编号（No.）：（IHEP-）HEPS-TS-CB-TS-2022-001

**PyDevice**

**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：PyDevice简介

PyDevice 是对 Python 解释器的 EPICS 设备支持。 它允许将 EPICS 数据库记录与 Python 代码连接起来。

该项目的目标是为 Python 开发人员提供非常简单的接口，以便将 Python 代码集成到 EPICS 控制系统中。 这是通过允许从 EPICS 记录中调用任意 Python 代码来实现的，包括但不限于内置函数、计算表达式、自定义函数等。此外，可以从 IOC shell 执行 Python 代码，这对于设置资源很有用或对Python 代码进行故障排除。 由于 PyDevice 只调用 Python 代码，这允许 Python 模块在独立的非 EPICS 环境中开发和测试，并最终连接到 EPICS PV。

# 二：IOC创建

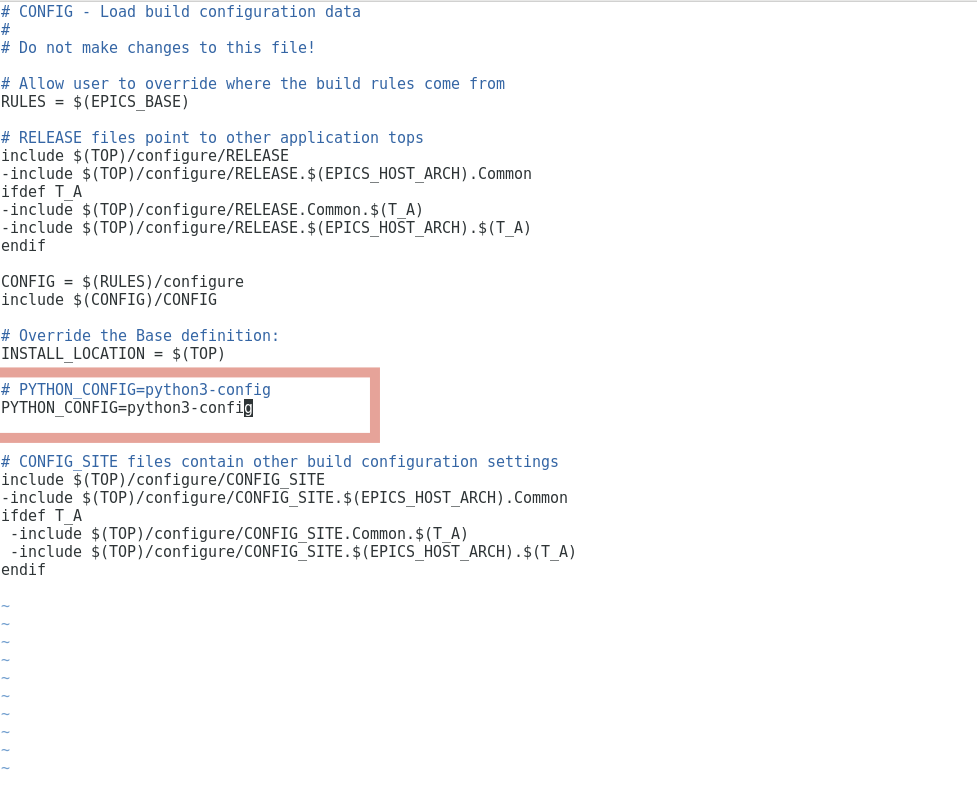
新建IOC，在

（1）在configure下的CONFIG文件中添加

# As soon as python3 is the default configuration

# PYTHON\_CONFIG=python3-config

PYTHON\_CONFIG=python-config（INSTALL\_LOCATION = $(TOP)的后面）



在configure下Makefile文件中

CONFIGS += $(subst ../,,$(wildcard $(CONFIG\_INSTALLS)))前添加：

TARGETS = $(CONFIG\_TARGETS) CONFIG.PyDevice

在include $(TOP)/configure/RULES后添加：

# Starting with Python 3.8, python-config needs --embed flag

ifeq (,$(findstring embed,$(shell $(PYTHON\_CONFIG) --help)))

