

CENTOS7 安装 epics , areadetector 模块文档

1.安装 epics

各种 epics 版本安装过程几乎差不多，为了便于下面的介绍，我们就以 EPICS7.0.1.1 为例，其次选择 EPICS7 还有一个原因就是后面 areaDetector 的安装会因为选择了 EPICS7 而简化很多，具体后面会解释。

从 EPICS 官网 <https://epics.anl.gov/download/base/index.php>

下载 EPICS7.0.1.1。

这里提前标注一下，我的当前目录为/home/xsc;在当前目录下新建 epics 目录，并切换到 epics 目录；

```
[~ epics]$
```

将下载的 base-7.0.1.1 移动到当前 epics 目录下，解压。入 base-7.0.1.1 目录，然后 make

```
[~base-7.0.0.1]
```

```
[xsc@localhost base-7.0.1.1]$ ls
bin          db          include     Makefile    startup
CentOS7-Base-163.repo  dbd         iocs       modules     templates
cfg          documentation  lib        README      test
configure   html          LICENSE    src
```

可能编译过程会出现缺少 readline.h 啥的，这时就需要我们安装

```
sudo yum install readline-devel-6.0-4.el6.i686
```

```
sudo yum install readline-devel-6.0-4.el6.x86_64(具体看系统是 64 位还是 32 位)
```

成功之后。在当前目录下新建 iocs 目录，mkdir iocs；

进入 iocs，新建 example 目录，进入 example 目录，记住当前目录为/home/xsc/epics/base-7.0.1.1/iocs/example；输入 makeBaseApp.pl -t example example;makeBaseApp.pl -l -t example example。ls 出现

下述结果

```
[xsc@localhost example]$ ls  
bin configure db dbd exampleApp include iocBoot lib Makefile
```

/iocBoot/iocExample

当前目录如下：

```
[xsc@localhost iocexample]$ ls  
envPaths Makefile README st.cmd
```

Chomd +x st.cmd 给 st.cmd 赋予可执行权限，然后./st.cmd 出现>>epics 提示符就 ok 了。

安装好了 EPICS 之后，我们还需要安装 extensions 的一些模块，先从官网下载 extensions，和 MEDM 模块，将 extensions 放在~/epics/目录下，如下：

```
[xsc@localhost epics]$ ls  
base-7.0.1.1 extensions module  
base-7.0.1.1.tar.gz extensionsTop 20120904.tar.gz
```

解压，再将 medm 放在/extensions/src 目录下，如下：

```
[xsc@localhost src]$ ls  
Makefile MEDM3_1_11.tar.gz medm-MEDM3_1_11
```

解压 MEDM3_1_11.tar.gz。参考

<https://epics.anl.gov/extensions/medm/index.php>

进行安装支持包。

在安装 MEDM 之前，我们呢需要安装一些 motif 支持包，这里由于是 centos7 我们呢安装 openmotif-devel。具体如下：

```
385 sudo yum install openmotif-devel  
386 sudo yum install openmotig  
387 sudo yum install openmotif  
388 sudo yum install libXt-devel  
389 sudo yum install libXt-devel.i686  
390 sudo yum install libXmu-devel.i686  
391 sudo yum install libXmu-headers  
392 sudo yum install libXp-devel.i686  
393 sudo yum install libmu-devel  
394 sudo yum install libMu-devel  
395 sudo yum install libXmu6  
396 sudo yum install X11proto-print-devel  
397 sudo yum install libX11-devel.i686  
398 sudo yum install x11proto-print-devel  
399 vim /etc/profile
```

```
新建epics目录: mkdir epics/  
解压base和extensions压缩包到epics/目录下, 解压medm压缩包到epics/extensions/src/目录下  
编辑base/startup/下的Site.profile文件, 修改MY_EPICS_BASE和EPICS_EXTENSIONS参数为正确的  
路径  
在base/目录下make编译  
在extensions/configure/目录下编辑RELEASE文件, 修改EPICS_BASE和EPICS_EXTENSIONS参数为正确  
路径  
在src/medm/目录下make编译  
在extensions/bin/linux-x86目录下即可运行medm: ./medm  
也可在系统环境变量中配置PATH变量, 添加medm的路径, 重启后即可在任何地方./运行medm
```

