

An Installation Guide to Pyro-Yocto using Ubuntu for Peary-image

**D M S Sultan***, University of Geneva, Switzerland*

[d.m.s.sultan@cern.ch](mailto:d.m.s.sultan@cern.ch)

Preface

A Quick Guide to Intersubband (IS) Absorbance Study

Page: ii

This document has been prepared with an aim to provide a guideline the installation process of embedded Yocto Linux-distribution with Peary image. Peary facilitates controlling the sensor module with **C**ontrol **a**nd **R**eadout **I**tk **BO**ard **u**nit (CaRIBOu) comprising Xilinx ZC706 development board. It would be also worth mentioning that the materials considered within this manual corresponds to an alternative development within Windows machine. The user can certainly try in solo-installed Linux machine. However, other Linux distributions than Ubuntu has mentioned in the git repository (<https://gitlab.cern.ch/Caribou/meta-caribou>) could be the alternative options.

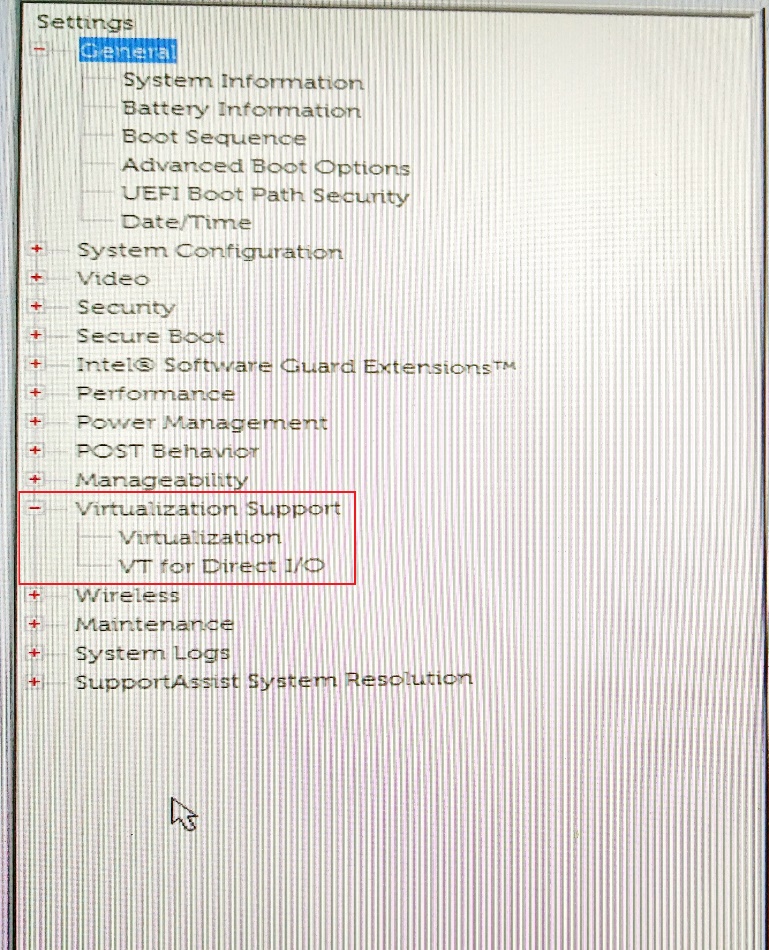
The author is acknowledging to Dr. Simon Spannagel and Adrian Fiergolski, EP-LCD group, CLIC for allowing the opportunity of the technical discussion at different stages. He also likes to make the specially mentioning to Dr. Mathieu, UniGE for sharing his experienced input. Nevertheless, it is worth mentioning here other colleagues or community in CERN, who are dedicatedly working to make the system powerful and functionally mature enough.

Content

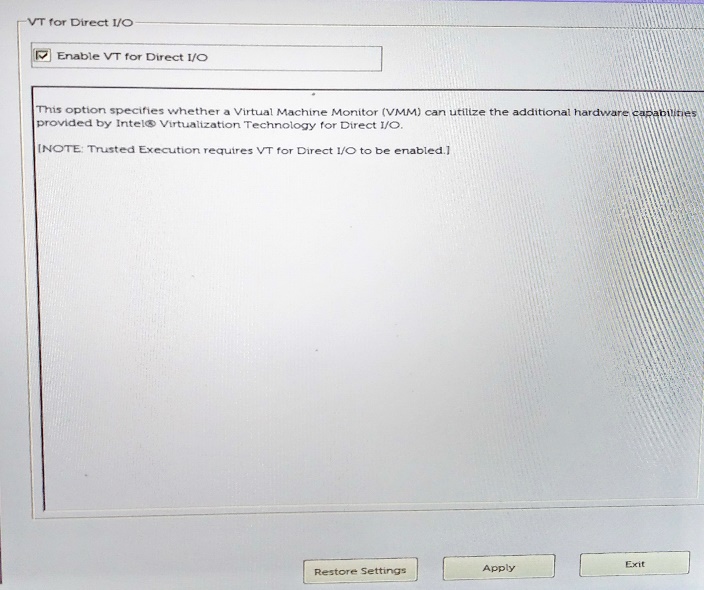
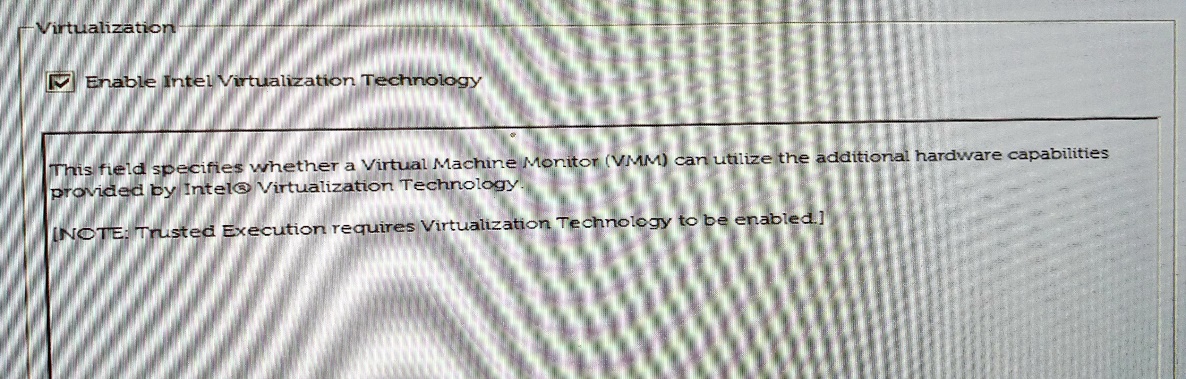
1. BIOS configuration 1
2. [Installing with VirtualBox](#_TOC_250002) 2
3. Import ready-image in VirtualBox 11
4. [Miscellaneous](#_TOC_250000) 16
   1. BIOS