PYDEV\_SYS\_PROD\_LIBS = $(patsubst -l%,%,$(filter -l%,$(shell $(PYTHON\_CONFIG) --ldflags)))

else

PYDEV\_SYS\_PROD\_LIBS = $(patsubst -l%,%,$(filter -l%,$(shell $(PYTHON\_CONFIG) --ldflags --embed)))

endif

PYDEV\_USR\_CXXFLAGS = $(filter -I%,$(shell $(PYTHON\_CONFIG) --cflags))

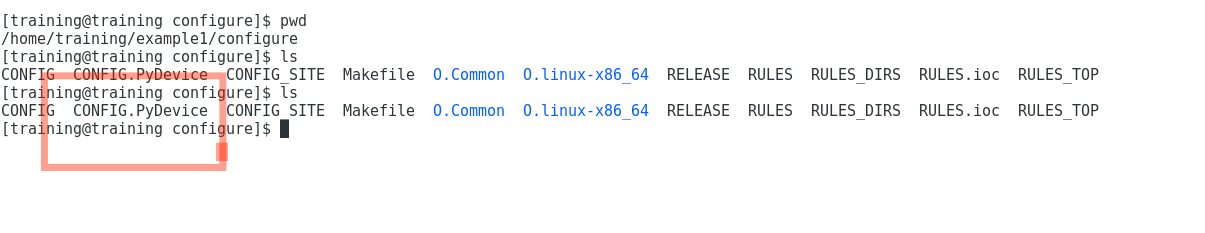
CONFIG.PyDevice:

@echo "SYS\_PROD\_LIBS += $(PYDEV\_SYS\_PROD\_LIBS)" > $(TOP)/configure/$@

@echo "USR\_CXXFLAGS += $(PYDEV\_USR\_CXXFLAGS)" >> $(TOP)/configure/$@



在configure下make后会自动生成



1. 在exampleApp的src下的Makefile文件中添加

include $(TOP)/configure/CONFIG.PyDevice

example\_DBD += pydev.dbd

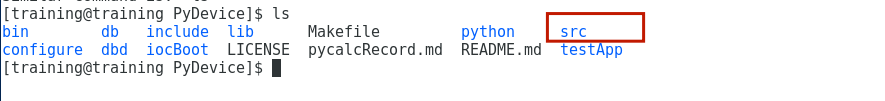
example\_DBD += pycalcRecord.dbd

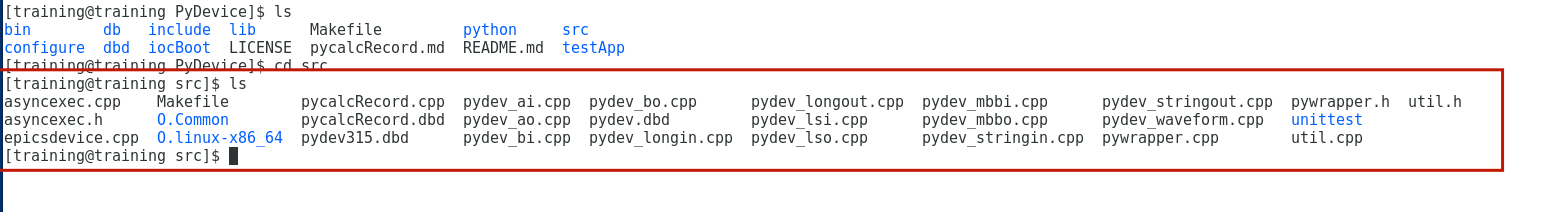
example\_LIBS += pydev

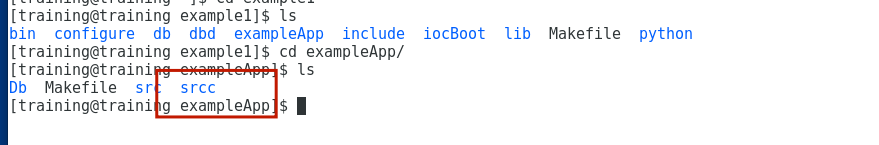
1. 将PyDevice下的src文件复制到exampleApp下，并修改src文件名（我修改为srcc），并在Makefile中添加

