YOCTO PROJECT FINAL REVIEW
RISHABH BHATIA 12BCE1029
VIRAT SARDANA 12BCE1123

Application development

Application developers mainly will need Yocto Project ADT cross development environment. Please follow the following steps for setup:

Setup Toolchain

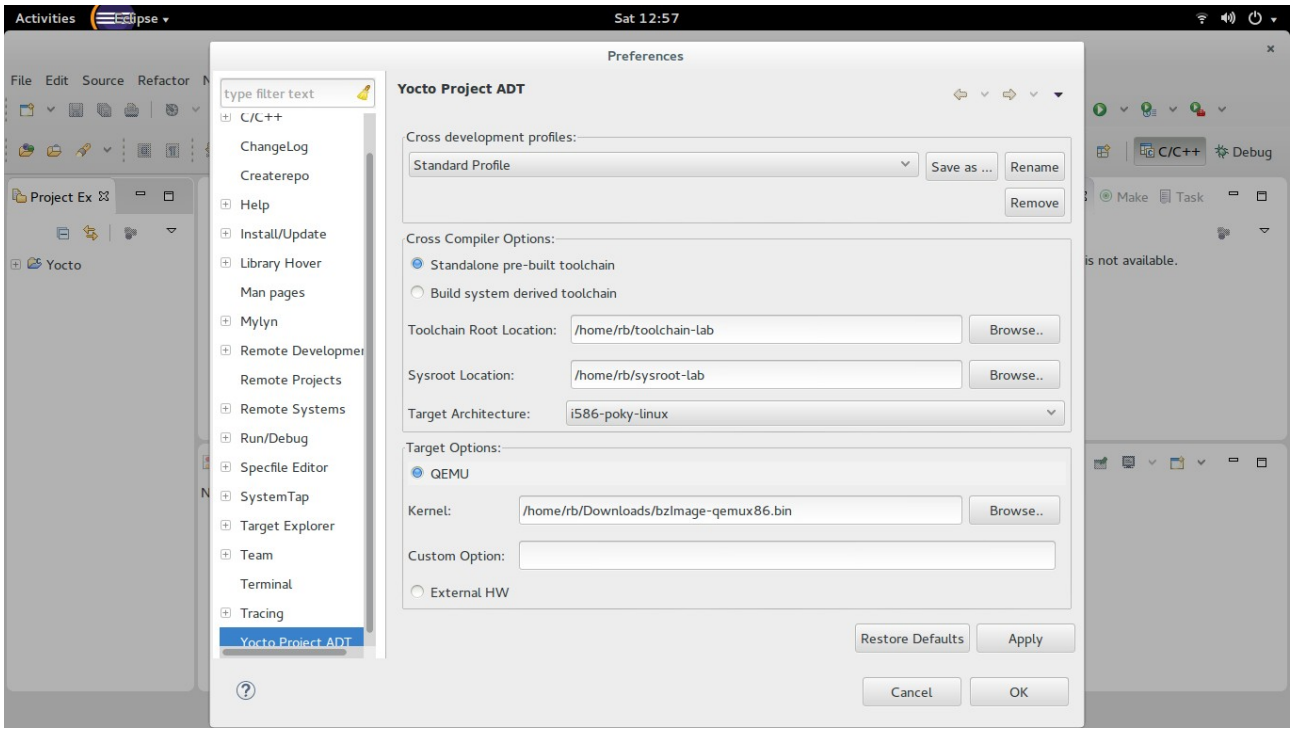
```
rb@rb-Inspiron-N4050:~$ cd /home/rb/Downloads/
rb@rb-Inspiron-N4050:~/Downloads$ chmod +x poky-eglibc-x86_64-i586-toolchain-gma
e-1.3.sh
rb@rb-Inspiron-N4050:~/Downloads$ ./poky-eglibc-x86_64-i586-toolchain-gmae-1.3.s
h
Enter target directory for SDK (default: /opt/poky/1.3): home/rb/toolchain-lab
You are about to install the SDK to "/home/rb/Downloads/home/rb/toolchain-lab".
Proceed[Y/n]?Y
Extracting SDK...done
Setting it up...done
SDK has been successfully set up and is ready to be used.
rb@rb-Inspiron-N4050:~/Downloads$
```

Setup Sysroot

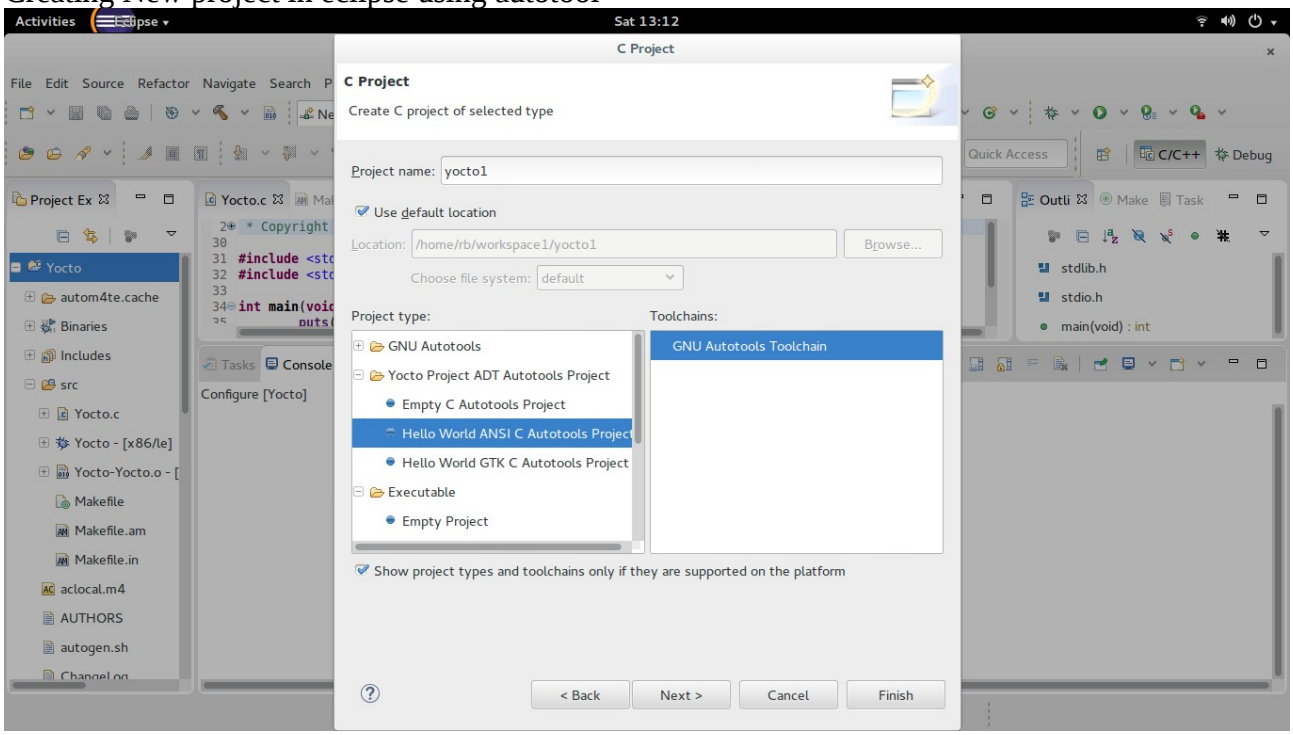
```
rb@rb-Inspiron-N4050:~$ cd toolchain-lab/
rb@rb-Inspiron-N4050:~/toolchain-lab$ source environment-setup-i586-poky-linux
rb@rb-Inspiron-N4050:~/toolchain-lab$ runqemu-extract-sdk ~/Downloads/core-image
-sato-sdk-qemux86.tar.bz2 ~/sysroot-lab
Error: /home/rb/sysroot-lab/../sysroot-lab.pseudo_state already exists!
Please delete the rootfs tree and pseudo directory manually
if this is really what you want.
rb@rb-Inspiron-N4050:~/toolchain-lab$ runqemu-extract-sdk ~/Downloads/core-image
-sato-sdk-qemux86.tar.bz2 ~/sysroot-lab1
Creating directory /home/rb/sysroot-lab1
Extracting rootfs tarball using pseudo...
/home/rb/toolchain-lab/sysroots/x86_64-pokysdk-linux/usr/bin/pseudo -P /home/rb/
toolchain-lab/sysroots/x86_64-pokysdk-linux/usr tar -C "/home/rb/sysroot-lab1" -
xjf "/home/rb/Downloads/core-image-sato-sdk-qemux86.tar.bz2"
SDK image successfully extracted to /home/rb/sysroot-lab1
rb@rb-Inspiron-N4050:~/toolchain-lab$
```

Use Yocto Project eclipse plug-in for cross development

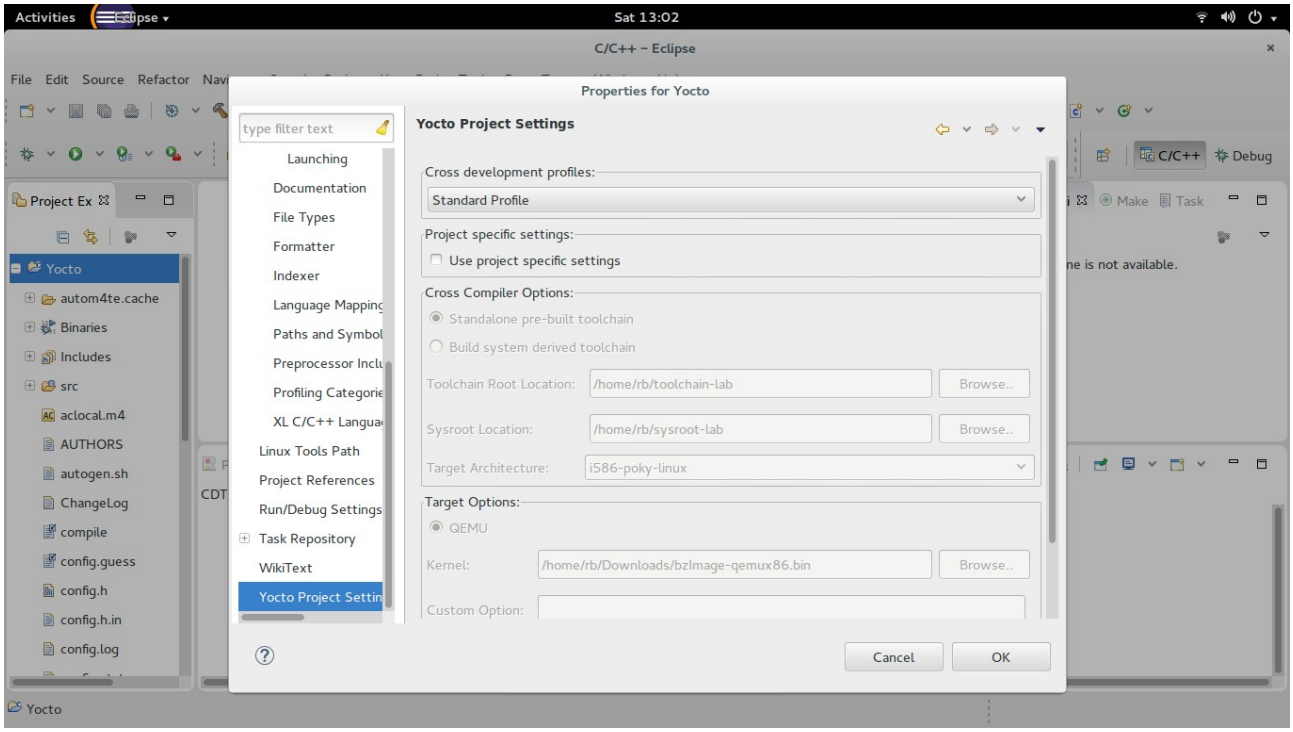
Configure Yocto Project ADT plug-in for cross development



