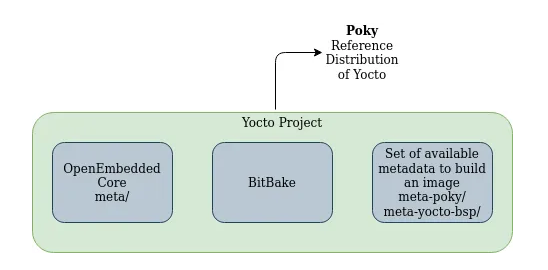
**Yocto Training Important Key Learnings**  **Documentation**

**What is Yocto** :- Yocto is an open-source collaboration project that provides tools, templates, and methods to help developers create custom Linux-based systems for embedded devices. The yocto project can generate images for many kinds of divces such as full system emulation for x86, x86-64, ARM, MIPS and PPCS-based architecture as well as specific hardware such as intel desktop board DH55TC.

**What is Poky :**- Poky includes a set of recipes, configuration files, and build tools that demonstrate how to create a minimal embedded Linux system. It can be used as a starting point for creating custom Linux distributions tailored to specific embedded devices or applications.

It serves as an example and a reference for building custom Linux distributions using the Yocto Project tools.



**What is Bitbake** :- The bitbake is the core of the yocto project. It is used to build the or package the recipes and create one image from it on any embedded hardware application. It takes the metadata from the application recipes and builds it and then further that image can run on any embedded hardware application. BitBake executes tasks, which are individual build or configuration steps. Tasks can be part of a recipe, and BitBake can be instructed to perform specific tasks or build specific targets (such as an image or a package).

**U-Boot with yocto** : - U-Boot, short for Universal Bootloader, is an open-source bootloader commonly used in embedded systems. It is designed to load and initialize the kernel and other necessary components before handing over control to the operating system

**CMake File : -** CMake is a cross-platform build system that is often used to build software projects. A CMakeLists.txt file is a script written in the CMake language that describes the build process for a software project. When it comes to Yocto, CMake is commonly used to build components of the software stack, and a CMakeLists.txt file is used to configure the build for a particular software component.

**Steps for downloading yocto image builder for our local system there are certain steps we have to follow** .

**Step 1** :- we must execute this command in our Linux terminal  **sudo apt-get install gawk wget git diffstat unzip texinfo gcc build-essential chrpath socat cpio python3 python3-pip python3-pexpect xz-utils debianutils iputils-ping python3-git python3-jinja2 libegl1-mesa libsdl1.2-dev pylint3 xterm python3-subunit mesa-common-dev**

**Step 2**  :-  **mkdir Yocto\_Project**

**Step 3 :- cd Yocto Project**

**Step 4 :- git clone** git://[git.yoctoproject.org/poky](https://eur03.safelinks.protection.outlook.com/?url=http%3A%2F%2Fgit.yoctoproject.org%2Fpoky&data=05%7C01%7Canirudha.ingole%40atlascopco.com%7Cafbca106ca4f4ffd218d08dbd04c37b0%7C556e6b1fb49d42788bafdb06eeefc8e9%7C0%7C0%7C638332797986658250%7CUnknown%7CTWFpbGZsb3d8eyJWIjoiMC4wLjAwMDAiLCJQIjoiV2luMzIiLCJBTiI6Ik1haWwiLCJXVCI6Mn0%3D%7C3000%7C%7C%7C&sdata=oCyZo2lTCAHt6RZxQ9y0FhETxZaDkgyQzPD%2FMn3bb7g%3D&reserved=0)