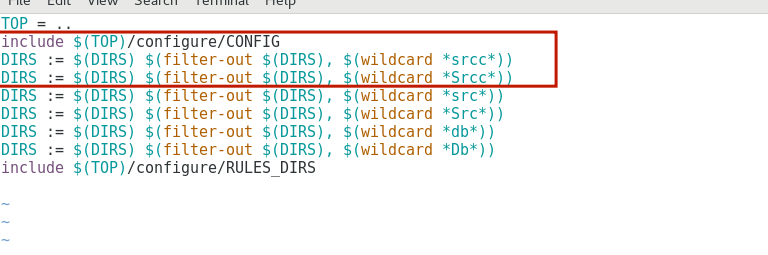
DIRS := $(DIRS) $(filter-out $(DIRS), $(wildcard \*srcc\*))

DIRS := $(DIRS) $(filter-out $(DIRS), $(wildcard \*Srcc\*))

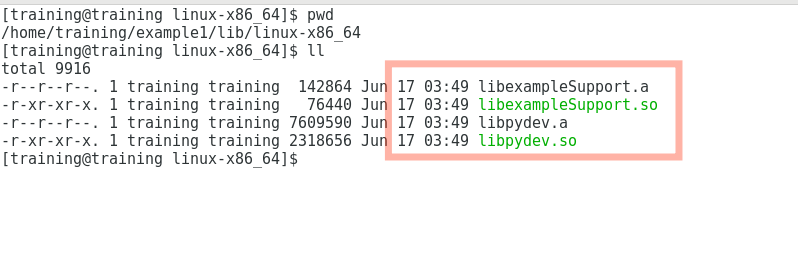






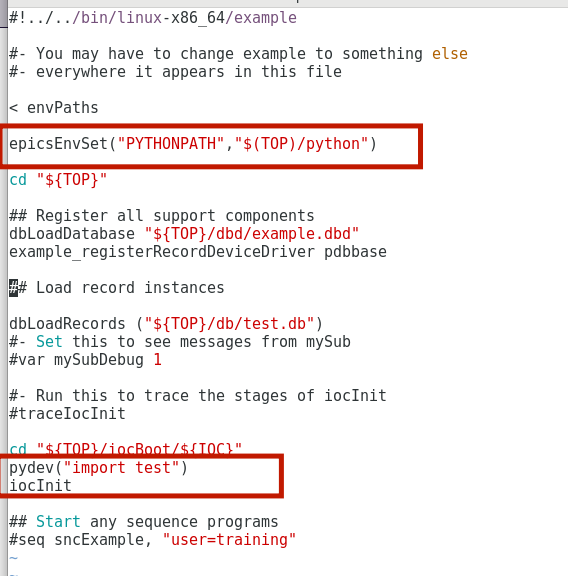


进入到exampleApp下make后会产生动态链接库

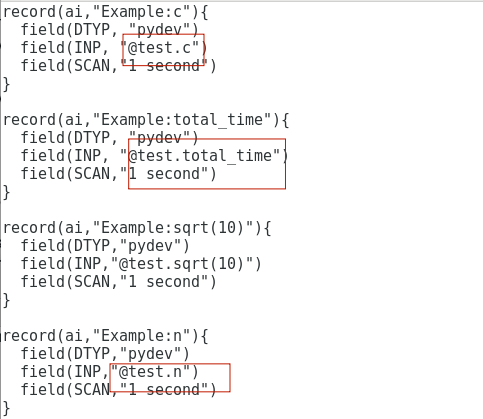
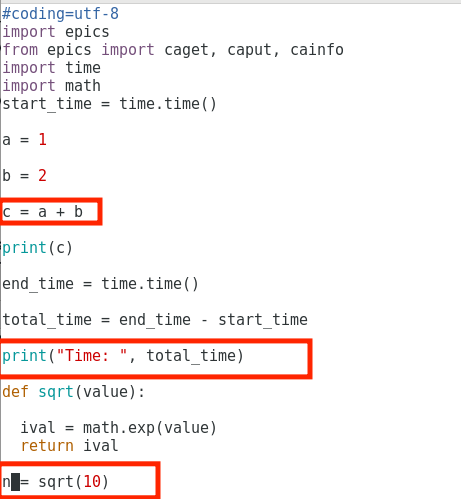


