***CROSS\_COMPILING TSLIB***

[root@linux](mailto:da3m0n@linux)**:** mkdir /usr/local/mini2440

[root@linux](mailto:da3m0n@linux): cd /usr/local/mini2440

***Download tslib source code:***

[root@linux](mailto:da3m0n@linux): mini2440# git clone <http://github.com/kergoth/tslib.git>

[root@linux](mailto:da3m0n@linux): mini2440# cd tslib

***Exporting path to installed cross-compiler:***

[root@linux](mailto:da3m0n@linux): mini2440# export CC\_MINI=/path/to/installed/cross-tool-chain/of/target/

Ex:

export CC\_MINI=/embedd/mini2440/Toolchain/buildroot-2014.02/output/host/usr/bin/arm-linux-

***OR***

[root@linux](mailto:da3m0n@linux): mini2440# PATH=$PATH:/embedd/mini2440/Toolchain/buildroot-2014.02/output/host/usr/bin/

***Cross compiling:***

[root@linux](mailto:da3m0n@linux): tslib# ./autogen-clean.sh

[root@linux](mailto:da3m0n@linux): tslib# ./autogen.sh

***Setup for compilation:***

[root@linux](mailto:da3m0n@linux): tslib# echo "ac\_cv\_func\_malloc\_0\_nonnull=yes" > arm-linux.autogen

[root@linux](mailto:da3m0n@linux): tslib# export CC=$CC\_MINI"gcc"

**(or)**

If u export path using PATH env variable then use bellow:

[root@linux](mailto:da3m0n@linux): tslib# export CC=arm-linux-gcc

[root@linux](mailto:da3m0n@linux): tslib# export CXX=$CC\_MINI"g++"

**(or)**

[root@linux](mailto:da3m0n@linux): tslib# export CXX=arm-linux-g++

[root@linux](mailto:da3m0n@linux): tslib# export CONFIG\_SITE=arm-linux.autogen

***Installing*:**

[root@linux](mailto:da3m0n@linux): tslib# ./configure --build=i386-linux --host=arm-linux --target=arm --enable-static --enable-shared --prefix=$PWD/build

[root@linux](mailto:da3m0n@linux): tslib# make

[root@linux](mailto:da3m0n@linux): tslib# make install

[root@linux](mailto:da3m0n@linux): tslib# ls build

bin etc include lib share

[root@linux](mailto:da3m0n@linux): tslib# ls build/\*

bin:

ts\_calibrate ts\_harvest ts\_print ts\_print\_raw ts\_test

etc:

ts.conf

include:

tslib.h

lib:

libts-1.0.so.0 libts.a libts.so ts/

libts-1.0.so.0.0.0 libts.la pkgconfig/

share:

man

**Testing TOUCHSCREEN using tslib:**

run **ts\_calibrate** then **ts\_print** then **ts\_test**

**/# ./ts\_calibrate**

Neccessary for device to capture correct pointer place(Precise.).

Otherwise some gap will be maintained b/w user pointed point with stylus

and mouse pointer displayed on screen.

**/# ./ts\_print**

touch and see co-ordinates

**/# ./ts\_test**

select draw

Draw pattern of ur choice.

***CROSS\_COMPILING ZLIB***

**Downloading zlib:**

[root@linux](mailto:da3m0n@linux): cd /usr/local/mini2440

[root@linux](mailto:root@linux): mini2440# wget <http://zlib.net/zlib-1.2.8.tar.gz>

[root@linux](mailto:root@linux): mini2440# tar xvf zlib-1.2.8.tar.gz

[root@linux](mailto:root@linux): mini2440# cd zlib-x.x.x

[root@linux](mailto:root@linux): zlib-x.x.x# prefix=./build CC=/path2/cross/compiled/toolchain/ CFLAGS="-O4" ./configure

**(OR)**

If path exported already then

[root@linux](mailto:root@linux): zlib-x.x.x# prefix=./build CC=arm-linux-cc CFLAGS="-O4" ./configure

The configure file will endup in creating Makefile

[root@linux](mailto:root@linux): zlib-x.x.x# make

[root@linux](mailto:root@linux): zlib-x.x.x# make install

Then the zlib will be installed in your given path.

[root@linux](mailto:root@linux): zlib-x.x.x# ls build/\*

include:

zconf.h zlib.h

lib:

libz.a libz.so libz.so.1 libz.so.1.2.8 pkgconfig

share:

man

**Installing zlib into tslib:**

[root@linux](mailto:root@linux): cp -Rfp /your/given/path/to/zlib/build/\* /your/given/path/to/tslib/build/

***Recepie:***

[root@linux](mailto:root@linux): ls tslib/build/\*

bin:

ts\_calibrate ts\_harvest ts\_print ts\_print\_raw ts\_test

etc:

ts.conf

include:

tslib.h zconf.h zlib.h

lib:

libts-1.0.so.0 libts.a libts.so libz.so libz.so.1.2.8 ts

libts-1.0.so.0.0.0 libts.la libz.a libz.so.1 pkgconfig

share:

man