进入 medm_MEDM3_1_11,

Base-7.0.1.1/startup/Site_profile 文件中设置如下：

```
# Location of epics base  
if [ -z "${MY_EPICS_BASE}" ] ; then  
    MY_EPICS_BASE=/home/xsc/epics/base-7.0.1.1  
fi  
  
# Location of epics extensions (medm, msi, etc.)  
if [ -z "${EPICS_EXTENSIONS}" ] ; then  
    EPICS_EXTENSIONS=/home/xsc/epics/extensions  
fi  
  
# Postscript printer definition needed by some extensions (
```

环境变量设置如下：

```
export EPICS_BASE=/home/xsc/epics/base-7.0.1.1  
export EPICS_HOST_ARCH=linux-x86_64  
export PATH=${PATH}:/home/xsc/epics/base-7.0.1.1/bin/linux-x86_64  
export LD_LIBRARY_PATH=${LD_LIBRARY_PATH}:${EPICS_BASE}/lib/${EPICS_HOST_ARCH}  
export EPICS_EXTENSIONS=/home/xsc/epics/extensions  
done
```

```
# Location of external products  
EPICS_BASE=$(TOP)/../base-7.0.1.1  
#EPICS_EXTENSIONS = /usr/local/epics/extensions  
EPICS_EXTENSIONS=$(TOP)  
  
OAG_APPS=$(TOP)/../oag/apps
```

然后，make

到这里我们就把 MEDM 安装好了，可以在 extensions/bin/linux-86_64 下面

找到 medm 可执行文件。./medm 运行即可

2 . 安装 areaDetector

下载 synApps,见网址:

<https://www.aps.anl.gov/BCDA/synApps/Where-to-find-it>

主要这里有个问题，对于 areaDetector 的安装我们可以采用先安装一些依赖

模块比如 asyn,busy,calc 和 sccan，autosave(这一点在官方安装指南里面

有明确的介绍)。但这样做麻烦的是你需要单独安装这些。这里我们采用在 synapps 中完成包括依赖模块在内的完整一次性安装。

下面一步一步操作：

我的 centos 当前路径为 *home/xsc/xsc/epics/*；

我的 epics 安装在 epics 目录下，我们在 epics 目录下新建一个 module 子目录。将下载好的 synApp 放在 module 目录下，解压：

```
[xsc@localhost support]$ ls
alive-R1-1-0          dac128V-R2-9          love-R3-2-6           seq-2-2-5
allenBradley-2-3      delaygen-R1-2-0       lua-R1-2-2            softGlue-R2-8-1
areaDetector-R3-3-1   documentation         Makefile              softGlueZynq-R2-0-1
assemble synApps.sh      dxp-R5-0              mca-R7-7              sscan-R2-11-1
asyn-R4-33            dxpSIT0R0-R1-1       measComp-R2-1         std-R3-5
autosave-R5-9         iocStats-3-1-15       modbus-R2-11          stream-R2-7-7b
busy-R1-7             ip330-R2-9            motor-R6-10-1         utils
calc-R3-7-1           ipac-2-15             optics-R2-13-1        vac-R1-7
camac-R2-7-1          ip-R2-19-1            quadEM-R9-1           vme-R2-9
caputRecorder-R1-7-1  ipUnidig-R2-11        README                xxx-R6-0
configure             LICENSE                README.md             Yokogawa_DAS-R1-0-0
```

