

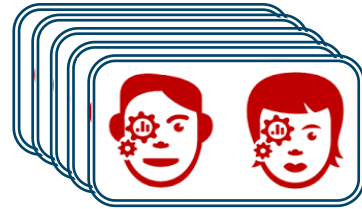


# Python Productivity for Zynq

Louis Liu  
Senior Application Engineer  
weli@xilinx.com



# PYNQ™ Python Productivity for Zynq



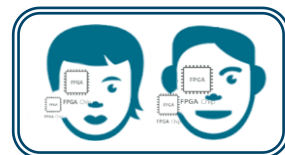
Domain Experts

Targeting the data center  
artificial intelligence,  
machine learning,  
data science



Embedded software Engineers

New users are not hardware designers,  
or embedded systems designers



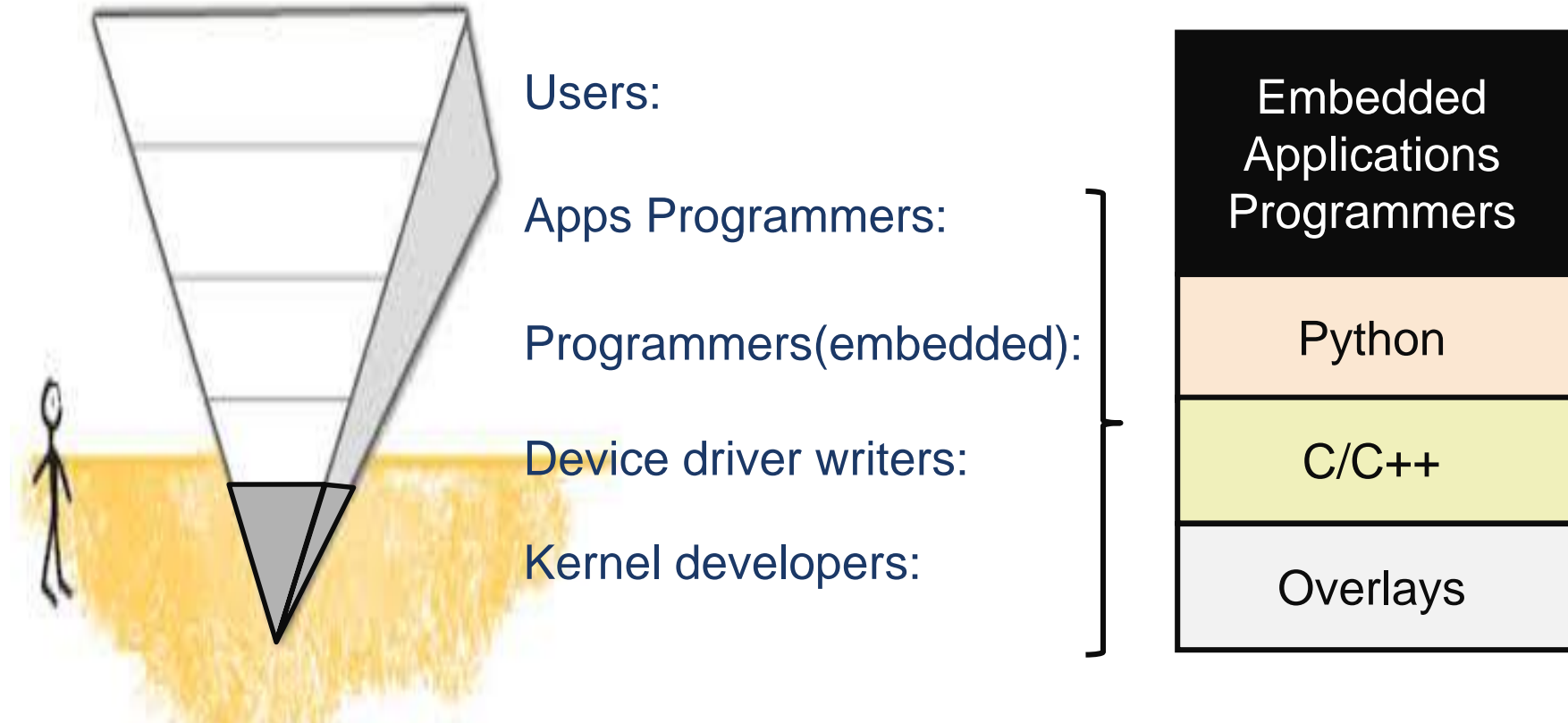
Hardware Engineers

PYNQ™

*Enable more people to program Xilinx  
processing platforms, more productively*

# Productivity Languages & Hardware Overlays

Zynq / Zynq UltraScale+



**Small group of experts create APSoC overlays and C API/drivers**  
**Many more users build applications in C/Python**

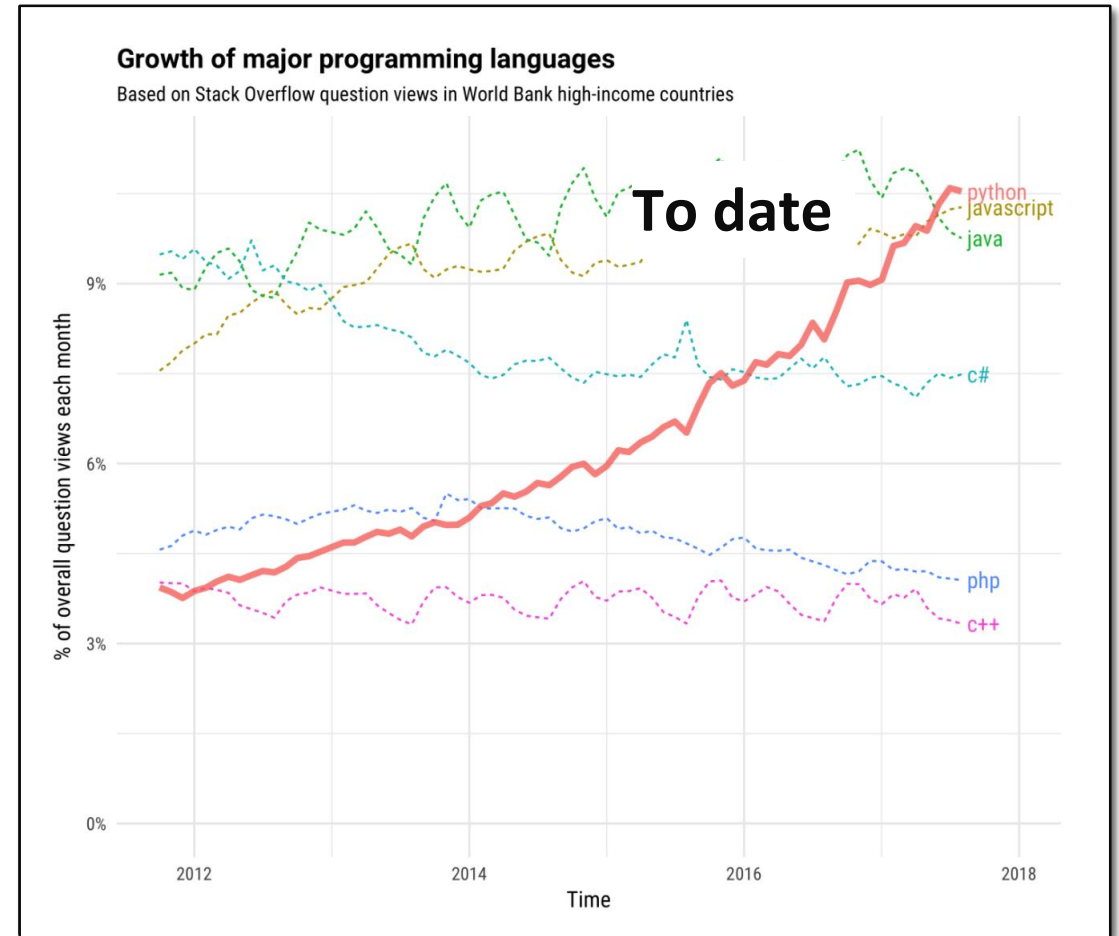
# Python is increasingly the Language of Choice

Top Programming Languages,  
IEEE Spectrum, July'18

Language Rank	Types	Spectrum Ranking
1. Python	🌐 🖥️ 📱	100.0
2. C++	📱 🖥️ 📡	98.4
3. C	📱 🖥️ 📡	98.2
4. Java	🌐 📱 🖥️	97.5
5. C#	🌐 📱 🖥️	89.8
6. PHP	🌐	85.4
7. R	🖥️	83.3
8. JavaScript	🌐 📱	82.8
9. Go	🌐 🖥️	76.7
10. Assembly	📡	74.5