The solution is intended to install ‘peary image’ within a windows machine. The technique will be followed here through the configuration of VirtualBox software in Windows interface. To facilitate the virtual full configuration, it would be worth enabling the virtual option from BIOS.

Please press ‘***F2***’ key while you be rebooting the machine. I am referring here a Dell machine XPS 15 and the BIOS interface could be different in end. One should find the section responsible for virtual machine support in BIOS. (See below)



Generally, both options remain disabled. One would need to enable them (see below) and the BIOS configuration pressing ‘***F10***’ key.



Reboot the machine and it would ready to install the virtual machine.

P.S.: Apology remains for the bad-quality images placed in this section.

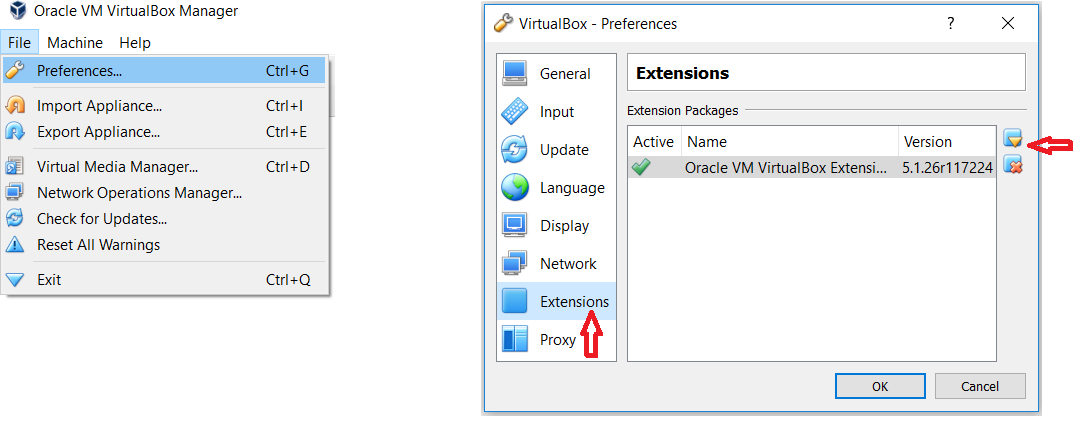
# Installing with VirtualBox

## 

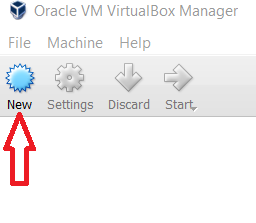
1. Download the VirtualBox for windows and extension pack. Install VirtualBox first.

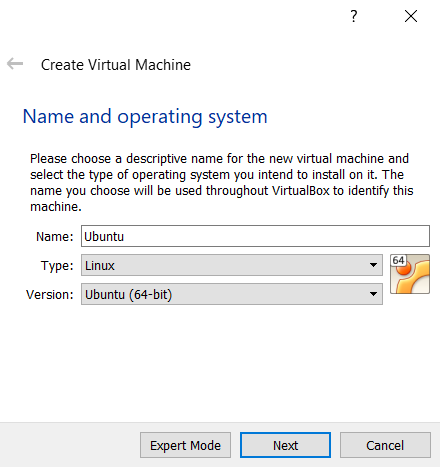
\

1. Once the VirtualBox is installed, click on preference from ‘***File***’ tab. Press the ‘***Extensions****’* and select extension package location and then press ok to install the extension pack for virtual box.

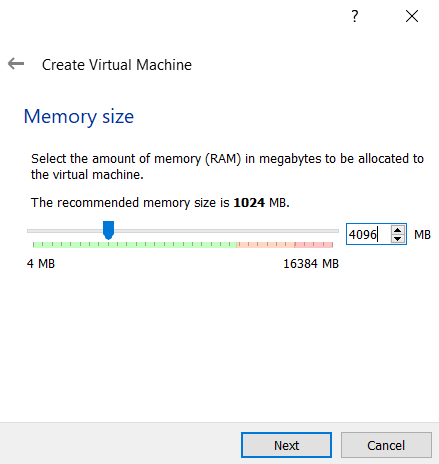


1. Click now on new to install a newer package. Give a proper description and click ***‘Next’***. However, you can even choose 32bit Ubuntu edition, if you like. I would propose using the Long-Term Edition (i.e. 16.04 LTS)