这里面除了 areaDetector 安装必须的模块之外，还有很多其他的支持模块，

在这里我们只需要安装 areaDetector 所需要的模块，分别

是:asyn,autosave,busy,calc,sscan,这里我们又需要考虑一个问题就是模块

依赖性的问题，因为 calc 和 asyn 模块依赖于 ipac 和 seq 模块，所以我们还

需要安装 ipac 和 seq 模块。至于为什么有依赖性，这一点我们可以参考当前

目录下的 Makefile 文件查看依赖性关系。

安装配置：首先进入当前目录下的 configure,出现一些配置文件

```
[xsc@localhost configure]$ ls
CONFIG
CONFIG_SITE
CONFIG_SITE.linux-x86_64
EXAMPLE_EPICS_BASE.win32-x86
EXAMPLE_EPICS_BASE.win32-x86-debug
EXAMPLE_EPICS_BASE.win32-x86-static
EXAMPLE_EPICS_BASE.windows-x64
EXAMPLE_EPICS_BASE.windows-x64-debug
EXAMPLE_EPICS_BASE.windows-x64-static
EXAMPLE_SUPPORT.cygwin-x86
EXAMPLE_SUPPORT.win32-x86
EXAMPLE_SUPPORT.win32-x86-debug
EXAMPLE_SUPPORT.win32-x86-static
EXAMPLE_SUPPORT.windows-x64
EXAMPLE_SUPPORT.windows-x64-debug
EXAMPLE_SUPPORT.windows-x64-static
EXAMPLE_SUPPORT.windows-x64-static
fonts.adl2edl
LICENSE
Makefile
makeReleaseConsistent.pl
README.md
RELEASE
RULES
RULES_DIRS
RULES_ioc
RULES_OPI
RULES_TOP
setup_epics_cygwin32
```

我们这里在 CONFIG_SITE ,RELEASE 文件进行配置：讲 CONFIG_SITE

里面的项全部注释掉，至于里面各项什么意思参考：

<https://epics.anl.gov/bcda/synApps/synApps.html>,注释掉 linux_USB_INSTALL 和

linux_NET_INSTALL。打开 RELEASE 首先配置 EPICS_BASE 改为我们的

EPICS base 所在目录，我这里是/home/xsc/xsc/epics/base-7.0.1.1,吧

SUPPORT 改为 support 所在目录，我这里

是/home/xsc/xsc/epics/module/support;

除此之外，将不需要的模块全部注释掉。只留下之前我们 areaDetector 需要的模块和依赖模块（再次强调，依赖模块一定要安装，不然会失败的，依赖关系之前已经说过请查看 Makefile 文件);

接下来我们在/home/xsc/xsc/epics/module/synApps/support 路径下 make erlease 使得我们的配置生效。

由于我们需要安装 areaDetector 模块，所以我们还需要到 areaDetector 模块下进行相关的修改;

之前 make release 会自动修改 areaDetector 目录中的相关配置，基本不用怎么修改，有个细节要注意带，在 areaDetector/configure/RELEASE_LIBS.local 里面涉及到一个 PVA 的设置，如果是 EPICS7 之前的版本是需要设置 PVA 模块的。EPICS7 之后的版本在 epics 中已经包含了 PVA 模块，所以注释掉就可以了，同事在 CONFIG_SITE 中 check_release 改为 YES(不然会很快出错哟)。

接下来把目录切换到 support 下可以进行激动人心的时刻了，make 然后等到结束没有 error 就 ok 了.第一次 make 可能出现 error，找不到命令 re2c

或者 lexer.c 文件啥的，不要慌，这是因为 seq 模块需要的支持文件，这时候，应该 x 下载 re2.org/install/insatll.html#download