Python is listed as an  
embedded language  
for the first time

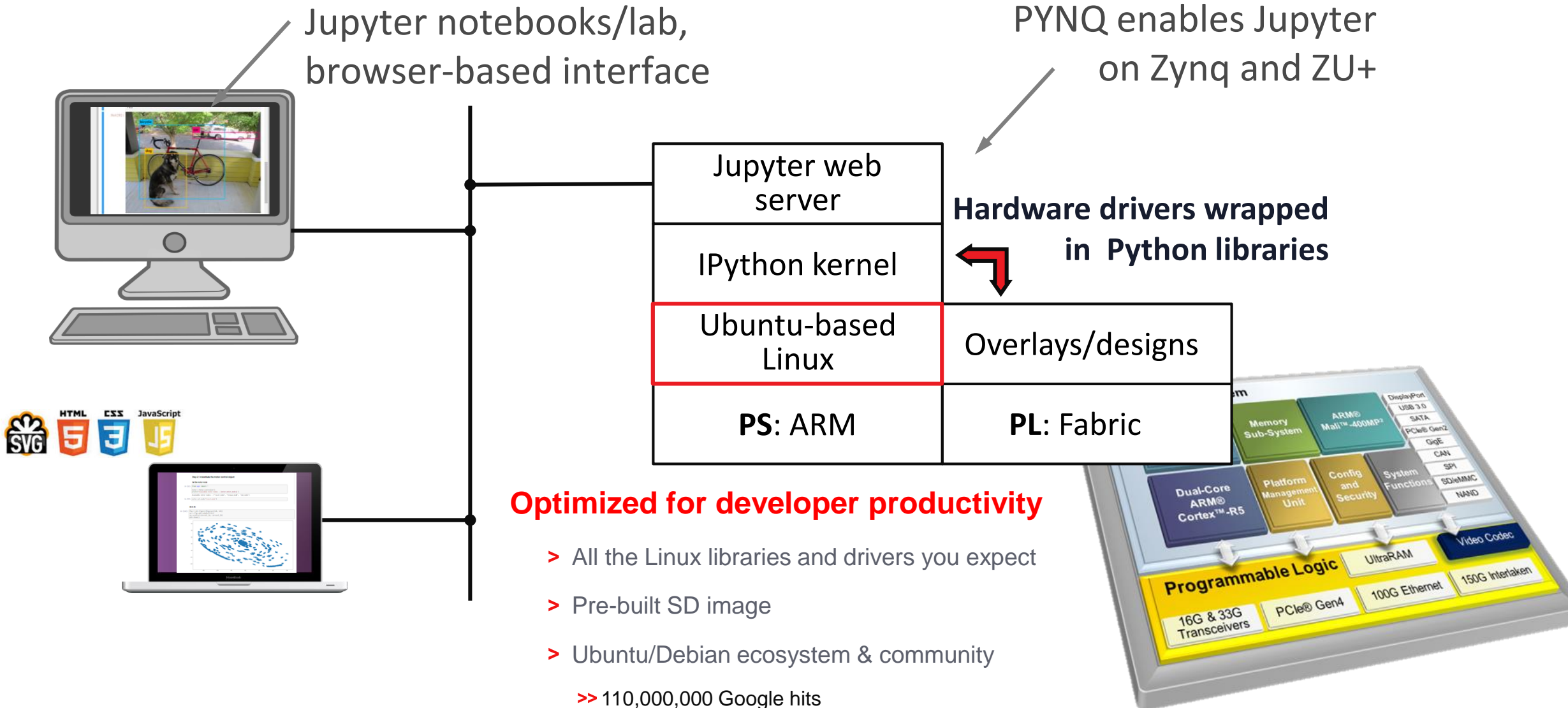
<https://spectrum.ieee.org/at-work/innovation/the-2018-top-programming-languages>



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

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

# Python productivity for Zynq



## Optimized for developer productivity

- > All the Linux libraries and drivers you expect
- > Pre-built SD image
- > Ubuntu/Debian ecosystem & community
- >> 110,000,000 Google hits

# Jupyter Notebooks to JupyterLab IDE

2017 ACM  
Software System Award

Code editor      Terminal

The screenshot displays the JupyterLab IDE interface. On the left, a code editor shows Python code for image processing using Darknet. The middle section displays the output of the code, including a classification result for a bicycle image with bounding boxes. On the right, a terminal window shows the execution of a program, and an output view displays a state transition diagram for an FSM generator.

Jupyter ... Julia, Python, R

Default engine of data science

Taught to 1,000+ Berkeley  
students every semester

2+ million notebooks  
on GitHub

Next-gen browser IDE

Includes Jupyter Notebooks

Jupyter notebooks

Visualization



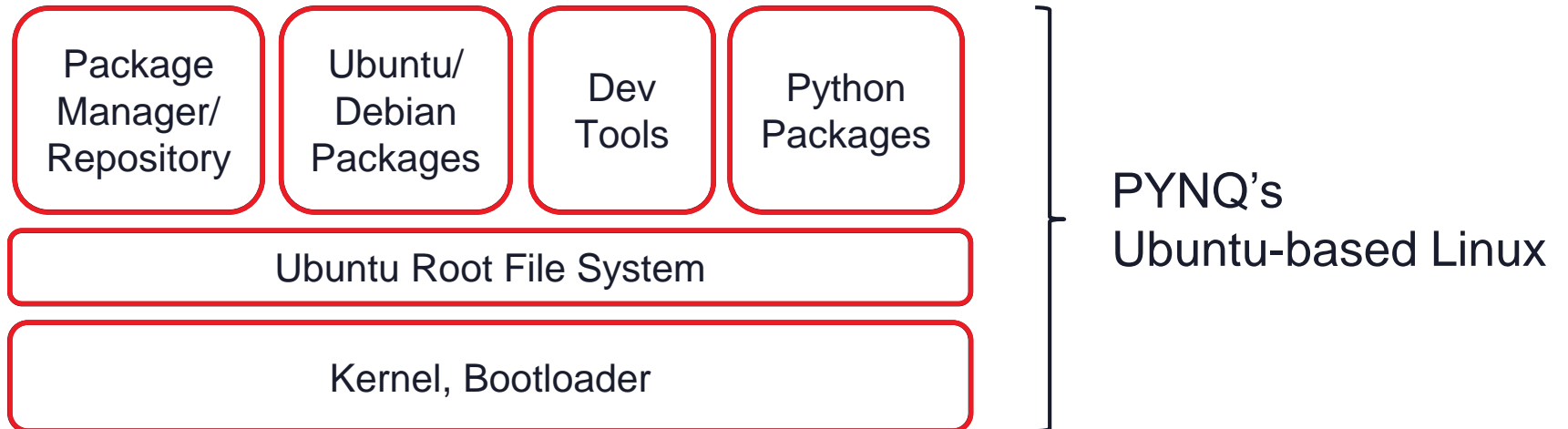
# PYNQ's Ubuntu-based Linux

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

# 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

Embedded Linux



➤ **Optimized for deployment efficiency**

- > Selective Linux libraries and drivers
- > Commonly delivered in flash memory on board
- > PetaLinux ecosystem:

>> 143,000 Google hits

3 orders of magnitude difference

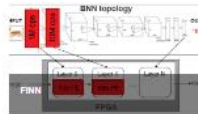


# Hundreds Overlays in PYNQ Community - Need your contributions too

## PYNQ Community

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

### Machine Learning on Xilinx FPGAs with FINN



PYNQ has been widely used for machine learning research and prototyping.

**FINN**, an experimental framework from Xilinx Research Labs to explore deep neural network inference on FPGAs. It specifically targets quantized neural networks, with emphasis on generating dataflow-style architectures customized for each network.

FINN makes extensive use of PYNQ as a prototyping platform. For more information see [pynq.io/ml](http://pynq.io/ml)

### PYNQ community projects

**PYNQ RFSoc**  
University Strathclyde  
QPSK demo on ZCU111



**PYNQ-PRIO**  
BYU  
Partial reconfiguration Input/Output



**PYNQ Hello World**  
Hardware accelerated image resizer example



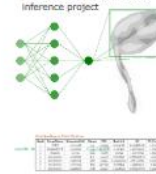
**RISC-V on PYNQ**  
UCSD



**Extended Kalman filter**  
University Sydney



**spoonNN**  
ETH Zurich  
FPGA-based neural network inference project



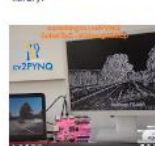
**iSmart DNN**  
FPGA-based neural network inference for DAC 2018 contest



**TGIIF**  
FPGA-based neural network inference for DAC 2018 contest



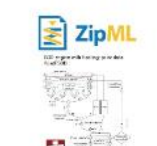
**cv2PYNQ**  
FAU  
Accelerated OpenCV image filtering library.



**Video processing**  
KU Leuven  
Hardware accelerated video processing



**ZipML-PYNQ**  
ETH Zurich  
Hardware accelerated compression



**PYNQ bot**  
IT Tallaght  
Control of robotic car from PYNQ

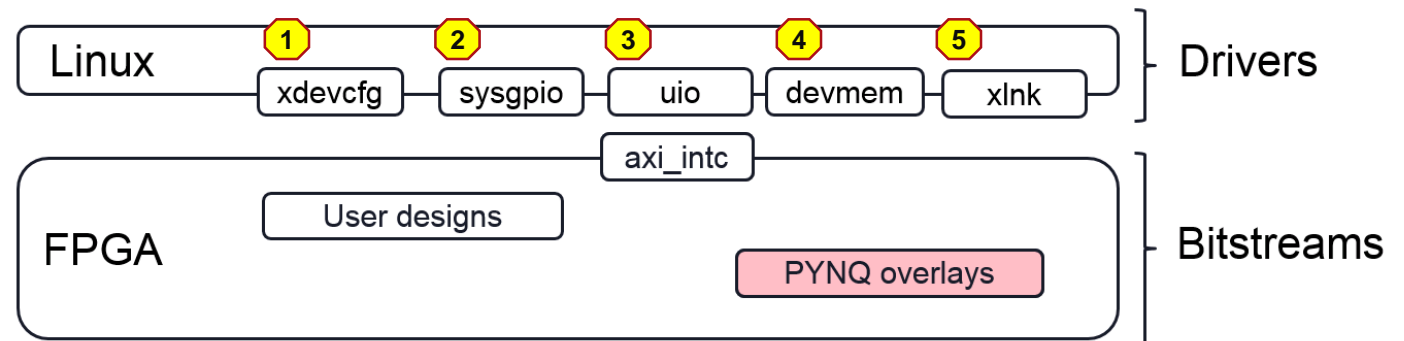
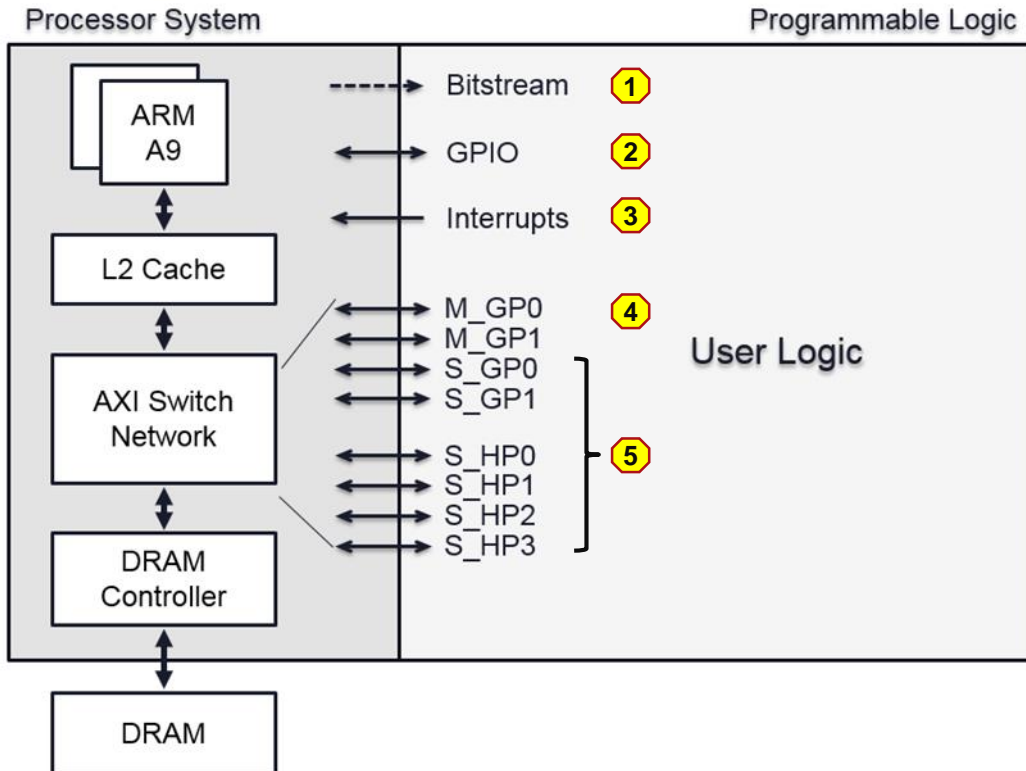


