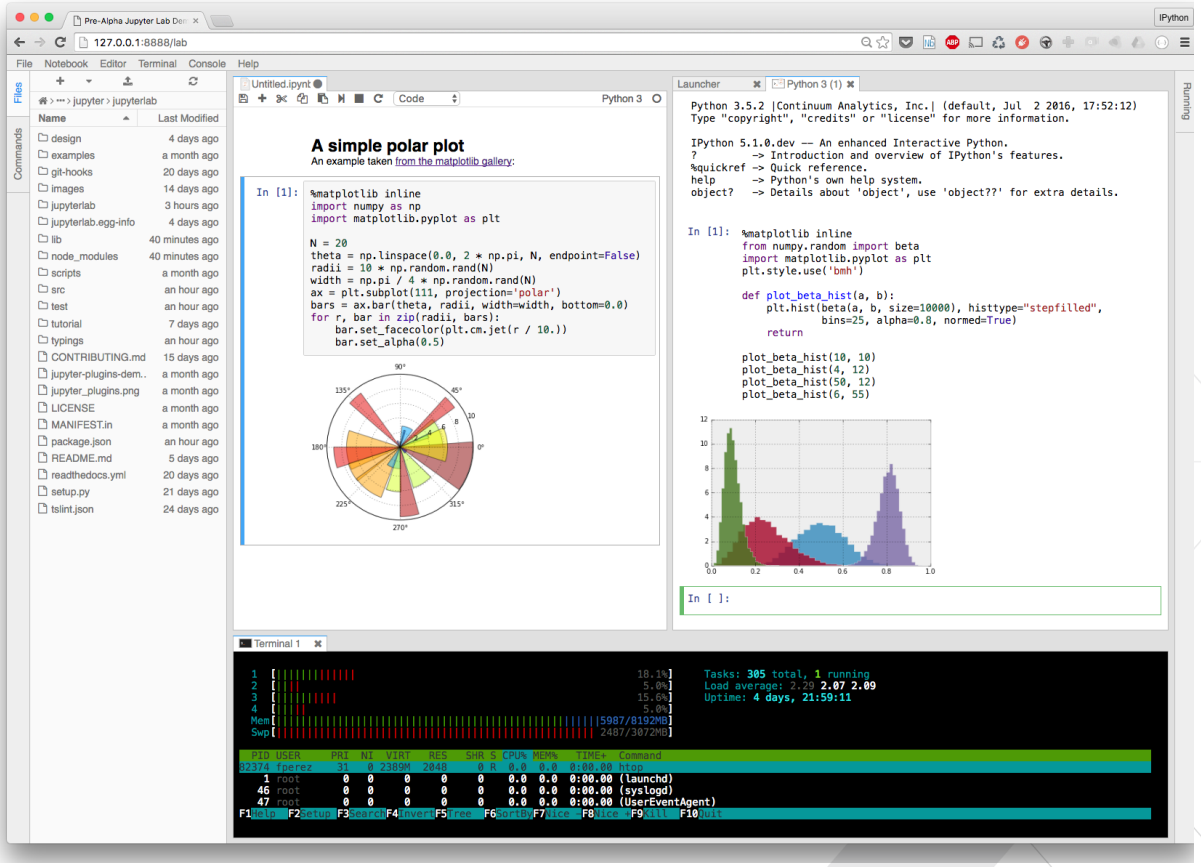
Taught to 1,000+ Berkeley data science students

© Copyright 2018 Xilinx

XILINX

JupyterLab: web-based IDE incl. Notebooks





Jupyter Notebook is now one of many plug-ins within the JupyterLab integrated development environment

JupyterLab - an open-source, extensible IDE in a browser

© Copyright 2018 Xilinx

XILINX.