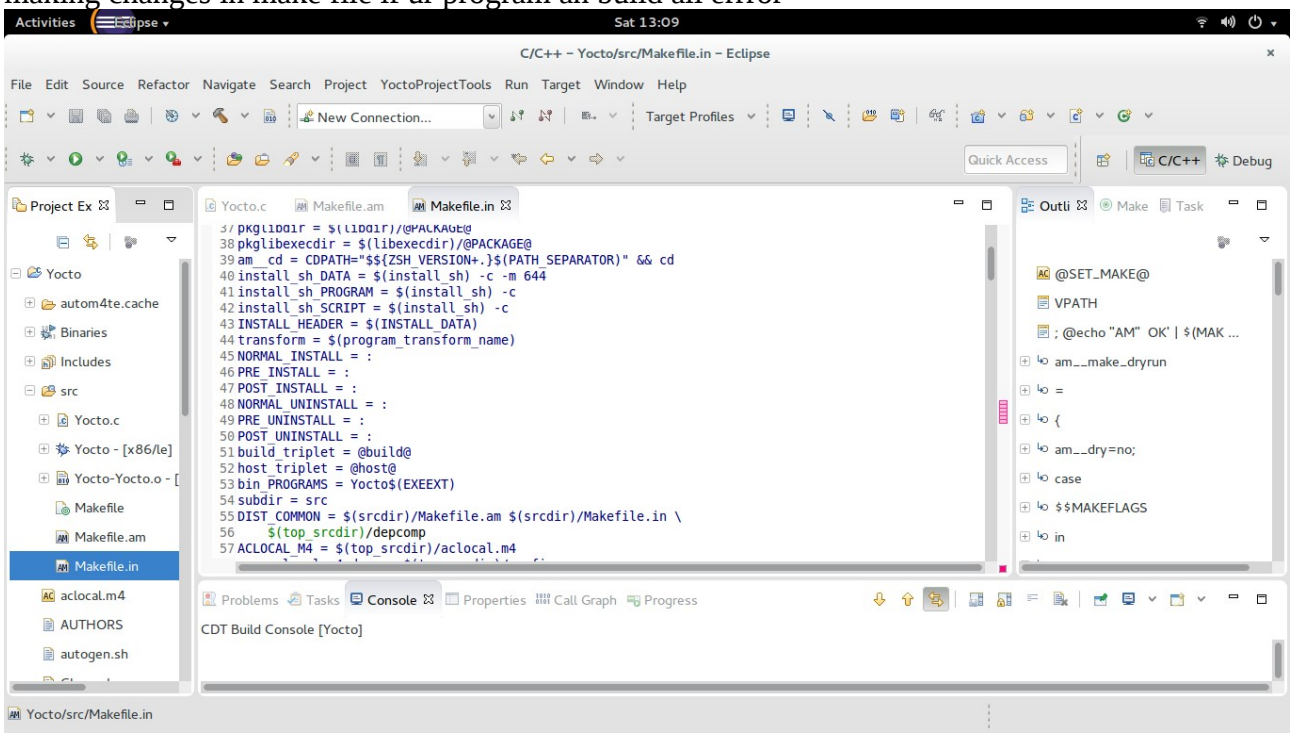
Creating New project in eclipse using autotool