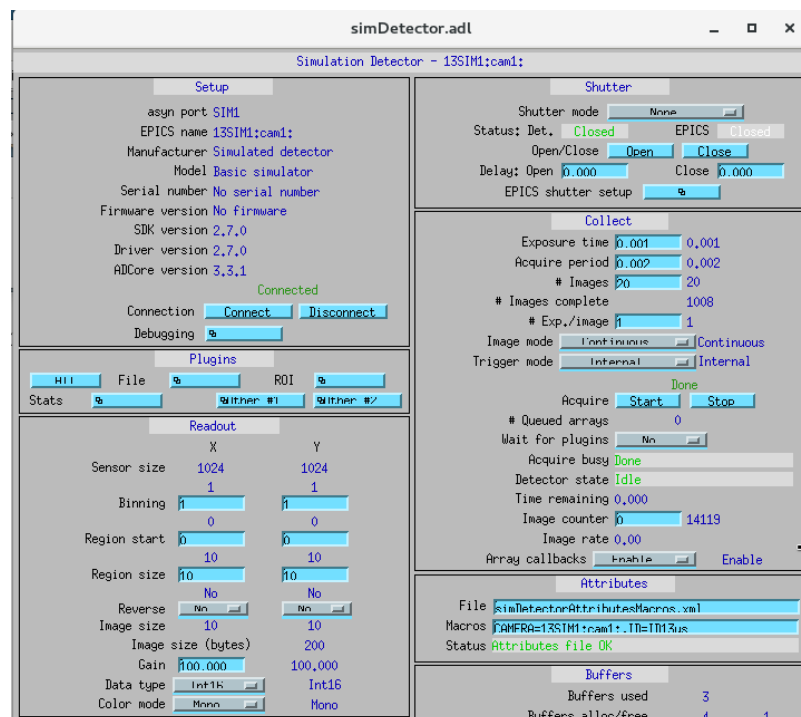
3. 使用 ADSimulator

为了模拟 areaDetector 存储数据的流程和细节，在没有真实相机的情况下，我们可以选择使用 ADSimulator 模拟数据的图片数据的产生并将其存入到 HDF5 等文件中。首先我们呢需要测试我们的 areaDetector,为此我们需要安装一些客户端工具，这里我安装的 medm，medm 可以加载 ADSimulator.adl 文件，导出 ADDriver 控制界面，实现对 Driver 的实时控制。此外我们还需要一个显示图片的 display 客户端，这里我安装的 ImageJ。

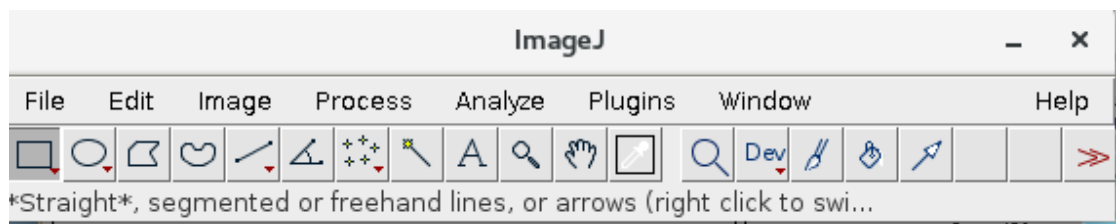
1. 安装 medm，下载最新版的 MEDM 安装包，见 <https://epics.anl.gov/extensions/medm/index.php>，
下载按安装即可。
2. 安装 ImageJ。下载 ImageJ 安装包，见 <https://imagej.nih.gov/ij/download.html>，
3. 解压即可。

现在工具已经齐全了，我们可以开始整了；进入
/home/xsc/epics/module/synApps/support/areaDetector-R3-3-1/ADSimDetector/iocs/simDetectorIOC/iocBoot/iocSimDetector 目录，
进入 st_base.cmd,将注释掉的全部 uncommet，为了方便 medm 的启动，
我们新建一个目录/home/xsc/epics/adls,运行 find
/home/xsc/epics/synApps/support -name '*.adl' -exec cp -fv {}
/home/xsc/epics/adls \;将所有的.adl 文件全部放在 adls 目录下，添加环境
变量：export EPICS_DISPLAY_PATH= /home/xsc/epics/adls