I WANT YOU

# PYNQ provides Linux Drivers for PS-PL Interfaces ...

## wrapped in Python Libraries

Zynq



# How MMIO works

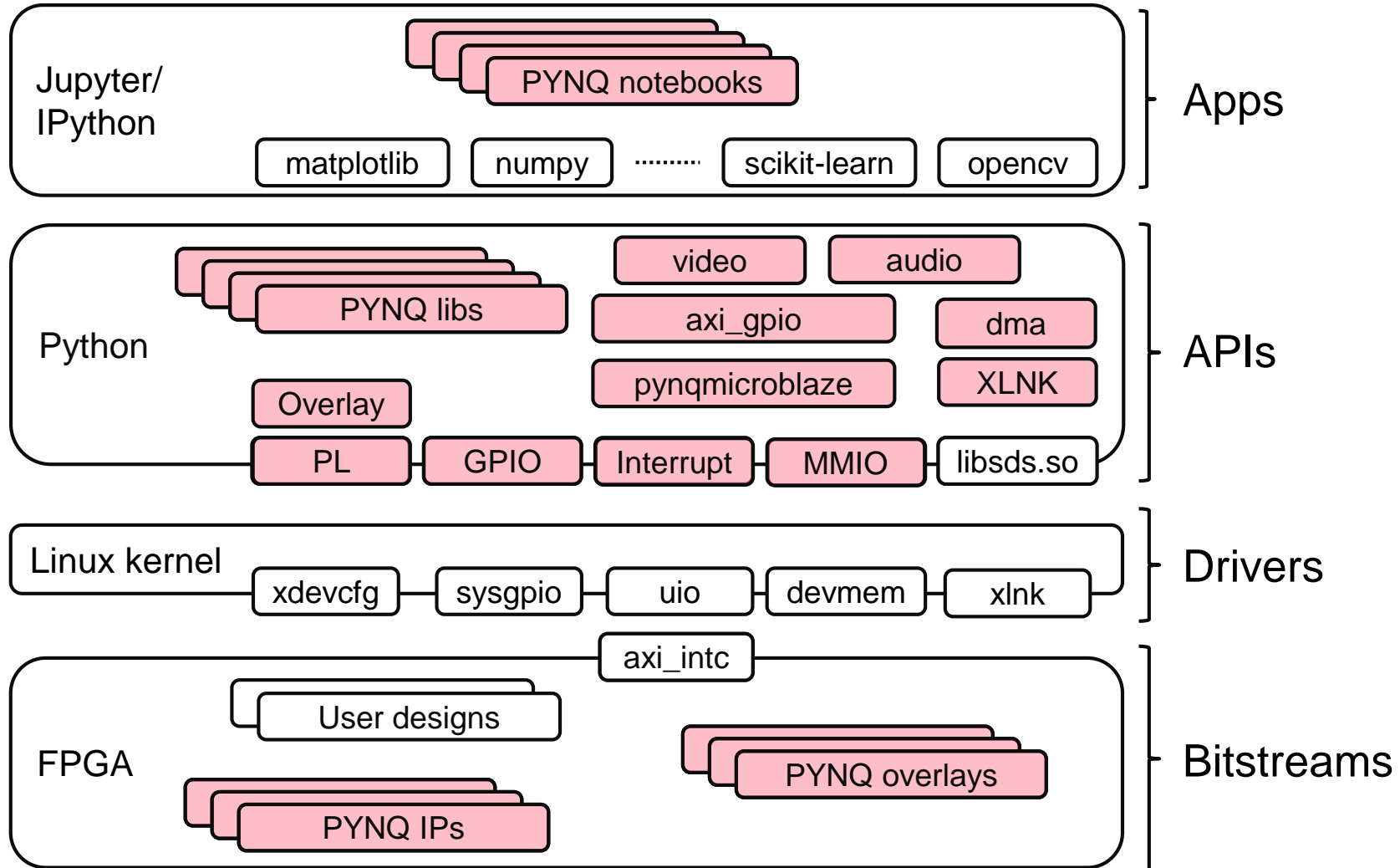
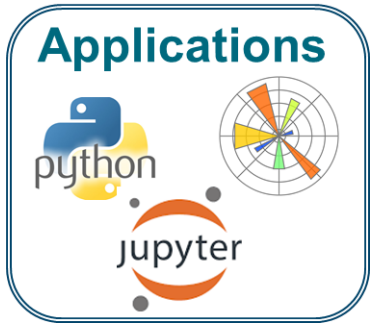
```
PYNQ/mmio.py at master · Xilinx · X +  
← → ↻ | GitHub, Inc. [US] | https://github.com/Xilinx/PYNQ/blob/master/pynq/mmio.py
```

```
17 #  
18 # THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUT  
19 # AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITE  
20 # THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PART  
21 # PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER C  
22 # CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPE  
23 # EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED  
24 # PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA,  
25 # OR BUSINESS INTERRUPTION). HOWEVER CAUSED AND ON ANY THEORY OF L  
26 # WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIC  
27 # OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, E  
28 # ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.  
29  
30 import os  
31 import mmap  
32 import numpy as np  
33  
34 __author__ = "Yun Rock Qu"  
35 __copyright__ = "Copyright 2016, Xilinx"  
36 __email__ = "pynq_support@xilinx.com"  
37  
38  
39 class MMIO:  
40     """ This class exposes API for MMIO read and write.  
41  
42     Attributes  
43     -----  
44     virt_base : int  
45         The address of the page for the MMIO base address.  
46     virt_offset : int  
47         The offset of the MMIO base address from the virt base
```

```
PYNQ/mmio.py at master · Xilinx · X +  
← → ↻ | GitHub, Inc. [US] | https://github.com/Xilinx/PYNQ/blob/master/pynq/mmio.py
```

```
92  
93     self.debug = debug  
94     self._debug('MMIO(address, size) = ({0:x}, {1:x} bytes).',  
95                 self.base_addr, self.length)  
96  
97     # Open file and mmap  
98     self.mmap_file = os.open('/dev/mem',  
99                             os.O_RDWR | os.O_SYNC)  
100  
101     self.mem = mmap.mmap(self.mmap_file, self.length + self.virt_offset,  
102                          mmap.MAP_SHARED,  
103                          mmap.PROT_READ | mmap.PROT_WRITE,  
104                          offset=self.virt_base)  
105  
106     self.array = np.frombuffer(self.mem, np.uint32,  
107                                length >> 2, self.virt_offset)  
108  
109     def __del__(self):  
110         """Destructor to ensure mmap file is closed  
111         """  
112         os.close(self.mmap_file)
```

## - The key of productivity is the Unified



# Software-style packaging & distribution of designs

Enabled by new *hybrid libraries*

The image displays four screenshots of GitHub repositories from Xilinx, each showing a different notebook design:

- QNN-MO-PYNQ:** Shows a notebook titled "3. Open image to be classified" with Python code for image classification and an output image of a dog.
- Bot-SPYN:** Shows a notebook titled "SPYN - III phase AC motor control" with objectives, a step to download the EDDP bitstream, and code for running the design.
- PYNQ-DL:** Shows a notebook titled "Resizing an image" with a hardware block diagram of an ARM processor connected to DRAM, a PL (Programmable Logic) block containing a Resize IP and DMA, and an AXI Interconnect.
- PYNQ-ComputerVision:** Shows a notebook titled "OpenCV Overlay: Filter2D and Dilate" with a list of steps for setting up the overlay and a code snippet for running the overlay.

