Initially written by: Eric Trautsch in April 2023. Reach out to erictrautsch@outlook.com with clarifiying questions.

Prerequisite Information and Tools

Environment and Image

We are running Pynq 2.6 on Ultra96v2 (Runs xilinx OS)

Note: Password for the download of these images is `vlc`, and will expire October 13, 2023 from my University OneDrive. (Will update docs when a different solution is found)

My image prior to installing vlc: here. This is the image that you should use if you want to compile VLC from scratch.

My inage post cross-compling vlc-lib here. This is a compile of VLC that works, but still has some issues.

In theory you should be able to do this from direct installation of Pynq 2.6. You may need to copy the `wifi.sh` script that handles connection to the network from the SW Github. (Note, you will need to edit that script to connect to your network).

Setting up Display

I am able to verify the display with this active DisplayPort cable here and a standard DisplayPort compatible monitor. As soon as I start the ultra96, I am getting a view of the main page.

You may need to set the Display environment variable, as shown:

```
export DISPLAY=":0"
```

`

is the value of the Display environment variable that worked for me with this setup.

Flashing SD Cards

To save images from my SD card, I used win32diskImager on my Windows computer to save the image to a file. To flash an image onto an SD card, I used balena etcher (You could probably use win32disk imager for this as well, but I did not).

Development Environment

For development, I use the VS-code remote host extention to connect to the Ultra96 over ssh.

Some notes about `dnf` Package Manager

Some prerequisite packages appear to be installable with the `dnf` installer. This tool has proven difficult for me to verify, but the standard process is:

use `dnf search *` to search packages of a name that you are looking for. For example, ffmpeg.

Then, a list of packages that match the search condition will appear, and you can `dnf install [package_name]` to install these. This tool appears to be a solution to some of the dependency problems we will run into as we get into compiling VLC.

Login Information

Following from other uses, I login as
`user: root`
`password: root`

VLC Compile Overview

From VLC wiki

These are the main steps to follow to compile VLC :

1. Prepare your compile environment,
2. Grab the source of VLC media player,
3. Bootstrap, using the ./bootstrap script, if you took the code from repository,
4. Prepare the extras libraries needed for VLC media player, read the list
5. Configure
6. Build, using make,
7. and Profit!

Prepare compile environment

```
cd /home/root
mkdir vlclib
cd vlclib
```

Get source

To get the VLC source files, run this series of commands. I did this starting at the root home directory (`cd ~`). Note, you can check the docs for the most recent tarball.

```
wget ftp://ftp.videolan.org/pub/videolan/vlc/3.0.12/vlc-3.0.12.tar.xz
tar xvJf vlc-3.0.12.tar.xz
cd vlc-3.0.12
```

Prepare dependencies

The following step may fail without some of these dependencies: `subversion yasm cvs cmake ragel`. You can dnf search and dnf install if there is an associated package.

The following should also be attempted to dnf install `ALSA and OpenGL.`

```
cd contrib
mkdir native
cd native
../bootstrap
make
```

need to pass BUILDCC as environment variable
`whereis gcc`
is the location of gcc (C compiler). It should include `/usr/lib/gcc`. If you do not have gcc, `dnf install gcc`

Then, run

```
export BUILDCC=/usr/lib/gcc
```

Configure

Now we are ready to configure the install, full configure syntax that worked for me is:

```
./configure --enable-x11 --enable-xvideo --disable-gtk --enable-sdl --enable-ffmpeg
--with-ffmpeg-mp3lame --disable-lua --disable-avcodec --disable-swscale
```

This is a moving target, we may want to try other configurations if there are issues with functionality.

Build

In the install directory:

```
make
```

Install

In the install directory:

```
make install
```

Optional: Install `python` bindings

Once the

```
pip3 install python-vlc
```

How to run C scripts with installed libVLC

After installing the libraries, we still need to include libVLC

```
export VLC_SRC=/home/root/vlclib/vlc-3.0.12
gcc -I ${VLC_SRC}/include/ -lvlc -L ${VLC_SRC}/src/.libs/ example.c -o example
```

Known issues and todos after install:

These are some things that I had to do to get libVLC running. Some of these might be need to be modified or not needed to run libVLC.

- VLC does not like to be run as root user. There is a way to get around this that involces these commands. If you search the error message there is a good stack overflow thread

```
sed -i 's/geteuid/getppid/' /usr/bin/vlc
sed -i 's/geteuid/getppid/' /usr/local/vlc
whereis vlc
sed -i 's/geteuid/getppid/' /usr/local/bin/vlc
```

- The libVLC files need to be linked to normal directories. Can be accomplished with these commands:

```
sudo ln -s /usr/local/bin/libvlc* /usr/bin/
sudo ln -s /usr/local/bin/vlc /usr/bin/vlc
sudo ln -s /usr/local/bin/libx264.a /usr/bin/
```

- The avcodec library is not avalible in my compliation of vlclib. This makes it difficult to verify if the installation is completely correct.
  - This is provided by ffmpeg... Might need to also cross-compile that!
- Used this to add libav*.pc files to path searching for. , if it comes up
- One dependency we do not have is the Lua (programming language) byte compiler. This has impacts on the final VLC, but may not matter for our uses.

References

- General UNIX compile instructions
- Configure Documentation