

YOCTO BSP LAYER

What is BSP Layer?	2
BSP Layer	2
How to find out what all hardware devices are supported	2
meta-ti	2
Hardware Configuration Supported	2
meta-ti vs meta-yocto-bsp	2
meta-yocto-bsp:	2
meta-ti	2
Adding Layers	2
Manual:	3
Automatic:	3
Steps for building	3
Flashing the image on the SD Card using wic	3
Challenge	3

What is BSP Layer?

- A collection of information(metadata) that defines how to support
 - a particular hardware device,
 - set of devices, or
 - hardware platform

BSP Layer

- Naming Convention: meta-<bsp_name>

How to find out what all hardware devices are supported

- conf/machine/*.conf list all the hardware devices supported by the BSP layer

meta-ti

- BSP Layer for Texas Instrument Hardware
 - \$ cd sources
 - \$ git clone git://git.yoctoproject.org/meta-ti

Hardware Configuration Supported

- Beagle Bone Black
- Beagle Board
- Panda Board
- OMAP boards

meta-ti vs meta-yocto-bsp

meta-yocto-bsp:

- provides "reference" BSPs for each of the supported architectures
- One for ARM (BeagleBone Black), one for MIPS, PPC and x86.
- it is based on the mainline kernel/bootloader
- does not support any advanced features or anything not in the upstream mainline kernel
- e.g. no capes, no power management, no hardware acceleration, no 3D, no PRU, etc.
- The purpose of this BSP is to have some basic out-of-box experience for the select hardware platforms within Poky to evaluate the Yocto Project and OpenEmbedded framework, but not the specific hardware platforms

meta-ti

- official Texas Instruments BSP that provides the latest WIP "staging" kernel and bootloader
- most of the advanced features and peripheral support for the wider range of latest TI platforms

Adding Layers

Two ways:

Manual:

- edit bblayers.conf file and add the new layer to BBLAYERS

Automatic:

- `$ bitbake-layers add-layer <path-to-new-layer>`
- `$ bitbake-layers add-layer ~/Yocto_Training/source/meta-ti/`

Steps for building

- Step1 : Source the environment script
 - `$ source poky/oe-init-build-env`
 - Add the meta-ti layer
 - `$ bitbake-layers add-layer ~/Yocto_Training/source/meta-ti/`
- Step2 : Open local.conf and set Machine to beaglebone
 - `MACHINE='beaglebone'`
- Step3 : Also add INHERIT += "rm_work" to save disk space
- Step4 : Generate an minimal image
 - `$ bitbake core-image-minimal`
- Step5 : Once the build finished, you will find the output images under
 - `$BUILDDIR/tmp/deploy/images/beaglebone`

Flashing the image on the SD Card using wic

- wic images are SD Card images and can be directly written into sd-card
- core-image-minimal-beaglebone.wic.xz is compressed wic image.
- It can be uncompressed using the unxz utility
 - `$ unxz core-image-minimal-beaglebone.wic.xz`
 - `$ ls -lh core-image-minimal-beaglebone.wic`
- Flash it to the sd card
 - `$ lsblk`
 - `$ sudo dd if=core-image-minimal-beaglebone.wic of=/dev/sdb bs=4096 status=progress && sync`

Challenge

- Generate an image for beaglebone using meta-ti, but remove meta-yocto-bsp layer