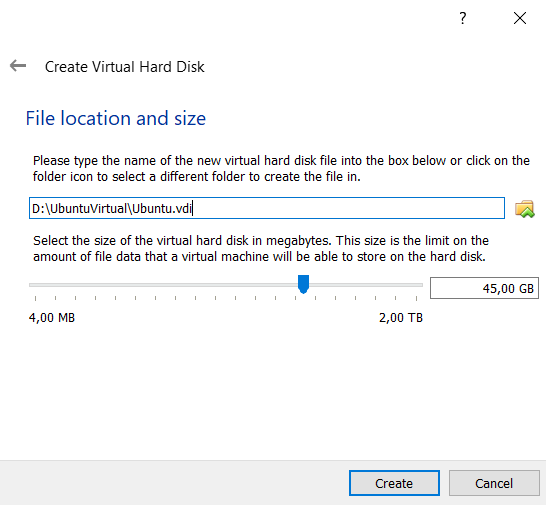




1. As an example 4GB Ram is chosen. The greater will you choose, would be better.

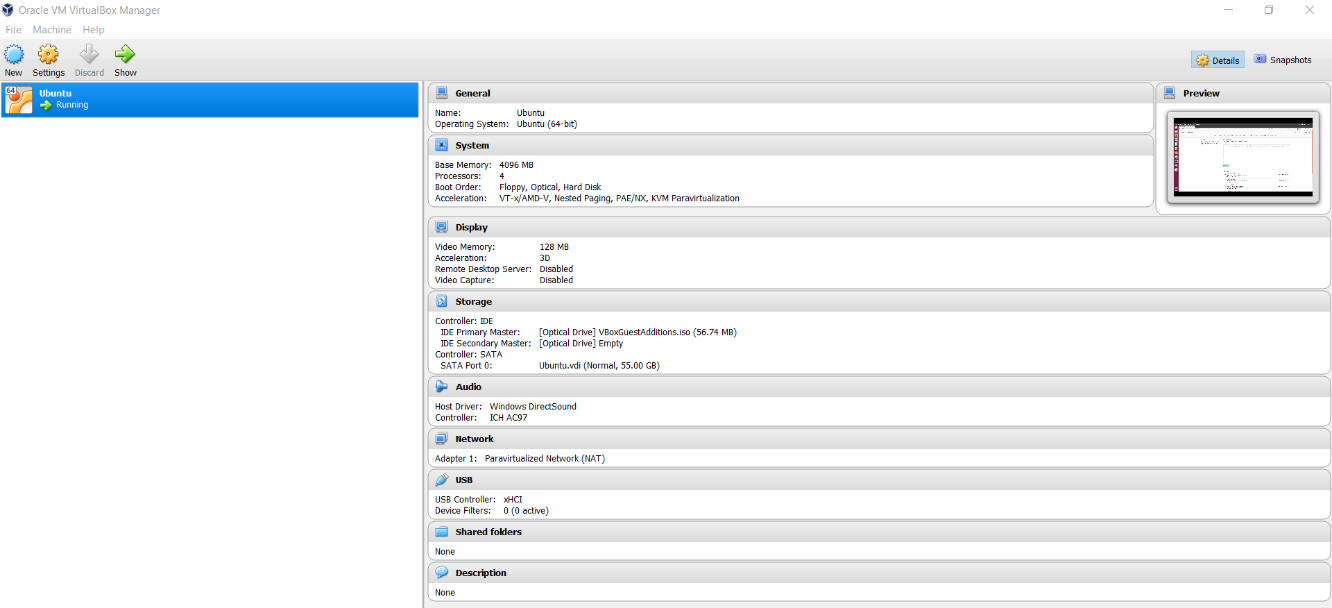


1. Choose the virtual drive with a fixed space of 55 GB at least. Choose the location in drive where you have already twice the size free-space available in windows file system.

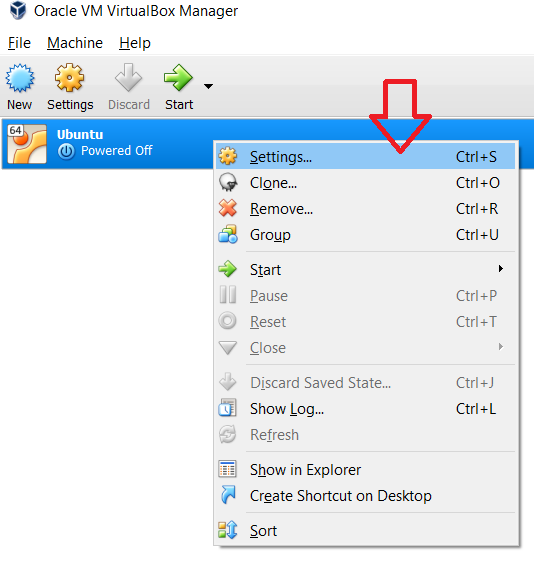


55,00 GB

1. After successful virtual HDD creation, one should see the following window.



1. Right-click on Ubuntu and select settings.



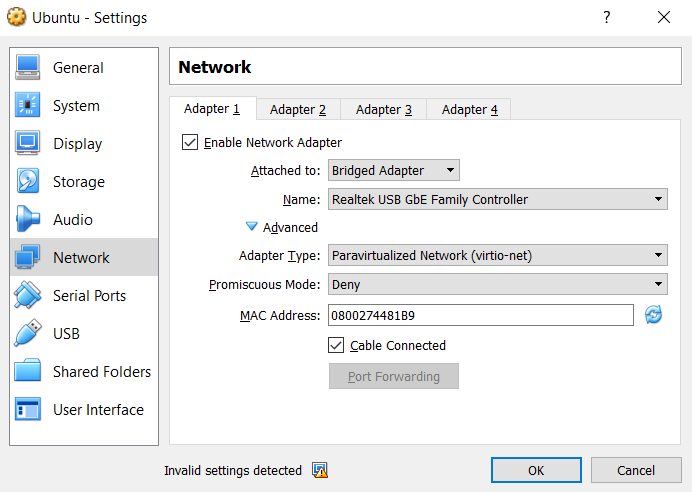
1. From the ‘***System***’ tab and following ‘***Processor***’ tab, choose the no. of processor cores (the more will be better). Keep remember, one shouldn’t allocate all processor cores, otherwise, your Windows host will start interrupting. Enable PAE/Nx option to access the processor physical address within virtual machine.