PYNQ enabled boards

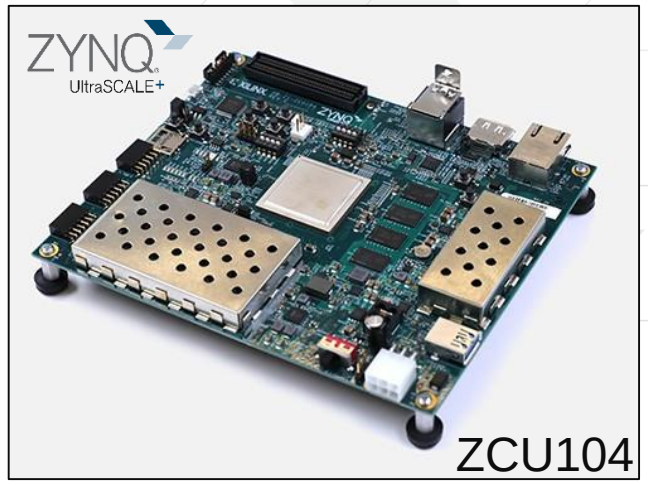
© Copyright 2018 Xilinx

PYNQ-enabled boards

- **Python productivity for Zynq**
  - Open source
  - Build image for other Zynq boards
- **Downloadable SD card image**
  - Zynq 7000
    - PYNQ-Z1 (Digilent)
    - PYNQ-Z2 (TUL)
  - Zynq Ultrascale+
    - ZCU104 (Xilinx)