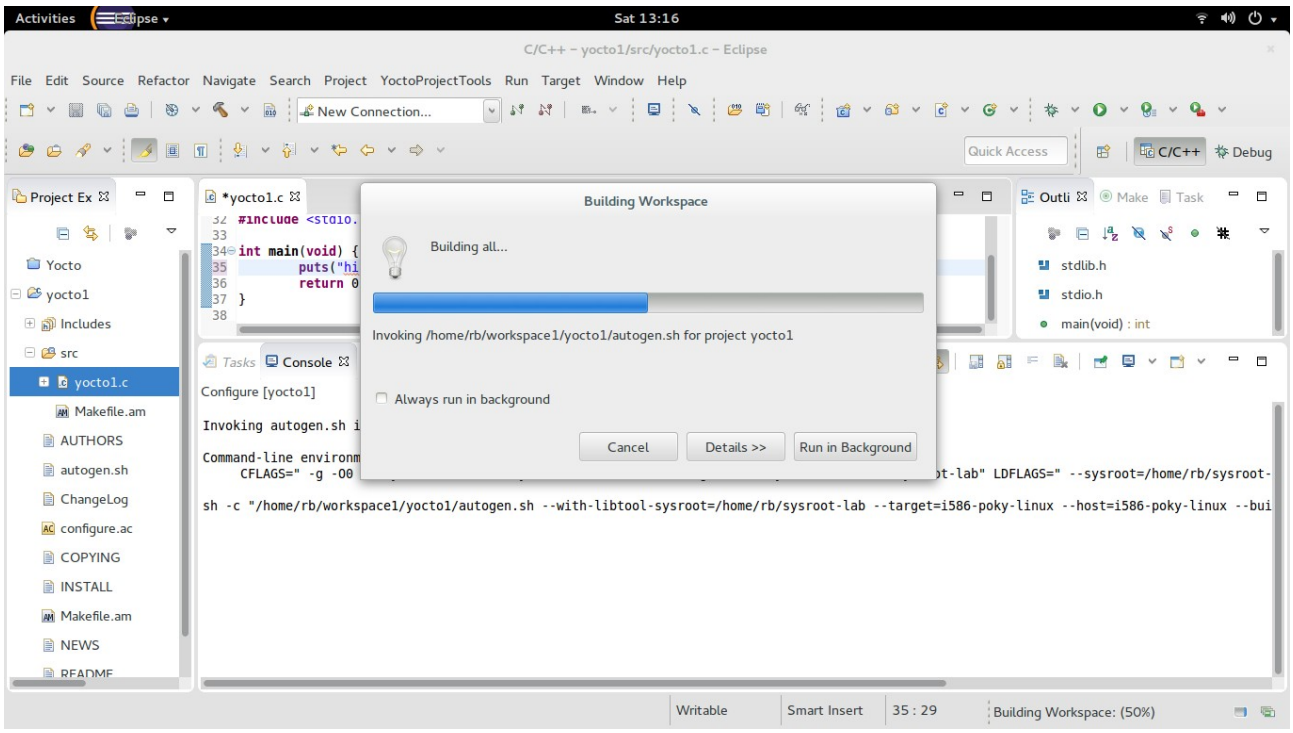
Change Yocto Project Settings

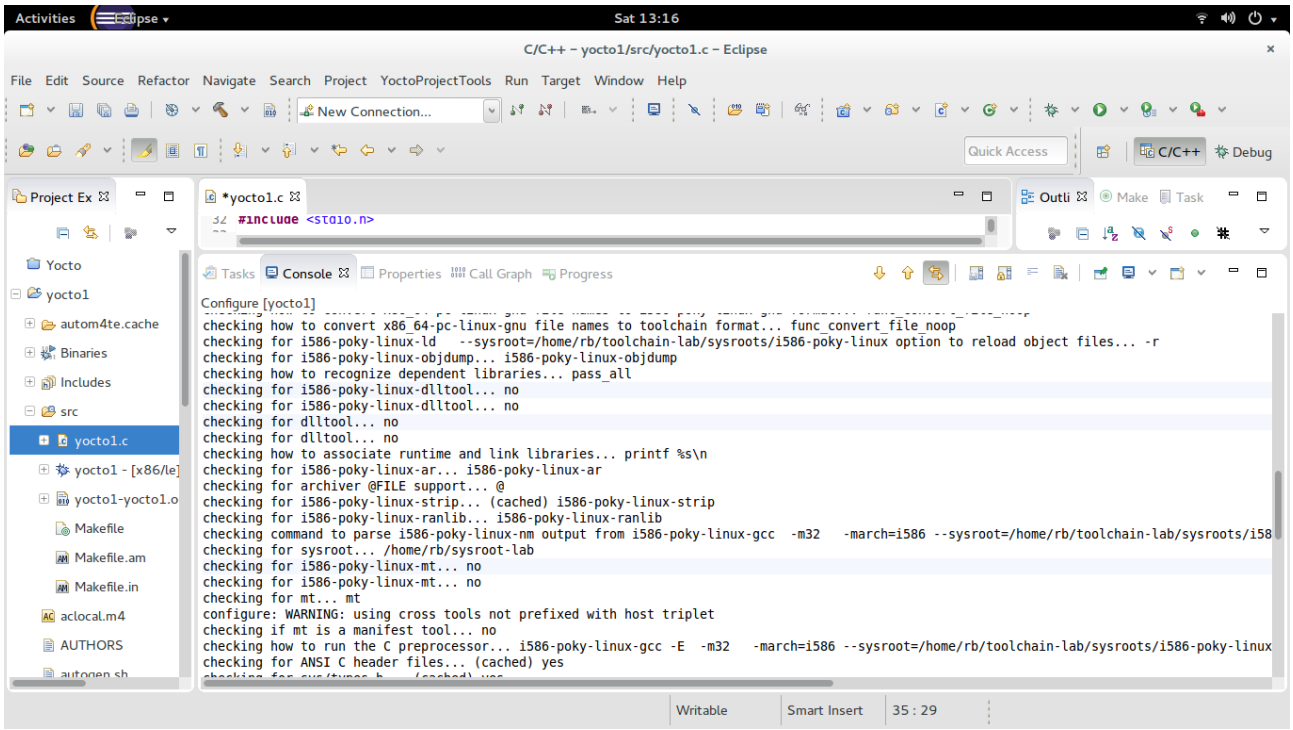


making changes in make file if ur program an build all errror



Building the project to print the text written

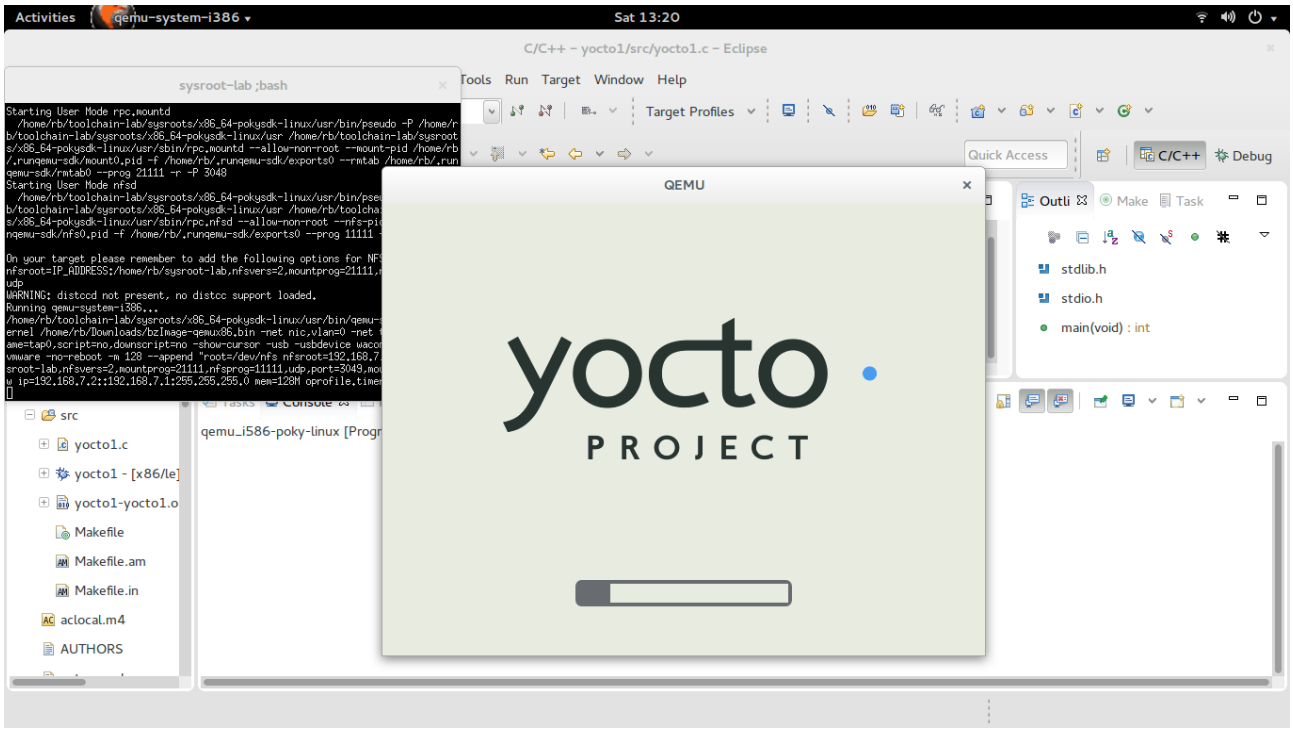




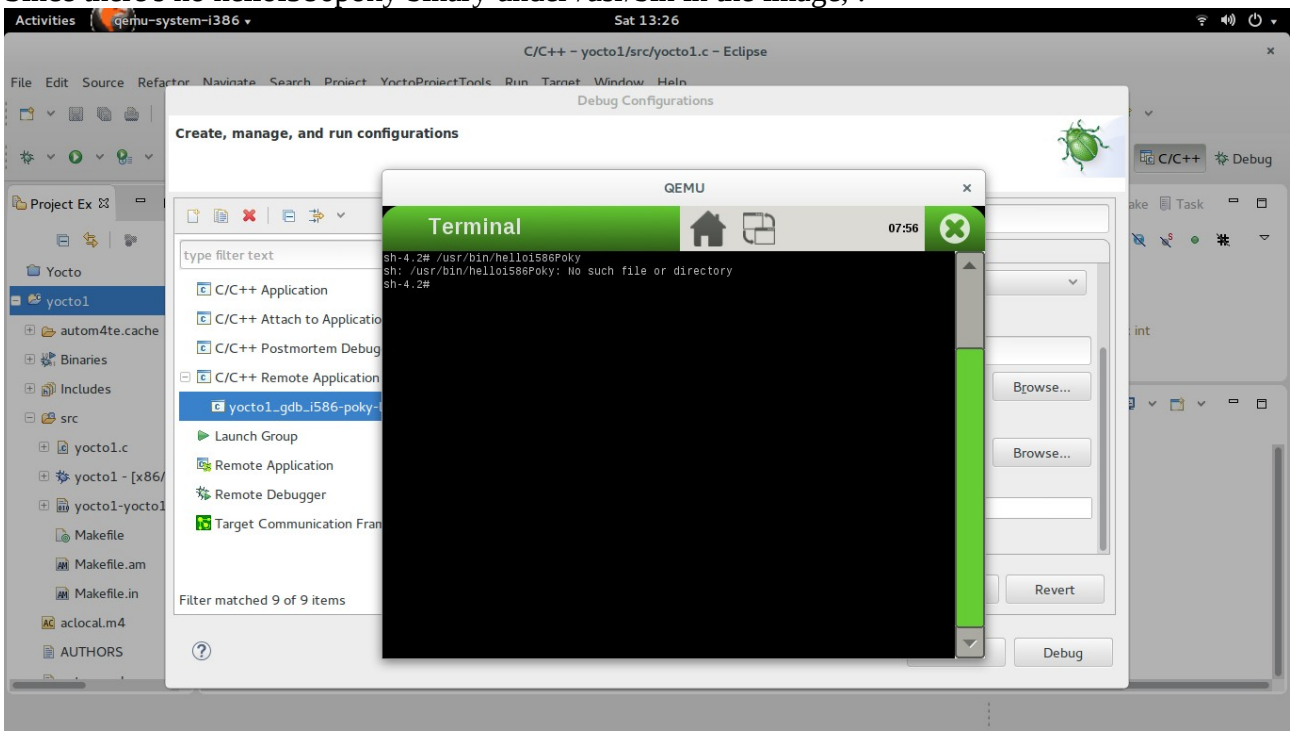
Conclusion

We have created a Yocto Project ADT autotools to print any string and successfully cross compiled it with your Yocto Project ADT cross development setup

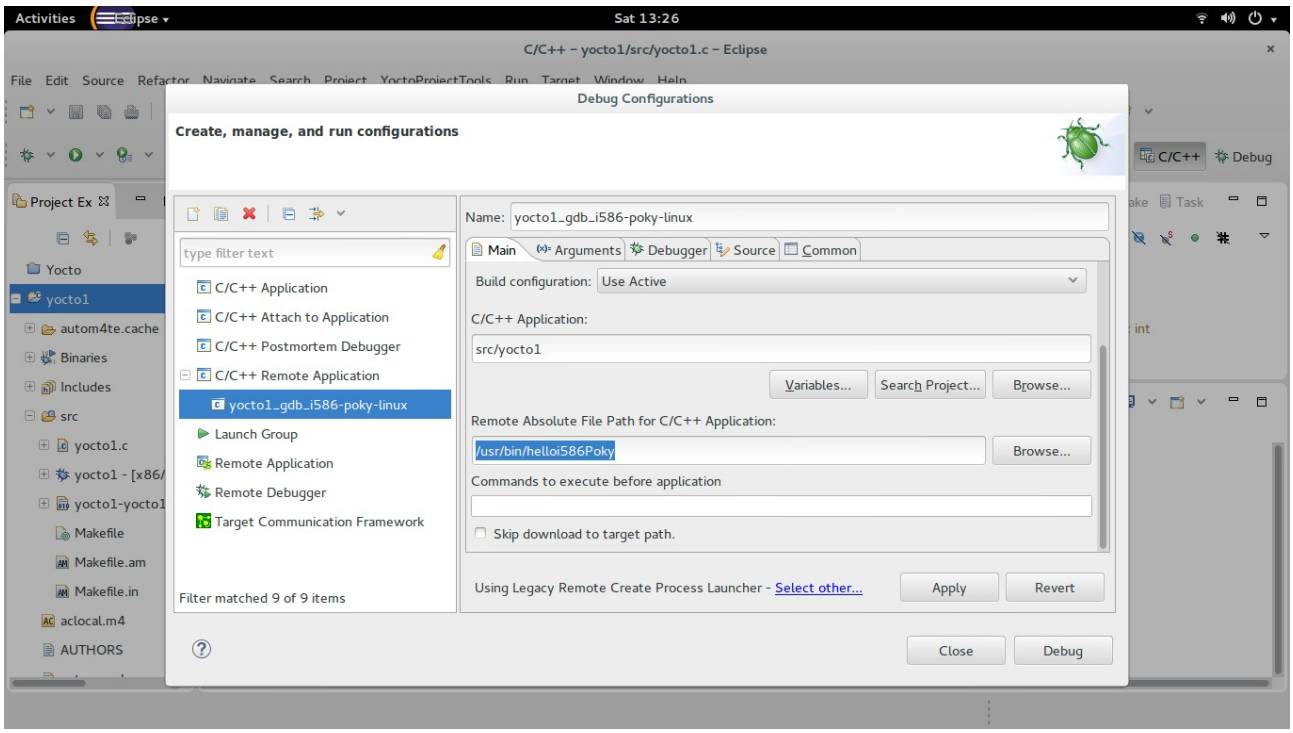
Use Yocto Project eclipse plug-in for cross debugging



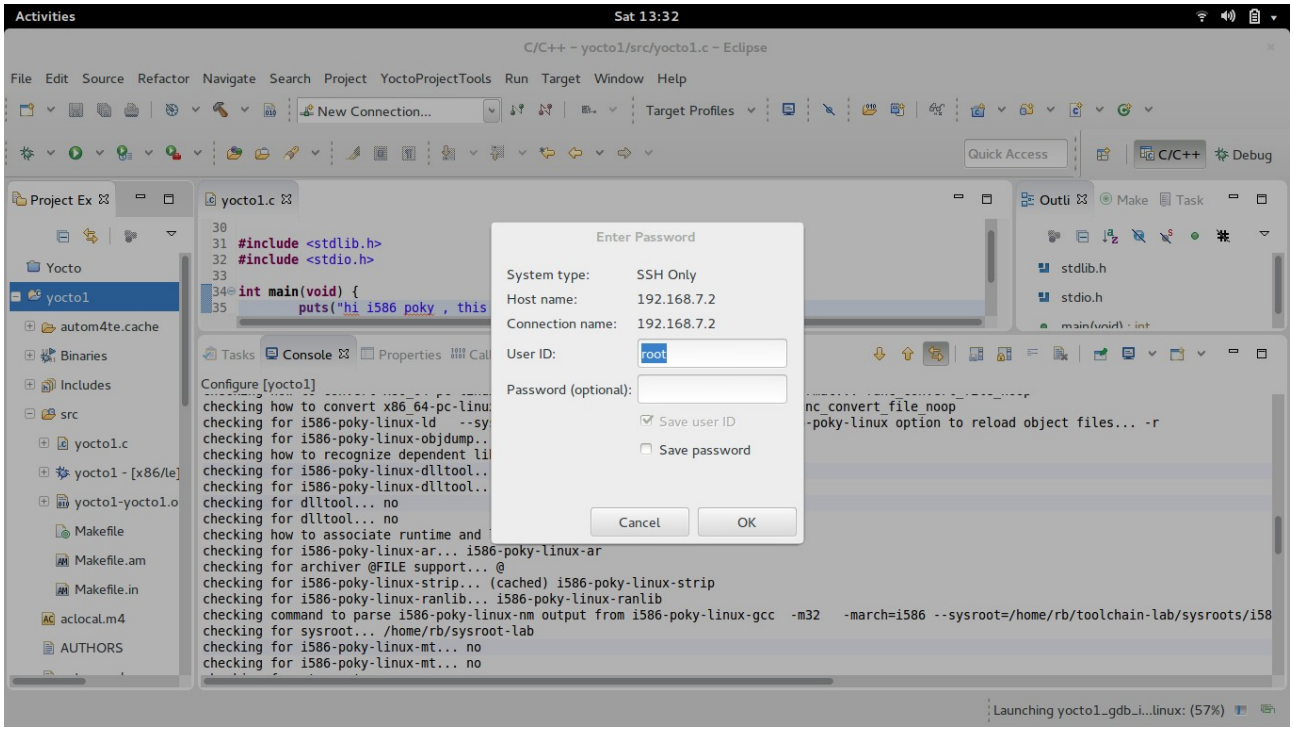
Since there's no helloi586poky binary under /usr/bin in the image, :



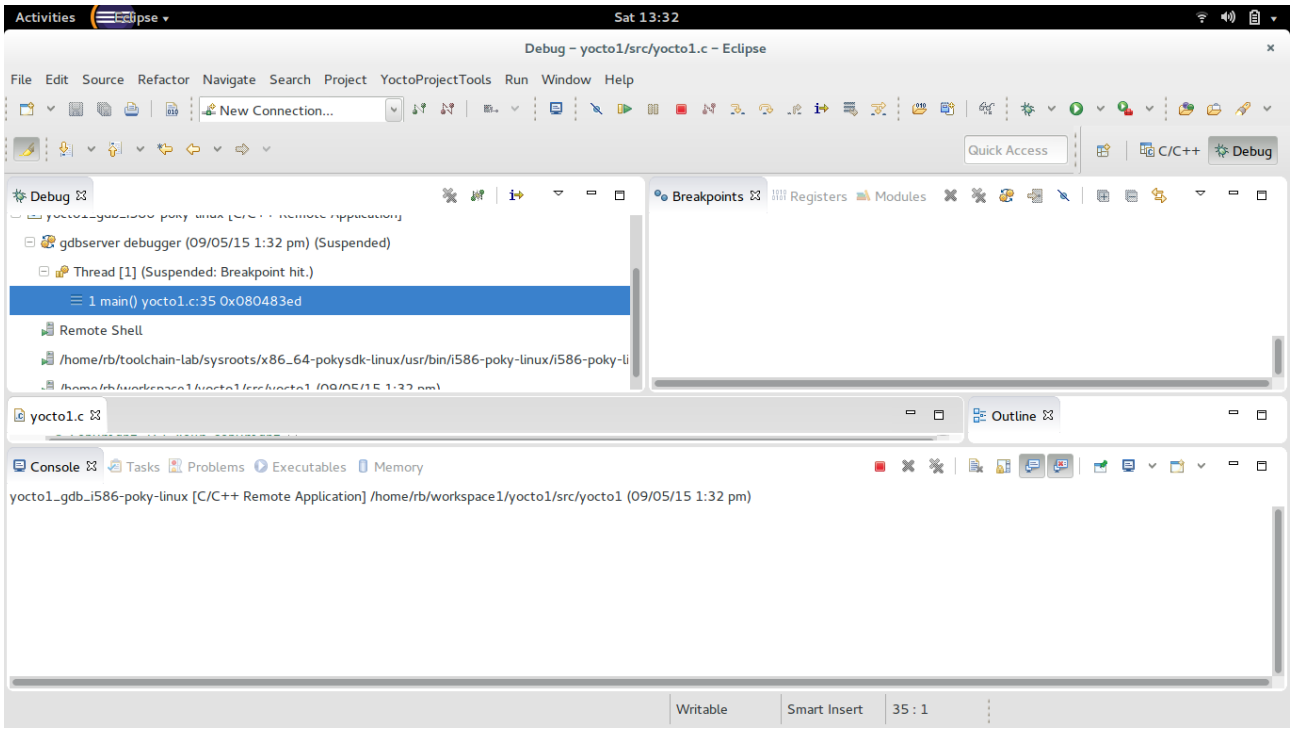
cross deploy and debug against qemu
configuring debug



