智能硬件
PYNQ入门

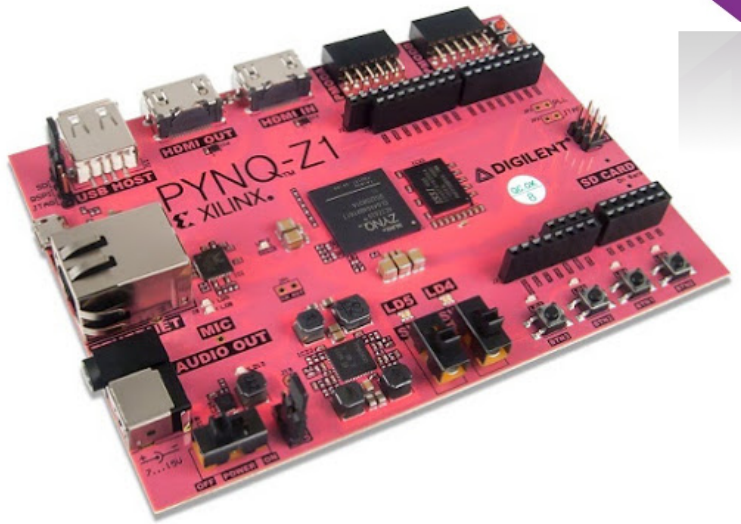
清华大学iCenter
闫泽禹

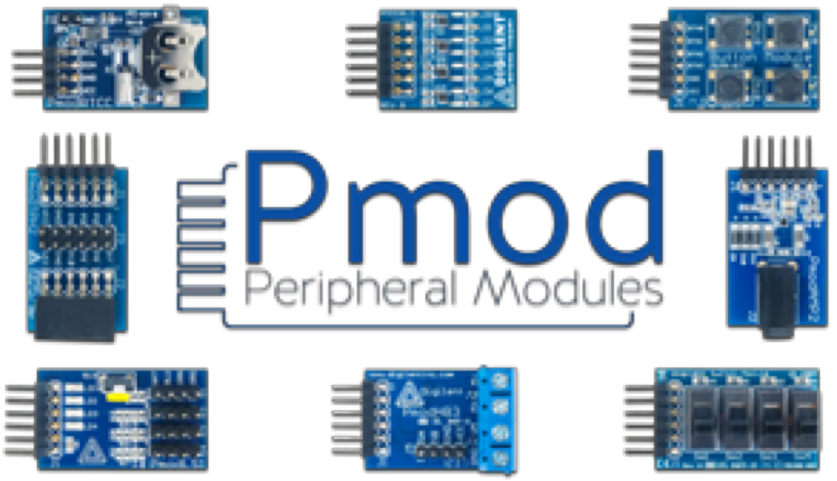
基础工业训练中心智能系统实验室
http://net.icenter.tsinghua.edu.cn

PYNQ是什么

- 是一款基于Python编程语言的智能硬件开发板**Python for Zynq**
- 对初学者友好，应用范围广
- 可结合Pmod系列传感器，轻易实现多样化的嵌入式功能
- 可与其他智能硬件开发板连接，实现更强大的功能
- 支持传统的硬件开发板的工具对其加强

The PYNQ-Z1 Board





PYNQ能干什么

PYNQ应用案例1

基于Xilinx 全可编程的 Zynq® SoC的深度学习无人机应用全球首秀：零度智控与深鉴科技合作智能无人机亮相CES

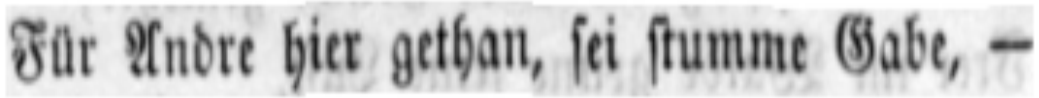
日期：2017-01-10　作者：佚名

- 实现重要分析与硬件加速
- 单个器件上高度集成多种功能
- 实时的多人检测，姿态识别，追踪等
- 深鉴科技Aristotle 深度学习处理器
  - 3W 左右的功耗
  - 每秒 2300 亿次深度学习运算性能