PYNQ-Z1

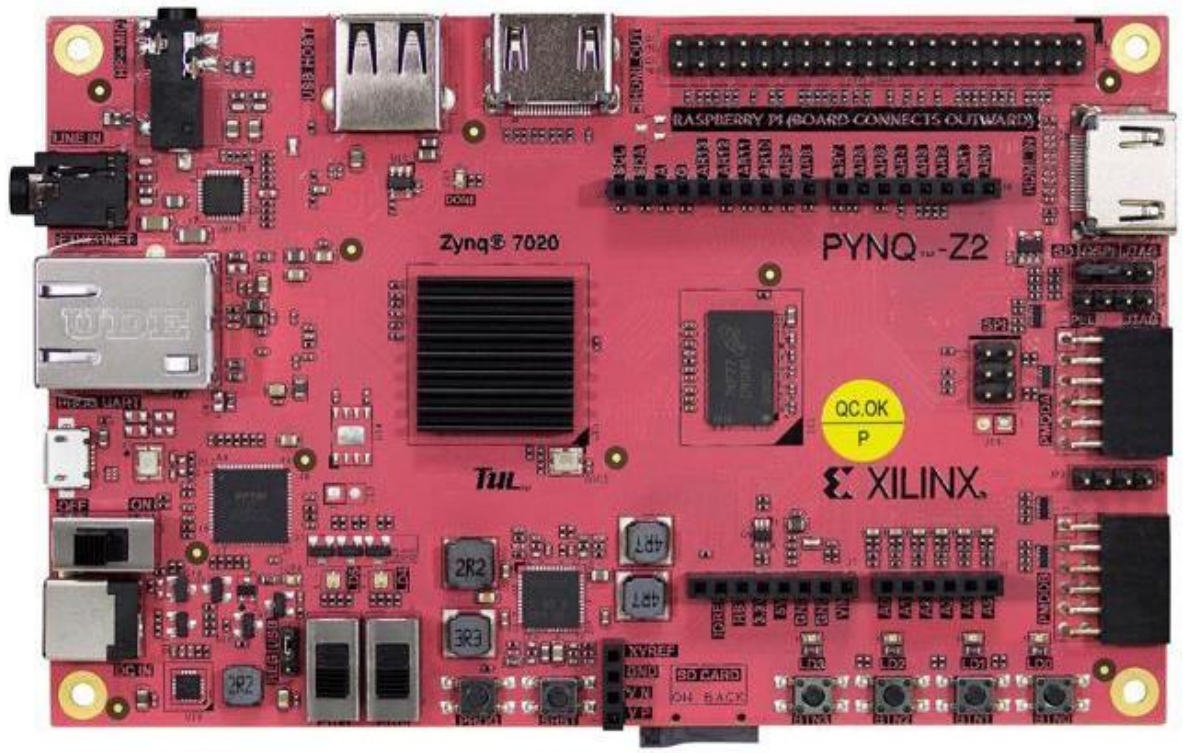
PYNQ-Z2



ZCU104

© Copyright 2018 Xilinx

New PYNQ-Z2 Board available now



$119/€119 or equivilent

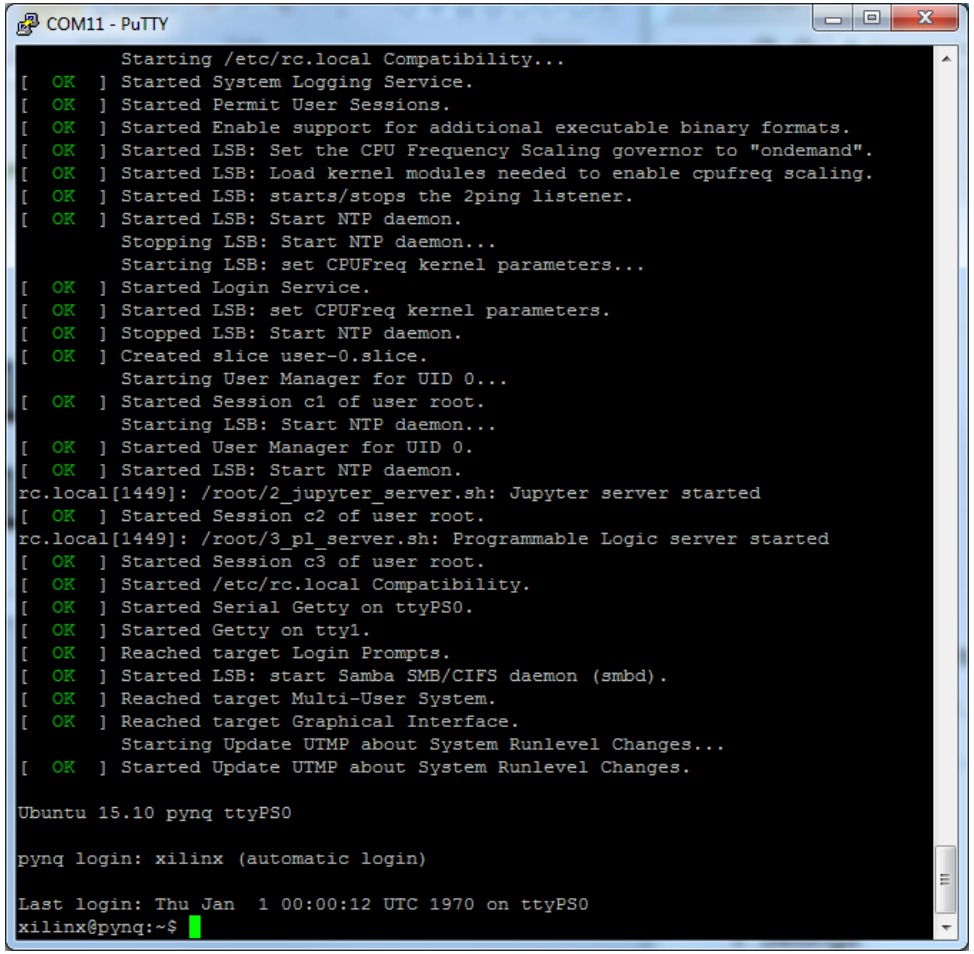
- New PYNQ reference platform
- New stereo audio with on-board codec
- New Raspberry Pi connector
- Open source design
- Manufactured in Taiwan by TUL
- Distributed globally by Premier Farnell
- Also Newegg in US
- Academic discounts & donations available