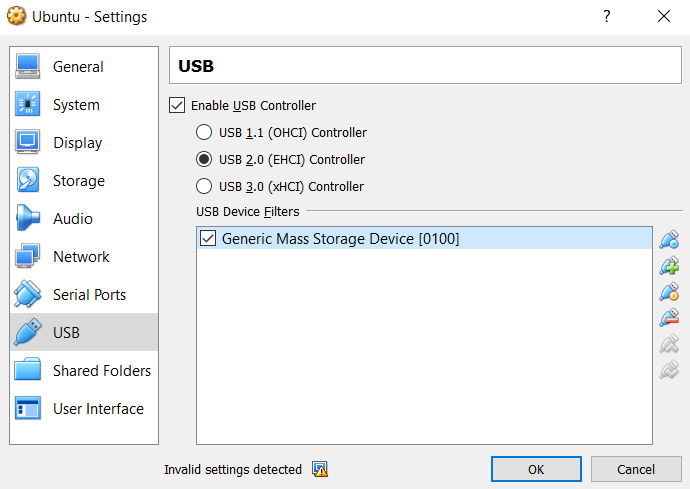
1. Choose the boot image location in storage tab.



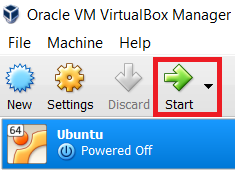
1. On Network tab, choose the Bridge-Network option with vitro net adapter. NAT option will also work fine.



1. From USB section, check if one can include the memory stick.



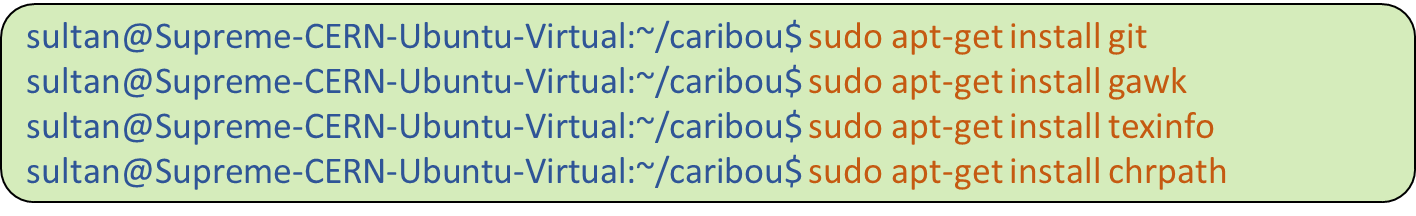
1. User should be ready now to ready to install Ubuntu. Press ‘OK’. Press the start and complete the Ubuntu install process done.



1. Once, you logged in to new installed ubuntu, create a folder at your desired direction. I made in *home/user* directory.
2. Using Terminal of Ubuntu go to current directory.