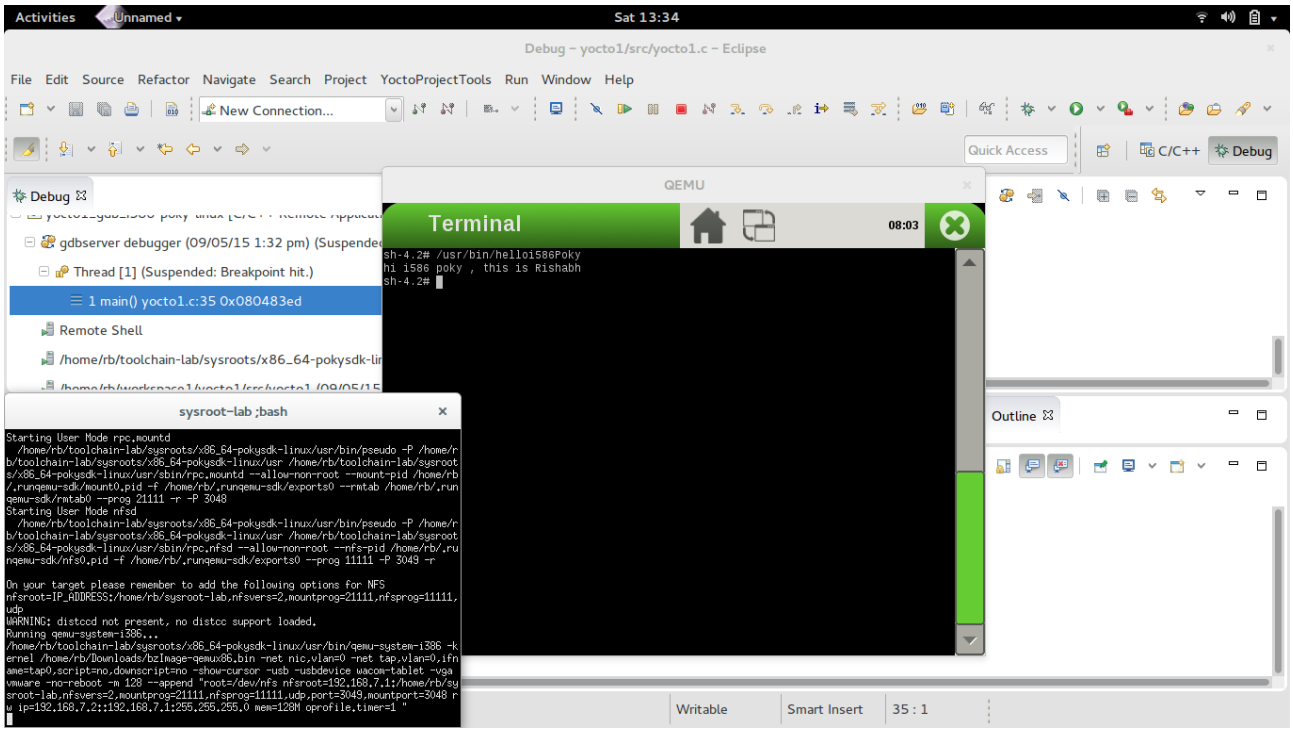
connecting to ssh:22



UPLOADING THE PROGRAM IN THE VIRUAL i586 processor



Getting the output in QEMU



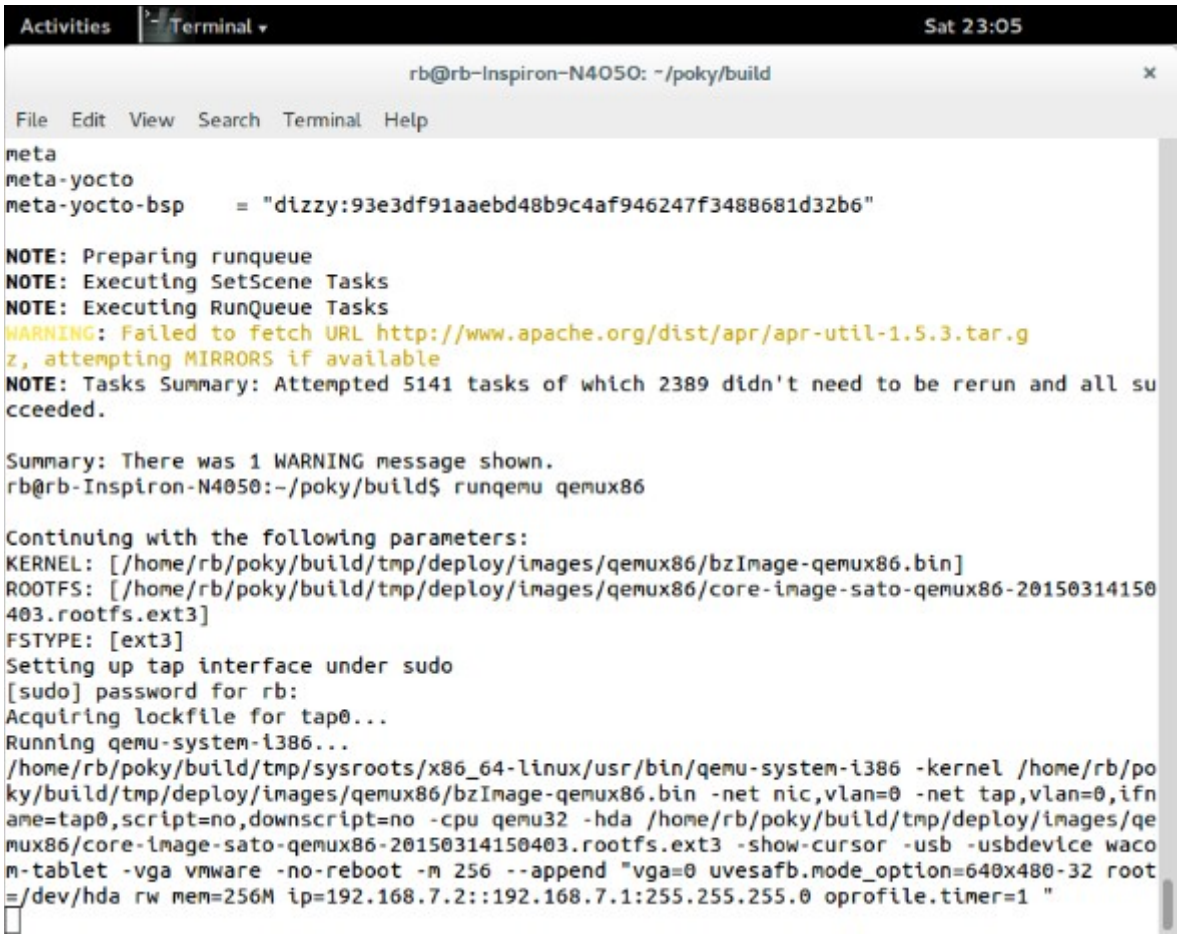
conclusion:

we exercised how to do remote cross debugging using Eclipse IDE.

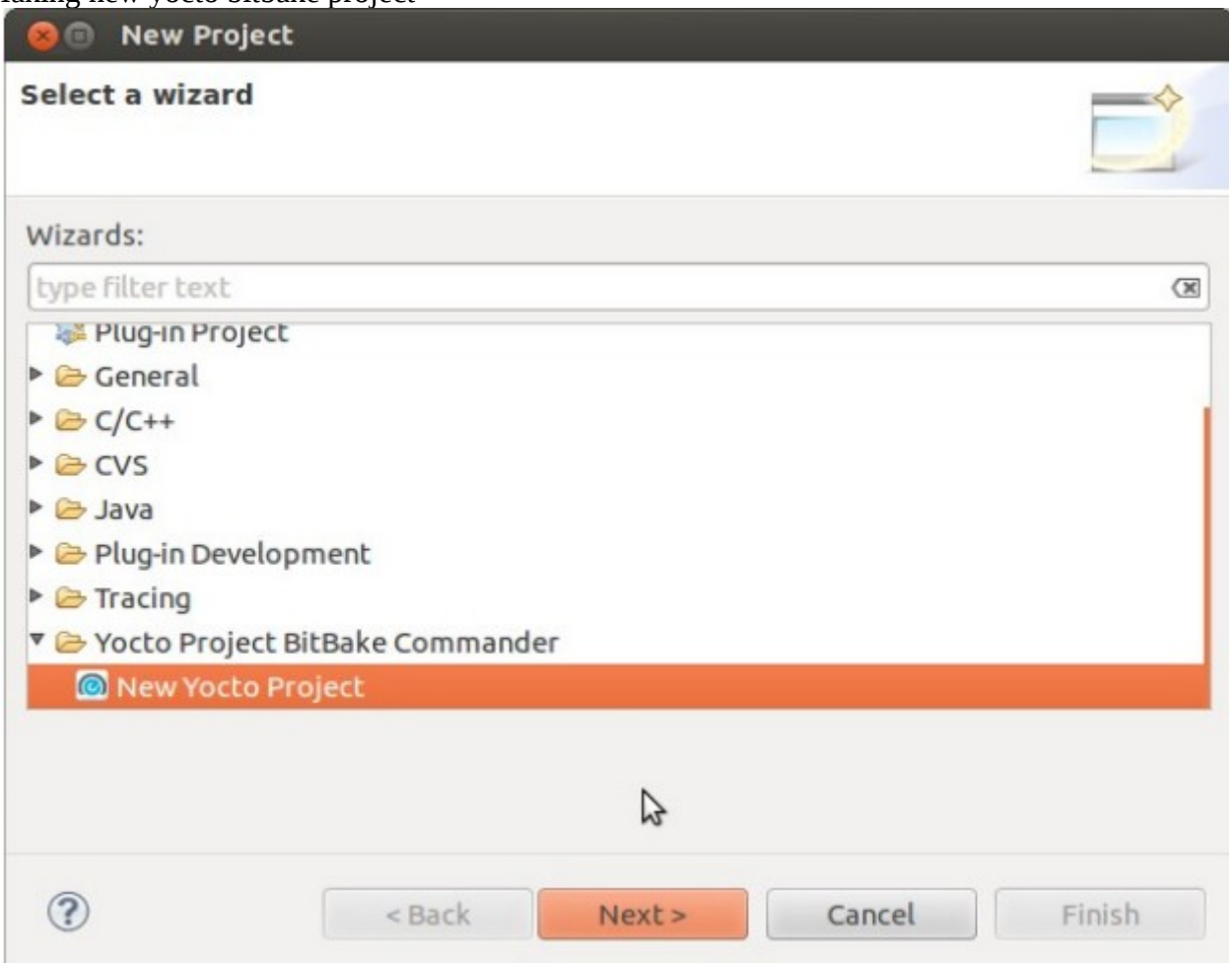
Yocto Bitbake Commander Project and HOB
i cancelled tje download since i already downloaded it once

```
rb@rb-Inspiron-N4050:~$ cd $HOME
rb@rb-Inspiron-N4050:~$ git clone git://git.yoctoproject.org/poky poky-lab
Cloning into 'poky-lab'...
remote: Counting objects: 269932, done.
remote: Compressing objects: 100% (66775/66775), done.
^L^Ziving objects:   5% (14612/269932), 5.18 MiB | 104.00 KiB/s
[1]+  Stopped                 git clone git://git.yoctoproject.org/poky poky-lab
rb@rb-Inspiron-N4050:~$ cd ~/poky
rb@rb-Inspiron-N4050:~/poky$ git checkout 1.4_M1
Branch 1.4_M1 set up to track remote branch 1.4_M1 from origin.
Switched to a new branch '1.4_M1'
rb@rb-Inspiron-N4050:~/poky$
```