1. 在st.cmd文件中添加python文件的路径

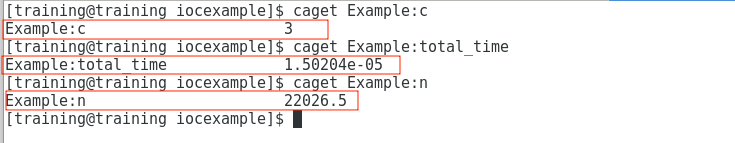
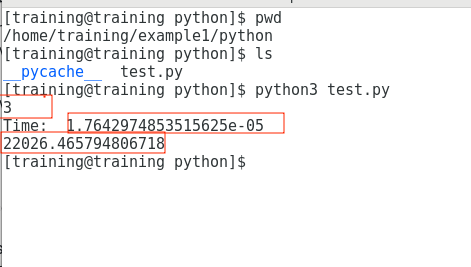
epicsEnvSet("PYTHONPATH","$(TOP)/python")



1. python程序和Db文件

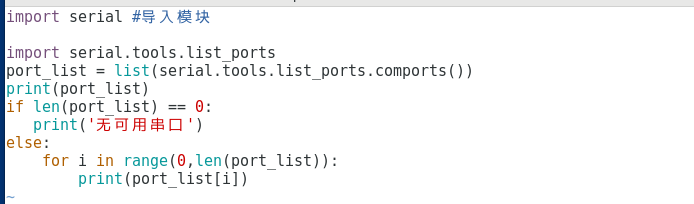


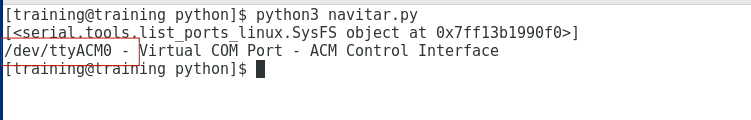
1. 执行结果为：



# 三：与设备串口通信（navitar为例）

1. 读取可用串口





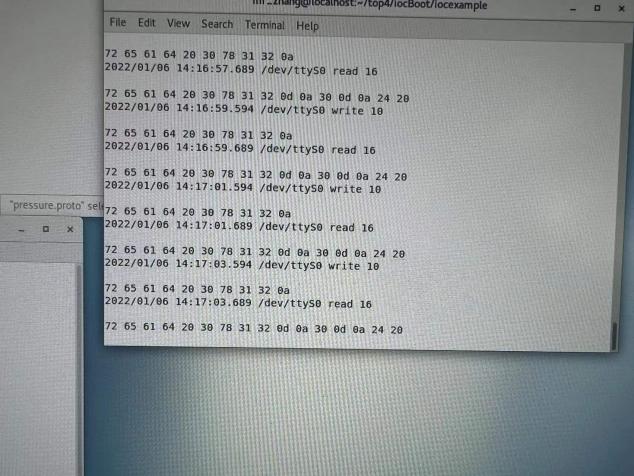
1. 与串口通信

当OutTerminator = LF时;可以得到数据，in可以得到数据，数据格式中间是0d 、 0a。

in =”read 0x12”