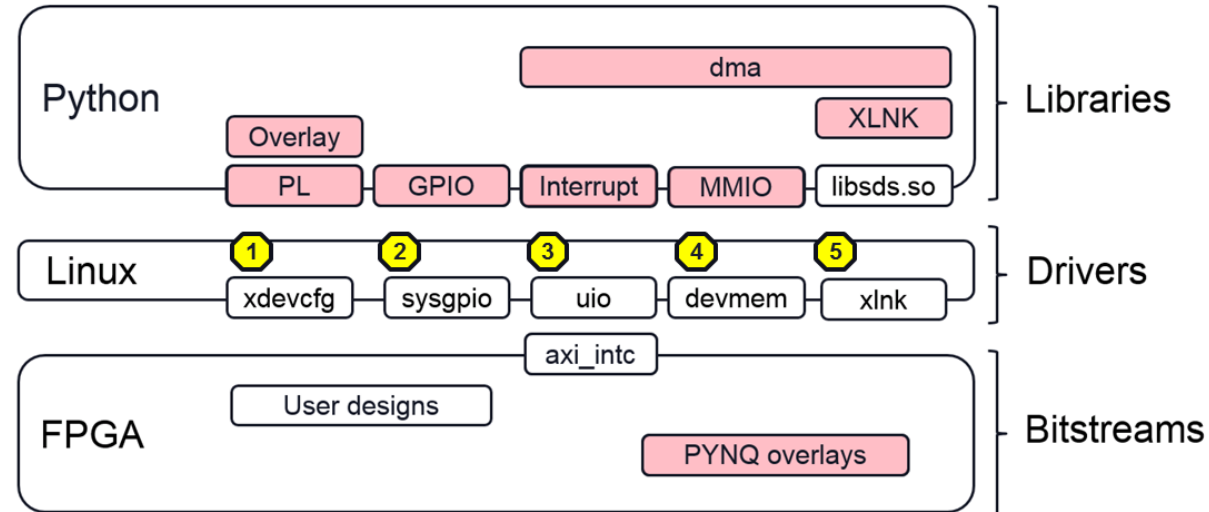
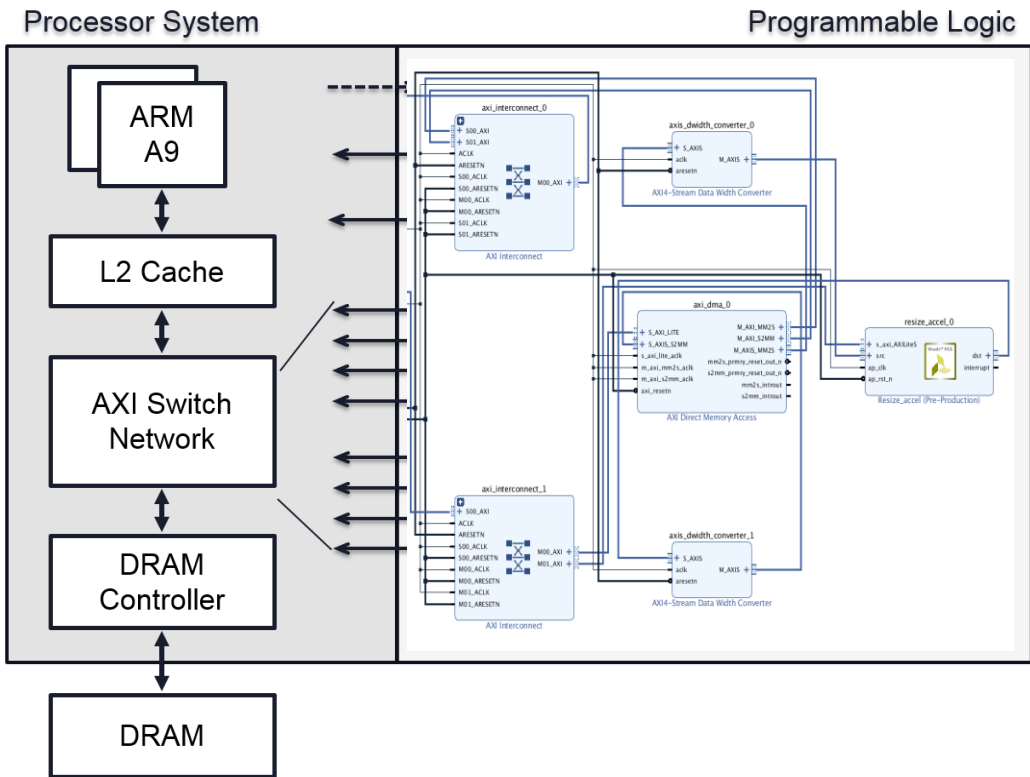
Download a design from GitHub with a single Python command:

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



# Loading a design into Zynq using PYNQ

```
from pynq import Overlay
resizer = Overlay('./resizer.bit')
```



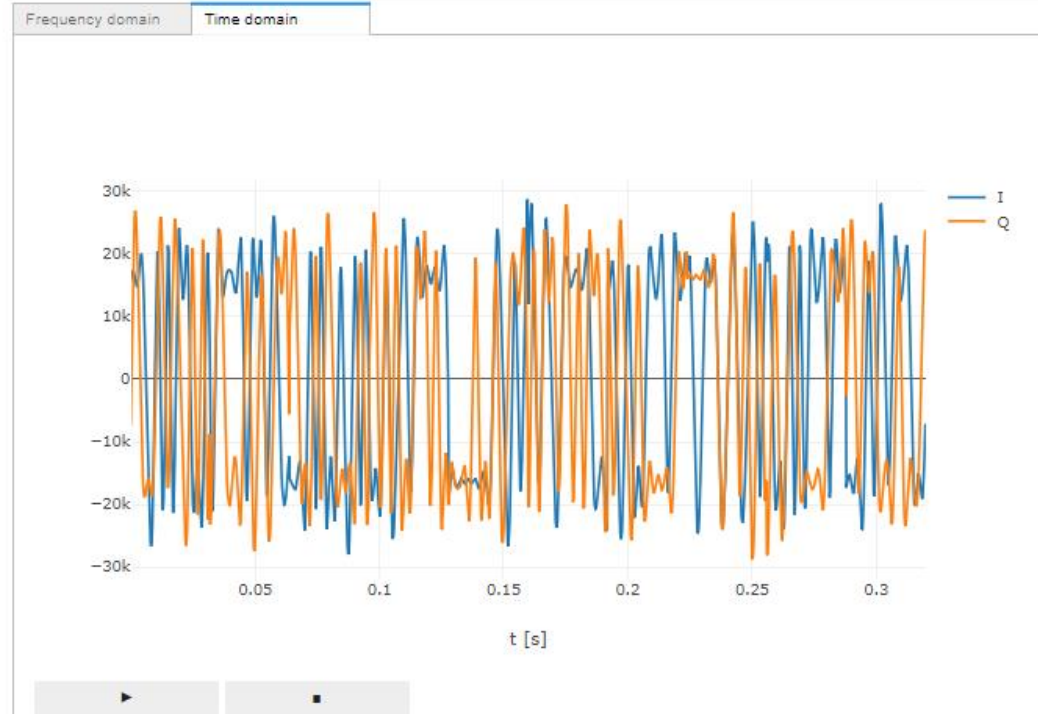
PYNQ automatically configures many design parameters based on data parsed from hybrid library

The image shows a web browser window with a dark theme. The address bar is empty. The page content features a film strip graphic with three frames showing a woman holding a camera. Below this is a search bar with the text "Search Google or type URL" and a microphone icon. Underneath the search bar is a 2x4 grid of eight gray placeholder boxes. The browser's taskbar at the bottom shows various application icons and the system tray with the date and time "8:44 AM 8/1/2018".



# Realtime and Graphic analysis

```
[7]: fs=4000  
  
iq_plot = sdr_plots.IQTimePlot(ol.qpsk_tx.get_many_shaped_time(N=10), fs, w=800)  
iq_dt = dma_timer.DmaTimer(iq_plot.add_data, ol.qpsk_tx.get_shaped_time, 0.05)  
  
fa_plot = sdr_plots.HWFreqPlot(ol.qpsk_tx.get_shaped_fft(), fs, avg_n=4, w=800)  
fa_dt = dma_timer.DmaTimer(fa_plot.add_frame, ol.qpsk_tx.get_shaped_fft, 0.3)  
  
tab1 = ipw.Tab([ipw.VBox([fa_plot.get_widget(), ipw.HBox(fa_dt.get_widget())]),  
               ipw.VBox([iq_plot.get_widget(), ipw.HBox(iq_dt.get_widget())])  
               ])  
tab1.set_title(0, 'Frequency domain')  
tab1.set_title(1, 'Time domain')  
tab1
```



**How Python helps, really a lot..**




# Ecosystem Advantage: there's a Library for that...

Standard Python comes with comprehensive libraries for common operations (web, regex, os, etc)  
In addition to this 'batteries included' strategy, there is a massive external ecosystem ...



149,433 projects    1,053,356 releases    1,433,310 files    256,727 users

 **python**™  
Package Index

The Python Package Index (PyPI) is a repository of software for the Python programming language.

PyPI helps you find and install software developed and shared by the Python community. [Learn about installing packages.](#)

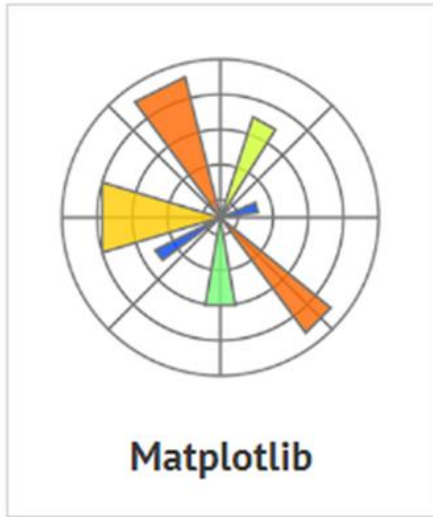
Package authors use PyPI to distribute their software. [Learn how to package your Python code for PyPI.](#)

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

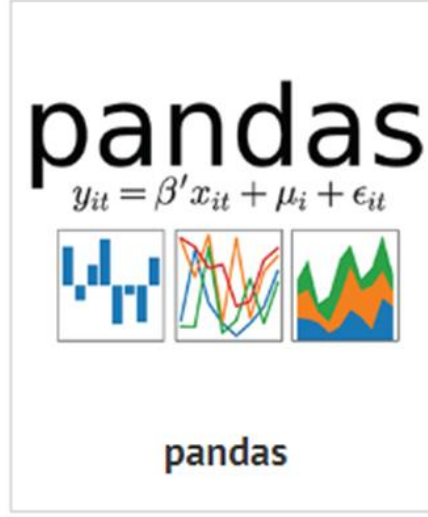
# Base Python libraries used for all Use Case



A Matlab™ like framework for **numerical computing**.