© Copyright 2018 Xilinx

Benefits of PYNQ

© Copyright 2018 Xilinx

Start using PYNQ out-of-the-box



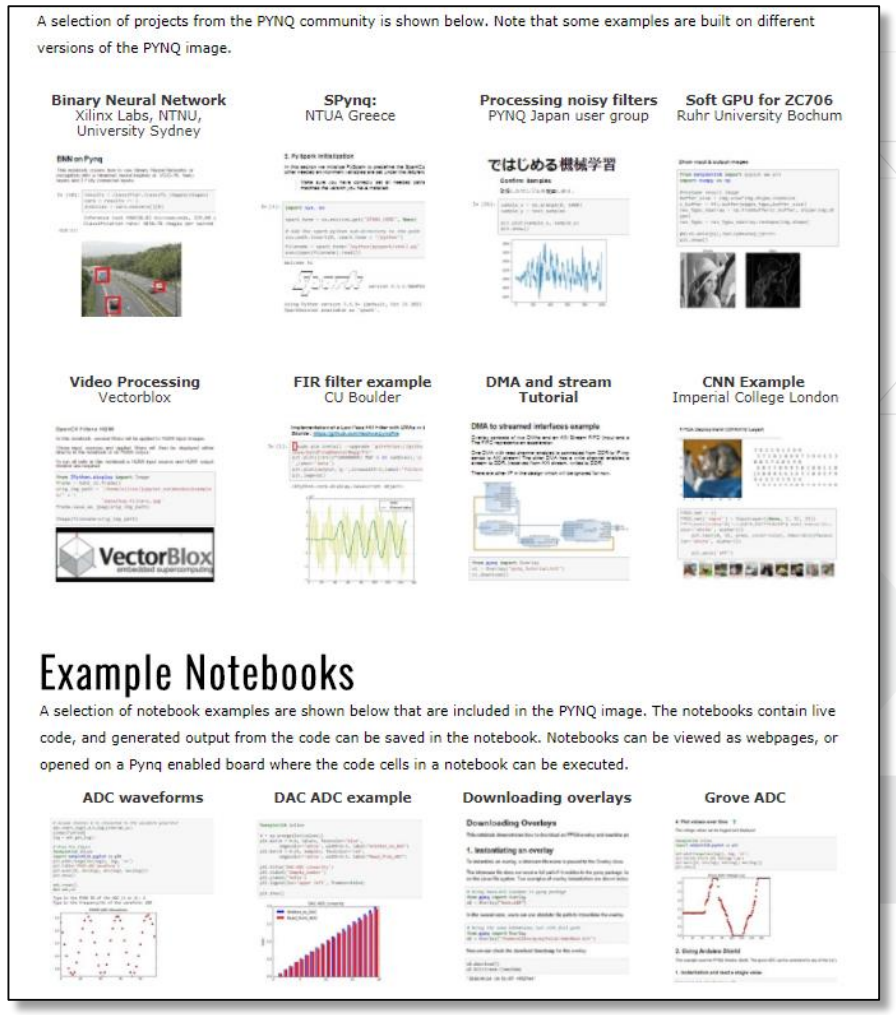
- **PYNQ delivered as downloadable SD card image**
  - Linux preconfigured
- **Additional packages and drivers pre-installed**
  - USB peripheral drivers: webcams, wifi modules …
- **PYNQ is for Zynq**
  - PYNQ image is portable to other Zynq boards

Start using Zynq out of the box ✓

© Copyright 2018 Xilinx

Desktop Linux

- **Network/Internet access**
  - “apt-get” to install packages from Ubuntu universe
  - Samba(Network drive)
  - Web services
- **Git directly on board**
- **Compilers and other development tools**
  - Gcc,, MicroBlaze, RISC-V ….
- **Python packages**
  - “pip install”
  - PYNQ Community examples



© Copyright 2018 Xilinx

Simplify downloading bitstreams to PL

- **PYNQ 'Overlay' class**
  - Simplifies downloading bitstream
  - two lines of code
  - No Xilinx tools required
- **Maintain many bitstreams on the SD card**
  - E.g. multiple different demos
- **Can execute Python in browser, or from command line**

```
from pynq import Overlay

ol = Overlay('gray.bit')
```

