ODroid KStars/Ekos/libindi Astronomy

Wayne Green

Thursday 25th June, 2020

Abstract

The ODroid OU4 is an 8-core Arm processor, here tricked out with a 64G eMMC card preloaded with Ubuntu 18.04. These instructions are how to load the ODroid to support KStars/Ekos/libindi for astronomy. It is more of a crib sheet. As an exercise the popular ds9 program compiled on the ODroid.

Contents

Overview

First acquire the ODroid XU4, in my case the one with the fan. Get a decent μSSD drive – a Verbatim Premium 10 32G or equivalent. I used BalenEtcher on a Win10 to load the image.

Boot the ODroid, expand the filesystem to the whole card and bring the image up to date. Below are notes I took and rearranged to be in the order of how you should do things. For example I loaded VIM (the vi editor) early, locate[1] late (because it is easy to load.) The next logical step is to get the all the development installs done – in this case I ran the INDI part first.

I prefer to use ds9 as an image viewer and wanted it installed on the ODroid. I did a sudo apt-get install saods9 – and the packaged program crashes when loading a fits file iamge even though the program started up fine. I downloaded the source, but the stock OS image was missing quite a few libraries. One by one they were uncovered, and the apt-get install lines placed to be right after the boot. But in my actual order, I did Section 1.3 before the ds9 libraries (I didn't know I needed them at that point).

I want the X11 functionalty, and I want Ekos to drop images where I want them. I installed NFS, and took notes on the steps to tie them to my main computer in Section 1.7.

1.1 Things to do once

Using the ssh command, remote login to the odroid on your main machine. You can do these things at the ODroid command prompt as well if you like. If you are doing another odroid for the first time, then you may have to remove its ssh-key – the message will tell you what to do:

```
ssh-keygen -f "/home/wayne/.ssh/known_hosts" -R "odroid.local"
```

Here is some code to cut/paste:

```
# If this is a different odroid might have to remove the
# old hash from the .ssh/known_hosts file.
#ssh-keygen -f "/home/wayne/.ssh/known_hosts" -R "odroid.local"
ssh -l odroid odroid.local  # see remove old key above

# fix up the ~/.bashrc to git rid of colors. (hard to see)
echo "export PS1=’(\j) \h \w [\!] ’" >> ~/.bashrc
echo "alias ls=’ls -FC’"            >> ~/.bashrc
echo "
```

1.2 Initialize for KStars

Here we remote login via ssh into the odroid, and follow the commands below. It may require a reboot or two. This is for a first time load. A short cut for the huge KStars libraries and some default stuff you may want will be developed below. (Thursday 25th June, 2020)

```
#Setup
ssh -l odroid odroid.local  # see remove old key above

sudo apt-get update
sudo apt-get upgrade              # N - keep the provider’s /etc/apt-fast.conf
sudo apt-get dist-upgrade
sudo apt-get install -y vim       # because,,, vi!
sudo apt install -y git           # because,,, git! (astrometry.net)

# Load these packages now, get it over with for ds9.
# Handy anyway.
sudo apt-get install -y locate       # bacause,,, locate!
sudo apt-get install -y libx11-dev   # needed for ds9
sudo apt-get install -y zlib1g-dev   # needed for ds9
sudo apt-get install -y libxml2-dev  # needed for ds9
sudo apt-get install -y libxslt1-dev # needed for ds9
sudo apt-get install -y autoconf     # needed for ds9

# for astrometry.net
sudo apt-get install -y swig
sudo apt-get install -y  python3-dev python3-pip python3-virtualenv
```

1.3 KStars/EkosINDI part

Load the packages, and then using the KStars interface initialize the site location etc, and load the internal things like lots of Tycho stars. Add the Atik support libraries.

```
# Indi/KStars
sudo apt-add-repository -y ppa:mutlaqja/ppa
sudo apt-get update
sudo apt-get install indi-full
sudo apt-get install kstars-bleeding

Atik Cameras:
sudo add-apt-repository -y ppa:mutlaqja/ppa
sudo apt-get update
sudo apt-get install indi-atik
sudo apt-get install libindi1     # moonlite NiteCrawler focuser
```

1.4 Remote X Desktop

This part will be idle most of the time – but still cycle-sucking. I did it anyway.

```
# https://linuxize.com/post/how-to-install-xrdp-on-ubuntu-18-04/
apt install xfce4 xfce4-goodies xorg dbus-x11 x11-xserver-utils
apt install xrdp

systemctl status xrdp
adduser xrdp ssl-cert
systemctl restart xrdp
ufw allow from 192.168.1.0/24 to any port 3389
ufw allow 3389
```

