Yocto for Dragoboard

Rakesh B

About yocto:

The Yocto Project is an open source collaboration project that provides templates, tools and methods to help you create custom Linux-based systems for embedded products regardless of the hardware architecture. It was founded in 2010 as a collaboration among many hardware manufacturers, open-source operating systems vendors, and electronics companies to bring some order to the chaos of embedded Linux development.

Why use the Yocto Project?
It's a complete embedded Linux development environment with tools, metadata, and documentation - everything you need. The free tools are easy to get started with, powerful to work with (including emulation environments, debuggers, an Application Toolkit Generator, etc.) and they allow projects to be carried forward over time without causing you to lose optimizations and investments made during the project's prototype phase. The Yocto Project fosters community adoption of this open source technology allowing its users to focus on their specific product features and development.

Platform Yocto supports

1. Raspberry pi
2. Beaglebone black
3. Digi Connect Core
4. Snickerdoodle(Zyng) etc. Yocto will supports for almost all hardwares
5. Other Supported platforms

Yocto Build for Dragonboard410c :

Note : Whatever is in yellow color type on terminal
The following packages are required for Ubuntu or Debian Linux distribution. The dependencies for a compatible environment include:
sudo apt-get install gawk wget git-core diffstat unzip texinfo gcc-multilib build-essential chrpath

1. Create directory structure to download source
mkdir -p ~/dragon/sources

2. cd into directory
cd ~/dragon/sources

3. Get the required layers

We will need bare minimum above 3 clones for building Linux for Dragonboard410c
– poky
– meta-openembedded
– meta-Qualcomm

git clone -b rocko git://git.yoctoproject.org/poky.git

git clone -b rocko git://git.openembedded.org/meta-openembedded

git clone -b rocko https://github.com/ndechesne/meta-qcom.git

4.Preparing the build configuration

Following command will create a build environment using the setup script that comes with poky, and will create a “dragon-build” directory in “~/dragon/” directory
cd ~/dragon/
source sources/poky/oe-init-build-env dragon-build
Now you will be in “~/dragon/dragon-build” directory

5. Add some configuration lines to Config files

Add following lines to change the build process for us
a) Changes in local.conf
echo 'MACHINE = "dragonboard-410c"' >> conf/local.conf
echo ‘PREFERRED_VERSION_linux-linaro-qcom = "4.9"’>>conf/local.conf
echo 'DISTRO_FEATURES_remove = "x11 wayland"' >> conf/local.conf
echo 'DISTRO_FEATURES_append = " systemd"' >> conf/local.conf
echo 'VIRTUAL-RUNTIME_init_manager = "systemd"' >> conf/local.conf
Default to setting automatically based on cpu count
echo 'PARALLEL_MAKE ?= " -j ${@oe.utils.cpu_count()}"' >> conf/local.conf
option determines how many tasks bitbake should run in parallel
echo 'BB_NUMBER_THREADS ?= " ${@oe.utils.cpu_count()}"' >> conf/local.conf

b) Changes the bblayers.conf file to following(this config file is available in Dragon-build/conf)

Note : bblayers.conf is a picking up the required layers from poky, meta-openembedded and meta-qualcomm as per you embedded application required
This file is available in dragon/dragon-build/conf/bblayer.conf
write the below contains in bblayers.conf if you want additional layers you can add.

```
# LAYER_CONF_VERSION is increased each time build/conf/bblayers.conf
# changes incompatibly
POKY_BBLAYERS_CONF_VERSION = "2"
BBPATH = "${TOPDIR}"
BBFILES ?= ""
BSPDIR := "/home/bathirak/dragon/"
BBLAYERS ?= " \
${BSPDIR}/sources/poky/meta \
${BSPDIR}/sources/poky/meta-poky \
${BSPDIR}/sources/poky/meta-yocto-bsp \
${BSPDIR}/sources/meta-openembedded/meta-oe \
${BSPDIR}/sources/meta-openembedded/meta-multimedia \
${BSPDIR}/sources/meta-qcom\
"
BBLAYERS_NON_REMOVABLE ?= " \
${BSPDIR}/sources/poky/meta \
${BSPDIR}/sources/poky/meta-poky \
```

6. Menuconfig changes for Kernel/Busybox

for menu config

```
bitbake linux-linaro-qcomlt -c menuconfig
```

for rootfs

```
bitbake -c menuconfig busybox
```

7. Only if you want to build specific receipe

```
bitbake <receipe name>
```

8. complete os Finally Build Image

bitbake core-image-base

you will get images in below path

/home/bathirak/dragom/dragon-build/tmp/deploy/images/dragonboard-410c

9. Set dragon board into fastboot method :

```
$ sudo fastboot flash boot boot-dragonboard-410c.img
$ sudo fastboot flash rootfs  core-image-base-dragonboard-410c.ext4
```