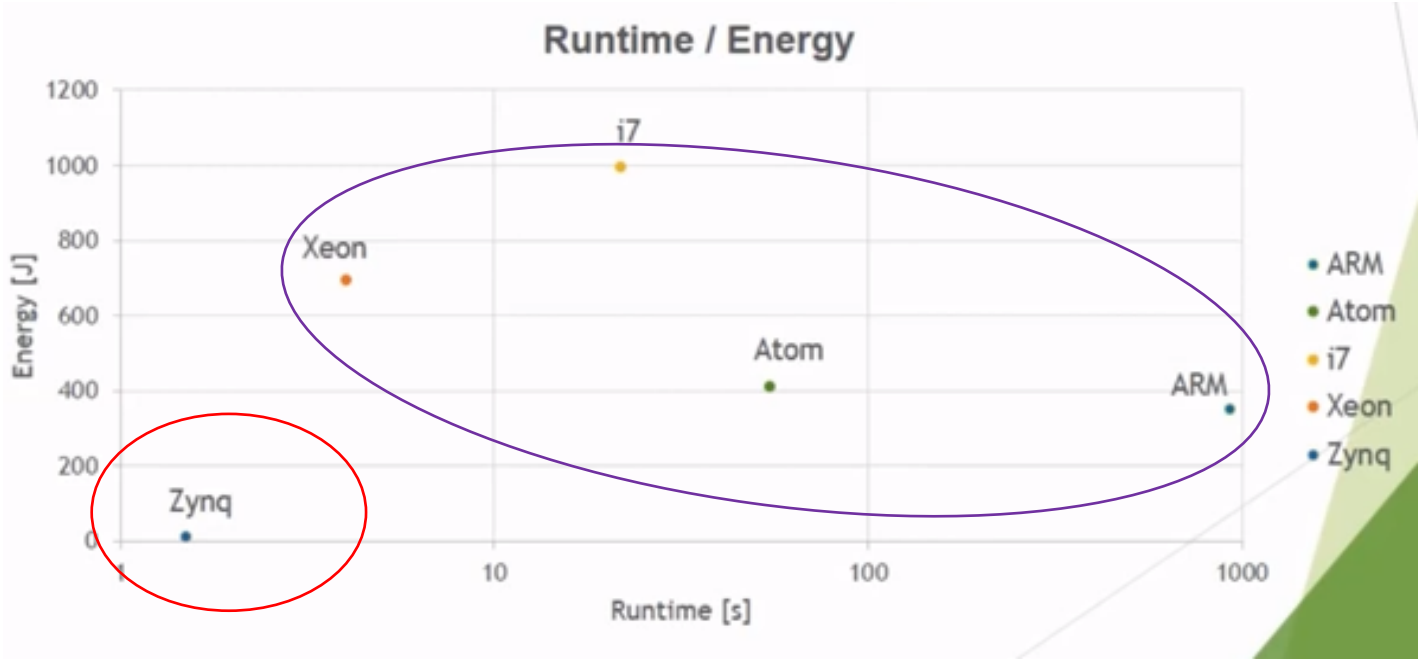
PYNQ应用案例2

- FPGA加速的BiLSTM神经网络的光学字符识别(OCR)
- 低功耗和高准确率(98.34%)

Out[2]:




```
HW OCRed text: Für Andre hier gethan, sei stumme Gabe, –
HW MOps/s: 54833.779575112705
HW inference time [ms]: 1.7328369617462158
```



PYNQ应用案例3

- HD肌电假肢控制器

➢使用256个HD肌电通道并达到<1毫秒的控制器延迟

➢效果好于纯软件的解决方案

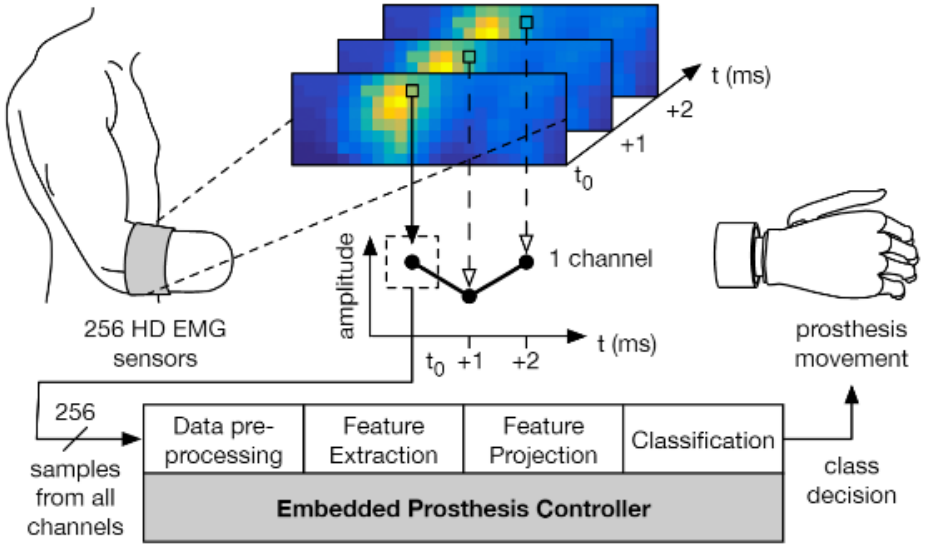
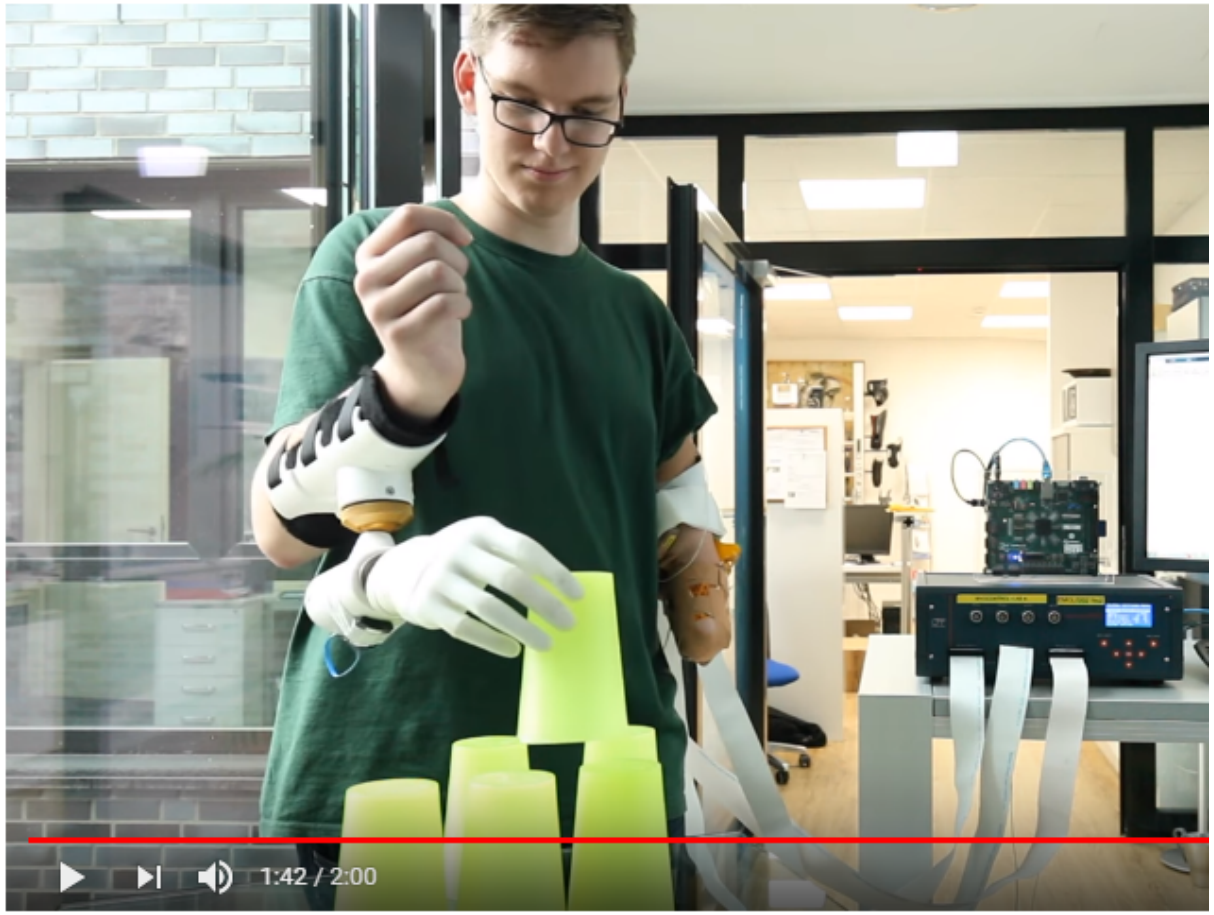