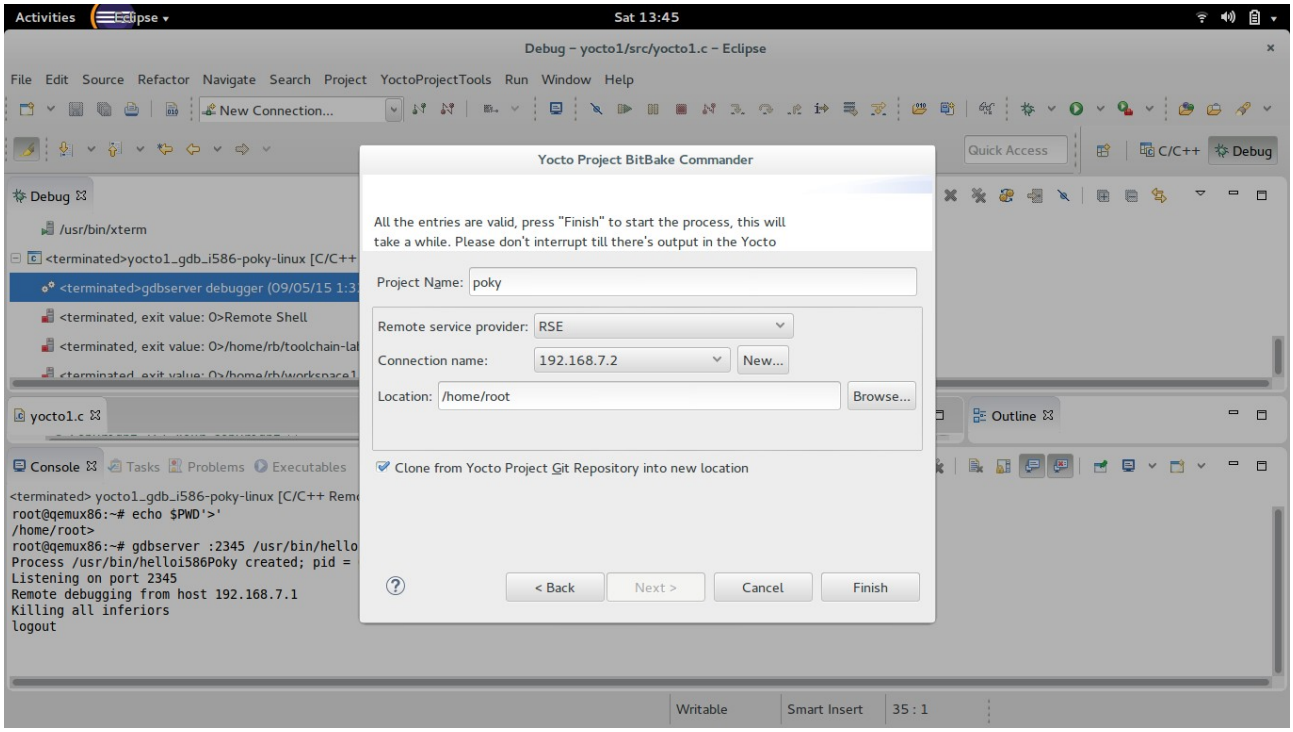
old screenshot i had of it

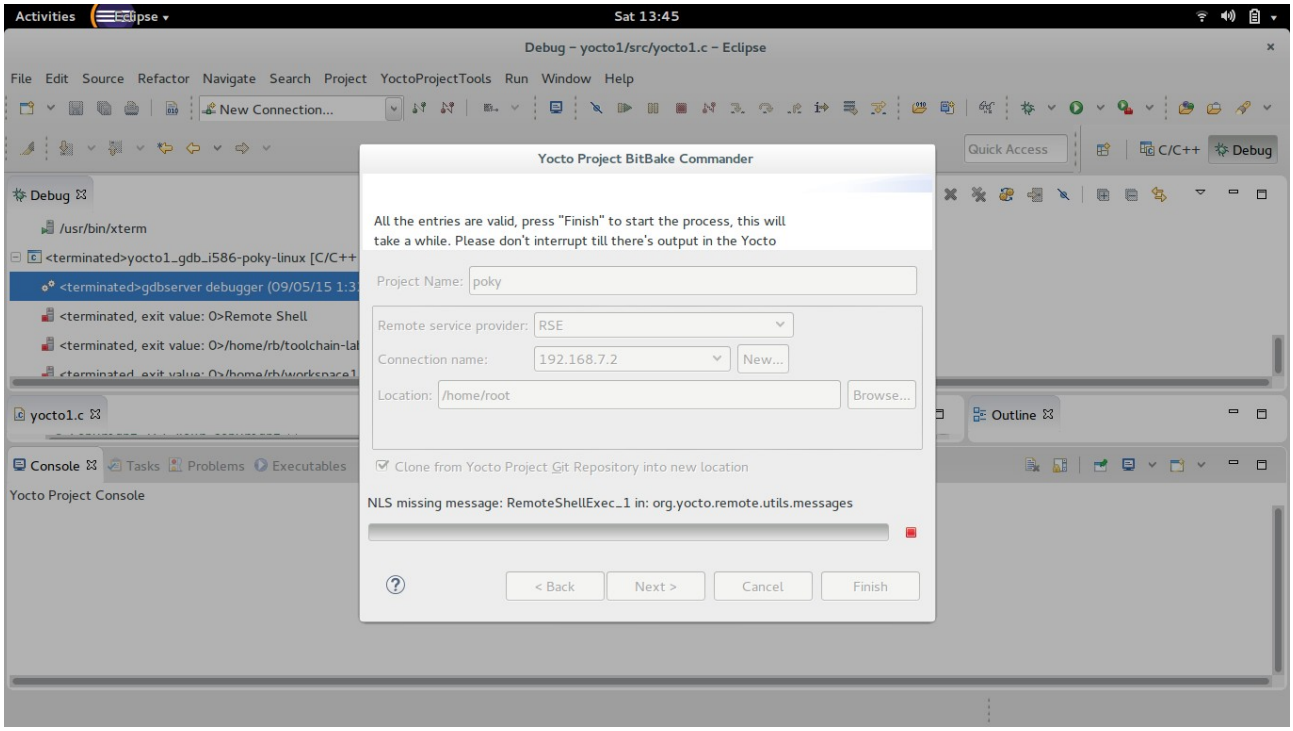


Making new yocto bitbake project

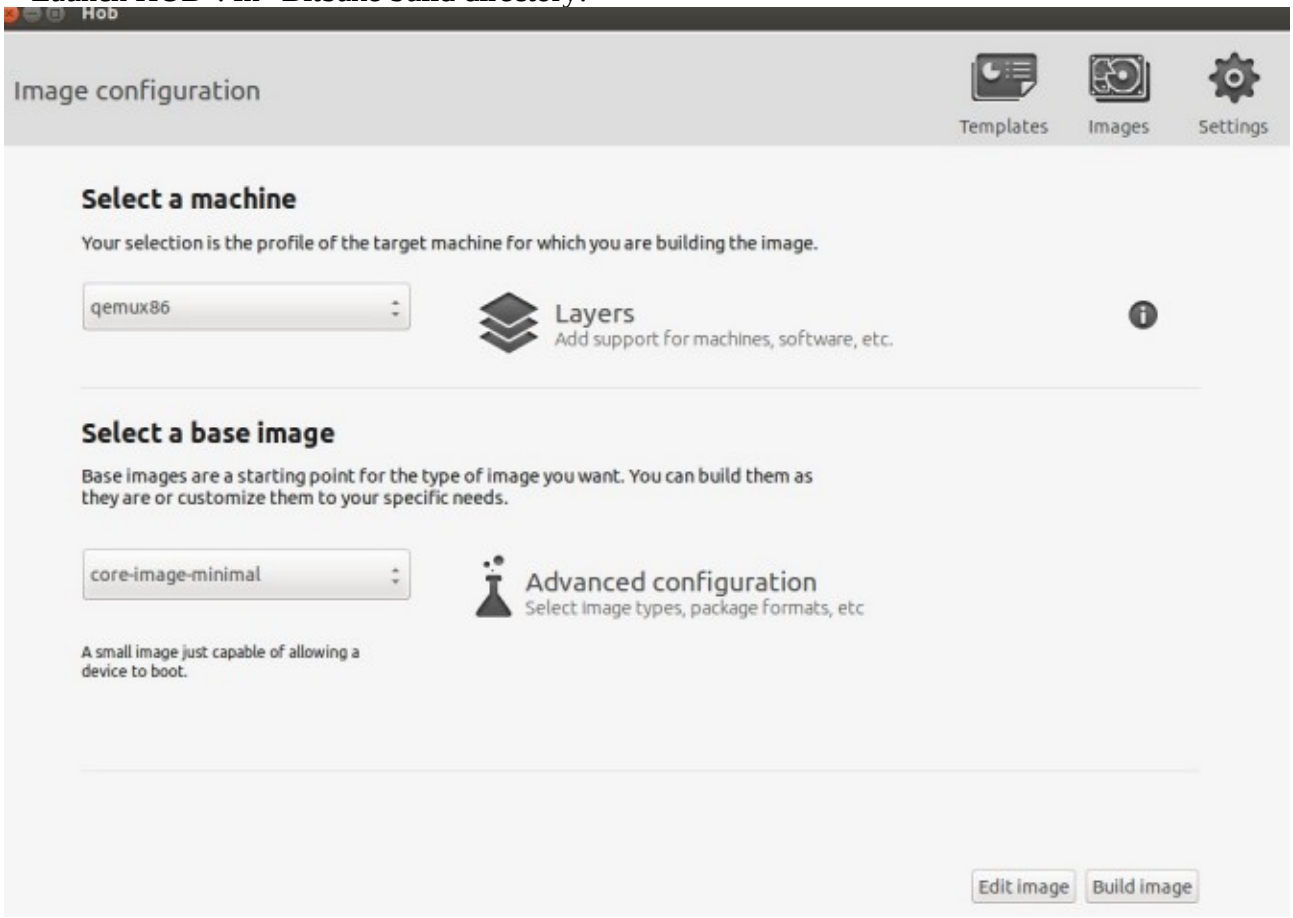


Configuring





Project
‣ Launch HOB". In "Bitbake build directory:"

