



Cross-Compiling Python & C Extensions for Embedded Systems

Christopher Lambacher

<http://whatschrisdoing.com/>

<http://twitter.com/lambacck>

<http://kclsoftware.com/>

What will we be covering today?

- *Not* covering porting Python to a new OS
- Compiling on Linux x86/x86_64 to target:
 - PPC
 - ARM
 - MIPS
- Examples today target ARM

Example devices

- Panda Board
 - Beagle (Bone/Board/Board MX)
 - Hawk Board
 - Panda Board
 - Others
-
- RaspberryPi (\$35)

Why not use Open Embedded?

- No recent Python version support (3.x)
- Missing 3rd party extensions
- Require custom extensions
- Required to use another platform

COMPILE



ALL THE THINGS.



**Python doesn't know about
cross-compiling!**

Don't worry, you just...

```
$ ./configure
$ make install
$ cp python hostpython
$ cp Parser/pgen Parser/hostpgen
$ make distclean
$ patch -p1 < Python-3.3.0a1-xcompile.patch
$ PLAT=arm-angstrom-linux-gnueabi-
$ PATH=$PATH:$HOME/work/PyConTalk2012/Tools/setup-scripts/build/tmp-
angstrom_2010_x-eglibc/sysroots/x86_64-linux/usr/bin/armv7a-angstrom-linux-
gnueabi
CC=${PLAT}gcc CXX=${PLAT}g++ AR=${PLAT}ar RANLIB=${PLAT}ranlib ./configure --
host=arm-linux --build=x86_64-linux-gnu --prefix=/opt/python33
$ CROSSBASE=/home/lambacck/work/PyConTalk2012/Tools/setup-scripts/build/tmp-
angstrom_2010_x-eglibc/sysroots/beaglebone CFLAGS="-I${CROSSBASE}/include -
I${CROSSBASE}/usr/include" LDFLAGS="-L${CROSSBASE}/lib -L${CROSSBASE}/usr/lib"
make HOSTPYTHON=./hostpython HOSTPGEN=./Parser/hostpgen BLDSSHARED="${PLAT}gcc
-shared" CROSS_COMPILE=${PLAT} CROSS_COMPILE_TARGET=yes HOSTARCH=arm-linux
BUILDARCH=x86_64-linux-gnu
$ make install HOSTPYTHON=./hostpython BLDSSHARED="${PLAT}gcc -shared"
CROSS_COMPILE=${PLAT} CROSS_COMPILE_TARGET=yes
prefix=$HOME/work/PyConTalk2012/python33
```

It looks worse than it is?

- Build Python for your host system
- Take Copies of `python` and `Parser/pgen`
- Clean build directory
- Patch Python
- `configure` (while telling it about target machine)
- `make` (while telling it about target machine)
- `make install` (while telling it about target machine and overriding `$prefix`)



**Python doesn't know about
cross-compiling!**

What could possibly go wrong?

- Must have correct patch
- Missing include/library search path
- Missing C dependencies

Include / Library Path

Python build finished, but the necessary bits to build these modules were not found:

```
_bz2                _curses
_curses_panel
_dbm                 _gdbm                _lzma
_sqlite3             _ssl                  _tkinter
readline             zlib
```

To find the necessary bits, look in `setup.py` in `detect_modules()` for the module's name.

3rd party extensions

```
export CC="${PLAT}gcc -pthread"
export LDSHARED="${CC} -shared"
export CROSSBASE=/home/lambacck/work/PyConTalk2012/Tools/setup-
scripts/build/tmp-angstrom_2010_x-eglibc/sysroots/beaglebone
export PYPREFIX=${HOME}/work/PyConTalk2012/python33
export CFLAGS="-I${PYPREFIX}/include/python3.3m -
I${CROSSBASE}/usr/include"
export LDFLAGS="-L${PYPREFIX}/lib/python3.3m -L${CROSSBASE}/lib -
L${CROSSBASE}/usr/lib"
export HOSTPYTHON=/home/lambacck/work/PyConTalk2012/Tools/setup-
scripts/build/tmp-angstrom_2010_x-eglibc/sysroots/x86_64-
linux/usr/local/python3.3/bin/python3
$HOSTPYTHON setup.py build
$HOSTPYTHON setup.py bdist_egg
mv dist/MarkupSafe-0.15-py3.3-linux-x86_64.egg MarkupSafe-0.15-
py3.3-linux-arm.egg
```



Python doesn't know about cross-compiling!

(and neither does Distutils / Setuptools / Distribute, or the library you are building)

What could possibly go wrong?

- Include/library search path
 - Hard Coded Search Locations
- Missing C dependencies
- Packages that use `configure`, `make` and `python-config` instead of `setup.py` (like `PyGObject` and `omniORBpy`)
- Platform-specific egg needs correct platform spec (e.g. `linux-arm` instead of `linux-x86_64`)

Include / Library Path

```
In file included from
/home/lambacck/work/PyConTalk2012/Tools
/setup-scripts/build/tmp-
angstrom_2010_x-eGLIBC/sysroots/x86_64-
linux/usr/local/python3.3/include/python
n3.3m/Python.h:50:0,
        from
markupsafe/_speedups.c:12:
/home/lambacck/work/PyConTalk2012/Tools
/setup-scripts/build/tmp-
angstrom_2010_x-eGLIBC/sysroots/x86_64-
linux/usr/local/python3.3/include/python
n3.3m/pyport.h:783:2: error: #error
"LONG_BIT definition appears wrong for
platform (bad gcc/glibc config?)."
```

Other things to worry about

- Application Dependencies
 - virtualenv
 - pip only installs source packages
 - Use `easy_install`
- Other platforms
 - iOS (<https://github.com/dennda/python-for-iphone>)
 - Android (<http://code.google.com/p/android-scripting>)

Where do we go from here?

- Is mobile the future of computing?
- Cross-compiling will become the norm
- Python should have a good story for doing cross-compiling.



Python 3.3 on BeagleBone



Questions?