切换到 iocSimDetecto 目录，运行./start_epics 即可出现如下界面：



接下来到 ImageJ 目录下，启动./ImageJ,出现：

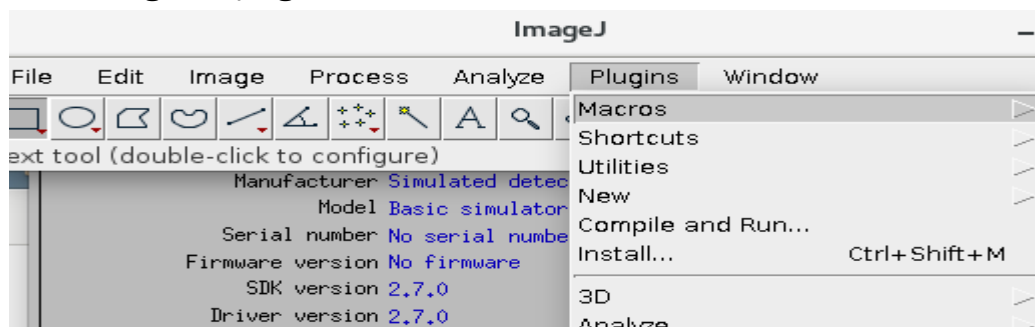


这个时候我们呢需要做很重要的一件事，就是下载

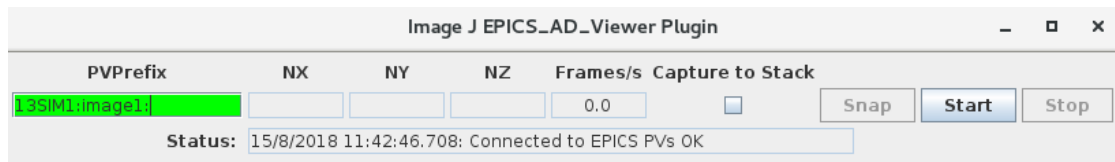
ADViewers (<https://github.com/areaDetector/ADViewers>)，并将里面

的 EPICS_AREADETECTOR 模块，添加到 ImageJ/plugin 目录下，然后就

可在 ImageJ 的 plugin 下面看到如下：



选择 Compile and Run, 进入选择 EPICS_AREADETECTOR , 悬着 EPICS_AD_Viewer.java,确定之后，弹出如下界面：



PVPrefixs 底色位绿色表示的是已经接连上改 PV ，接下来就是控制 Simulator.adl 界面，同时在 ImageJ 点击开始即可显示图片。

4.使用 ADProsilica

```
$cd areadetector
$cd ADProsilica
$make
$cd iocs/prosilica/iocboot/iocProsilica
$cp envpath envpath.linux
```

接下来我们要开始设置一些细节。为了方便我们将/synapps/support 目录下的所有.adl 文件拷贝到一个新建的/home/xsc/epics/adls 目录下。

```
$find /home/xsc/epics/synApps/support -name '*.adl' -exec cp -fv
{} /home/xsc/epics/adls \
$pwd
/home/xsc/epics/module/synapps/support/areaDetectorxxx/ADProsilica
a/iocs/prosilicaloc/iocboot/iocProsilica
$vim start_epics
```

取消注释第一行 medm...,将 ADProsilica.adl 前面加上路

径/home/xsc/epics/adls

即绝对路径为/home/xsc/epics/adls/ADProsilica.adl

保存退出

下面开始相机设置，相机的 ip 地址需要自己测试，我这里的相机的 ip 为 192.168.121.111

于是我的计算机的静态 ip 地址就设置成 192.168.121.112（这里并不唯一，

只需要在同一个子网内即可)

关闭防火墙,systemctl stop firewalld

启动相机

\$/start_epics(开启脚本)

弹出界面 (如果相机信息没有的花应该就是没有连接上, 需要检测以下计算机和相机之间是否已经畅通了如果相机信息没有的花应该就是没有连接上, 需要检测以下计算机和相机之间是否已经畅通了)

\$cd /home/xsc/epics/extensions/src/ImageJ

\$/ImageJ(开启 ImageJ)

(具体就不说了, 很简单的操作)

下面只需要在 ADprosila.adl 界面设置参数之后点击 start,在 ImageJ 上就可以看到图片了。

这里还有一些 ImageJ 环境环境变量的设置, 列出如下:

```
export EPICS_CA_AUTO_ADDR_LIST=NO
export EPICS_CA_MAX_ARRAY_BYTES=10000000
export EPICS_CA_ADDR_LIST=192.168.121.112
```

到这里整个流程就完成了。