1.5 SAOImage/ds9

Odroid:

```
cd /Downloads
wget http://ds9.si.edu/download/source/ds9.8.1.tar.gz
tar xzvf ds9.8.1.tar.gz  # -> makes SAO
cd SAO
make unix/install > log0 2>&1
make >> log0 2>&1
tail -f log0  # ties the window up watching the make
```

This went on for quite a while, each failure ment tracking down the missing library, adding that to the top of this file, and going again. It was painful, fortunately I knew ds9 when it was sane, and could decipher the twisted package names.

Eventually:

```
ssh -X odroid@odroid # from titan
ds9 theOdroidFits.fits # pops up an X display on Titan!
```

But, downloads bog down the ODroid, so we want INDI to drop the file via the network onto titan! So on to NFS.

1.6 ODroid in the Network

In this example, my main machine titan lives on my network at 192.168.0.216. Currently the ODroid uses DHCP and is at 192.168.0.218 – I maintain long DHCP leases with renewal on my network host.

The ODroid is gig-Ether.

I added 192.168.0.216 to the odroid /etc/hosts file. I added 192.168.0.218 to titans /etc/hosts file.

Titan has the nfs-server installed, has been for years.

Titan has one-and-only one file that matters: the /etc/exports file. It needs one line added to allow the ODroid to come in as a user.

The ODroid needs to have the nfs-common package installed, one line added to its /etc/fstab, and a little helper script added to /etc/network/if-up.d/fstab to mount the nfs connection during odroid-boot. (I suspect late rc2, before rc3). I don't care, I want it ready with run-level 5 hits.

1.7 NFS

Third things first: This machine is behind at least 3 routers in my system, the last is the WAN gateway.

Getting NFS was a bit painful to implement and I need to dig into some security issues. I should do the export on titan ONLY for ODroid's IP address for example. It's been 30 years since I had to do this for a living.

Firewalls need to be explored and opened. My IPtables are a bit goofy on titan as it does Strange Things™ with Docker containers, and the rest of my network.

When the camera takes an image, it will put the image into ordoid:/mnt/titantoday that winds up on titan:/export/wayne/titantoday. So, AS the camera is downloading – the network is churning and the file's write finishes on titan. While the next hammering download is taking place, titan does not feel the impact so much.

Note: This will have to change for high cadence work, but it is not too bad.

NFS is a Unix/Unix network file sharing program. I trust it more than SMB – but SMB has made progress. It is the one I choose, as my analysis routines are Unix based on a remote server anyway.

I suspect a NFS/Tunneling might be slower.

Or, I may implement a "pull" strategy from titan, to get the files over when I want them. This means titan will not see the camera download at all.

```
# Now to get NFS working
#On odroid
# safer? sudo ufw allow from 192.168.0.216/32 to any port nfs
sudo ufw allow from 192.168.0.0/24 to any port nfs
sudo ufw status
sudo apt install nfs-common
---------------------------------
# odroid /etc/fstab
```

```
UUID=e139ce78-9841-40fe-8823-96a304a09859 / ext4 errors=remount-ro,noatime 0 1
LABEL=boot /media/boot vfat defaults 0 1
titan:/export/wayne/ /mnt/titantoday nfs rw,async,user,auto 0 0

mount -a
# manual way... mount titan:/export/wayne/ /mnt/titantoday/

# add a config script to mount fs
cat >/etc/network/if-up.d/fstab
#!/bin/sh
mount -a
^D
chmod +x /etc/network/if-up.d/fstab
---

#On titan: edit /etc/exports
/export       192.168.0.1/24(rw,fsid=0,insecure,no_subtree_check,async)
/export/users 192.168.0.0/24(rw,nohide,insecure,no_subtree_check,async)
/export/wayne 192.168.0.0/24(rw,async,no_subtree_check,anonuid=1000,anongid=1000)
sudo exportfs -av

# I suspect the line CIDR /32 = netmask of 255.255.255.255 or precise macine.
# anonuid=1000,anongid=1000 is a bit concerning! It requires agreement in
# user's ids between machines and 1000 is default linux install.
/export/wayne 192.168.0.216/32(rw,async,no_subtree_check,anonuid=1000,anongid=1000)
```

2 Summary/Things to do

Three-D print a case that will hold on to something. Probably my Alpy-600 to start. That fan is a bit questionable, but another is hammering away at the back of the camera.