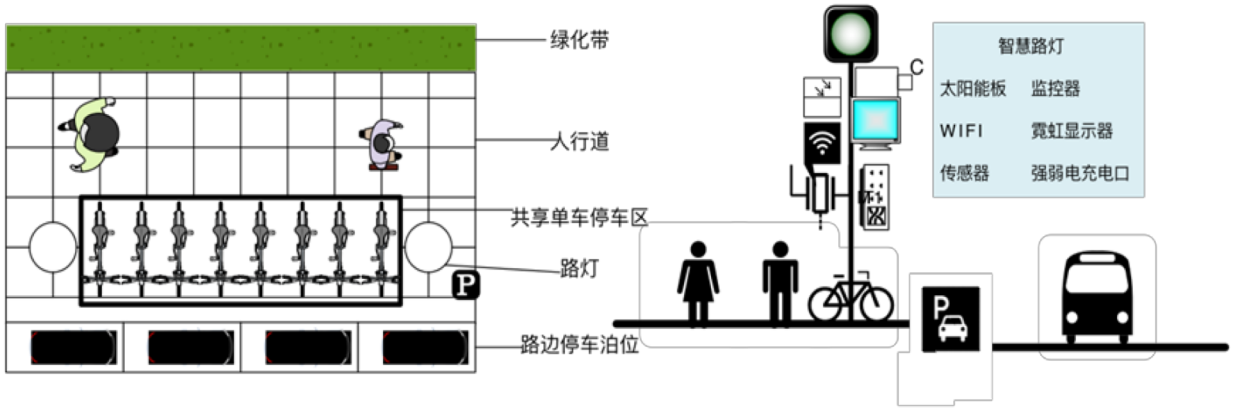
Figure 1: ZYNQ-based prosthesis controller: HD EMG input signals, four processing steps on ZYNQ System-on-Chip. Class decision as controller output.



https://www.youtube.com/watch?v=RZGHkCOCRC4&feature=youtu.be

PYNQ应用案例4

- 共享单车电子围栏系统
  - 共享单车的停车区域用于集中停靠、管理共享单车
  - 放入其中的共享单车会被检测停放是否规范，如果不规范会被检测到并提醒租借人摆放整齐
  - PYNQ内置OpenCV库，处理摄像头传回图像并返回租赁人相关信息