Simply and fast way to configure Programmable Logic ✓

© Copyright 2018 Xilinx

Simplify IP debug and prototyping

- **Debug of IP typically uses C/C++**
- **SDK tools used to:**
  - Compile test application
  - Download application to board
  - Step through code
- **PYNQ MMIO class allows peek/poke of IP registers from Python**
  - Python executes directly on the board
  - No offline compilation, download loop
  - No SDK tools

C code – compile, debug from host

```
/*****************************************************************************/
* This function does a selftest on the IIC device and XIic driver as an
* example.
* @param    DeviceId is the XPAR_<IIC_instance>_DEVICE_ID value from
*       xparameters.h.
*****************************************************************************/
int IicSelfTestExample(u16 DeviceId)
{
    Status = XIic_CfgInitialize(&Iic, ConfigPtr, ConfigPtr->BaseAddress);
    if (Status != XST_SUCCESS) {
        return XST_FAILURE;
    }

    /* Perform a self-test to ensure that the hardware was built */
    Status = XIic_SelfTest(&Iic);
    if (Status != XST_SUCCESS) {
        return XST_FAILURE;
    }
```

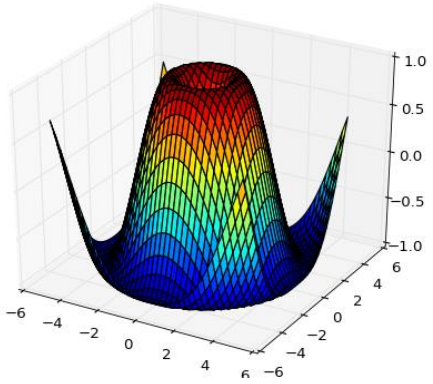
Python executes directly on target

```
from pynq import MMIO

# Map registers to MMIO instance
my_ip = MMIO(my_ip_addr, LENGTH)

# Write 0x1 to start IP
my_ip.write(CONTROL_REGISTER, 0x1)
# Check status register
my_ip.read(STATUS_REGISTER)
```

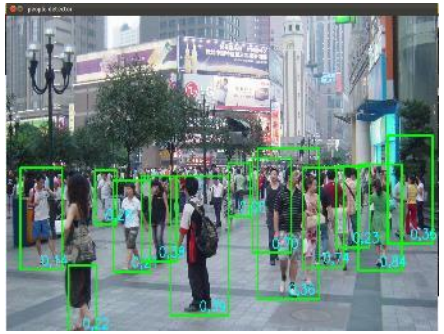
Rapid testing and prototyping ✓

© Copyright 2018 Xilinx

Python packages for data analysis and visualisation



Matplotlib



OpenCV

- **Take advantage of Python for data analysis and processing**
  - NumPy
    - Scientific computing package for Python
  - Matplotlib
    - Python 2D plotting library
  - Pandas
    - Data analysis tools for Python
  - OpenCV
    - Computer Vision and machine learning software

Optimized open-source software libraries ✓

© Copyright 2018 Xilinx

Example

6. Detailed Classification Information

In addition to highest ranked class, it is possible to get the r
couple of images of a car, an airplane, and a bird and place