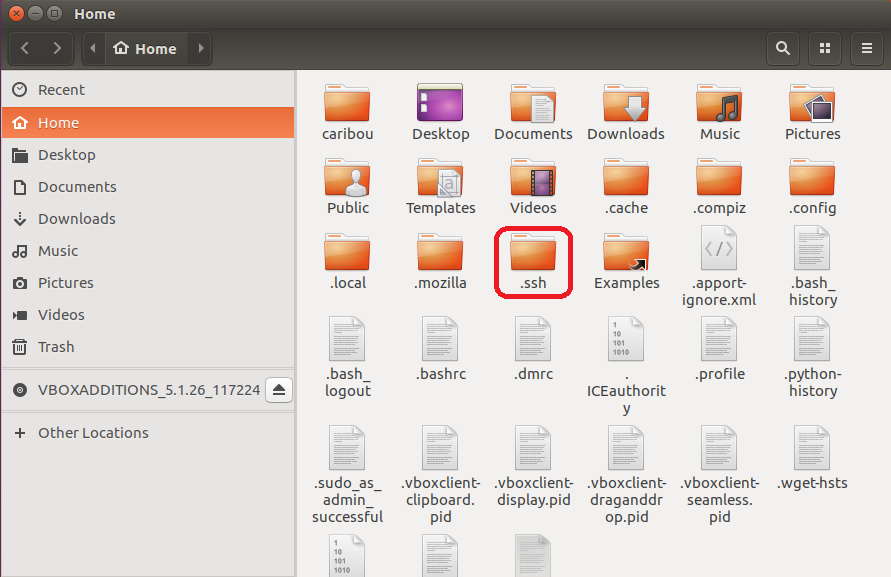
1. Install the following four packages

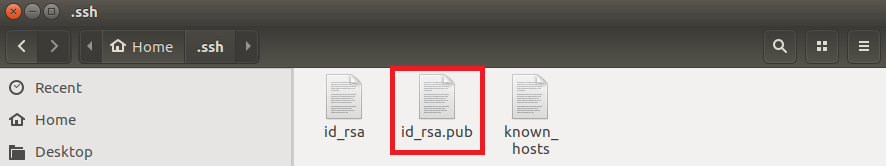


1. Now create the ssh key to access or clone the contents hosted at meta-caribou git-hub location.



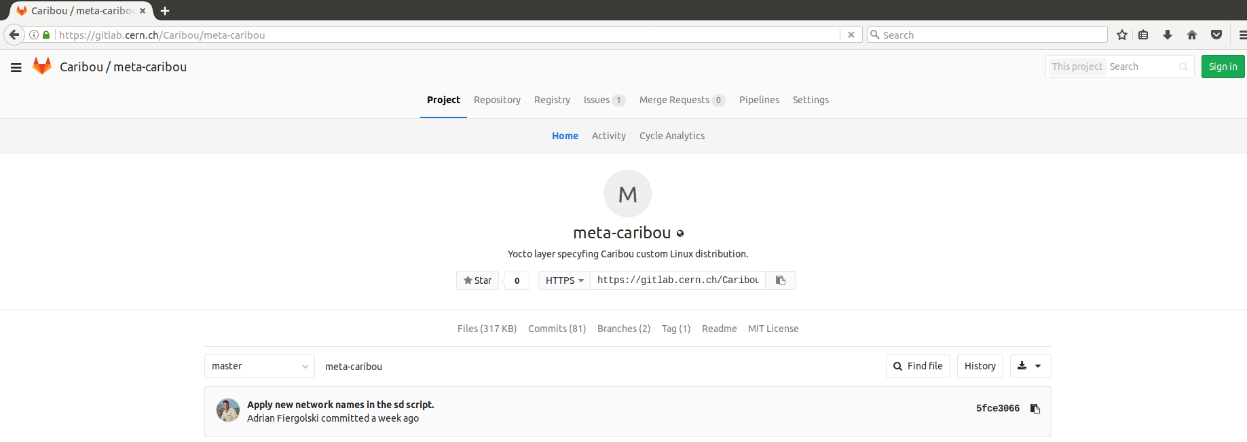
1. If you have chosen the default location to generate key, it will be in *home/user* directory. From the file manager, go to that directory. *Ctrl+h* key will allow to see hidden file. Open it. Open the ‘*id\_rsa.pub*’ file with any editor. Double click on it shall make this document open in default installed ***LibreOfficeWriter***. Of course, your email address will be different than mine, as you could see from public key example.