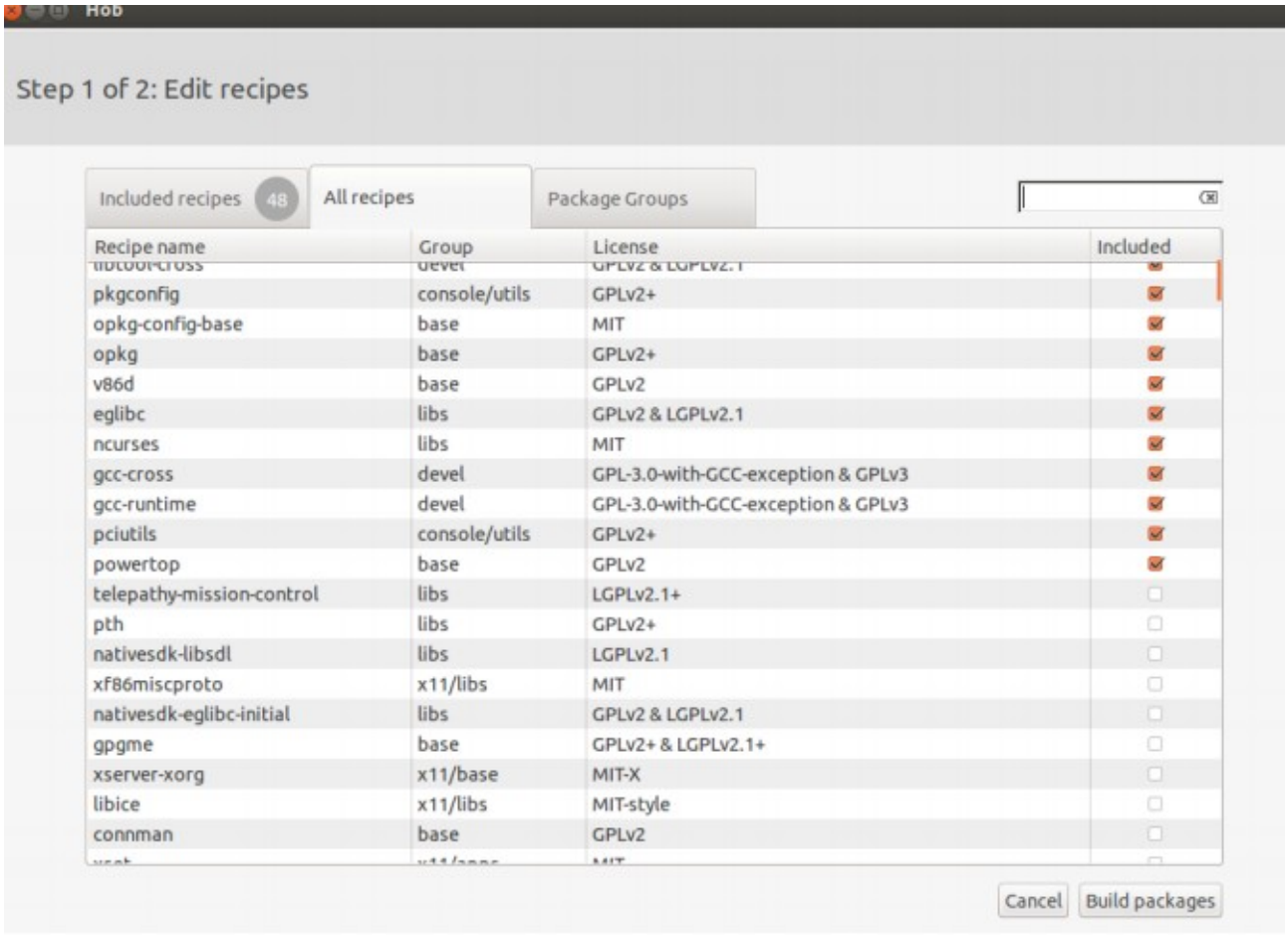


Conclusion

You have created a bitbake commander project in Eclipse IDE
and built a qemu image using hob against the bitbake commander project meta data. Logged into your built qemu image and ran couple commands.

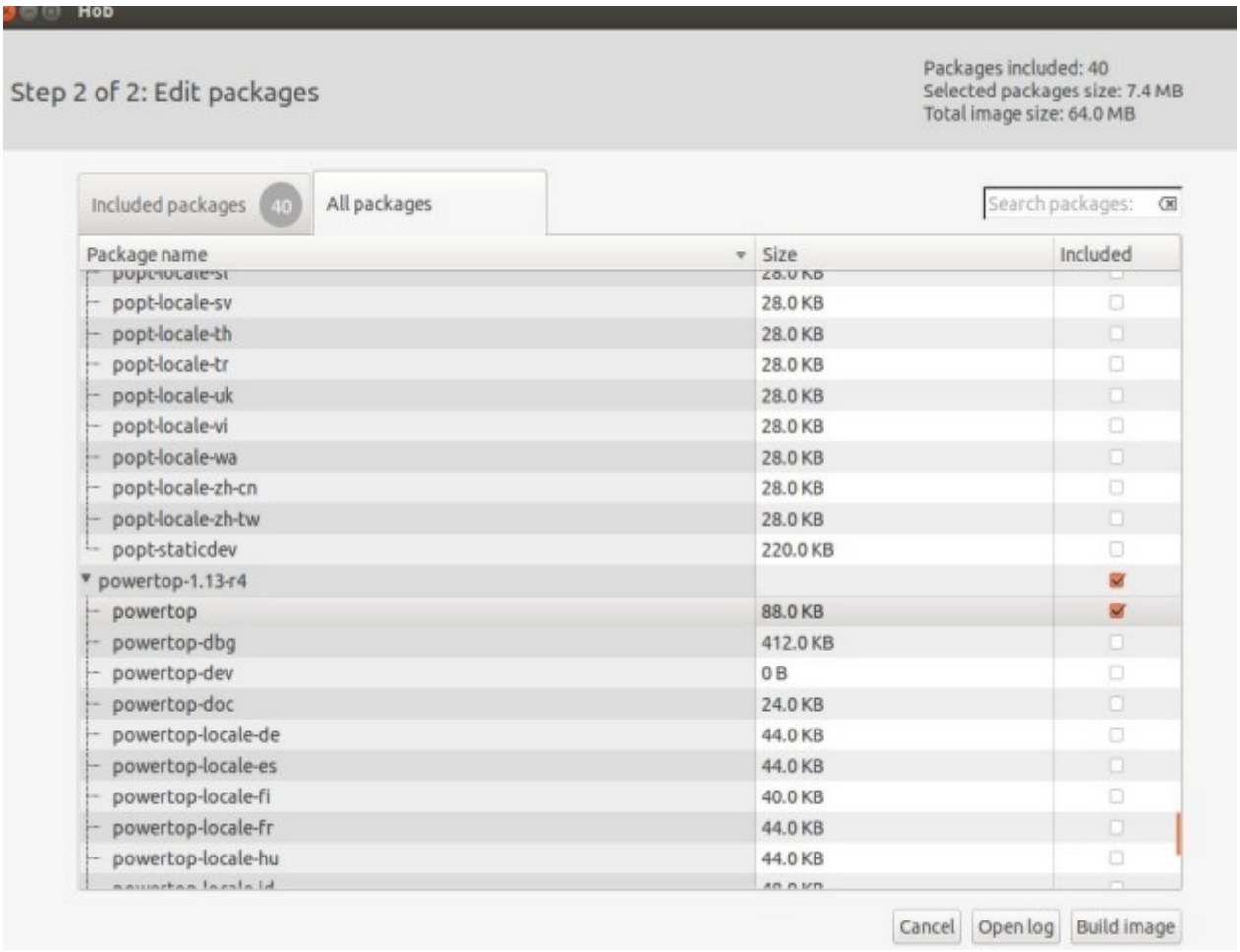
Customize image using hob

Switch back to hob. And click on “Build new image”, this will bring us back to the hob first window.
Click on “Edit Image” to bring up the recipe view. Then, click on “All recipes” tab. In top right hand “Search recipes:” box, type “powertop” and hit enter. This will jump to powertop recipe entry. You'll notice the “Included” box is not ticked. Click on that box to ensure it's ticked. Then click on “Build packages”.
Here are the screen shots



This will bring up the package build window, which you can see all the related packages for powertop will be built. Wait till the build is done ...it will take a really long time

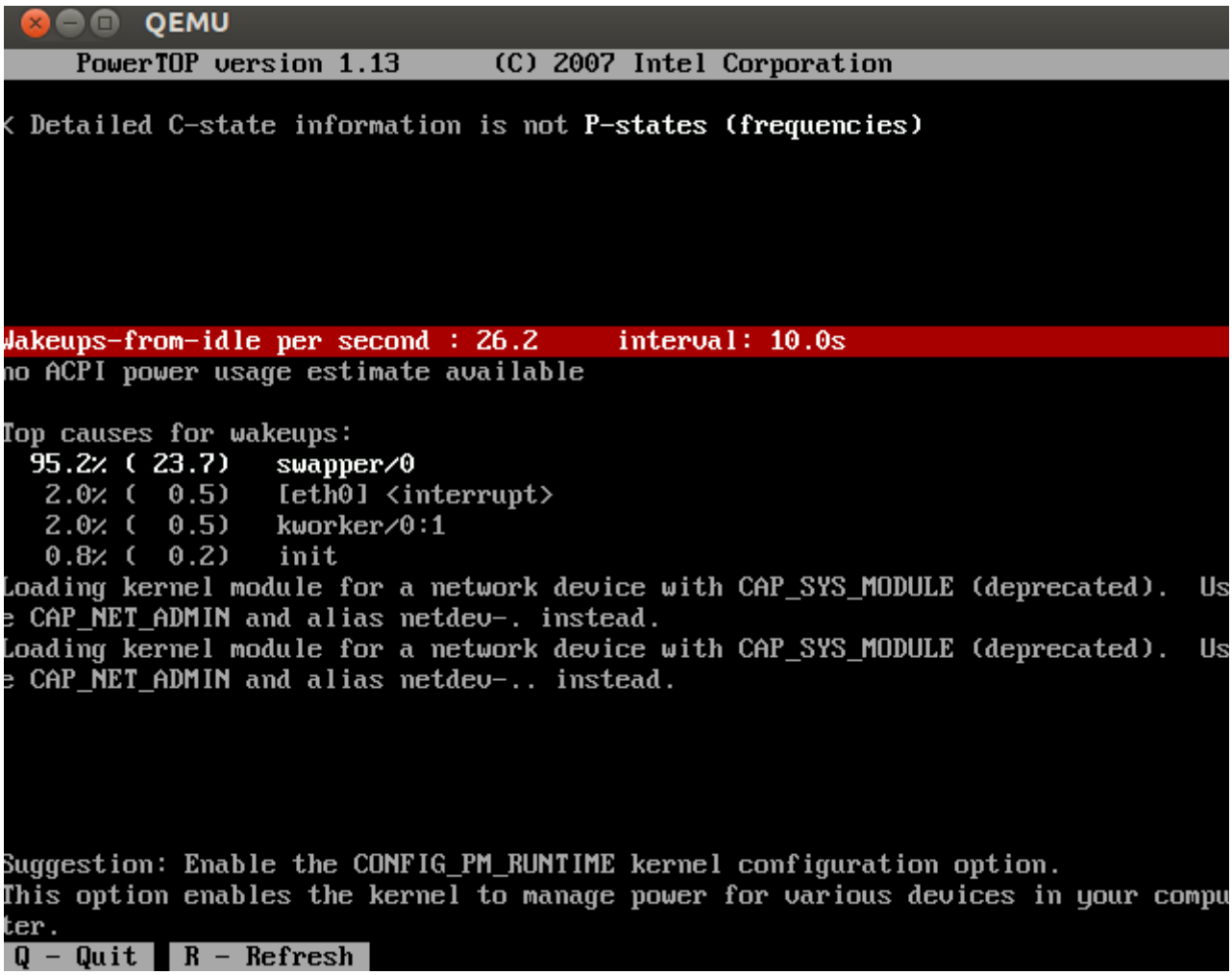
After the package build is done. The “Edit Packages” window will be showing. Here, click on “All packages” tab. In top right hand “Search packages:” box, type “powertop” and hit enter. This will bring us to the powertop packages list. There, only click on the included box for powertop.



Click on “Build image”, this will build a new customized image base on coreimage-minimal that include powertop. After the build is successfully done,

click on “Run image” .login to qemu as root password “”.