2D plotting library for **static and interactive data visualizations**



Data wrangling for easy-to-use data **ingestion, transformation, and export functions**

**Acquire,  
Transform,  
Organize,  
Display**



# What is NumPy

> **NumPy is the fundamental package for scientific computing with Python. It contains among other things:**

- >> a powerful N-dimensional array object
- >> sophisticated (broadcasting) functions
- >> tools for integrating C/C++ and Fortran code
- >> useful linear algebra, Fourier transform

> **NumPy can also be used as:**

- >> An efficient multi-dimensional container of generic data.
- >> Arbitrary data-types can be defined.
- >> This allows NumPy to seamlessly and speedily integrate with a wide variety of databases.

```
In [1]: import numpy as np

#Array multiplication...
a = np.array([2,3,4,5]) # in MatLab a = [2 3 4 5]

b = np.array([6,7,8,9]) # in MatLab b = [6 7 8 9]

c = a * b # as in MatLab
c

Out[1]: array([12, 21, 32, 45])

In [2]: #Matrix multiplication...
m1 = np.array([[1, 2, 3, 4.0], [5.1, 6.2, 7.3, 8.4], [9, 10, 11, 12]])
m2 = np.array([[1.3, 2.5, 3.7, 4.9], [5.21, 6.22, 7.33, 8.44], [12, 10, 11, 10]])

m3 = m1 * m2
m3

Out[2]: array([[ 1.3 ,  5. , 11.1 , 19.6 ],
               [ 26.571, 38.564, 53.509, 70.896],
               [108. , 100. , 121. , 120. ]])
```

# How Numpy interacts with Programmable Logic?



↑ Array, Matrix

```
In [7]: import numpy as np
import pynq

def get_pynq_buffer(shape, dtype):
    """ Simple function to call PYNQ's memory allocator with numpy attributes
    """
    return pynq.XInk().cma_array(shape, dtype)
```

Virtual memory

PYNQ™

XInk

Physical memory

Contiguous Array  
in DDR Memory  
Shared Object

Physical memory

Hard Real Time  
Offload CPU and OS  
Dedicated "Accelerator"



# Provide a Numpy Array (or Arrays) with collected data Or Pandas structure

This notebook makes the Exploratory Data Analysis:

We acquire the Data

We display the Data

We process the Data

```
In [3]: PLdata #it is the Numpy Array
```

```
Out[3]:
```

	Mot-ID	cycle	Setup1	Setup2	Setup3	la_mA	lb_mA	Total_Current	s4	s5	s6	s7	s8	s9	Acc	Freq	ttf
0	1	1	-0.0007	-0.0004	100	518.67	641.82	1589.70	1400.60	14.62	21.61	554.36	2388.06	9046.19	1.3	47.47	191
1	1	2	0.0019	-0.0003	100	518.67	642.15	1591.82	1403.14	14.62	21.61	553.75	2388.04	9044.07	1.3	47.49	190
2	1	3	-0.0043	0.0003	100	518.67	642.35	1587.99	1404.20	14.62	21.61	554.26	2388.08	9052.94	1.3	47.27	189
3	1	4	0.0007	0.0000	100	518.67	642.35	1582.79	1401.87	14.62	21.61	554.45	2388.11	9049.48	1.3	47.13	188
4	1	5	-0.0019	-0.0002	100	518.67	642.37	1582.85	1406.22	14.62	21.61	554.00	2388.06	9055.15	1.3	47.28	187
5	1	6	-0.0043	-0.0001	100	518.67	642.10	1584.47	1398.37	14.62	21.61	554.67	2388.02	9049.68	1.3	47.16	186
6	1	7	0.0010	0.0001	100	518.67	642.48	1592.32	1397.77	14.62	21.61	554.34	2388.02	9059.13	1.3	47.36	185
7	1	8	-0.0034	0.0003	100	518.67	642.56	1582.96	1400.97	14.62	21.61	553.85	2388.00	9040.80	1.3	47.24	184
8	1	9	0.0008	0.0001	100	518.67	642.12	1590.98	1394.80	14.62	21.61	553.69	2388.05	9046.46	1.3	47.29	183
9	1	10	-0.0033	0.0001	100	518.67	641.71	1591.24	1400.46	14.62	21.61	553.59	2388.05	9051.70	1.3	47.03	182
10	1	11	0.0018	-0.0003	100	518.67	642.28	1581.75	1400.64	14.62	21.61	554.54	2388.05	9049.61	1.3	47.15	181
11	1	12	0.0016	0.0002	100	518.67	642.06	1583.41	1400.15	14.62	21.61	554.52	2388.09	9049.37	1.3	47.18	180
12	1	13	-0.0019	0.0004	100	518.67	643.07	1582.19	1400.83	14.62	21.61	553.44	2388.12	9046.82	1.3	47.38	179

Your Job is done  
Customer can take from it





# Connect other Python Libraries -

**pandas**  
 $y_{it} = \beta' x_{it} + \mu_i + \epsilon_{it}$

**scikit-learn**  
 Machine Learning in Python

- Simple and efficient tools for data mining and data analysis
- Accessible to everybody, and reusable in various contexts
- Built on NumPy, SciPy, and matplotlib
- Open source, commercially usable - BSD license