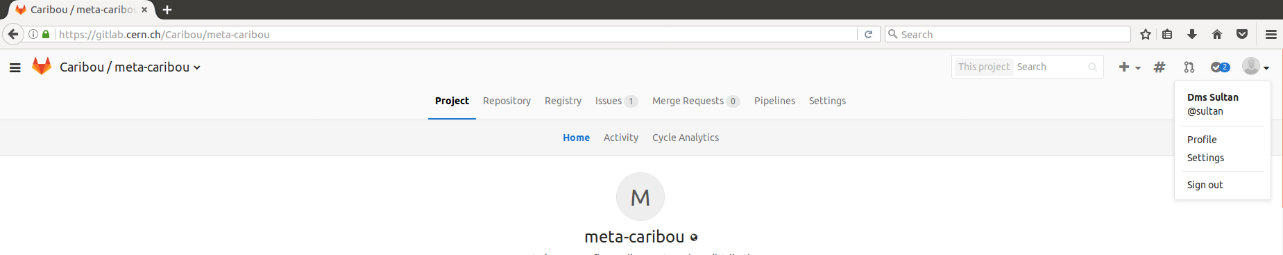




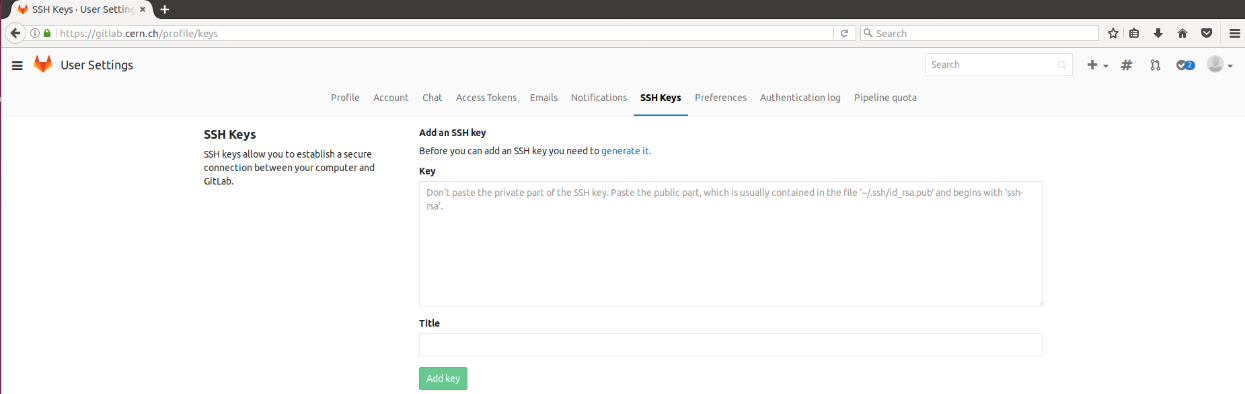


1. Sign in with your user login-id into meta-caribou git location. Click on *Settings.*

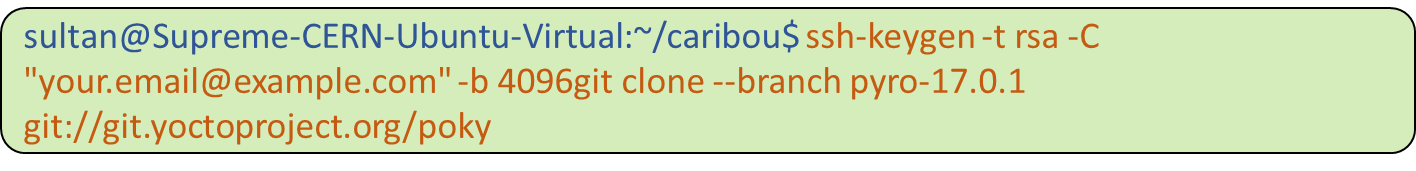




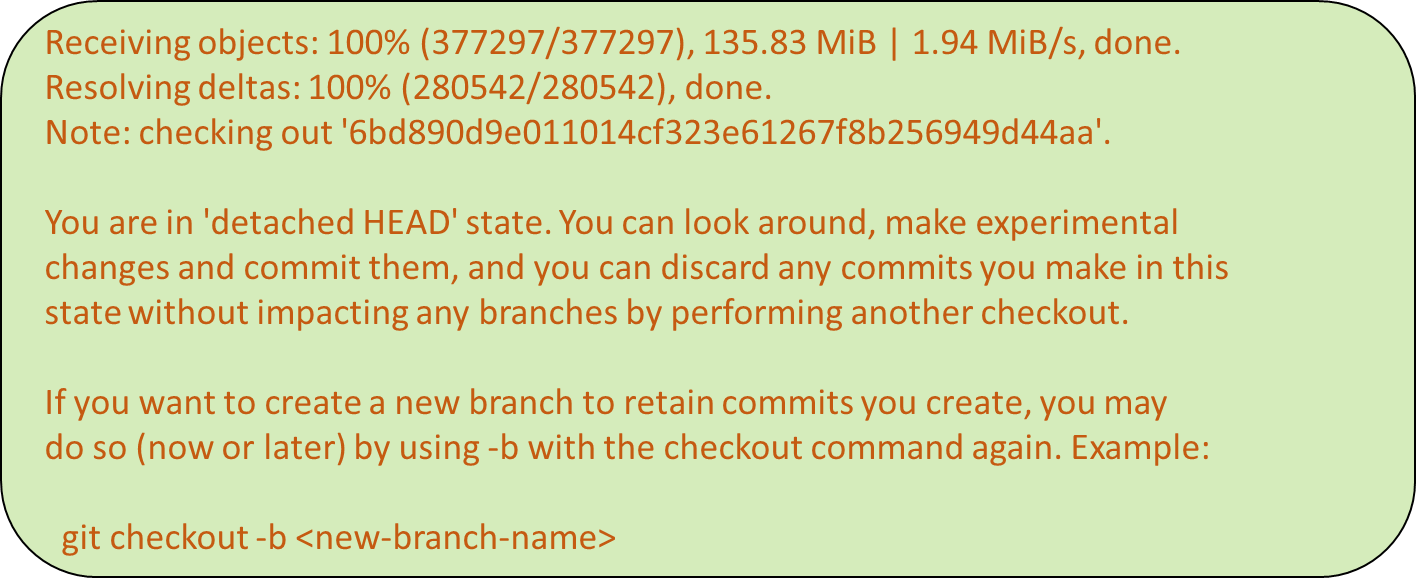
1. Click on ***SSH keys*** tabs. Copy and paste the content of *id\_rsa.pub* key into the key section, as shown in following image. Give a proper title. Add key at the end pressing *Add key*.



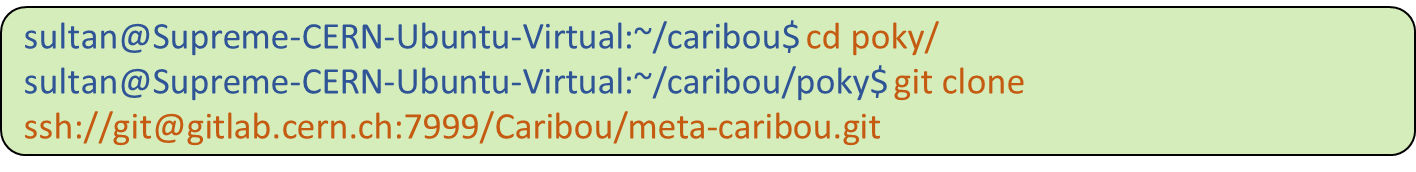
1. Please wait few seconds to receive an email at your given email address that will be confirming that the key is created.
2. Using Terminal of Ubuntu starting cloning now the pyro-yocto files.



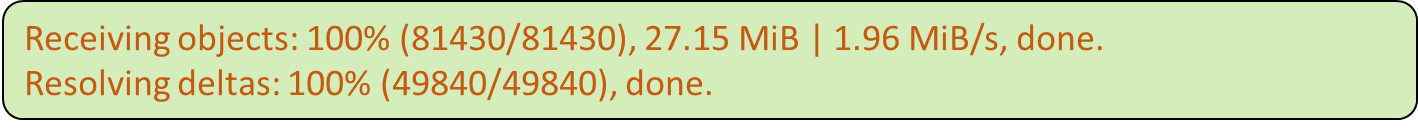
It would ask you for ssh authentication, if had created password during ssh-key generation. If so, use that password and proceed. At successful cloning, you should receive the following message.



1. ***poky*** folder will be created consisting all cloned contents. Go there and clone the meta-caribou contents.