After login qemu as root, type
powertop
since the new image has powertop, Here is the screenshot

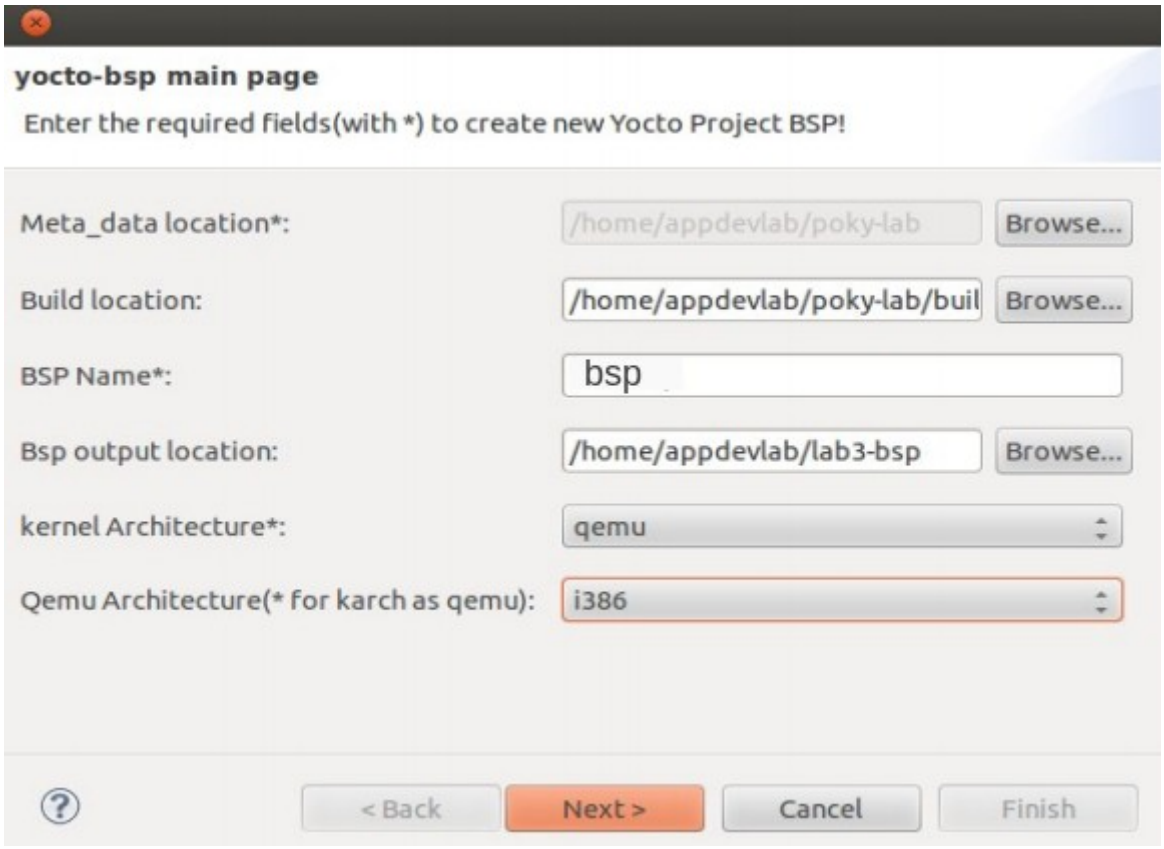


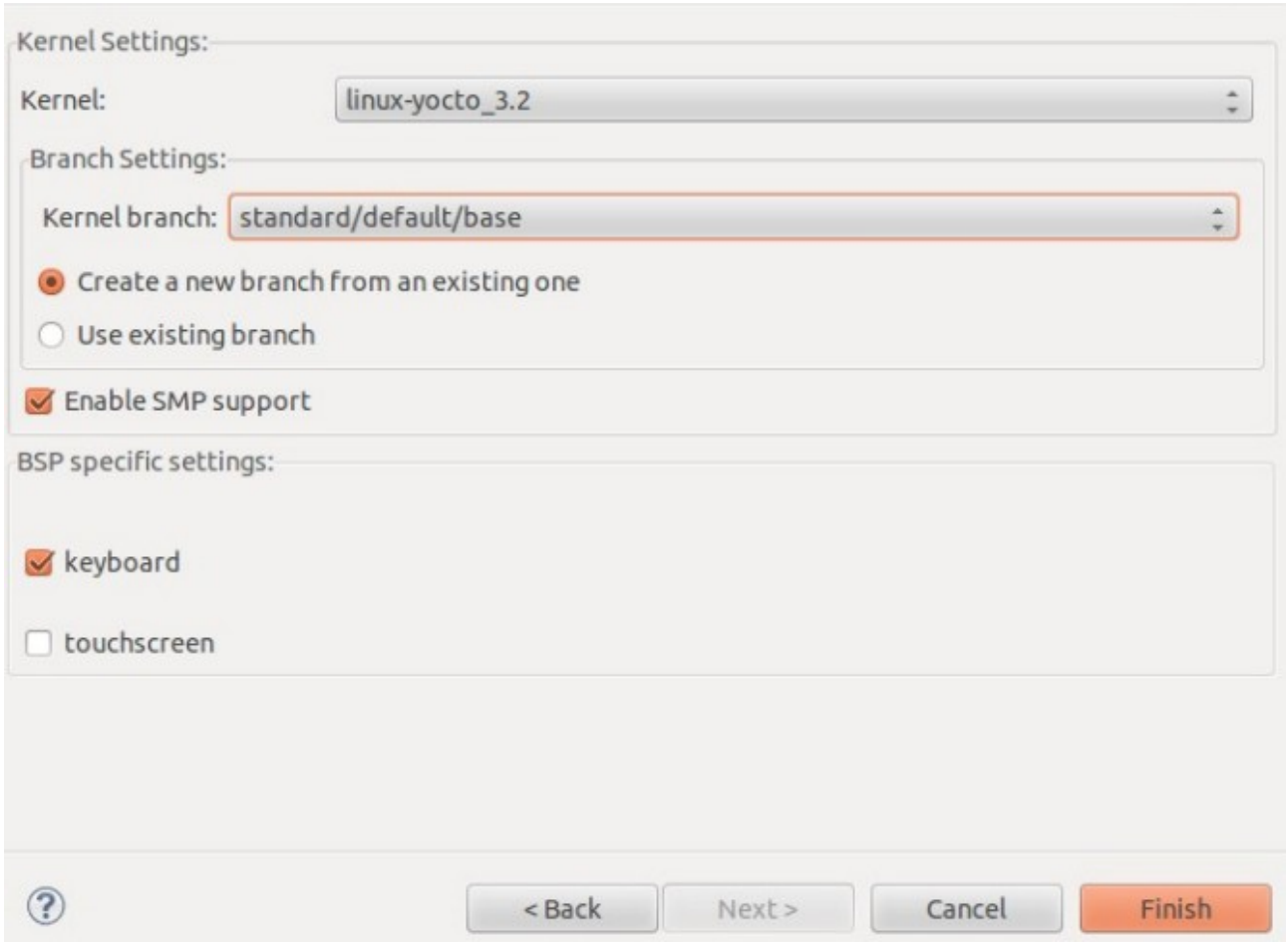
Conclusion

In this we have used hob and success did an image customization by adding powertop to your customized image that based on core-image-minimal. The steps involved in image customization is: first, you need to select powertop recipe to build the related packages; next, we need to select the wanted packages to be include in the image an build the image.

Create a BSP using yocto-bsp plug-in and hob

Create the bsp layer
Inside eclipse IDE, do the following :
YoctoProjectTools ▸ yocto-bsp to bring up yocto-bsp plug-in interface.
Fill as in screenshot
replacing appdevlab with your true userid

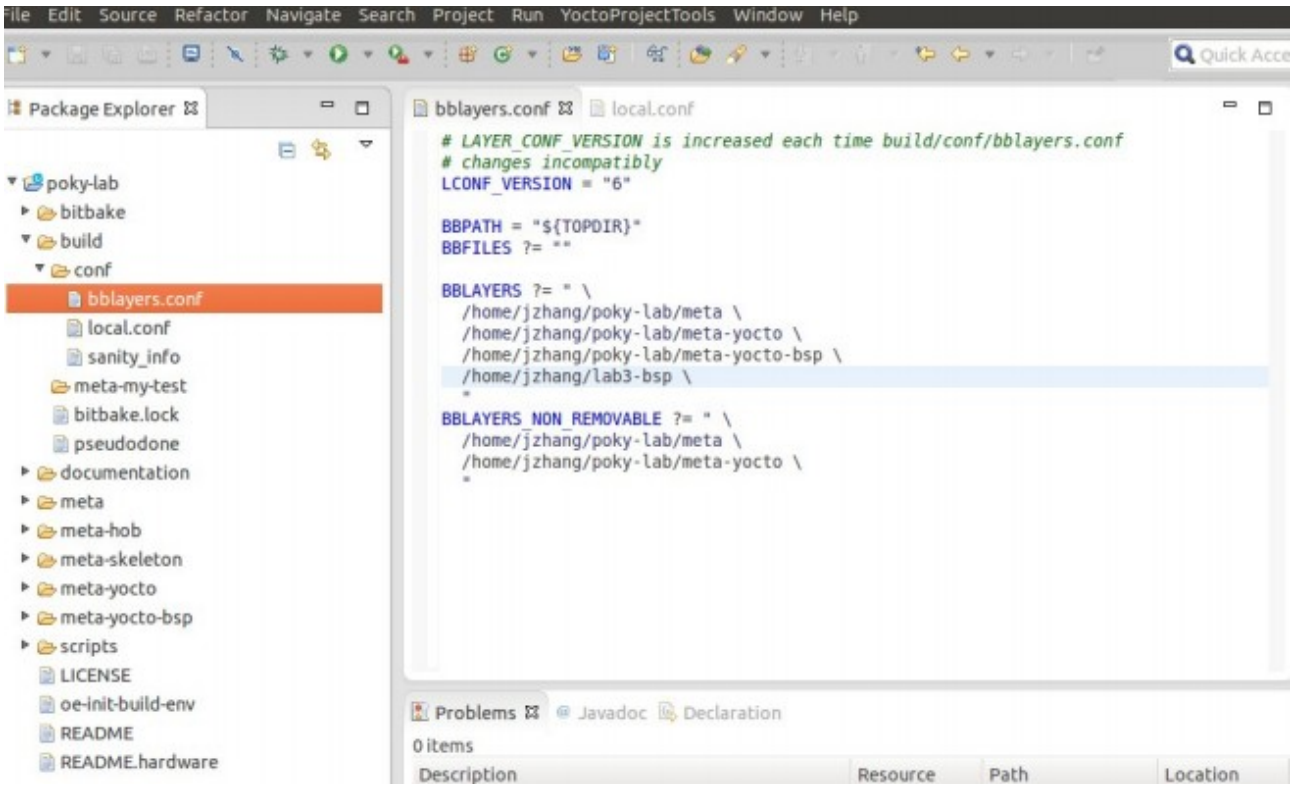


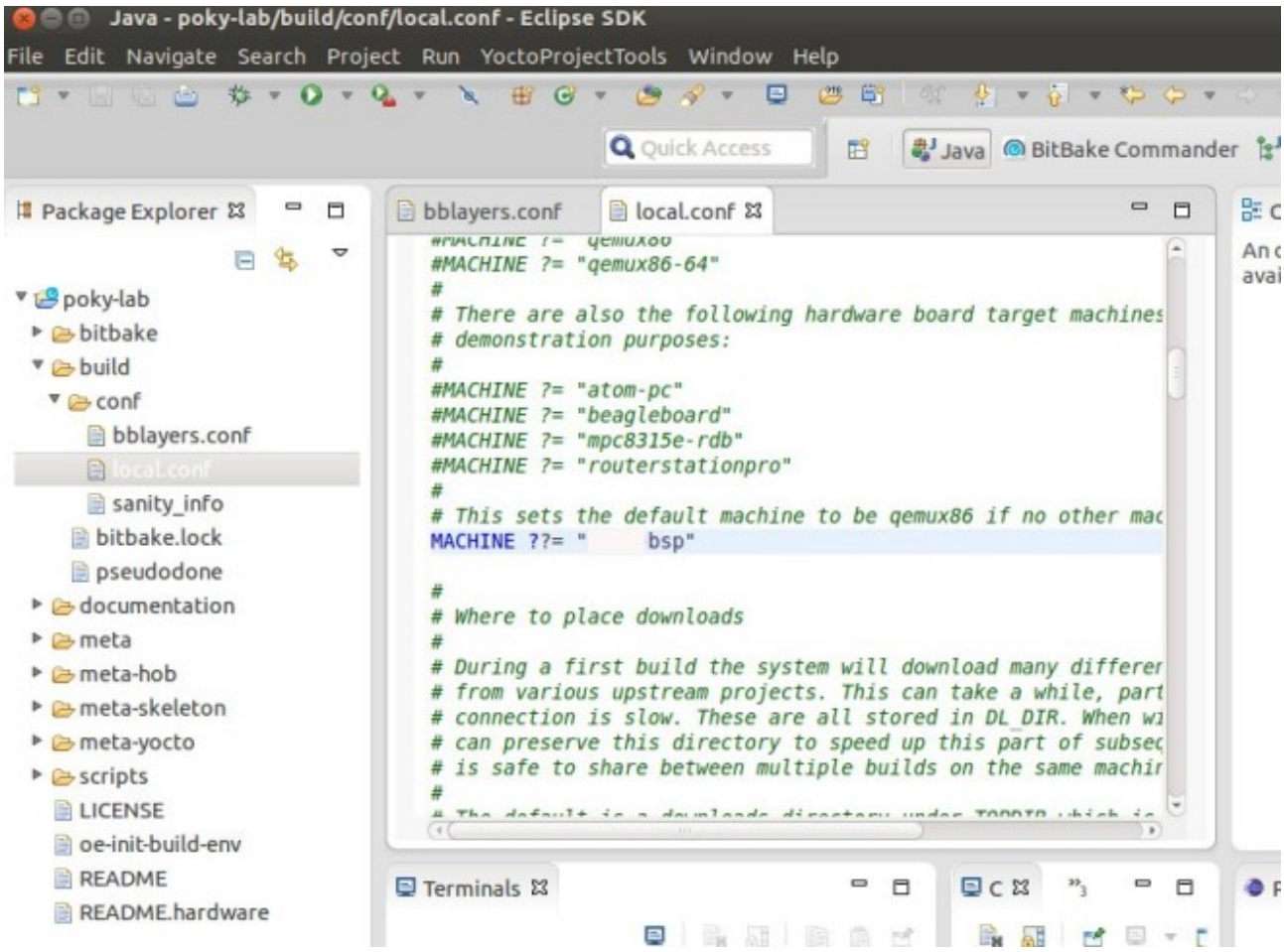


click finish

Create the bsp layer

Extend the poky-lab bitbake commander project in a tree view by clicking on the
Then further extend the build directory under poky-lab and conf directory under build.
Double click “bblayers.conf” file to bring its content to the bitbakecommander editor.
As shown in screenshots

