Yocto

03 November 2021 18:43

https://www.yoctoproject.org/docs/2.0/yocto-project-qs/yocto-project-qs.html#yp-resources
https://george-calin.medium.com/how-to-prepare-a-helloworld-c-recipe-with-yocto-project-1f74c296a777
https://nagarro.sharepoint.com/sites/EmbeddedTraining/SitePages/Yocto001FirstRecipe.aspx?ct=1636950059454&or=Teams-HL
https://www.yoctoproject.org/docs/2.0/yocto-project-qs/yocto-project-qs.html#yp-resources

Video Tutorials :
What is poky
"Introduction to the Yocto Project and Bitbake" by Behan Webster - Part 1
"Introduction to the Yocto Project and Bitbake" by Behan Webster - Part 2

```
sudo apt-get install gawk wget git-core diffstat unzip texinfo gcc-multilib \
    build-essential chrpath socat libsdl1.2-dev xterm
git clone git://git.yoctoproject.org/poky
Cd poky
Git checkout dunfell
source oe-init-build-env
bitbake core-image-minimal
runqemu qemux86   /    runqemu qemux86-64 nographic

 runqemu qemux86-rj nographic

bitbake core-image-sato
```

http://book.yoctoprojectbook.com/page/code-chapter-8

Installing tiff library on ubuntu:
sudo apt-get install libtiff-dev

Yocto Project Overview :

Collection of tools and methods enabling
◆ Rapid evaluation of embedded Linux on many popular off-the-shelf boards
◆ Easy customization of distribution characteristics

YP builds packages - then uses these packages to build bootable images

Intro to OpenEmbedded :

➢ The OpenEmbedded Project co-maintains OE-core build system:
◆ bitbake build tool and scripts
◆ Metadata and configuration ➢ Provides a central point for new metadata

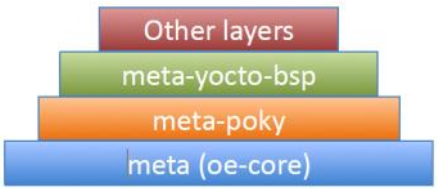
What is Bitbake?

Bitbake
◆ Powerful and flexible build engine (Python)
◆ Reads metadata
◆ Determines dependencies
◆ Schedules tasks

Metadata – a structured collection of "recipes" which tell BitBake what to build, organized in layers.

What is Poky?

- Poky is a reference distribution
- Poky has its own git repo
  - git clone git://git.yoctoproject.org/poky
- Primary Poky layers
  - oe-core (poky/meta)
  - meta-poky (poky/meta-poky)
  - meta-yocto-bsp
- Poky is the starting point for building things with the Yocto Project



Poky in Detail:

➢ Contains core components
- ◆ Bitbake tool: A python-based build engine
- ◆ Build scripts (infrastructure)
- ◆ Foundation package recipes (oe-core)
- ◆ meta-poky (Contains distribution policy)
- ◆ Reference BSPs
- ◆ Yocto Project documentation

Tiff Image :-

Installing of tiff library :-
sudo apt install libtiff-dev

Command to run to generate:
g++ -Wall tiff_image.cpp -o tiff_image -ltiff
./tiff_image

Create a **customized image**, as the output you should able to build the image by bitbaking this custom image recipe.

Ex - bitbake yourname-core-image

Try to add packages to the customized image

Ex - the package group we created, ssh, opencv ... etc.

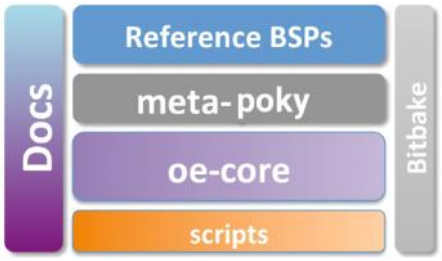
Go through bellow documentation and get a proper idea about the image and implement. Find more documentations from third party as well. Share them here if you found any useful.

https://docs.yoctoproject.org/ref-manual/images.html#
https://docs.yoctoproject.org/dev-manual/common-tasks.html#customizing-images

https://developer.toradex.com/knowledge-base/custom-meta-layers-recipes-images-in-yocto-project#Create_an_Image
https://hub.mender.io/t/how-to-create-custom-images-using-yocto-project/902
https://www.digikey.bg/en/maker/projects/intro-to-embedded-linux-part-4-create-custom-layer-and-image-in-yocto/aac0ab17e0c64ae482675abea00b328d

https://tutorialadda.com/yocto/create-your-own-linux-image-for-the-raspberry-pi-board-using-yocto-project#Raspberrypi_layer
https://jumpnowtek.com/rpi/Raspberry-Pi-Systems-with-Yocto.html

git clone git://git.yoctoproject.org/meta-raspberrypi -b dunfell



In Summary :

➢ Yocto Project is a large collaboration project
➢ OpenEmbedded is providing most metadata
➢ Bitbake is the build tool
➢ Poky is the Yocto Project's reference distribution
➢ Poky contains a version of bitbake and oe-core from which you can start your project

Metadata and bitbake :

➢ Most common form of metadata: The Recipe
➢ A Recipe provides a “list of ingredients” and “cooking instructions”
➢ Defines settings and a set of tasks used by bitbake to build binary packages

What is Metadata?

➢ Metadata exists in four general categories:
➢ Recipes (*.bb)
    ◆ Usually describe build instructions for a single package
➢ PackageGroups (special *.bb)
    ◆ Often used to group packages together for a FS image
➢ Classes (*.bbclass)
    ◆ Inheritance mechanism for common functionality
➢ Configuration (*.conf)
    ◆ Drives the overall behavior of the build process

Other Metadata:

➢ Append files (*.bbappend)
    ◆ Define additional metadata for a similarly named .bb file
    ◆ Can add or override previously set values
➢ Include files (*.inc)
    ◆ Files which are used with the include directive
    ◆ Also can be included with require (mandatory include)
    ◆ Include files are typical found via the BBPATH variable

What is a Recipe?

➢ A recipe is a set of instructions for building packages,
including:
    ◆ Where to obtain the upstream sources and which patches to apply (this is called “fetching”)
        o SRC_URI
    ◆ Dependencies (on libraries or other recipes)
        o DEPENDS, RDEPENDS
    ◆ Configuration/compilation options
        o EXTRA_OECONF, EXTRA_OEMAKE
    ◆ Define which files go into what output packages
        o FILES_*