In [8]: `from IPython.display import display`

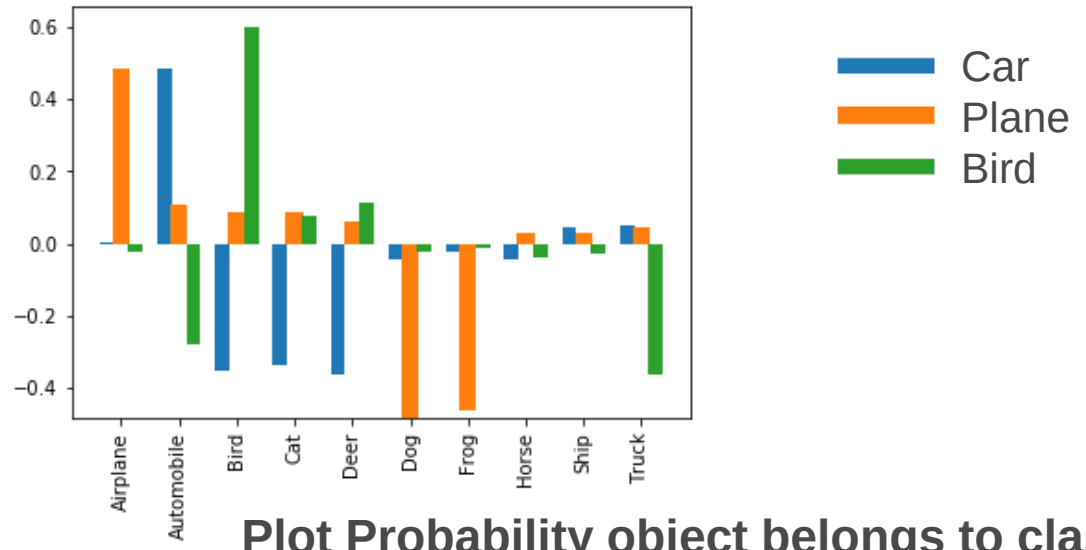
Probability object belongs to class

```
oseconds
 images per second
69 163 244 249 244 267 268]
```

```
oseconds
 images per second
277 277 271 132 137 262 263 266]
```

```
oseconds
 images per second
274 284 249 252 245 248 163]
```

```
%matplotlib inline
import matplotlib.pyplot as plt

x_pos = np.arange(len(car_class))
fig, ax = plt.subplots()
ax.bar(x_pos - 0.25, (car_class/255)-1, 0.25)
ax.bar(x_pos, (air_class/255)-1, 0.3)
ax.bar(x_pos + 0.25, (bird_class/255)-1, 0.25)
ax.set_xticklabels(classifier.bnn.classes, rotation='vertical')
ax.set_xticks(x_pos)
ax.set
plt.show()
```