**NumPy**

**BULLET PHYSICS LIBRARY**

**ROS.org**

**SimPy**  
 Discrete event simulation for Python

data analysis

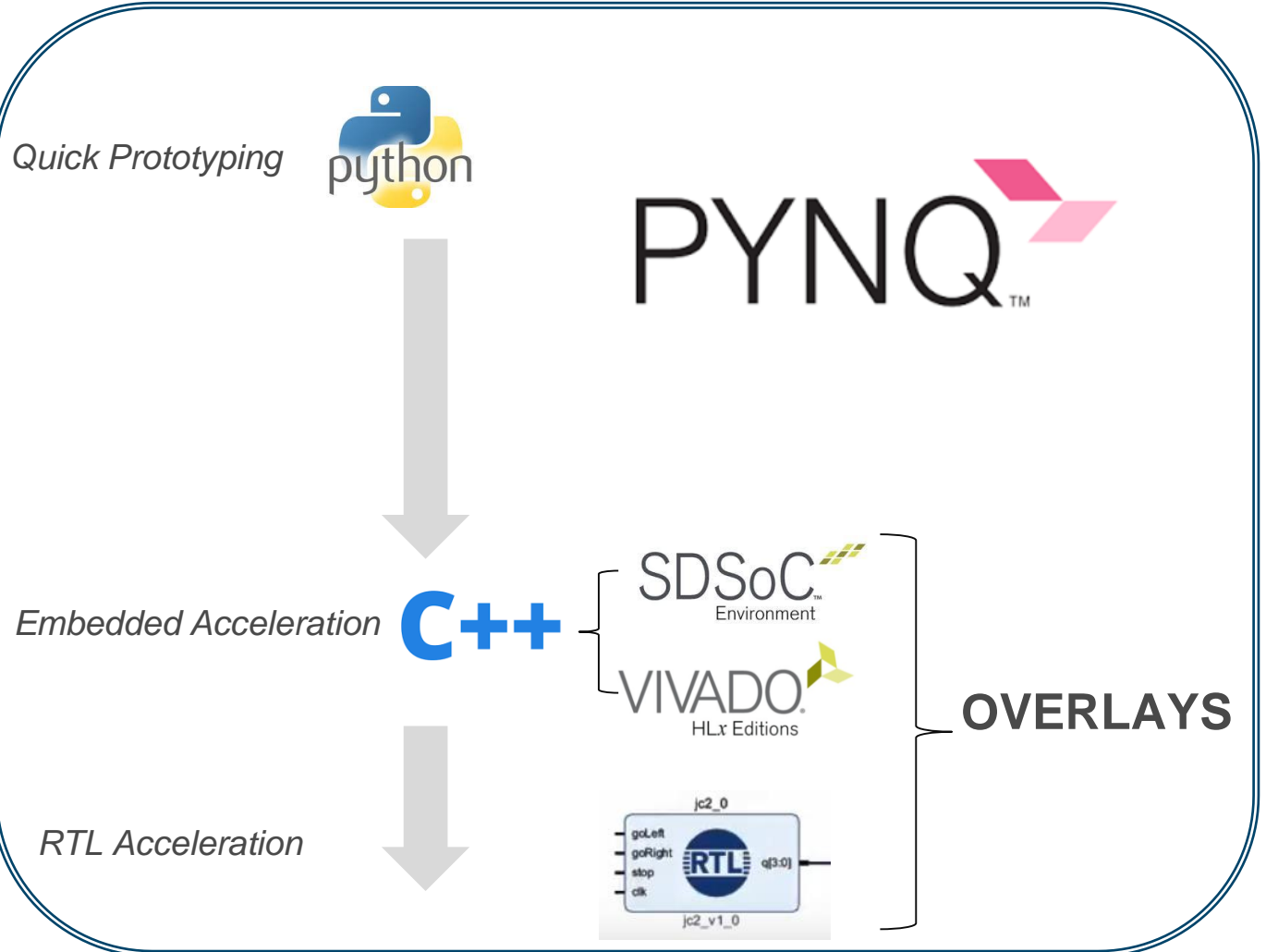
Statistical machine Learning

data analysis

sensors

robots

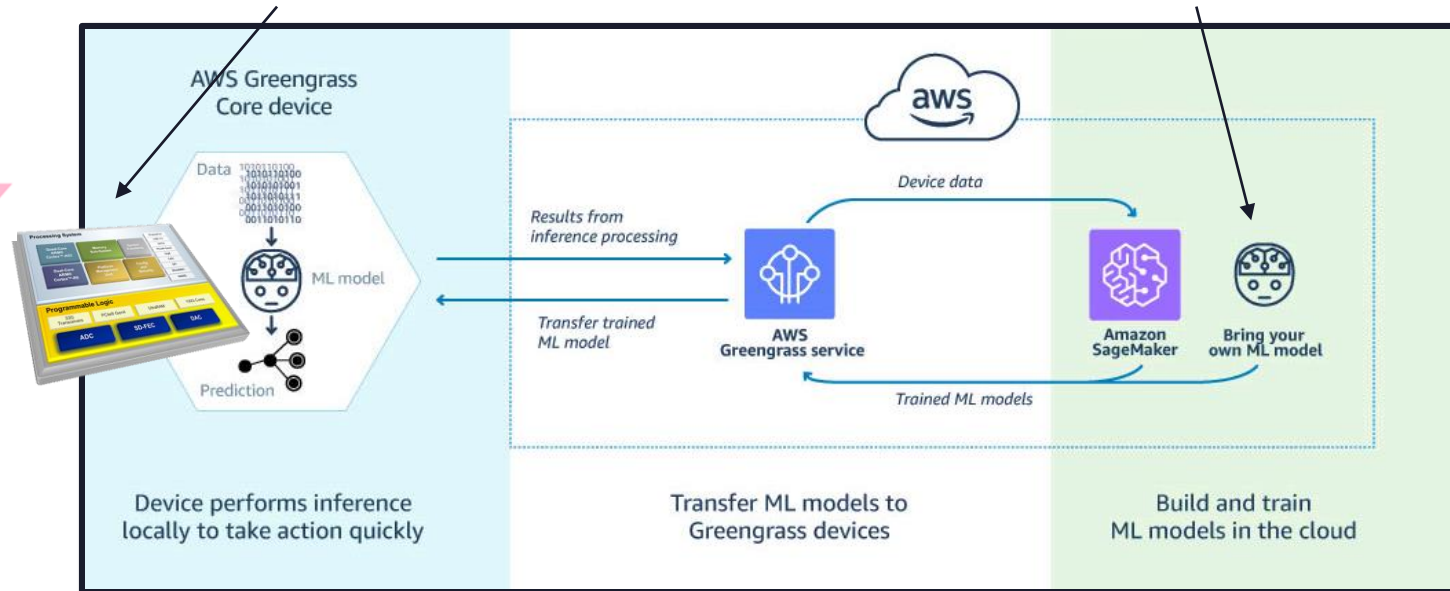
discrete event simulation



# Edge-to-cloud co-design

Common JupyterLab tooling at edge and cloud

PYNQ™

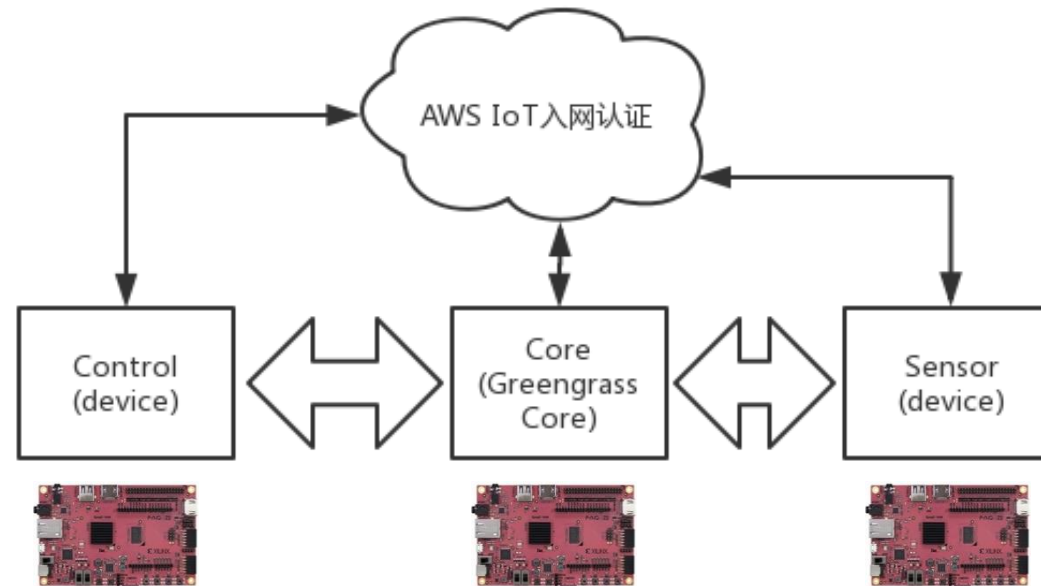
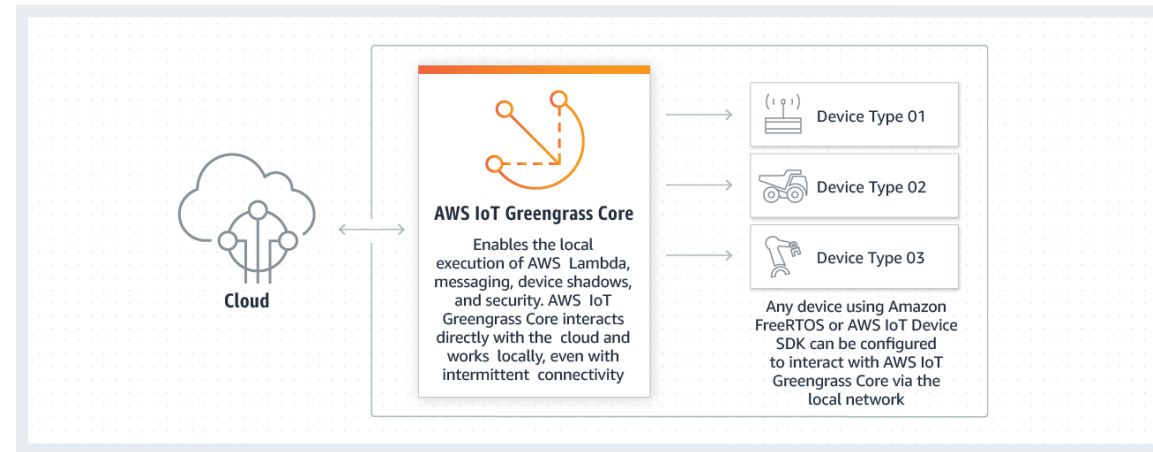


PYNQ enables ML experts and radio engineers to focus on their 'value-add'

