编号（No.）：（IHEP-）HEPS-TS-CB-TS-2022-001

**pythonSoftIOC**

**摘 要**

Abstract

|  |  |
| --- | --- |
| **起 草（Prepared by）** | 张群 |
| **校 核（Checked by）** |  |
| **会 签（Concurred by）** |  |
| **审 核（Reviewed by）** |  |
| **审 定（Verified by）** |  |
| **批 准（Approved by）** |  |

中国科学院高能物理研究所

Institute of High Energy Physics

Chinese Academy of Science

2022年06月

北京

*（This is an internal publication and does not necessarily reflect the views of the HEPS management.）*

# **一：安装过程:**

#### python3 --version（python 3.7以上）

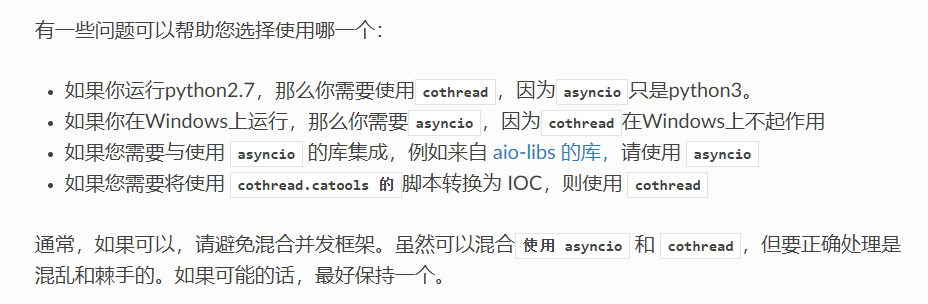
