Software Development for Embedded GNU Radio Applications

Philip Balister
philip@opensdr.com

Open SDR

May 28, 2015

1. Embedded SW Development
2. OpenEmbedded
3. Develop Software
4. Creating a Finished Product

Who Am I

- Open source hippie since Linux 0.95a
- MSEE from Virginia Tech in 2007
- Ported OSSIE to an OMAP
- Heavily involved with OpenEmbedded
- Responsible for the GNU Radio Embedded working group
- Currently working with Ettus on the embedded products

Challenges

- Smaller systems (CPU and RAM)
- GNU Radio requires large amounts of RAM to compile
- Cross compiling is traditionally complicated
- Testing!
- Integrating finished flow graph back into image

Resource Issues

- CPU Speed verus power
- Memory Size. 4GB Address limit
- No dedicated swap space
- Inspite of advances in mobile processors, still not SW dev platforms!

Compiling GNU Radio on Embedded Device

- The dreaded Out Of Memory Killer
- (User sees "internal compiler error message")
- Workaround by adding a swapfile
- Pay attention to compiler options
- Compile time still an issue

Cross Compiling

- Build software on one machine to run on another
- Use powerful computers to solve resource issues
- Contamination from host software
- Some packages assume native compile
- Benefits out weigh the problems!

What is OpenEmbedded

- A system for creating custom Linux distributions
- Collects source
- Compiles packages
- Creates image files for flashing
- Helps manage software licenses and license compliance
- Member organization of the Yocto Project

Using OpenEmbedded to Generate an SDK

- SDK's are complicated beasts
- Toolchain - compiler, assembler, linker etc
- Libraries - libc, stdlibc++, and application specific libraries
- Host tools - don't depend on the ones on the dev machine
- OpenEmbedded creates installable sdk's based on an image recipe

Adding extra packages to sdk

- How to add packages that run on the host

```
TOOLCHAIN_HOST_TASK_append = " \
    nativesdk-python-netserver \
    nativesdk-python-pickle \
    nativesdk-python-subprocess \
    nativesdk-orc nativesdk-swig \
    nativesdk-python-xml \
    nativesdk-cmake \
    "
```

Using the SDK

- http://gnuradio.org/redmine/projects/gnuradio/wiki/Cross_compile_an_OOT_and_install_on_target
- SDK environment file

```
export CC="arm-oe-linux-gnueabi-gcc \
-march=armv7-a -mfloat-abi=hard \
-mfpu=neon --sysroot=$SDKTARGETSYSROOT"
export AS="arm-oe-linux-gnueabi-as "
export LD="arm-oe-linux-gnueabi-ld  \
--sysroot=$SDKTARGETSYSROOT"
export CFLAGS=" -O2 -pipe -g \
-feliminate-unused-debug-types"
export CXXFLAGS=" -O2 -pipe -g \
-feliminate-unused-debug-types"
export LDFLAGS="-Wl,-O1 -Wl,--hash-style=gnu \
-Wl,--as-needed"
```

Using the SDK - compiling

```
$ cmake –Wno–dev –DCMAKE_TOOLCHAIN_FILE= \
../../gnuradio/cmake/Toolchains/oe–sdk_cross.cmake
–DENABLE_DOXYGEN=OFF ../
$ make
```

Using the SDK - Toolchain file

- I was going to show the the toolchain file .....
- Toolchain file uses the variables from the SDK enviroment
- Cmake only uses headers and libraries from the SDK

Testing the Output

- Use sshfs to mount targets file system
- Use `make install DESTDIR=$HOME/mydevice`
- Many alternatives to accomplish this
- Pay attention to install paths
- May need to use LD_PRELOAD or LD_LIBRARY_PATH for some cases

Finished Product

- Standard GNU Radio images focused on development
- Real products should focus application without excess baggage
- Read the contents of packagegroups
- Drop uneeded packagegroups
- Verify IMAGE_FEATURES
- Don't forget passwords

Modifying an Image Recipe

```
SUMMARY = "A console-only image with a development/debug \
environment suitable for building GNURadio out of tree blocks installed."

require version-image.inc

IMAGE_FEATURES += "splash ssh-server-openssh tools-sdk \
                   tools-debug debug-tweaks \
                   dev-pkgs dbg-pkgs \
                  "

EXTRA_IMAGE_FEATURES += "package-management"

LICENSE = "MIT"

require recipes-images/images/native-sdk.inc

CORE_IMAGE_EXTRA_INSTALL = "\
    packagegroup-sdr-base-extended \
    packagegroup-sdr-debug \
    packagegroup-sdr-devel \
    packagegroup-sdr-python \
    packagegroup-sdr-gnuradio \
    packagegroup-sdr-hardware \
    xauth \
    "

#    zeroc-ice

inherit core-image
```

Questions

Questions?

License Links

- https://www.youtube.com/watch?v=Tm8axkXu2R4
- https://sfconservancy.org/
- http://www.gnu.org/licenses/
- http://mil-oss.org/resources/articles-papers-presentations
- http://mil-oss.org/resources/suggested-reading