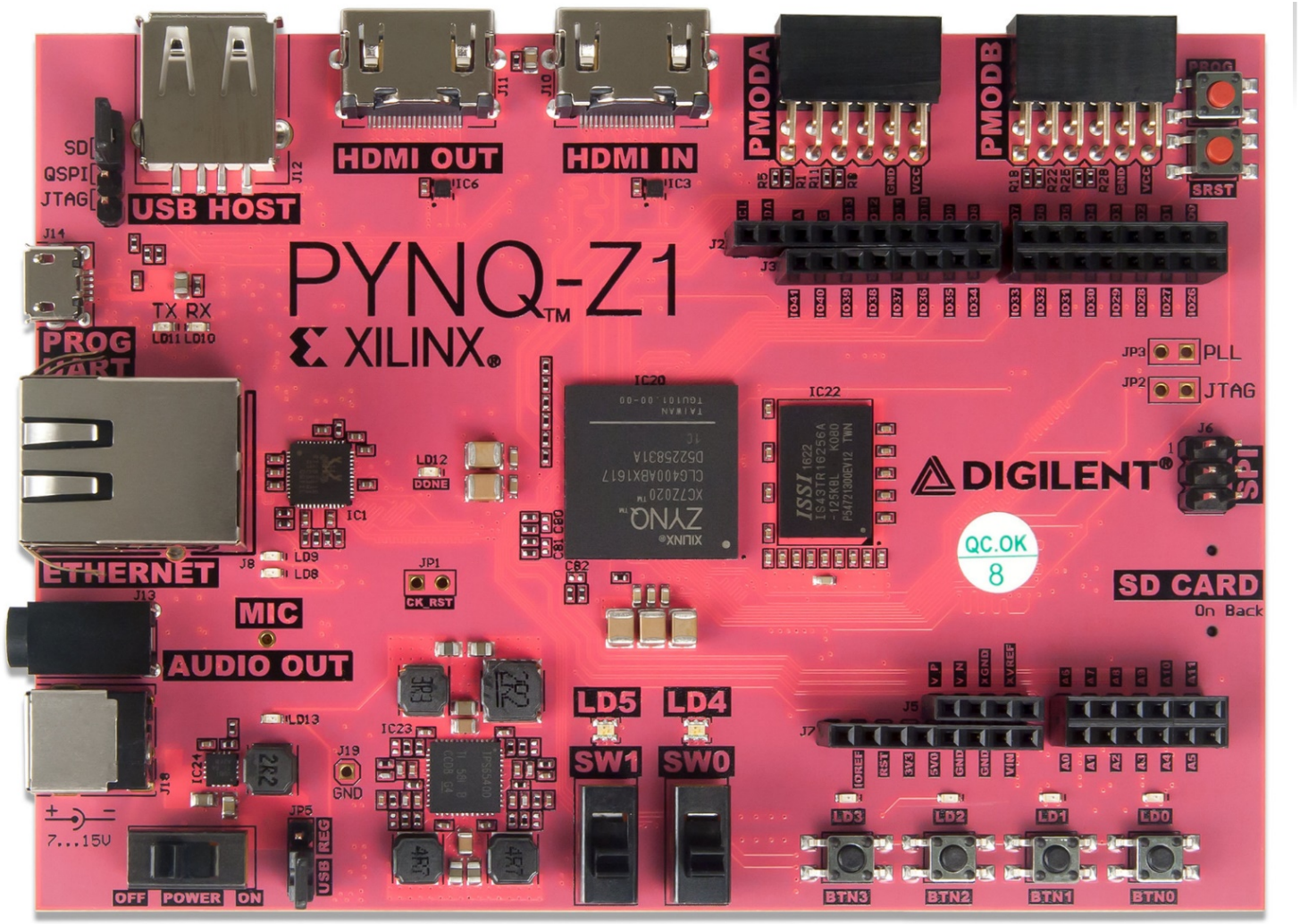


PYNQ应用案例5

- 比特币挖掘系统 By California State Univ.
- FPGA配置更加灵活
- Zynq价格低廉
- 日后加入更多core挖掘

PYNQ基本硬件结构

- Power
  - from USB or any 7V-15V source
- USB and Ethernet
- Audio and Video
  - Electret microphone
  - 3.5mm mono audio output jack
  - HDMI sink and source port
- Switches, push-buttons, and LEDs:
  - 4 push-buttons, 2 slide switches, 4 LEDs, 2 RGB LEDs



PYNQ基本硬件结构

- Expansion Connectors:
  - Two standard Pmod ports
    - 16 Total FPGA I/O
  - Arduino/chipKIT Shield connector
    - 49 Total FPGA I/O
    - 6 Single-ended 0-3.3V Analog inputs to XADC
    - 4 Differential 0-1.0V Analog inputs to XADC