#### python3 -m pip install softioc

#### python3 -m pip install cothread

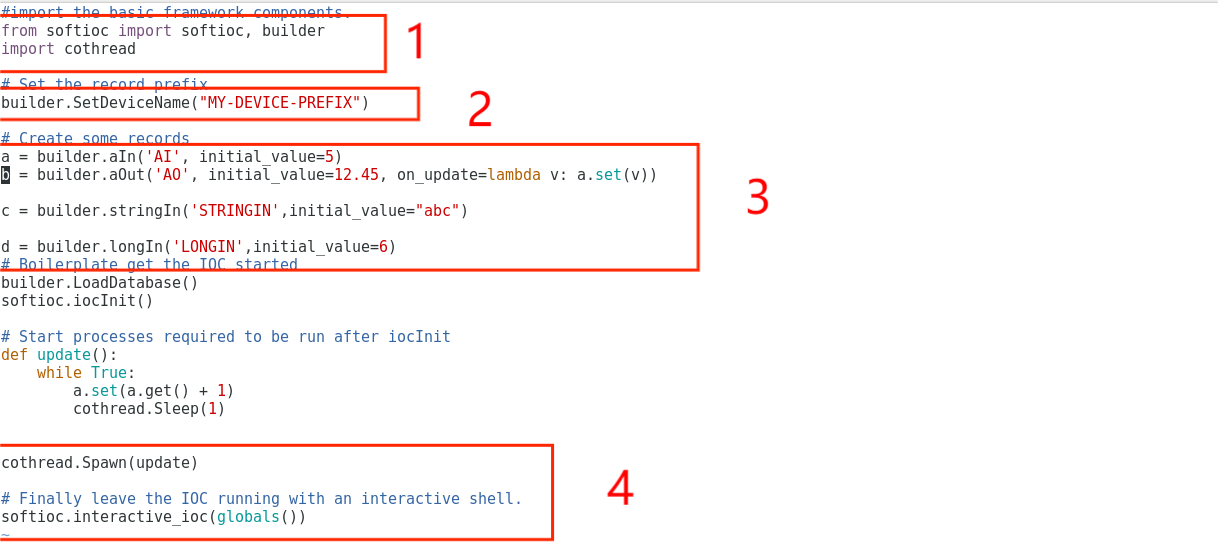
#### pythonSoftIOC --version

# 二：创建IOC （两种方式-cothread和asyncio）

1. 创建方式选择：



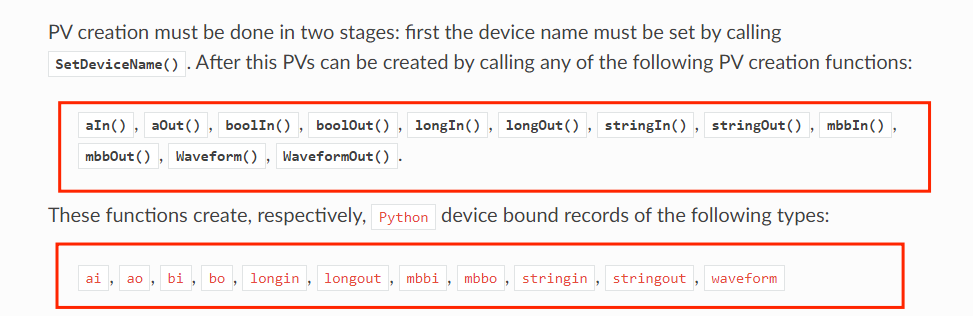
（2）cothread方式：



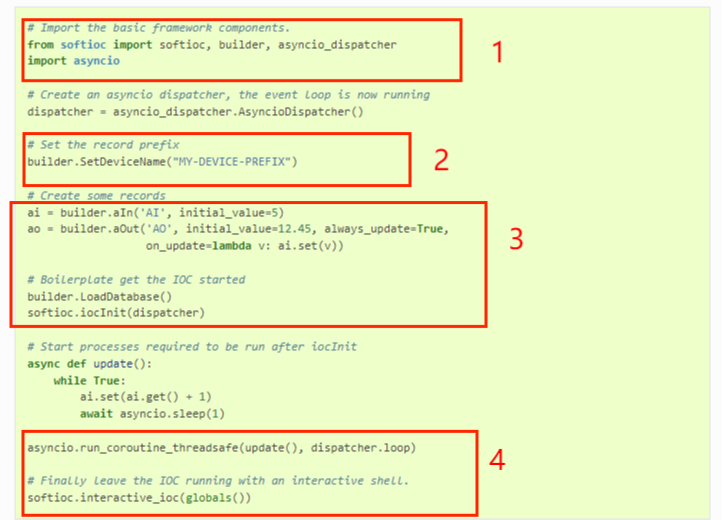
第一步：导入包，[softioc.softioc](https://dls-controls.github.io/pythonSoftIOC/master/reference/api.html" \l "module-softioc.softioc" \o "softioc.softioc) 和 [softioc.builder](https://dls-controls.github.io/pythonSoftIOC/master/reference/api.html" \l "module-softioc.builder" \o "softioc.builder) 这两个子模块为 Python 软 IOC 提供了基本功能，并且是通常使用的子模块。