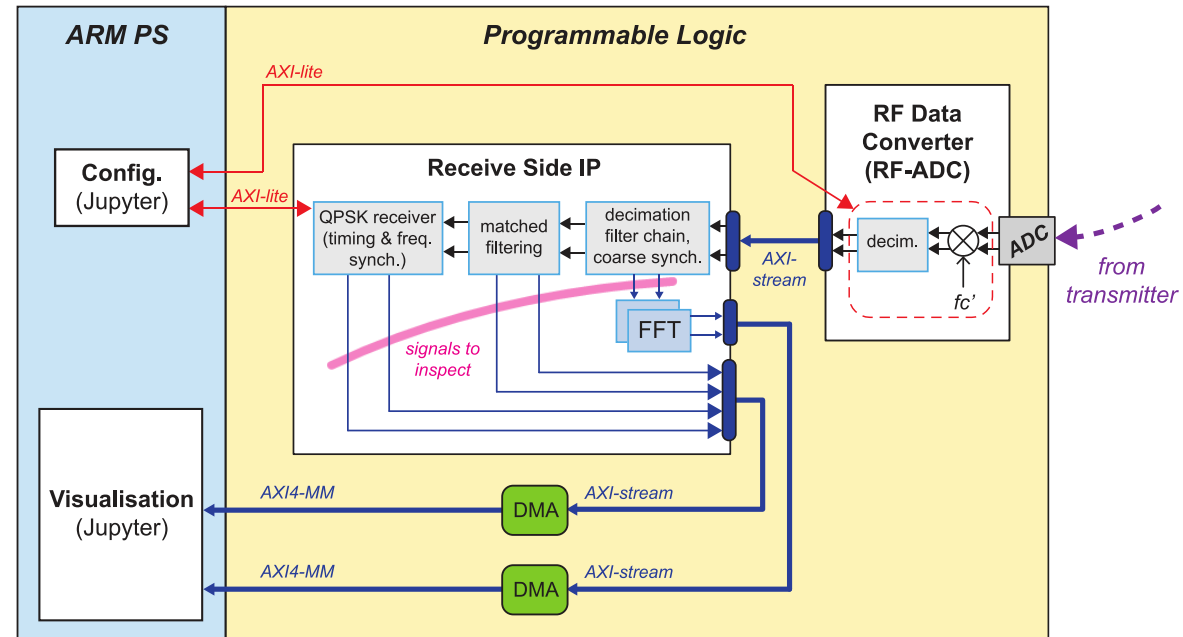
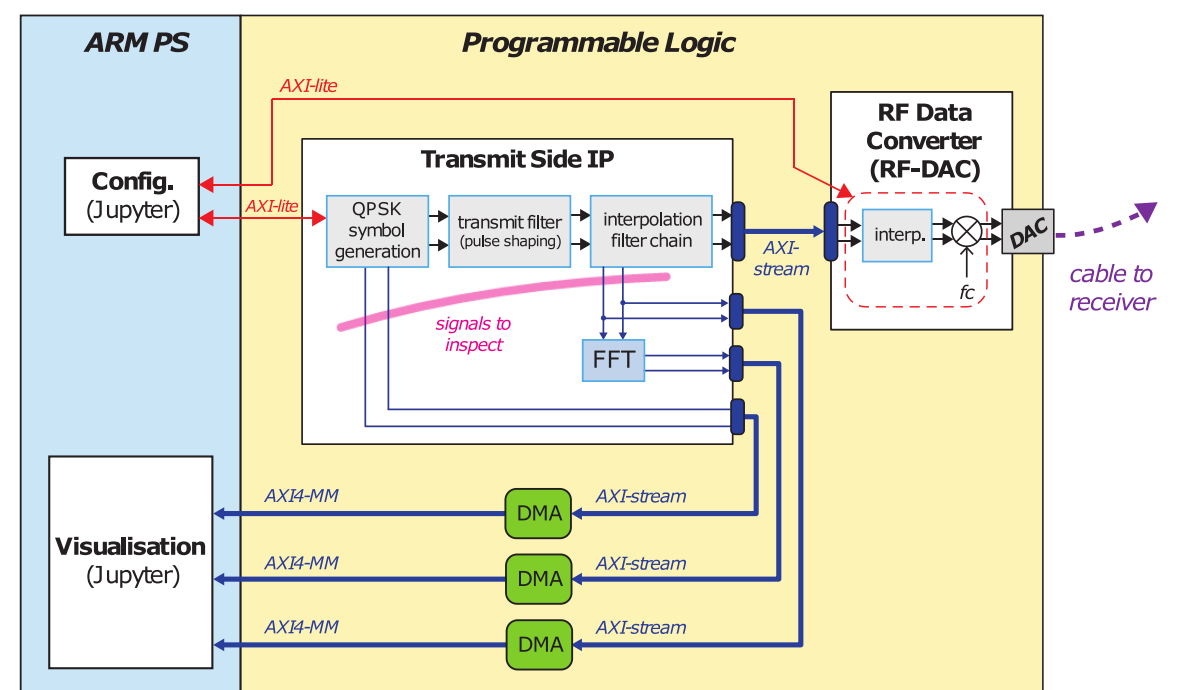
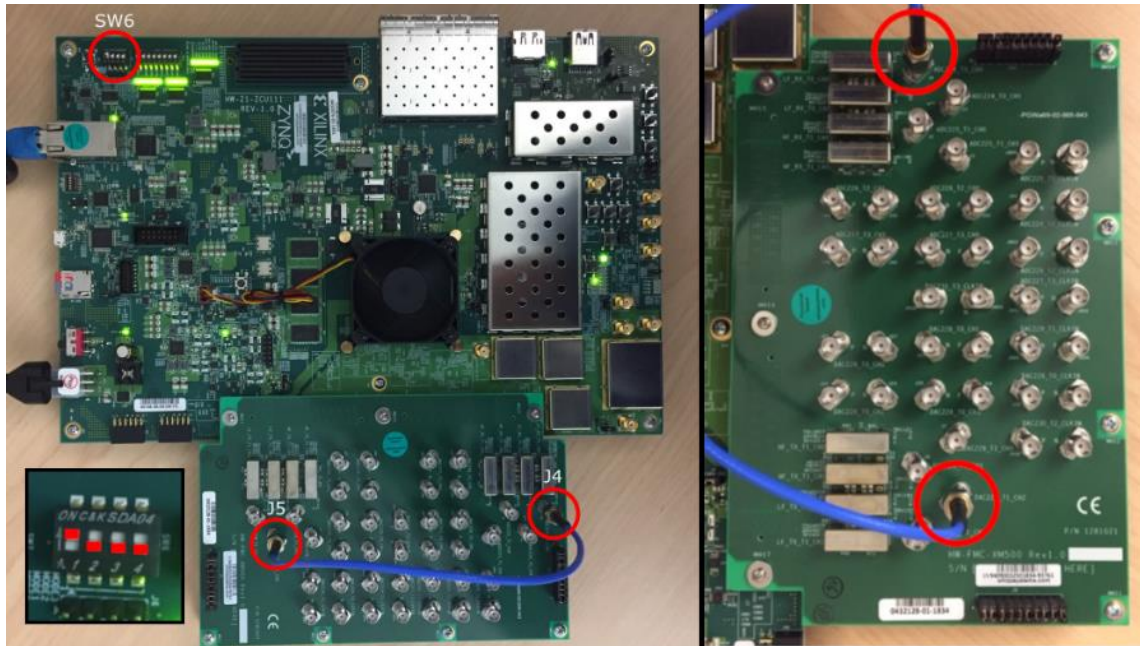
## Edge-to-cloud co-design trade-offs:

- Maximize on-chip processing
- Minimize edge-to-cloud data exchange
- Exploit scalability of cloud processing
- Aggregate intelligence between and across multiple edge nodes
- Co-optimize the above for best system performance

# AWS IoT Greengrass ( Base on MQTT )



# RF\_QPSK Demo





# Student contest designs base on PYNQ

## More and more are on the way

XILINX OpenHW

返回首页

简体中文

登录 注册

飞行器备份仪表显示系统  
该项目硬件采用惯性传感器、FPGA芯片...  
2019-03-12

Proccesed Image  
前列腺增生双极电切AI预警  
我们构建了一套前列腺增生双极电切手...  
2019-03-12

辅助救援机器人  
本作品着眼于辅助救援人员，在灾后获...  
2019-03-12

基于 PYNQ 的智能多媒体播放器  
本作品着眼于表情识别技术，语音识别...  
2019-03-12

基于 FPGA 的机器博弈五子棋游戏  
人工智能是近年来很活跃的研究领域之...  
2019-03-12

微光图像实时处理系统  
利用 PYNQ 实现微光图像的实时增强。...  
2019-03-12

基于FPGA的DDE算法运算加速  
2019-03-12

基于 FPGA 的绘图机械臂  
本作品着眼机器视觉，集合图像技术、...  
2019-03-12

基于 FPGA 的双目测距仪  
从目前设备的发展情况出发，在普通双...  
2019-03-12

基于PYNQ的双目机器视觉系统  
本文设计了一个低成本、高可靠性、便...  
2019-03-12

基于PYNQ-Z2的地图轮廓识别拼...  
我们采用了提取图像轮廓特征的方法，...

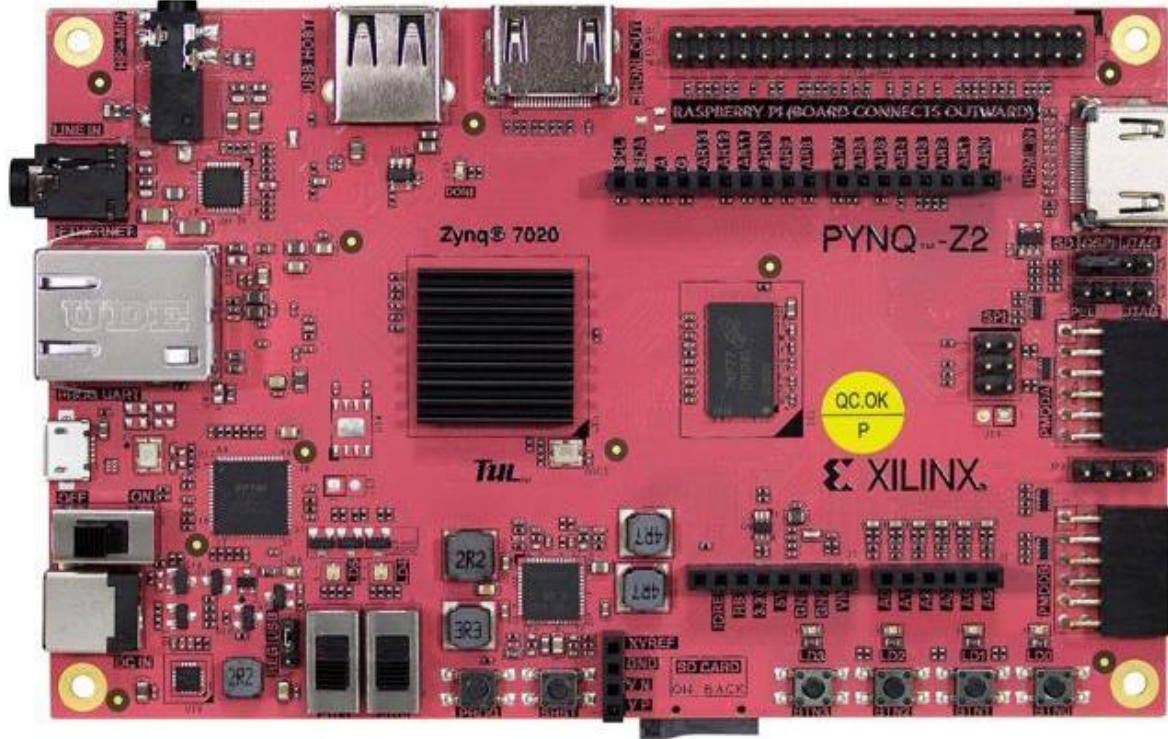
基于背景重建的运动检测系统  
作品基于赛方提供的 PYNQ-Z2 开发板...

基于神经网络的语音识别系统  
用神经网络的算法去代替传统的算法去...