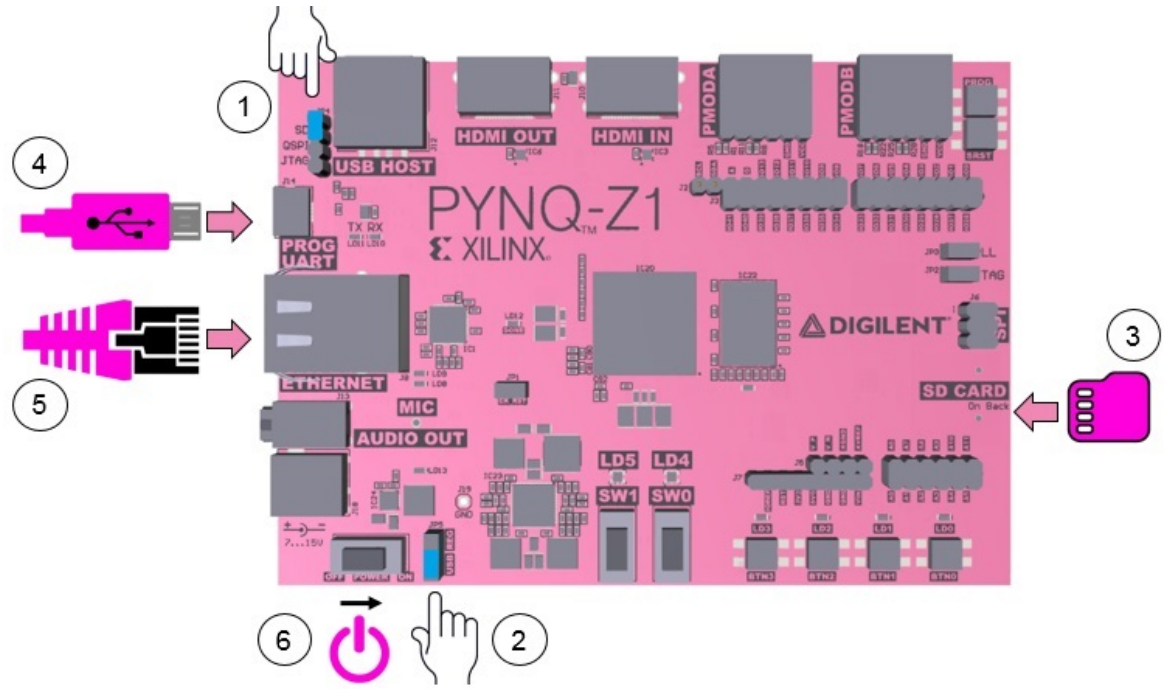
※XADC:zynq芯片内部进行温度和电压检测的模块

如何将PYNQ板连接到电脑上

- 准备材料：
  - 带网口的笔记本电脑，安装有Chrome，Safari，Firefox浏览器之一
  - PYNQ板，系统SD卡
  - 网线，Micro USB充电线

- 连接方法(windows系统):

1. 对PYNQ板做调整:
   - ✓SD卡启动模式
   - ✓USB供电模式



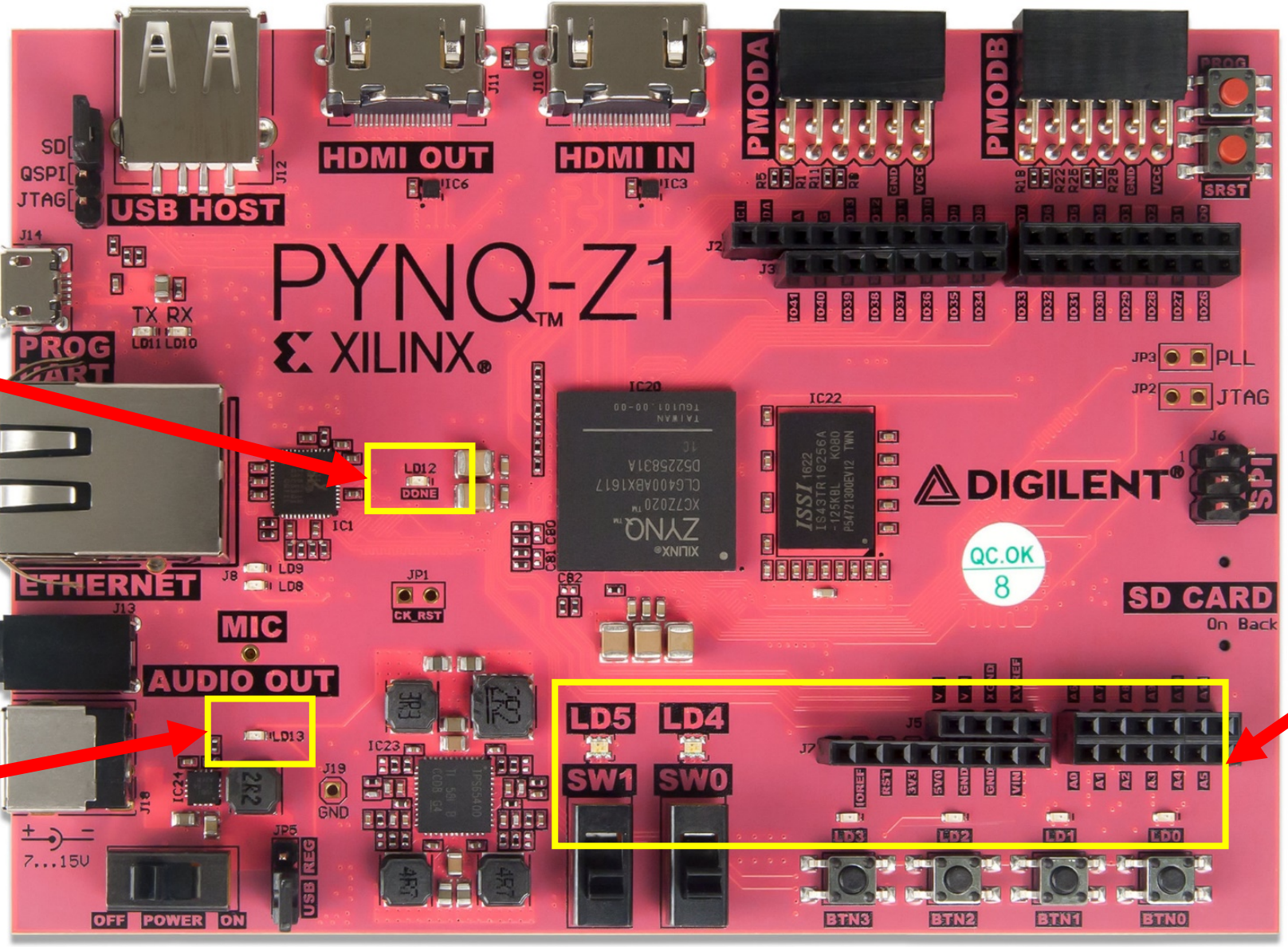
2. 将PYNQ板连接到电脑上

- 控制面板-网络和Internet-网络和共享中心
- 找到相应的以太网网络名称，右键点击菜单选择“属性”
- 选择IPv4并打开属性
- 更改IP地址为192.168.2.1，子网掩码为255.255.255.0并确定