第二步:设置PV前缀

第三步：创建PV



（3）asyncio方式



第一步：导入包

第二步：设置PV前缀

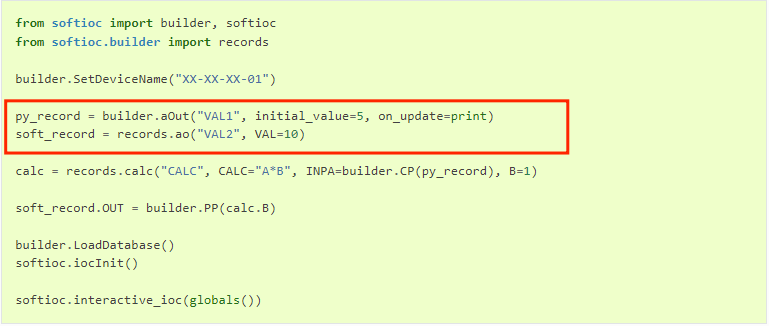
第三步：创建PV

# 三：获取PV值

使用caget、caput、camonitor进行获取pv值

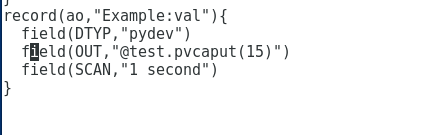
# 四：与PyDevice混合使用

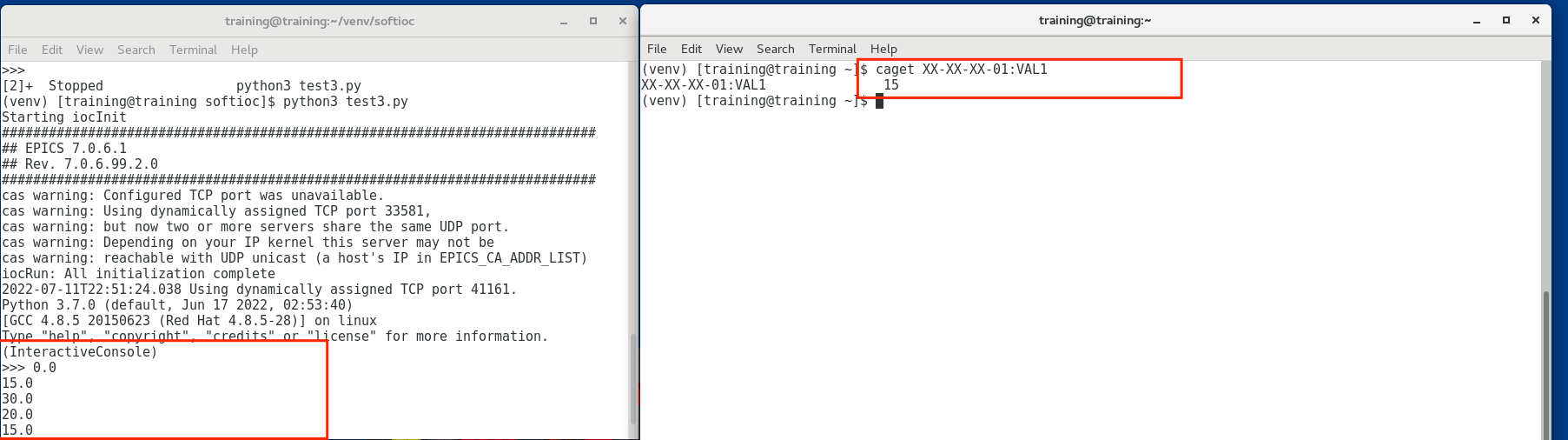
PythonSoftIOC



PyDevice

Python文件和DB文件

****

****

**五：API**

**[softioc.softioc](https://dls-controls.github.io/pythonSoftIOC/master/reference/api.html" \l "module-softioc.softioc" \o "softioc.softioc):** 此模块将基本接口包装到 EPICS IOC

**[softioc.asyncio\_dispatcher](https://dls-controls.github.io/pythonSoftIOC/master/reference/api.html" \l "module-softioc.asyncio_dispatcher" \o "softioc.asyncio_dispatcher):** 基于异步的应用程序的调度程序

**[softioc.alarm](https://dls-controls.github.io/pythonSoftIOC/master/reference/api.html" \l "module-softioc.alarm" \o "softioc.alarm):** PV警报值的定义

**[softioc.builder](https://dls-controls.github.io/pythonSoftIOC/master/reference/api.html" \l "module-softioc.builder" \o "softioc.builder):** 此模块提供用于创建 PV 的工具

1. [initial\_value](https://dls-controls.github.io/pythonSoftIOC/master/reference/api.html" \l "initial-value):这用于指定每条记录的初始值
2. [on\_update](https://dls-controls.github.io/pythonSoftIOC/master/reference/api.html" \l "on-update):这仅适用于 OUT 记录（包括由 [WaveformOut（）](https://dls-controls.github.io/pythonSoftIOC/master/reference/api.html" \l "softioc.builder.WaveformOut" \o "softioc.builder.WaveformOut) 创建的记录）。这指定了在记录处理完成后将调用的函数。
3. [always\_update](https://dls-controls.github.io/pythonSoftIOC/master/reference/api.html" \l "always-update)：

**[softioc.pvlog](https://dls-controls.github.io/pythonSoftIOC/master/reference/api.html" \l "module-softioc.pvlog" \o "softioc.pvlog):** 导入此模块的操作将 IOC 配置为记录到数据库的每个外部 put。

**[softioc.device](https://dls-controls.github.io/pythonSoftIOC/master/reference/api.html" \l "module-softioc.device" \o "softioc.device):** 实现对所有受支持的记录类型的设备支持

# 注意：

1.pyepics中得caget、caput等在python脚本中可以直接使用

2.PyDevice中使用其他IOC的PV值需要使用cothread中的caget、caput。

3.Pythonsoftioc中若使用其他的IOC的PV值，也只能使用cothread中的caget、caput。

# 专用光测试小结：

1. 运动控制相关PV值已保存到archiver中
2. ImageJ相关数据已保存为csv文件和图片，后续处理数据。

# 不足：

1.未确定EPICS的PV与ImageJ中数据之间的关系。

2.此次专用光期间对于图像相关的PV值没有保存。

3.确定万兆网转换器与笔记本电脑自带网口不能同时设置ip问题。