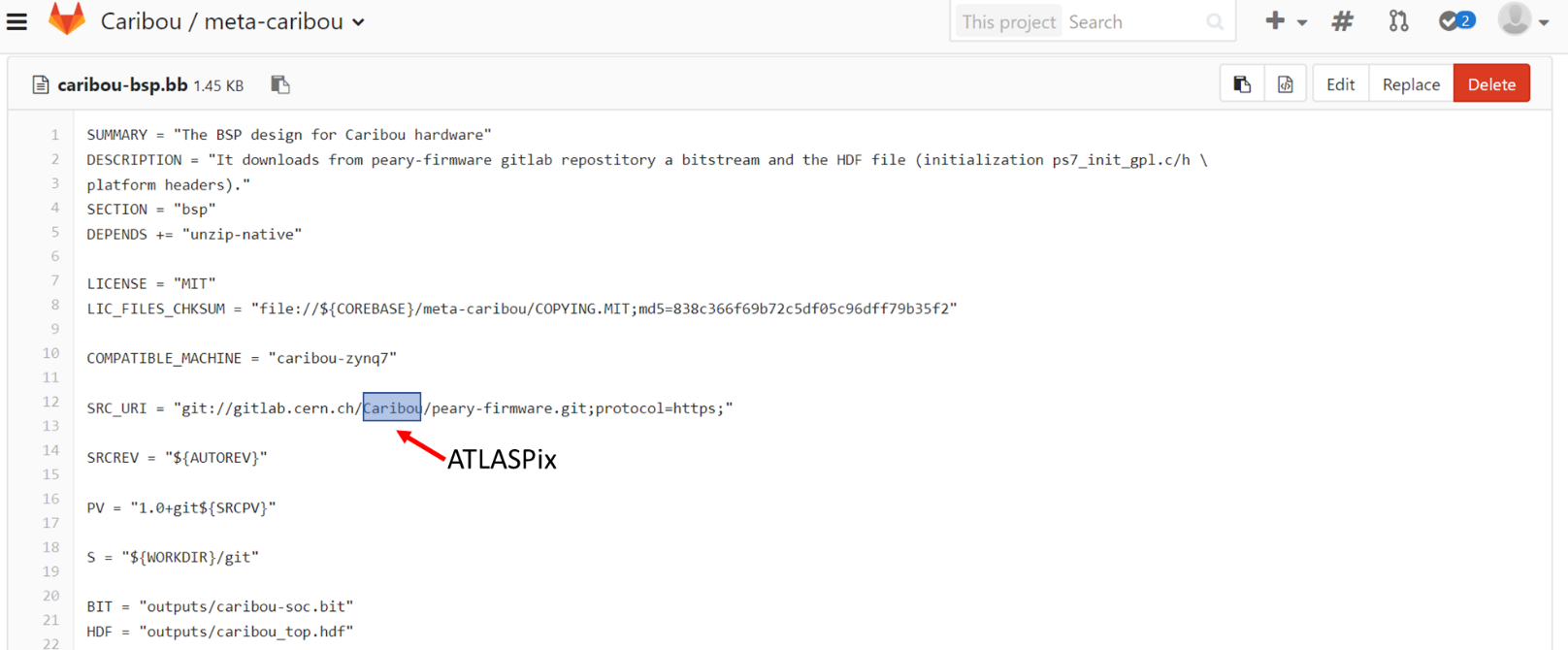
At successful cloning, you should receive the following message.



1. **Optional** (Important mainly for ATLASPix system developer): To build the yocto image with modified peary and firmware for ATLASPix chips, one must modify the two .bb file in the cloned meta-caribou layer.



Change in the following ***SRC\_URI*** section: replace ***Caribou*** with ***ATLASPix***.

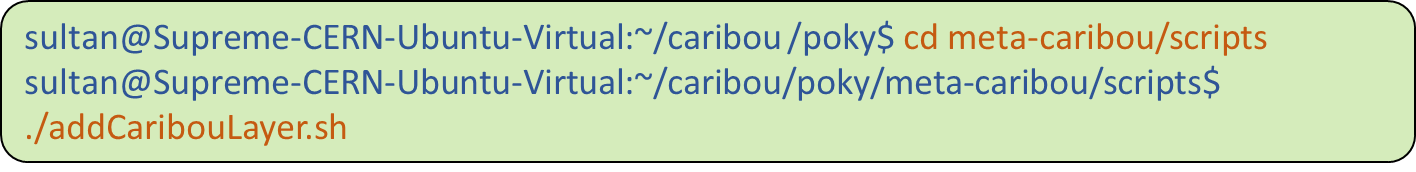


The same thing is also needed to be repeated with ***peary\_git.bb***.





1. To use the caribou layer, one need to make the build system aware of it. Moreover, one need to install and add all the dependency layers. Change directory to *meta-caribou/scripts* and run *addCaribouLayer.sh*.

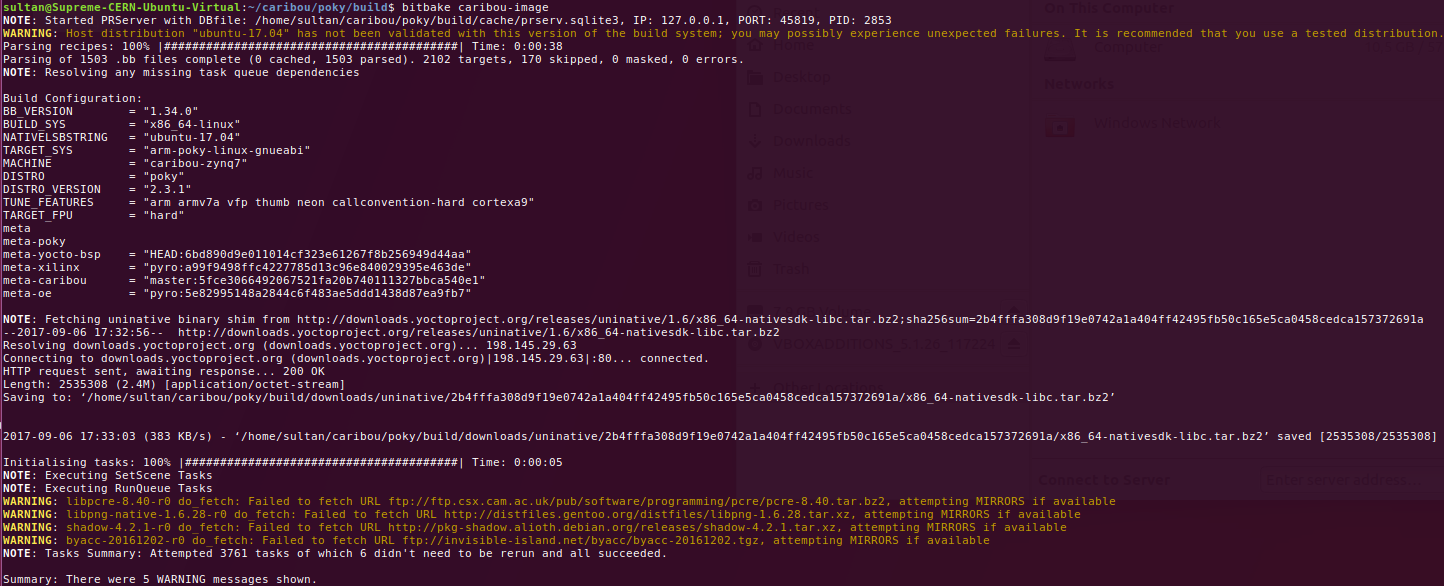


1. Once the successful caribou layer scripts completion, one need to be back at poky directory and initialize the build environment.