3. 打开PYNQ板

- ✓LD13 LED变亮，说明已通电
- ✓稍后，LD12 / Done LED变亮，说明设备工作状态
- ✓LD4 & LD5 LEDs 和LD0-LD3 LEDs 同时亮起
- ✓随后LD4 & LD5 LEDs 灭掉，LD0-LD3 LEDs则保持不变
- ✓系统已经启动成功准备好

(2)PYNQ设备准备就绪



(1)PYNQ已通电

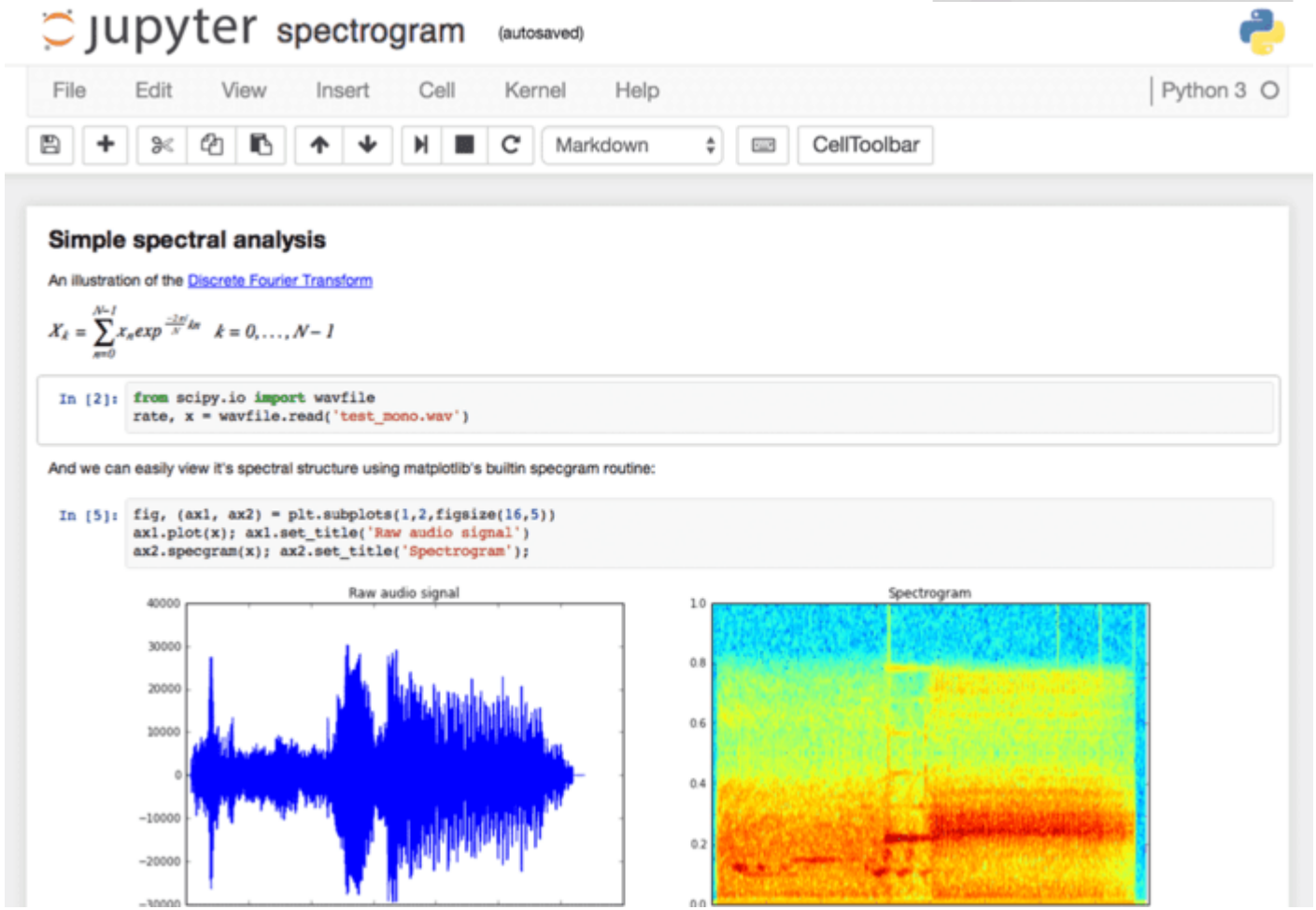
(3)LD5和LD4熄灭后，系统准备完成

如何将PYNQ板连接到电脑上

- 连接到Jupyter Notebooks
  - 打开电脑的浏览器
  - 输入IP地址http://192.168.2.99:9090
  - 登陆名称和密码均为xilinx
- Last Step, START CODING！