Write 72 65 61 64 20 30 78 31 32

Read 72 65 61 64 20 30 78 31 32 0d 0a 30 0d 0a 24 20（16位）



**Python程序**

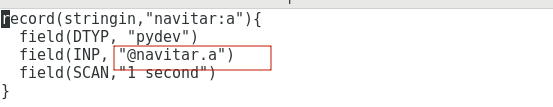


**运行python程序**

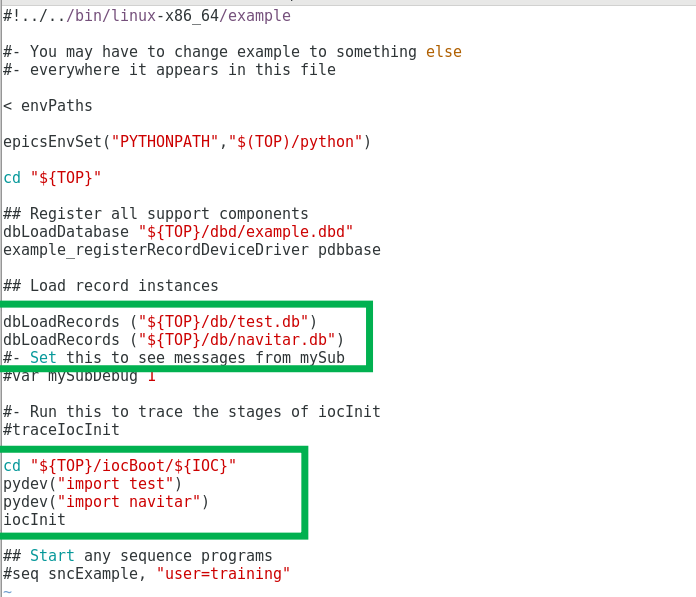


1. 关联IOC

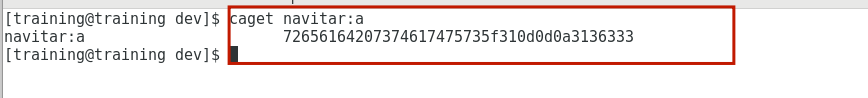
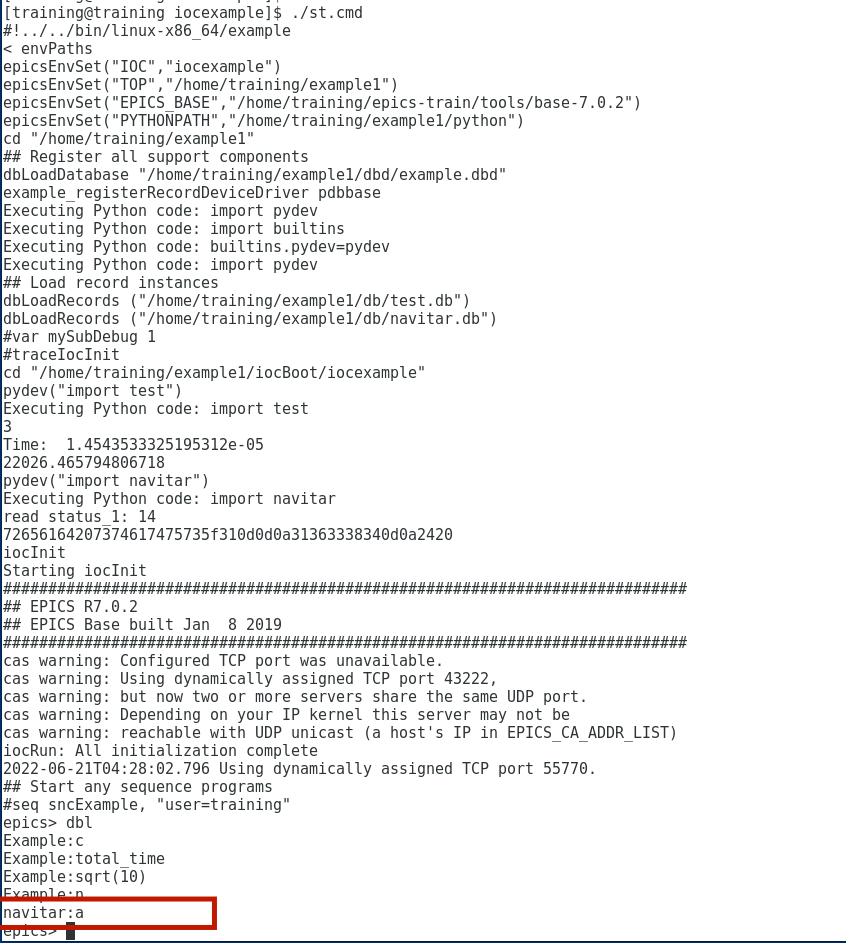
**DB文件**



**st.cmd文件**

****

**IOC运行**



# 四：应用场景

1. Python的串口通信（pyserial）

<http://t.zoukankan.com/dongxiaodong-p-9992083.html>

1. Python的网口通讯（socket）

<https://blog.csdn.net/qq_34162147/article/details/121661083>

1. Python 西门子PLC的读写（python-snap7）

<https://zhuanlan.zhihu.com/p/373721969>