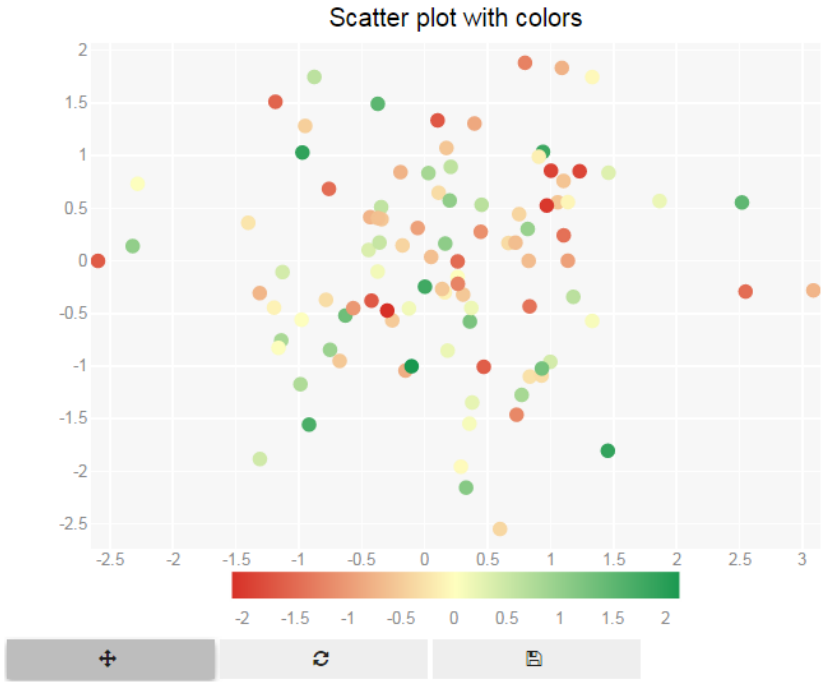


Plot Probability object belongs to class

© Copyright 2018 Xilinx

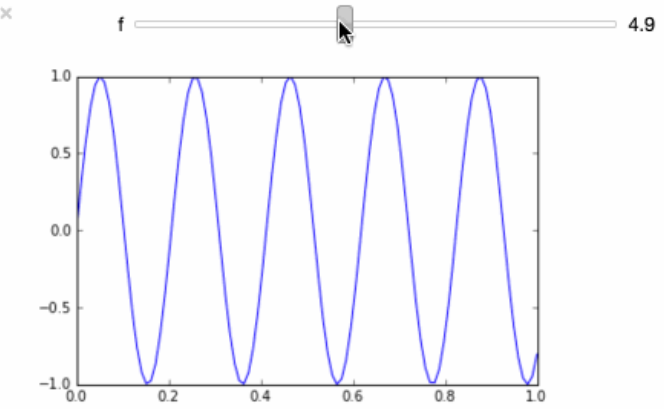
XILINX.

Take advantage of interactive widgets



http://jupyter.org/widgets.html

```
In [22]: from IPython.html.widgets import *
t = arange(0.0, 1.0, 0.01)

def pltsin(f):
    plt.plot(x,sin(2*pi*t*f))
    plt.show()

interact(pltsin, f=(1,10,0.1))
```



https://blog.dominodatalab.com/interactive-dashboards-in-jupyter/

```
In [7]: p2 = IntProgress(max=56)
p2.value += 10
p2.description = 'Running'
display(p2)
```

Running

ThingA ThingA (modified)
ThingB ThingB
ThingC ThingC (modified)
ThingD ThingD

show Annotator themes
group_by How often do you use Jupyter Notebook?
values Percentages

stereo fullscreen

0

Add intuitive graphical interfaces ✓

© Copyright 2018 Xilinx

XILINX.

Why PYNQ is a Game-changer!

- **PYNQ makes Zynq/ZynqU+ accessible to non-traditional customers**
- **PYNQ delivers open source benefits**
  - Huge ecosystem
  - Extensive knowledge base
  - Amazing community support
- **PYNQ enables highly-productive …**
  - Prototyping
  - Debug
  - Verification
  - Evaluation
- **PYNQ powers awesome demonstrators**
- **PYNQ documentation flows are amazing**
  - Capture your own work
  - Capture work you want to re-use
- **PYNQ designs can be …**
  - Packaged, published and distributed

just like software

- **“PYNQ makes FPGAs FUN again!”,**
  J. Gray, Xilinx Power User*

*http://fpga.org/wp-content/uploads/2017/09/GRVI-Phalanx-on-Pynq-FCCM2017-Pynq-Workshop.pdf

© Copyright 2018 Xilinx

Community

© Copyright 2018 Xilinx

pynq.io

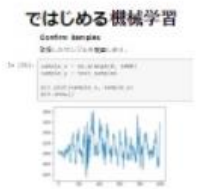
Community Projects

A selection of projects from the PYNQ community is shown below. Note that some examples are built on different versions of the PYNQ image.

Binary Neural Network
Xilinx Labs, NTNU, University Sydney

SPynq:
NTUA Greece

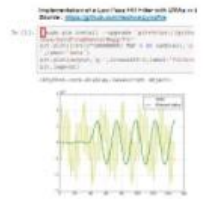
Processing noisy filters
PYNQ Japan user group



Soft GPU for ZC706
Ruhr University Bochum

Video Processing
Vectorblox

FIR filter example
CU Boulder



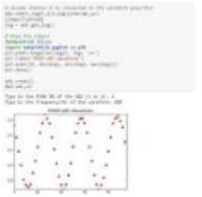
DMA and stream Tutorial

CNN Example
Imperial College London

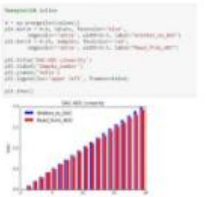
Example Notebooks

A selection of notebook examples are shown below that are included in the PYNQ image. The notebooks contain live code, and generated output from the code can be saved in the notebook. Notebooks can be viewed as webpages, or opened on a Pynq enabled board where the code cells in a notebook can be executed.