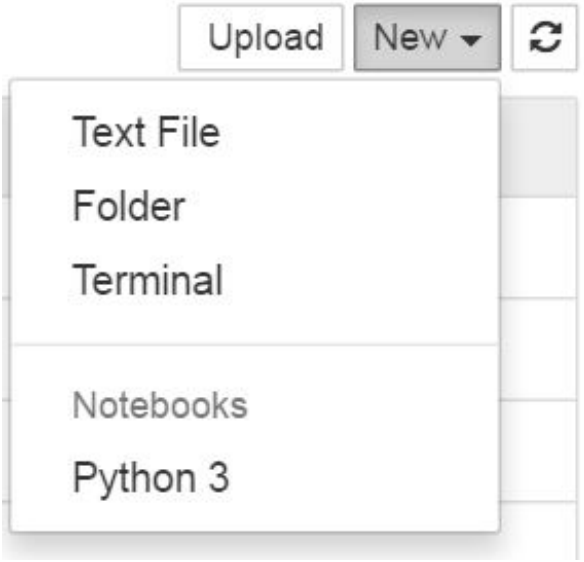
PYNQ编程环境

- Jupyter Notebook
  - 主要为Python而生
  - 基于浏览器的编程环境
  - IPython 内核和库
  - 提供许多examples
- Linux系统
- 包含FPGA的基本库与接口



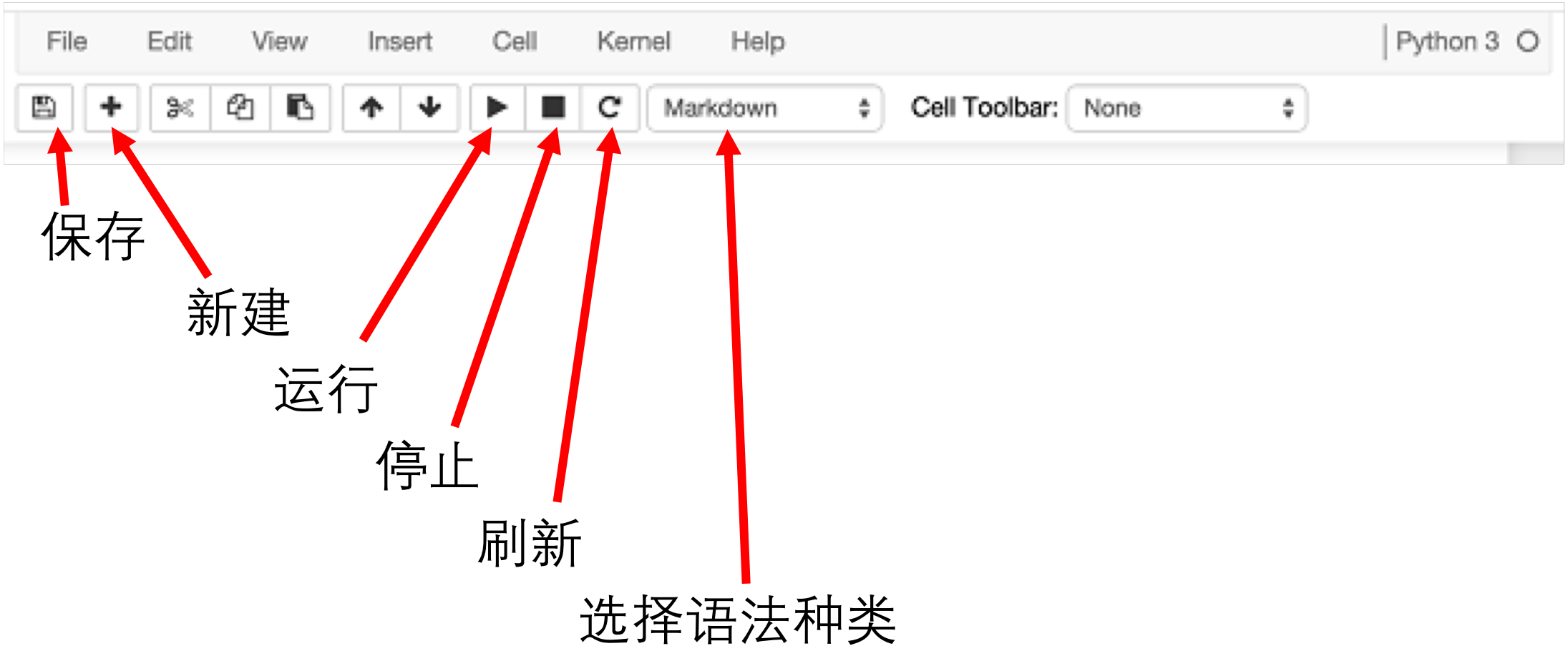
Python编程环境

- Jupyter Notebooks
  - 一个在浏览器中的交互式编程环境，支持python和markdown语法
  - 在浏览器中即可编辑和运行代码，实时查看代码运行结果
  - 如何新建一个文件:
    - 可从下落菜单中选择
      - 文档
      - 文件夹
      - python文件



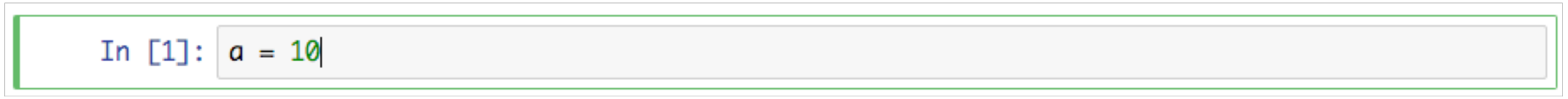
Jupyter Notebook

- 菜单栏：



Jupyter Notebook

- Cell简介
  - 代码以cell为单位运行，In[]表示输入



  - cell可使用markdown或LaTeX语法，多样化文档内容
  - 运行cell使用Shift+Enter或按菜单栏上的播放键
  - 输出或报错信息会紧跟在In cell后面显示

