Getting Arduino / XLR8 Setup On Linux

Unlike the Mac and Windows XLR8 packages, there are a couple of extra steps for getting XLR8 to run on Linux. There are three main focuses concentrated on in this tutorial: installation of PySerial, installation of the Arduino IDE, and modifying port permissions. When you reach the end, you will be able to jump to Step 3 in the Quick Start Guide for XLR8. This tutorial was created & tested on Ubuntu 16.04 LTS 64bit.

Assumptions

- You have used a terminal application before. You do not need to be an expert, but if you have used a terminal application before that experience will be helpful here.

- You know how to download applications from the internet.

- You already have Python installed. If you do not have Python installed your computer will let you know in Step 1.

- You know your computer's user account name, password, and have admin privileges. (The user account is who you sign in as when logging onto your computer.)

Tools You Will Need

- A computer running Linux. Note that if you are interested in building your own XBs, you will need to be running on a 64bit platform to use the recommended software. However, if you do not plan to build your own FPGA images and just what to use your XLR8 "out of the box," a 32-bit system will work fine.

- Internet connection

- Terminal application

- Python Installed (this tutorial is using Python 2.7)

Let's get started…

1. Install pySerial

If you do not know if you have pySerial installed, the first steps listed here will help determine if you do or not.

1.1. Open a terminal window.

1.2. Make sure pySerial is not already installed.

    1.2.1.   Type the command python and the press enter/return key. (Note, if you do not receive a message telling you what Python version is installed you do not have Python installed.)

    1.2.2.   Type the python command help().

    1.2.3.   Type the help option serial.

    1.2.4.   You should now see a message saying that no documentation for serial has been found. If you already have pySerial installed the terminal will open a file showing you the documentation on the module. Since the serial module is already installed, you can and skip down to Step 2. An example terminal view of exiting a file can be found in section 1.6.4 – 1.6.6).

    1.2.5.   Press both the control and "d" key twice to exit the Python menus.

TERMINAL VIEW: All commands that the steps above instructed you to type are highlighted in blue.

```
yourusername@ubuntu: ~$ python
Python 2.7.11+ (default, Apr 17 2016, 14:00:29)
[GCC 5.3.1 20160413] on linux2
Type "help", "copyright", "credits" or "license" for more information.
>>> help()
Welcome to Python message …
help> serial
no Python documentation found for 'serial'
help>
control key + d key
control key + d key
yourusername@ubuntu: ~$
```

1.3. Choose your method of installing pySerial. There are two ways to install. First, there is the express way which uses apt-get. Second is the long way, but if your Linux distribution does not have pySerial prepackaged it is the way you will have to take.

1.4. Express pySerial Install

    1.4.1.   Type sudo apt-get install python-serial and press return/enter key.

    1.4.2.   Enter your password when promted.

TERMINAL VIEW: All commands that the steps above instructed you to type are highlighted in blue.

```
yourusername@ubuntu: ~$ sudo apt-get install python-serial
Python 2.7.11+ (default, Apr 17 2016, 14:00:29)
[ sudo ] password for yourusername: ***
…
Setting up python-serial (3.0.1.1) …
yourusername@ubuntu: ~$
```

1.4.3.   Done! Go to 1.6 to verify install was successful.

1.5. Install pySerial The Long Way
   1.5.1.   In your browser go to *https://pypi.python.org/pypi/pyserial*.
   1.5.2.   If necessary, scroll down to the downloads.
   1.5.3.   Locate the file ending in ".tar.gz:
   1.5.4.   Click on the file to download it. (This tutorial is using the download file pyserial-3.1.1.tar.gz)
   1.5.5.   Back in your Terminal application navigate to your downloads directory. (Or go to whatever directory where you downloaded the pySerial package. This tutorial's examples will assume the default download location is the Downloads directory.) To navigate to your Downloads directory type cd Downloads/.
   1.5.6.   Extract the files from the package using the following command: tar –zxvf pyserial-3.1.1.tar.gz. Note that if you are installing a different version of python the version number will be different. The Python version number shown in this example is 3.1.1.
   1.5.7.   Navigate inside the pySerial directory you just extracted by typing cd pyserial-3.1.1/. Note, once again the numbers following the "-" after pySerial may be different if you are installing another version of python.

