# 1 Create An Overlayer

To build your own layer you need to create:

## 1.1 layer.conf:

```
#######start#######
#conf/layer.conf

BBFILES += "${LAYERDIR}/recipes/*/*.bb"
BBPATH .= ":${LAYERDIR}"
BBFILE_COLLECTIONS += "mcm"
BBFILE_PRIORITY_mcm = "2"
BBFILE_PATTERN_mcm = "^${LAYERDIR}/"

#######end#######
```

## 1.2 Recipe(s):

```
#######start#######

#recipes/mcm440-phone-app_git.bb

DESCRIPTION = "MCM440-phone-app Application"
LICENSE = "GPL"
# Increase the version counter if this file get changed, after bitbake has built it already once
successfully
PR = "r12"


SRC_URI = "git://github.com/christophpurrer/mcm440-phone-app.git;protocol=http;tag=master"
# SRC_URI = "git:///home/feistling/workspace/mcm440-phone-app/;protocol=git;tag=master"
# SRC_URI= "file:///home/feistling/workspace/mcm440-phone-app/"

inherit qt4x11

# S >> locate the resources
# S = "${WORKDIR}/${P}"
S = "${WORKDIR}/git/src/mcm440-phone-app-mpp/"

# S = "/home/feistling/workspace/mcm440-phone-app/src/mcm440-phone-app-mpp"
```

```
# compile task is always the same so we do not define it here

# add example from the slides
do_install() {
        export INSTALL_ROOT=${D}
        make install
}

#######end#######
```

### 1.3 Image

```
#######start#######

#recipes/mcm440-phone-app-image.bb
#gives you a an awesome beagleboard image with fancy functionality
PR = "r15"

require recipes/images/base-image.bb

XSERVER = "xserver-kdrive-fbdev"

IMAGE_INSTALL += " \
        libqtdeclarative4 \
        ${XSERVER} \
        mcm440-phone-app \
        hello \
        "
# mcm440-phone-app \

export IMAGE_BASENAME = "mcm440-phone-app-image"

#inherit image

# change some configuration files in the rootfilesystem
mcm440_phone_app_image_rootfs_postprocess() {
        curdir=$PWD
        cd ${IMAGE_ROOTFS}

        # create/overwrite network configuration
        echo "auto usb0" > ./etc/network/interfaces
        echo "iface usb0 inet static" >> ./etc/network/interfaces
        echo "address 192.168.0.202" >>  ./etc/network/interfaces
        echo "netmask 255.255.255.0" >>  ./etc/network/interfaces
        echo "network 192.168.0.0" >>    ./etc/network/interfaces
        echo "gateway 192.168.0.200" >>  ./etc/network/interfaces
```

```
        # activate the SSH dropbear daemon
        echo "DROPBEAR_KEYTYPES=\"rsa\"" >> ./etc/default/dropbear

        # change opk package management system feed urls
        rm -rf ./etc/opkg/*-feed.conf
        echo "src/gz all http://192.168.0.200/all" >                ./etc/opkg/all-feed.conf
        echo "src/gz armv7a http://192.168.0.200/armv7a" >          ./etc/opkg/arm7a-feed.conf
        echo "src/gz beagleboard http://192.168.0.200/beagleboard" > ./etc/opkg/beagleboard-feed.conf

        cd $curdir
}

# register the above command for execution
ROOTFS_POSTPROCESS_COMMAND += "mcm440_phone_app_image_rootfs_postprocess"

#######end#######
```

## 2 Further Configuration

In the build directory of your openembbed you must add your own layer to the BBLAYERS in the bblayers.conf.

```
#######start#######

#build/conf/bblayers.conf
# LAYER_CONF_VERSION is increased each time build/conf/bblayers.conf
# changes incompatibly
LCONF_VERSION = "5"

BBFILES ?= ""

# Add your overlay location to BBLAYERS
# Make sure to have a conf/layers.conf in there
BBLAYERS = " \
  /home/alex/Hagenberg/EmbbedLinux/openembbed/sources/openembedded \
  /home/alex/Hagenberg/EmbbedLinux/openembbed/mcm440-layer \
  "

#######end#######
```

# 3 BitBake Your Overlayer

It is never a bad idea to source the OE and bitbake environment variables.

$ source /home/USER/.oe/environment

After that go to the build directory and build your image with:

$ bitbake YOURIMAGE -g

In our case it was:

$ bitbake mcm440-phone-app-image -g

With the param -g, the dependency tree is written in a dot file.

# 4 Move Your Image To A Beagleboard

After a long time waiting until the build process of your image is finished, you can copy your image and the necessary kernel modules to your SD card on the beagleboard.

The image and modules file can be found in the TMPDIR directory. The TMPDIR directory can be specified in the local.conf in the conf/build directory.

Copy and extract the content of this two files on your SD. Don't forget, superuser rights are need to copy and extract the content.

# 5 Start Your Application

Connect to your beagleboard via SSH or Serial, login and start the X server:

$ export DISPLAY=:0
$ Xfbdev --screen  1920x1080@25 -mouse mouse

To use a mouse for your application you need an usb hub. The hub is directly connected with the beagleboard. After that you can start your application. In our case it was the application:

$ mcm440-phone-app

If you have attached your beagleboard to a display via HDMI, switch to the HDMI channel and you will see your application.