

Cmpt433 Final Project

Nap Time Systems

How To Guide:

Will it Build? Part 1



Chrome notebook image from willitblend.com

Tcl? Yes! Tk? Maybe...

Building and using Tool Control Language (tcl) on the EM2440

Tcl is a powerful scripting language used in many applications in industry that can be built for the EM2440. For our purposes it can be used to produce dynamic web content which in concert with the boa web server gives us another way to control the board via a graphical user interface.

Building TCL:

1. Download tcl:

```
wget http://prdownloads.sourceforge.net/tcl/tcl8.4.19-src.tar.gz
```

2. Unpack it:

```
tar xzf http://prdownloads.sourceforge.net/tcl/tcl8.4.19-src.tar.gz
```

3. Configure it:

```
cd tcl8.4.19-src/unix
```

run the attached configure script based on the instructions found here:

<http://objectmix.com/tcl/15449-how-cross-compile-tcl8-4-tk8-4-arm-linux.html>

(look for the 3rd post in the thread)

Script contents (assumes \$PUBLIC points to your public directory):

```
#!/bin/sh
# reference
# http://objectmix.com/tcl/15449-how-cross-compile-tcl8-4-tk8-4-arm-linux.html
export CROSS_COMPILE=arm-linux
export CC=arm-linux-gcc
export ac_cv_func_strtod=yes
export tcl_cv_strtod_buggy=1
./configure --prefix=$PUBLIC/tcl \
--target=arm-linux \
--host=arm-linux \
--build=i386-pc-linux \
```

```
--with-endianness=little \  
--with-cc=arm-linux-gcc \  
--with-ar=arm-linux-ar \  
--with-install-prefix=../.. \  
--with-cflags="-O0 -msoft-float -march=armv4t -mtune=arm920t -mcpu=arm920t"
```

4. Build it:

make && make install

If you cannot copy the tclsh8.4 binary and the tcl libraries to the board mount an sd card or nfs share and modify your LD_LIBRARY_PATH and PATH variables to point to the lib and bin directories.

Tcl also usually comes with Tk a portable windowing library. As Tk depends on X11 you have to build X11 to use it. Here is a good step by step guide <http://www.x.org/wiki/CrossCompilingXorg>. The X consortium have created very easy to use tools for building this humongous beast. However, the build broke for me in two (drm/test and mesa/glsl) out of the 59 packages. The first was able to simply pass over the error in the build and in the second I believe a small change to the make file would fix the problem. In general if you get an error from make saying “expected ')' but got text” that is because the make process is trying to run something compiled for another host architecture.

To use Tcl you can run it from the command line:

```
tclsh8.4  
  
>> echo "hello world!"  
  
hello world!
```

Or use the `#! {path to tcl bin}/tclsh8.4` at the top of a script.

Here is a tutorial on how to use tcl:

<http://wiki.tcl.tk/1304>

For dynamic web programming (cgi):

<http://wiki.tcl.tk/1389>

LAME? Yes

Does it run quickly – no sadly. A 5 minute song might take 40 minutes to encode or decode.

To build it download the sources as usual:

<http://sourceforge.net/projects/lame/files/lame/3.99/lame-3.99.2.tar.gz/download>

Configure options:

```
./configure '--host=arm-linux' '--prefix=/home/cal/cmpt433/private/lame' 'CFLAGS=-O0 -O0 -msoft-float -march=armv4t  
-mtune=arm920t -mcpu=arm920t' 'CPPFLAGS=-O0 -O0 -msoft-float -march=armv4t -mtune=arm920t -mcpu=arm920t' 'host_alias=arm-  
linux'
```

Make and make install as usual.

Linphone? Yes

Linphone is a full featured SIP protocol IP phone that is designed to be cross compiled. For our project we use a component of linphone, oRTP, to provide network streaming with the Real Time Protocol.

Another potentially very useful library is the mediastreamer2 library which allows an application developer to chain together encoding and sending over the network via a set of independent modules.

Linphone download:

<http://mirror.csclub.uwaterloo.ca/nongnu/linphone/stable/sources/>

Main website:

<http://www.linphone.org/>

Mediastreamer:

<http://www.linphone.org/eng/documentation/dev/mediastreamer2.html>

Mediastreamer depends on libogg:

<http://www.xiph.org/downloads/>

oRTP depends on speex (even if you –disable-speex in the configure step)

<http://www.speex.org/downloads/>

The mediastreamer.sh script encapsulates the build process for mediastreamer, oRTP and speex:

```
#!/bin/sh
```

```
# run in the directory where you wish to download files

# assumes the $PR variable is set which points to our private directory

# this does not build linphone as ncurses does not seem to work on the board

Wget http://mirror.csclub.uwaterloo.ca/nongnu/linphone/stable/sources/linphone-3.4.3.tar.gz

tar xzf linphone-3.4.3.tar.gz

Cd linphone-3.4.3/

Cd oRTP

Make clean

./configure --prefix=$PR/ortp --host=arm-linux CFLAGS="$ARMOPTS" CPPFLAGS="$ARMOPTS"

Make

Make install

Cd ../../

Wget http://downloads.xiph.org/releases/ogg/libogg-1.3.0.tar.gz

tar xzf libogg-1.3.0.tar.gz

Cd libogg-1.3.0/

Make clean

./configure --prefix=$PR/speex --host=arm-linux CFLAGS="$ARMOPTS" CPPFLAGS="$ARMOPTS"

Make

Make install

Wget http://downloads.xiph.org/releases/speex/speex-1.2rc1.tar.gz

tar xzf speex-1.2rc1.tar.gz

Cd speex-1.2rc1/

Make clean

./configure --prefix=$PR/speex --host=arm-linux CFLAGS="-I$PR/speex/include $ARMOPTS" CPPFLAGS="-I$PR/speex/include $ARMOPTS" LDFLAGS="-L$PR/speex/lib"

Make

Make install

Cd ../linphone-3.4.3/oRTP/

Make clean

./configure --prefix=$PR/ortp --host=arm-linux CFLAGS="-I$PR/speex/include $ARMOPTS" CPPFLAGS="-I$PR/speex/include $ARMOPTS" LDFLAGS="-L$PR/speex/lib"

Make

Make install
```

```
Cd ../mediastreamer2/
```

```
Make clean
```

```
Export SPEEX_CFLAGS="-I$PR/speex/include" SPEEX_LIBS="-L$PR/speex/lib -lspeex -lm -lspeexdsp"
```

```
./configure --prefix=$PR/ortp --host=arm-linux CFLAGS="-I$PR/speex/include -I$PR/ortp/include -I$PR/alsa/include $ARMOPTS"  
CPPFLAGS="-I$PR/speex/include -I$PR/ortp/include -I$PR/alsa/include $ARMOPTS" LDFLAGS="-L$PR/speex/lib -L$PR/ortp/lib  
-L$PR/alsa/lib" --disable-video --disable-pulseaudio
```

```
Make
```

```
Make install
```

The most useful items for sending and receiving real time data were the test scripts in oRTP/tests.

The mediastreamer encoding and decoding process are most likely excessive for the EM2440. With raw pcm data being piped into an oRTP test program we found that it was not able to reliably handle anything more than a 4khz sample rate.

If you wish to build linphone you will need to get and build ncurses. However, if you have built alsamixer you will already have these libraries. Note that the more advanced terminal features do not work on the EM2440 III. To actually use the phone you would have to use another type of interface such as the touch screen.