TERMINAL VIEW: All commands that the steps above instructed you to type are highlighted in blue.

```
Please wait a moment while I gather a list of all available modules…
List of modules
…
>>>
control key + d key
control key + d key
yourusername@ubuntu: ~$ cd Downloads/
yourusername@ubuntu: ~/Downloads$ tar –zxvf pyserial-3.1.1.tar.gz
pyserial-3.1.1/
                                                          continued ->
```

```
pyserial-3.1.1/setup.py
pyserial-3.1.1/CHANGES.rst
…
yourusername@ubuntu: ~/Downloads$ cd pyserial-3.1.1/
```

1.5.8.   Build the package by typing python setup.py build.

1.5.9.   Type sudo python setup.py install to install the package. You will need to enter your computer account's password when prompted.

TERMINAL VIEW: All commands that the steps above instructed you to type are highlighted in blue.

```
…
yourusername@ubuntu: ~/Downloads$ cd pyserial-3.1.1/
yourusername@ubuntu: ~/Downloads/pyserial-3.1.1$ python setup.py build
running build
running build_py
creating build
…
yourusername@ubuntu: ~/Downloads/pyserial-3.1.1$ sudo python setup.py install
[sudo] password for yourusername: ***
running install
running build
running build.py
…
yourusername@ubuntu: ~/Downloads/pyserial-3.1.1$
```

1.6. Check that installation was successful

1.6.1.   Type the command python and the press enter/return key.

1.6.2.   Type the python command help().

1.6.3.   Type the module name serial.

1.6.4.   If your installation of serial was successful, the terminal will open a file containing the known information on the serial module.

1.6.5.   Press the q to exit the file.

1.6.6.   Press both the control and "d" key twice to exit the Python menus.

TERMINAL VIEW: All commands that the steps above instructed you to type are highlighted in blue.

```
yourusername@ubuntu: ~$ python
Python 2.7.11+ (default, Apr 17 2016, 14:00:29)
[GCC 5.3.1 20160413] on linux2
Type "help", "copyright", "credits" or "license" for more information.
>>> help()
Welcome to Python message …
help> serial
Help on the package serial:
NAME
        Serial
FILE
        /home/yourusername/downloads/pyserial-versionnumber/serial/__init__.py
DESCRIPTION
…
: q
>>>
control key + d key
control key + d key
```

1.6.7.   Congratulations PySerial is now installed!

2. Check if Arduino IDE is Installed

This is a cautionary step, at the time of this writing, the included Arduino Package on Ubuntu 16.04 is dated and not compatible with XLR8. XLR8 requires Arduino version 1.6.4+. Therefore, if you have installed an older version because it is what came prepackaged on your system the steps below will help you remove it. However, if you are the type that runs multiple versions of the Arduino IDE feel free to skip this step.

The following steps will check and see if an Arduino package was installed with the app-get command. Note, if you installed a version of Arduino and did not use apt-get for the install, the commands below will produce terminal output displaying that Arduino IDE is not installed even if it is. To check if Arduino is installed anywhere on your computer, you need to conduct a search. How to perform this search is out of scope for this tutorial since all distributions of Linux have their individual path structures.

2.1. First, check to make sure you did not already install Arduino IDE with apt-get command. Type the command apt-cache policy arduino.
2.2. This command will produce any available information.
2.2.1.   If you see something similar to this "Installed: 2:1.0.5+dfsg2-4" printed to your window, you know that the Arduino package is installed.

2.2.2.   If you see "(none)" next to the "Installed:" on the list of data, you know that the Arduino package is not installed. You can skip to Step 4.

2.3. Uninstall the Arduino package with the following command: sudo apt-get --purge remove arduino. Enter your password when prompted.

2.4. Type apt-cache policy arduino one more time to be sure Arduino is uninstalled.

TERMINAL VIEW: All commands that the steps above instructed you to type are highlighted in blue.

```
yourusername@ubuntu: ~$ apt-cache policy arduino
arduino:
    Installed: 2:1.0.5+dfsg2-4
    Candidate: 2:1.0.5+dfsg2-4
    Version table:
        2:1.0.5+dfsg2-4 500
            500 http://us.archive.ubuntu.com/ubuntu xenial/universe amd64 Packages
            500 http://us.archive.ubuntu.com/ubuntu xenial/universe i386 Packages
yourusername@ubuntu: ~$ sudo apt-get --purge remove arduino
[sudo] password for yourusername: ***
Reading package lists … Done
…
Do you want to continue? [Y/n] y
(Reading database … 242198 files and directories currently installed.)
Removing arduino (2:1.0.5+dfsg2-4) …
…
Processing triggers for mime-support (3.59ubuntu1) …
yourusername@ubuntu: ~$ apt-cache policy arduino
arduino:
    Installed: (none)
    Candidate: 2:1.0.5+dfsg2-4
    Version table:
        2:1.0.5+dfsg2-4 500
            500 http://us.archive.ubuntu.com/ubuntu xenial/universe amd64 Packages
            500 http://us.archive.ubuntu.com/ubuntu xenial/universe i386 Packages
yourusername@ubuntu: ~$
```