**Step 5** : - **cd Poky**

**Step6** :-  **git checkout Krickstone**

**Step 7** :- **cd .**.

**Step 8** : - **source poky/oe-init-build-env rpi-build**

**Step 9** :- **pdate bblayers.conf "Add Raspberry Pi Meta“**

**Open bblayers.conf**

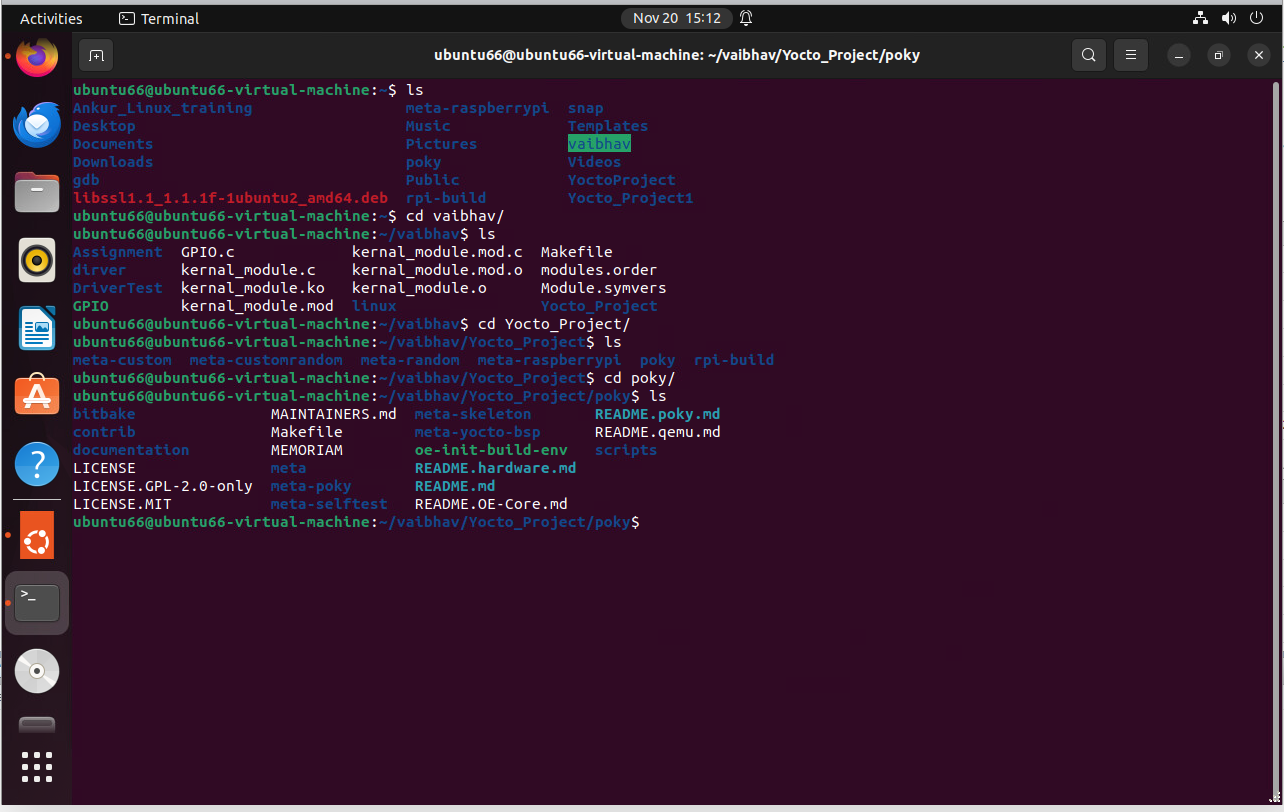
**$ vi conf/bblayers.conf**

**Add following line : /path/to/meta-raspberrypi**

**Step 10:**- **bitbake core-image minimal**

Alternative:

* alternative to this if doesnt work [https://git.yoctoproject.org/git/poky](https://eur03.safelinks.protection.outlook.com/?url=https%3A%2F%2Fgit.yoctoproject.org%2Fgit%2Fpoky&data=05%7C01%7Canirudha.ingole%40atlascopco.com%7Cafbca106ca4f4ffd218d08dbd04c37b0%7C556e6b1fb49d42788bafdb06eeefc8e9%7C0%7C0%7C638332797986658250%7CUnknown%7CTWFpbGZsb3d8eyJWIjoiMC4wLjAwMDAiLCJQIjoiV2luMzIiLCJBTiI6Ik1haWwiLCJXVCI6Mn0%3D%7C3000%7C%7C%7C&sdata=5YUPu96Qik7nKvHhCiVK38sNQ5b6riSG4Tnzbk5IA2Q%3D&reserved=0)
* alternative [https://git.yoctoproject.org/git/meta-raspberrypi](https://eur03.safelinks.protection.outlook.com/?url=https%3A%2F%2Fgit.yoctoproject.org%2Fgit%2Fmeta-raspberrypi&data=05%7C01%7Canirudha.ingole%40atlascopco.com%7Cafbca106ca4f4ffd218d08dbd04c37b0%7C556e6b1fb49d42788bafdb06eeefc8e9%7C0%7C0%7C638332797986658250%7CUnknown%7CTWFpbGZsb3d8eyJWIjoiMC4wLjAwMDAiLCJQIjoiV2luMzIiLCJBTiI6Ik1haWwiLCJXVCI6Mn0%3D%7C3000%7C%7C%7C&sdata=5bsQbcF3ZjsshYM3BCmMh3ajBFjMWj5uI7Xh%2BPDF%2FX0%3D&reserved=0)



**By following this documentation, we can setup the yocto build into our system successfully**

**Creating a new-meta layer in yocto**

**step1:-** $ source poky/oe-init-build-env

**Step2:-** *To create new custom model following is the command we have to follow* **bitbake-layers create-layer../meta-custom Creating New Meta-Layer In Yoct .**

**Step 3 :-** 2-To add new custom layer : **$ bitbake-layers add-layer ../meta-custom** Run below command to check if your layer is added $ bitbake-layers show-layer.

**Yocto DAS U-BOOT**

**Step 1** :- -**To enable u-boot in raspberrypi yocto image edit local.conf and add following in it : RPI\_USE\_U\_BOOT = “1” and add ENAB** **LE\_UART = “1**

**Step 2** :- **Build image with bitbake core-image-base.**