非接触式静脉影像增强现实眼镜

基于FPGA的智能轮椅控制系统

# PYNQ-Z2 Board -design for starter



\$119 to everyone in US

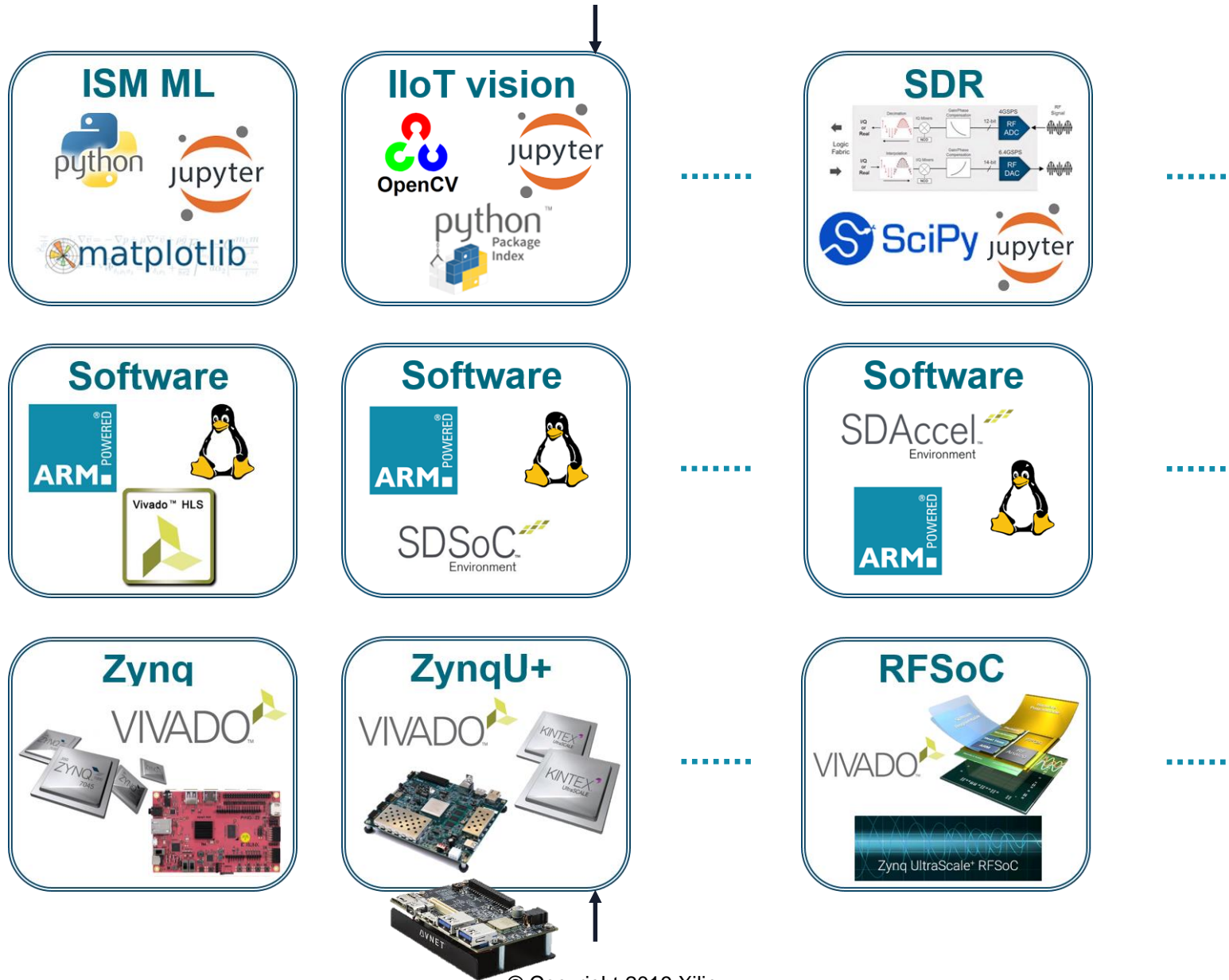
- New PYNQ reference platform
- New stereo audio with on-board codec
- New Raspberry Pi connector
- Open source design
- Z2 manufactured in Taiwan by TUL
- Distributed globally by Premier Farnell
- Also Newegg in US
- Academic discounts & donations available

# PYNQ on other Boards

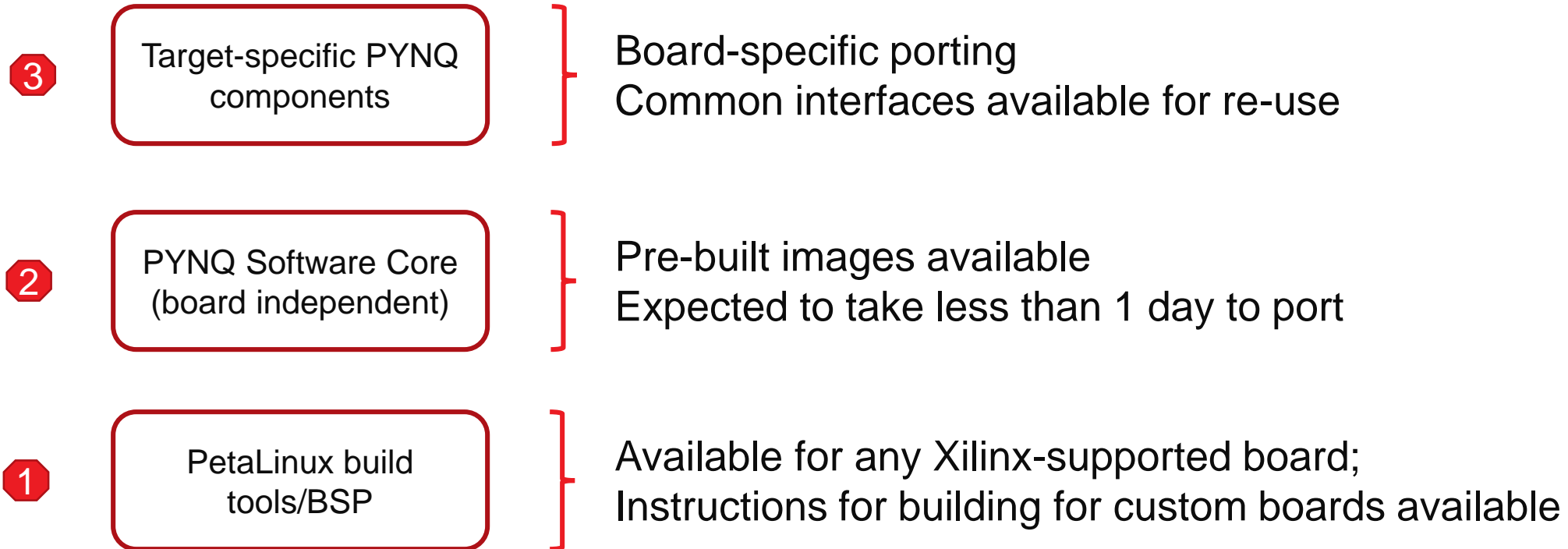




# Next steps: scaling across platforms and domains



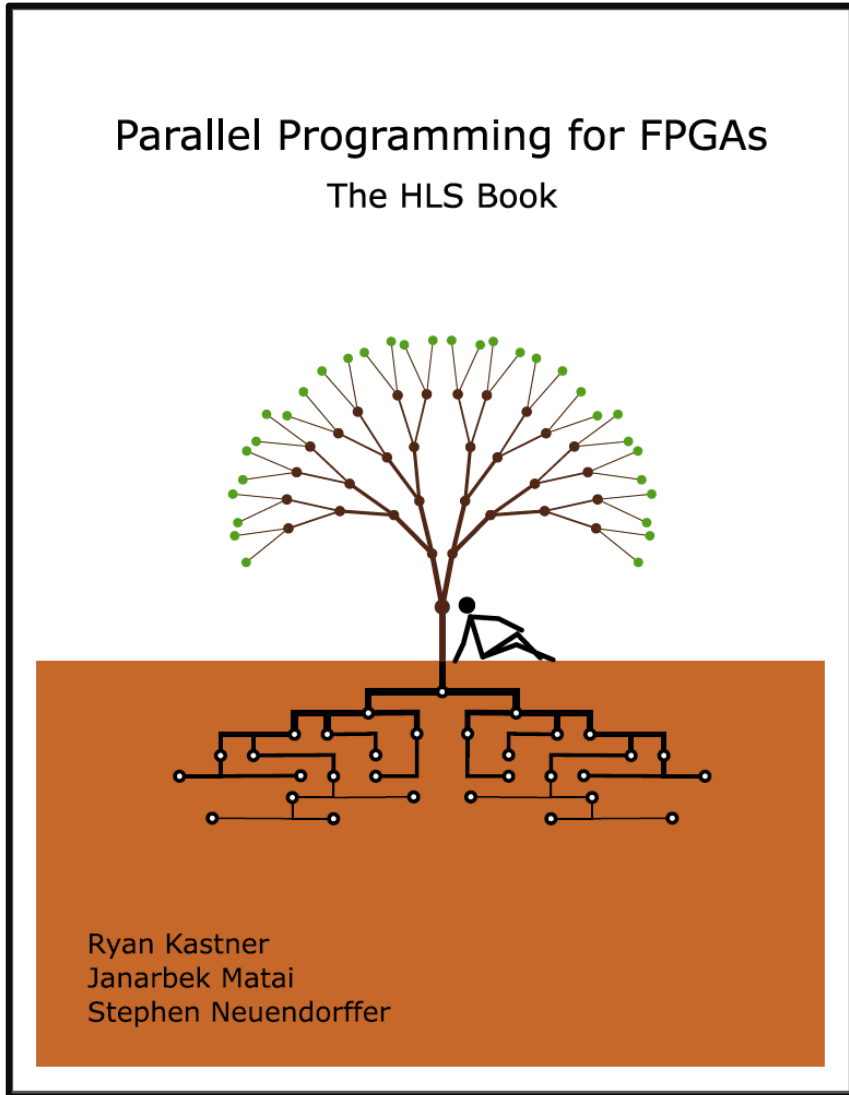
# Efficient porting PYNQ to any Zynq-based platform





A grid of 16 project highlights, each with a title, description, and a small image or screenshot. The projects include: spoonNN (ETH Zurich), iSmart DNN (FGA-based neural network), TGIIF (1st place in the DAC 2018 design contest), cv2PYNQ (FAU, Accelerated OpenCV image filtering library), Video processing (KU Leuven), ZipML-PYNQ (ETH Zurich), PYNQ bot (IT Tallaght), PYNQ LED cube (Fudan University), SPYN (Xilinx ISM), PYNQ networking (Xilinx labs), QNN (Xilinx labs), LSTM (TU Kaiserslautern), BNN (NTNU), Video filters with PR (BYU), PYNQ computer vision (Xilinx labs), GZip on PYNQ (University Bucharest), SPynq (NTUA), FIR filter example (CU Boulder), CNN on PYNQ (Imperial College London), and VectorBlox (HDMI Video processing).

# New open source HLS book



*Parallel Programming for FPGAs* is an open-source book aimed at teaching hardware and software developers how to efficiently program FPGAs using high-level synthesis



Reply “pp4fpgas” in wechat console

<http://kastner.ucsd.edu/hlsbook/>

**Adaptable.**  
**Intelligent.**