2.5. The Arduino package included on your system is now uninstalled.

3. Install Arduino IDE

If you already have the IDE installed, you can skip to Step 3.

3.1.1. Go to *https://www.arduino.cc/en/Main/Software* on their website and download the Linux package appropriate for your operating system.

3.1.2. Type cd ~/Downloads/ in your Terminal application to navigate to your Downloads directory.

3.1.3. Type the command tar –xvf arduino-1.6.11-*.tar.xz to extract the package. Note that if you are installing a different version of the Arduino IDE, the version number will be different. The IDE version number shown in this example is 1.6.11.

3.1.4. To be sure you can use the Arduino IDE globally, move it to your /opt directory. Do this by typing sudo mv arduino-1.6.11 /opt (note the space before "/opt"). When prompted enter your computer account's password.

3.1.5. Type cd /opt/arduino-1.6.11 to navigate to the location you just placed it in.

3.1.6. Change the permissions for the install script so the installation can take place. Type chmod +x install.sh

3.1.7. Type ./install.sh to run the installation script. This will also put a shortcut icon on your Desktop.

TERMINAL VIEW: All commands that the steps above instructed you to type are highlighted in blue.

```
yourusername@ubuntu: ~$ cd ~/Downloads/
yourusername@ubuntu: ~/Downloads$ tar –xvf arduino-1.6.11-linux64.tar.xz
arduino-1.6.11/
arduino-1.6.11/lib/
arduino-1.6.11/lib/jmdns-3.4.1.jar
…
yourusername@ubuntu: ~/Downloads$ sudo mv arduino-1.6.11 /opt
[sudo] passwaord for yourusername: ***
yourusername@ubuntu: ~/Downloads$ cd /opt/arduino-1.6.11/
yourusername@ubuntu: ~/opt/arduino-1.6.11$ chmod +x install.sh
yourusername@ubuntu: ~/opt/arduino-1.6.11$ ./install.sh

Adding desktop shortcut, menu item and file associations for Arduino IDE… touch
…
done!
yourusername@ubuntu: ~/opt/arduino-1.6.11$
```

3.1.8. Arduino IDE is now installed!

4. Modify Port Permissions

To communicate your XLR8 or normal Arduino on Linux you need to tell your computer that it is okay for you to talk to your device through your USB port.

4.1. Type ls -l /dev/tty* to get a list of ports

4.2. Find the group name of your port files. You should now see a list of ports beginning with tty. If you look closely, you should see some listings with the beginning string "crw-rw---- 1 root dialout" or "crw-rw---- 1 root uucp." Note if your ports are listed contain "dialout" or "uucp." This tutorial will use "dialout" for its examples. If your ports are listed with "uucp" just use that term whenever you see "dialout."

4.3. Add your computer's user account to the group by typing sudo usermod –a –G dialout <yourusername> into the terminal. When prompted type the password for your account.

> **TERMINAL VIEW:** All commands that the steps above instructed you to type are highlighted in blue.

```
done!
yourusername@ubuntu: ~/opt/arduino-1.6.11$ ls –l /dev/tty*
crw-rw---- 1 root dialout 4, 0 5 apr 23.01 ttyACM
…
yourusername@ubuntu: ~/opt/arduino-1.6.11$ sudo usermod –a –G dialout <yourusername>
[sudo] password for yourusername: ***
yourusername@ubuntu: ~/opt/arduino-1.6.11$
```

4.4. Log out of your computer's user account and then log back in. Logging out is needed to complete the changes made to your groups in Step 3.3.

You Made It!

You can now open the Arduino IDE from your Desktop's shortcut icon, hook up your XLR8 or Arduino via USB, and program away. When you want to take advantage of the XLR8's custom hardware, you can go to Step 3 in the Quick Start Guide for XLR8.