I still need to write the task that drives ds9 when new images shows up. Did a lot of documentation on that as I was trying to get that to work from inside a Docker container. Futile I suspect, but I am sure learning networking on Docker's virtual net.

3 Operations

It is possible to store images both 'locally' on the ODroid, and remotely (see NFS above.)

the ˜/Desktop/Today is where all the software has been configured (in the past!) to use. At the end of the night, Rename this and copy to local machine as needed.

With NFS, the copies are done one-by-one in realtime.

```
ssh -l odroid@odroid.local
...pw...

cd Desktop

ls -l Today          # see if exists,
mv Today when.Today  # move it to the 'right' date
mkdir Today          # make todays.
```

4 Use with The Sky/X (TSX)

The remote use requires the TCP/IP address of the mount controler. TSK Tools->TCP Server, the port is 3040 by default. On the same computer, set the TCP/IP address to localhost 127.0.0.1.

`https://www.indilib.org/devices/mounts/bisque-paramount.html`

5 Astrometry.net Local

The main documentation is at:

`http://astrometry.net/doc/index.html`

The Astrometry.net package can be obtained via git.

```
mkdir -p ~/git/{clones,external}
mkdir -p ~/clones
cd ~/clones
wget http://heasarc.gsfc.nasa.gov/FTP/software/fitsio/c/cfitsio-3.48.tar.gz
tar xzvf cfitsio-3.48.tar.gz
cd cfitsio-3.48
./configure
make > log1 2>&1 &
tail -f log1
view Makefile # make sure install dir is right. Ha! prefix=/usr needed
sudo make prefix=/usr install

#git clone ....the clone from github
cd ~/git/external
git clone https://github.com/dstndstn/astrometry.net.git
cd astrometry.net
# in the background, run make and watch for errors with tail -f
make > log1 2>&1 &
tail -f log1
# ... take a minute! ...
# making this documentation, cfitsio was missing, see above
make > log2 2>&1 &
tail -f log2
# OK had to pause and install swig (above)
make > log3 2>&1 &
tail -f log3

# OK had to pause and install python dev (above)
make > log4 2>&1 &
tail -f log4

sudo make install INSTALL_DIR=/usr/local >install.log 2>&1
less install.log

# get some index files
# first-off download them to a local place on your
# network, then copy them here to the ODroid.
# get wayne's handy scripts.
```

5.1 Index Files

The pagkage is installed at a given root location, here /usr/local/share/astrometry. The data files are controled by a file called etc/astrometry.cfg.

```
add_path /usr/local/astrometry/data
autoindex
```

The two lines use all the files in the /usr/local/astrometry/data location. There are three strategies:

- Keep all the files together in the data directory.

Acquire the INDEX-42 files.

Make GAIA index files. `https://github.com/LCOGT/gaia-astrometry-net`

Get Pre-made GAIA files for scales 00 through 07. 194 files. `http://data.astrometry.net/5000/`

6 Summary

The KStars planetarium packages leaves quite a bit to be desired, but allows one to add their own libraries, and control cameras etc. The ODroid/KStars/Ekos/INDI suite, with configuration tricks here allows the odroid to happily live on the telescope without keyboard/mouse/monitor. IT requires power and Ethernet from the outside – and connections to the instrument: USB for the camera and perhaps USB for the filter wheel.

Scale	File	Range		Range	
		(arcmin)		(degrees)	
00	index-4200-*.fits	2.0	2.8		
01	index-4201-*.fits	2.8	4.0		
02	index-4202-*.fits	4.0	5.6		
03	index-4203-*.fits	5.6	8.0		
04	index-4204-*.fits	8	11		
05	index-4205-*.fits	11	16		
06	index-4206-*.fits	16	22		
07	index-4207-*.fits	22	30		
08	index-4208.fits	30	42		
09	index-4209.fits	42	60		
10	index-4210.fits	60	85		
11	index-4211.fits	85	120	1.42	2.00
12	index-4212.fits	120	170	2.00	2.83
13	index-4213.fits	170	240	2.83	4.00
14	index-4214.fits	240	340	4.00	5.67
15	index-4215.fits	340	480	5.67	8.00
16	index-4216.fits	480	680	8.00	11.33
17	index-4217.fits	680	1000	11.33	16.67
18	index-4218.fits	1000	1400	16.67	23.33
19	index-4219.fits	1400	2000	23.33	33.33

Table 1: Index File Scales