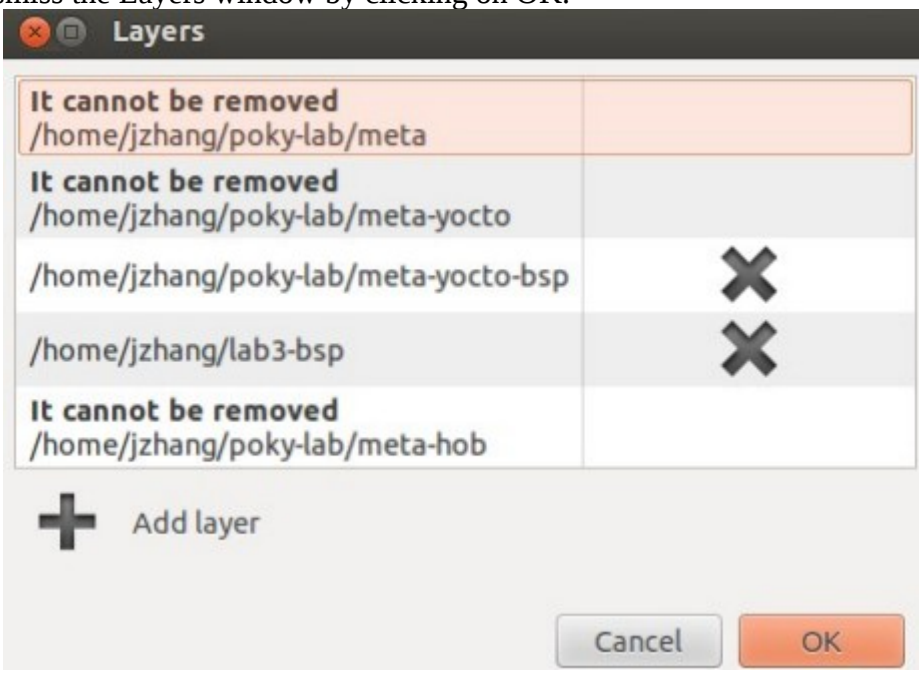

File Edit Source Refactor Navigate Search Project Run YoctoProjectTools Window Help
Package Explorer
poky-lab
bitbake
build
conf
bblayers.conf
local.conf
sanity_info
meta-my-test
bitbake.lock
pseudodone
documentation
meta
meta-hob
meta-skeleton
meta-yocto
meta-yocto-bsp
scripts
LICENSE
oe-init-build-env
README
README.hardware
bblayers.conf
local.conf
# LAYER_CONF_VERSION is increased each time build/conf/bblayers.conf
# changes incompatibly
LCONF_VERSION = "6"
BBPATH = "${TOPDIR}"
BBFILES ?= ""
BBLAYERS ?= " \
/home/jzhang/poky-lab/meta \
/home/jzhang/poky-lab/meta-yocto \
/home/jzhang/poky-lab/meta-yocto-bsp \
/home/jzhang/lab3-bsp \
"
BBLAYERS_NON_REMOVABLE ?= " \
/home/jzhang/poky-lab/meta \
/home/jzhang/poky-lab/meta-yocto \
"
Problems
Javadoc
Declaration
0 items
Description
Resource
Path
Location

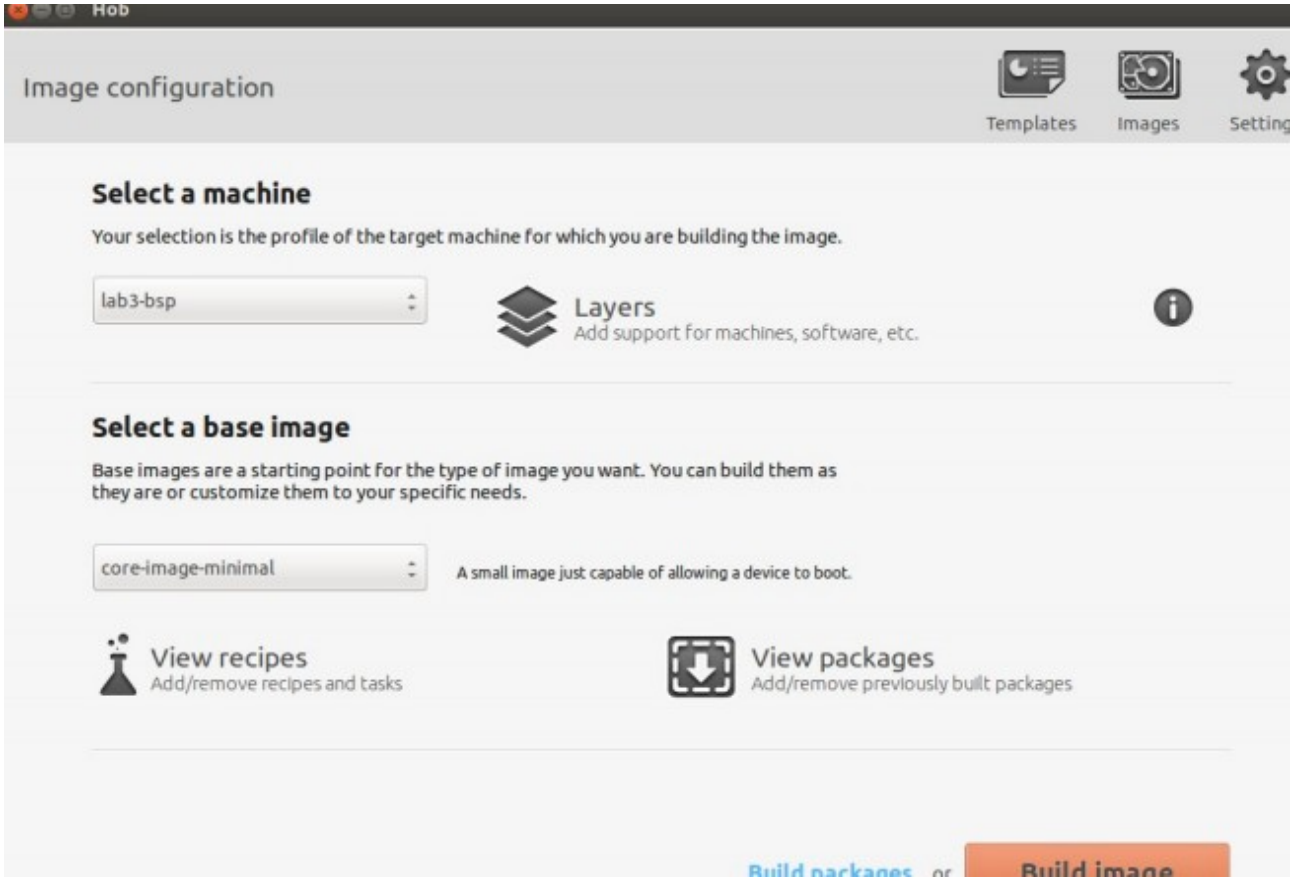

Java - poky-lab/build/conf/local.conf - Eclipse SDK
File Edit Navigate Search Project Run YoctoProjectTools Window Help
Quick Access
Java
BitBake Commander
Package Explorer
poky-lab
bitbake
build
conf
bblayers.conf
local.conf
sanity_info
bitbake.lock
pseudodone
documentation
meta
meta-hob
meta-skeleton
meta-yocto
scripts
LICENSE
oe-init-build-env
README
README.hardware
bblayers.conf
local.conf
#MACHINE ?= "qemux86-64"
#
There are also the following hardware board target machines
demonstration purposes:
#
#MACHINE ?= "atom-pc"
#MACHINE ?= "beagleboard"
#MACHINE ?= "mpc8315e-rdb"
#MACHINE ?= "routerstationpro"
#
This sets the default machine to be qemux86 if no other mac
MACHINE ??= "   bsp"
#
Where to place downloads
#
During a first build the system will download many differer
from various upstream projects. This can take a while, part
connection is slow. These are all stored in DL_DIR. When wi
can preserve this directory to speed up this part of subsec
is safe to share between multiple builds on the same machir
#
Terminals

Use hob build the bsp image

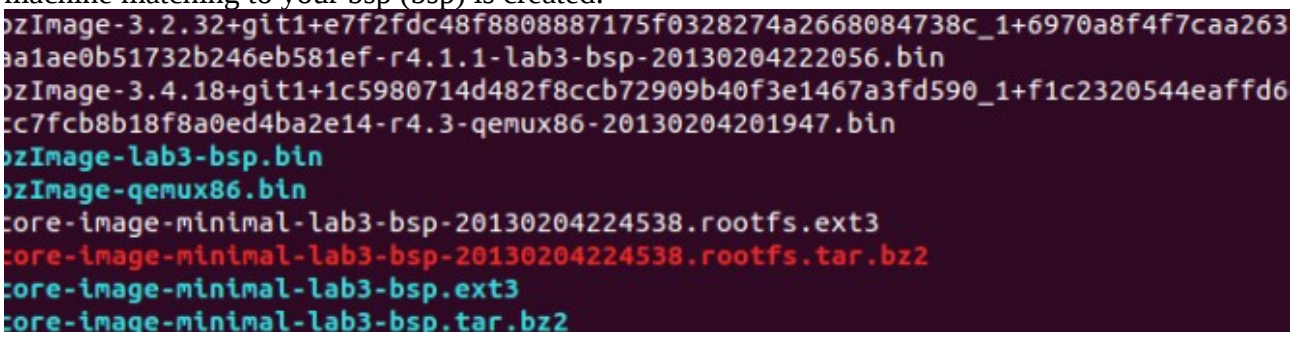
Highlight “poky-lab” project, then go to “ Project Launch HOB”. In “Bitbake build directory:” browse to /home/appdevlab/poky-lab/build. Then click “OK” to bring up hob.

In hob, for “Select a machine” drop down list, “bsp” is listed. Select it. Click on layer icon. You'll see “/home/appdevlab/bsp” is listed. All of these reflect the changes we just made to poky-lab meta-data. Dismiss the Layers window by clicking on OK.





Monitoring the build progress in the hob build window.
Once the build is successfully
exit hob.
Step 5: Bring up a terminal, and type following command:
cd ~/poky-lab
ls build/tmp/deploy/images
and you should see the new kernel file and rootfs images file for your new
machine matching to your bsp (bsp) is created:



Conclusion
Now we have used the yocto-bsp tool to generate a complete BSP layer. Modified meta data to include the layer and built the bsp image