※程序无反应时可使用菜单栏中的Kernel-Restart Kernel强行结束程序

Python基本介绍

- 一种解释型、面向对象、动态数据类型的高级程序设计语言
- 支持交互式编程: Jupyter Notebooks
- 清晰的缩进格式，易于阅读

- 拥有大量的第三方package
  可以完成各种各样的功能:
  - Numpy, opencv-python, matplotlib…
  - 使用import命令在代码开头导入相应的package

```
import math       #入导math模块
math.floor()      #调用math模块中的floor()函数
```

```python
#!/usr/bin/env python
class Fibs():
    def __init__(self):
        self.a = 0
        self.b = 1
    def __iter__(self):
        return self
    def next(self):
        fib = self.a
        self.a,self.b = self.b,self.a + self.b
        return fib
if __name__ == "__main__":
    fibs = Fibs()

    for f in fibs:

        if f > 7000:
            print f
            break
```

图: 输出斐波那契数列

PYNQ的python基本库介绍

- 基础modules:
  - pynq.ps: 管理处理系统和PS/PL接口
  - pynq.pl: 管理可编程逻辑
  - Pynq.overlay: 管理状态，驱动和表层的内容
  - pynq.gpio: 管理通用IO接口
  - pynq.interrupt: 管理PYNQ异步IO

- Sub-packages:
  - pynq.lib: 包括所有的Pmod，Arduino和其他通信控制器的驱动
    - 也是我们的主要使用package

- pynq.lib package分为两部分
  - 自带硬件modules和扩展modules
- 自带:
  - .audio，.button，.led，.rgbled，.switch，.video等等
- 扩展: .arduino, .pmod

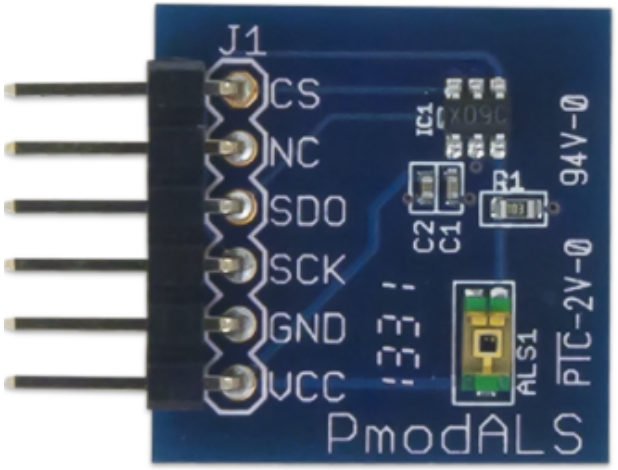
- 例: 如何让PYNQ亮起来
  - 导入board里的类定义
  - 实例化各个类
  - 使用完后关闭led
  - delete实例

In [1]:
```
from time import sleep
from pynq.board import LED
from pynq.board import RGBLED
from pynq.board import Button

Delay1 = 0.3
Delay2 = 0.1
color = 0
leds = [LED(index) for index in range(4)]
rgbleds = [RGBLED(index) for index in [4,5]]
btns = [Button(index) for index in range(4)]
```

```
for led in leds:
    led.off()
for rgbled in rgbleds:
    rgbled.off()

del leds,btns
```

- 智能路灯
  - 未来的路灯不仅可以照明，还可以根据环境的亮暗程度决定是否开灯并自动调节灯的亮度
  - 既省去了人工成本，又提高了利用率
  - 而对路灯亮度的准确调节离不开对环境光的检测，环境光传感器(Ambient Light Sensor)是必要的

  - ALS: 传感器的光电流与环境光强度成正比
  - 通过对电流的检测，实现对灯光的控制



深圳LED智能路灯现身街头！可连wifi 还能显示车位、路况！

中国经济新闻网 2017-11-07 11:15:23

近日，在深圳沙河西路转深圳湾科技生态园的入口处，树立起了两根颇具科技感的路灯杆，这路灯杆名为"智慧道路共同杆"，多杆合一，功能强大~!设备都在杆里面，外人看不到，被业界称为路灯杆中的隐形"钢铁侠"。未来的智慧道路只须树一根杆，就能实现几乎所有功能。

- 智能路灯的实现
- 如何使用Pmod_ALS传感器监测环境光

```
In [1]: from pynq import Overlay
        Overlay("base.bit").download()

In [2]: from pynq.iop import Pmod_ALS
        from pynq.iop import PMODB

        # ALS sensor is on PMODB
        my_als = Pmod_ALS(PMODB)
        my_als.read()
```

① 导入overlay初始化硬件系统

② 导入pmod传感器驱动模块和其连接的pmod接口驱动

③ 初始化pmod_als传感器实例并读入数据

④ 对数据进行处理并显示变化图像

◆ 使用完成后关闭信号记录

◆ 不需要回收传感器的代码

※ For more: http://pynq.readthedocs.io/en/latest/

谢谢

基础工业训练中心智能系统实验室（编制）

http://net.icenter.tsinghua.edu.cn