ADC waveforms

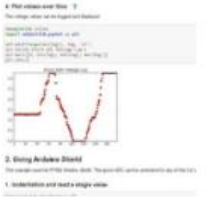


DAC ADC example



Downloading overlays

Grove ADC



Selection of projects and notebooks

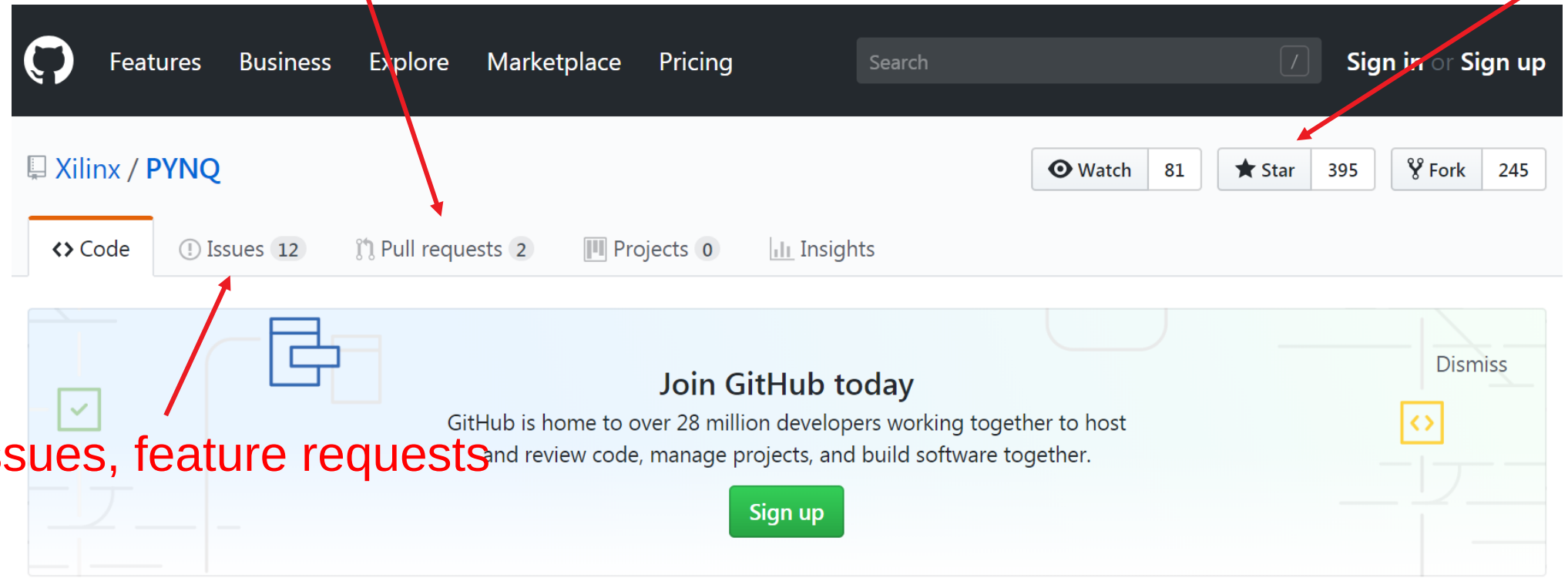
© Copyright 2018 Xilinx

XILINX.

All Feedback helps

Contribute

If you like it, star it!



Issues, feature requests

Python Productivity for ZYNQ http://www.pynq.io/

pynq

© Copyright 2018 Xilinx

XILINX.

Summary

PYNQ™

- PYNQ is Python productivity for Zynq
- Everything runs on Zynq, access via a browser
- Support for Zynq Ultrascale+
- Overlays are hardware libraries and enable software developers to use Zynq
- Provides a rapid prototyping framework for hardware developers

pynq.io

pynq.readthedocs.org

github.com/Xilinx/PYNQ

tul.com.tw/ProductsPYNQ-Z2.html

pynq.io/support

© Copyright 2018 Xilinx

XILINX.

Adaptable.
Intelligent.

XILINX

PYNQ

© Copyright 2018 Xilinx