1. Now one should be ready to build the peary image. ‘bitbake’ tool will be used to build the image.



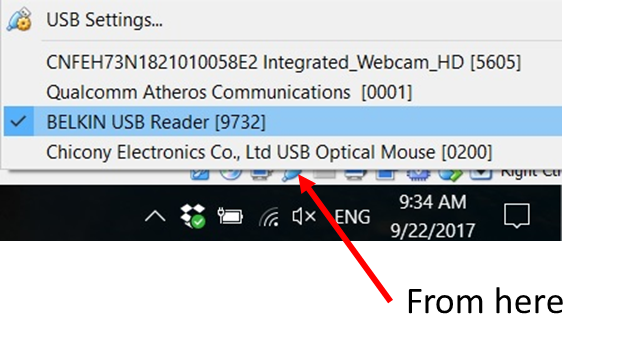


The above peary image has produced with Ubuntu 17.04 desktop version. However, as recommended earlier, Ubuntu 16.0.4 stable version will definitely work.

1. After successful completion of the image creation, it would be worth baking wic-tools, that generates the *.direct* file to be loaded into your SD card.

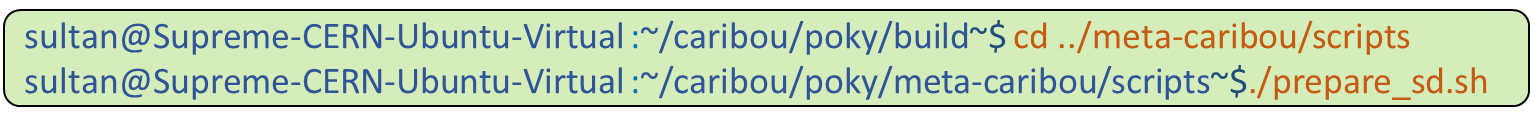


1. One must connect the SD card reader in a **USB** port with a SD card with it. Make it active from virtual box interface.

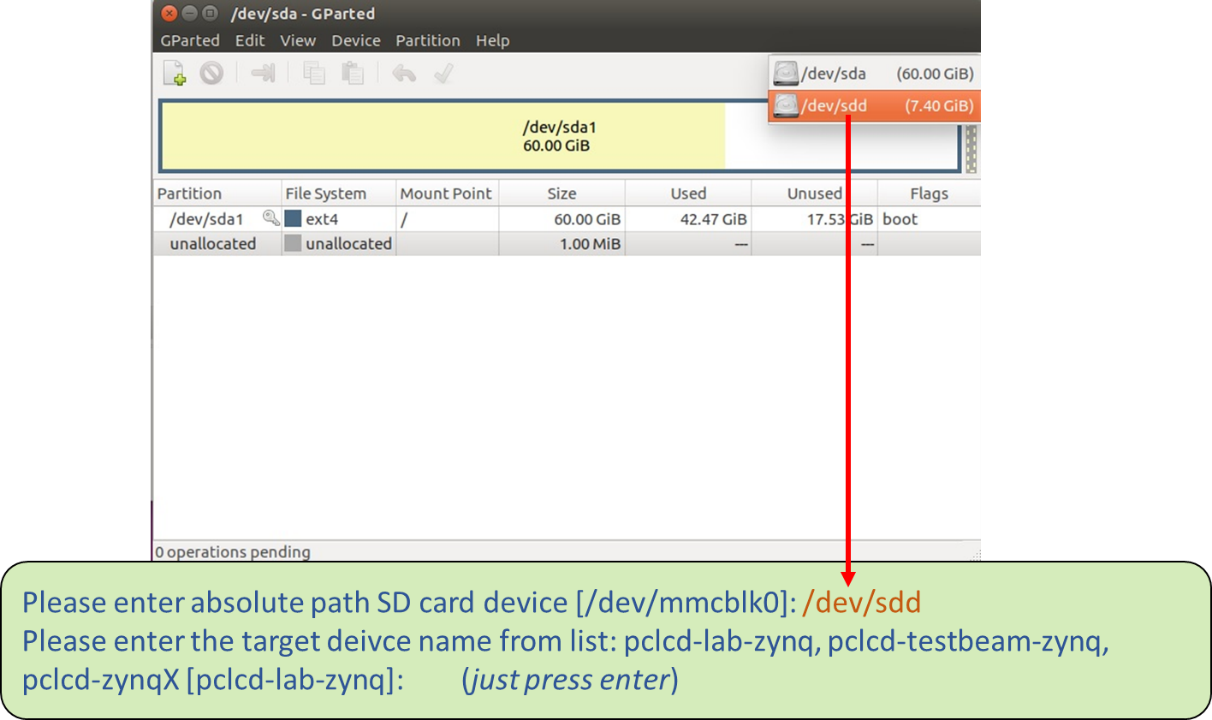


P.S.: *If one would like to use the default laptop memory card reader, he must install relevant driver to make the virtual ubuntu know about its existence.*

1. Image-producer is now almost set to prepare your SD card. Please go back /meta-caribou/scripts and run the scripts of SD card preparation.

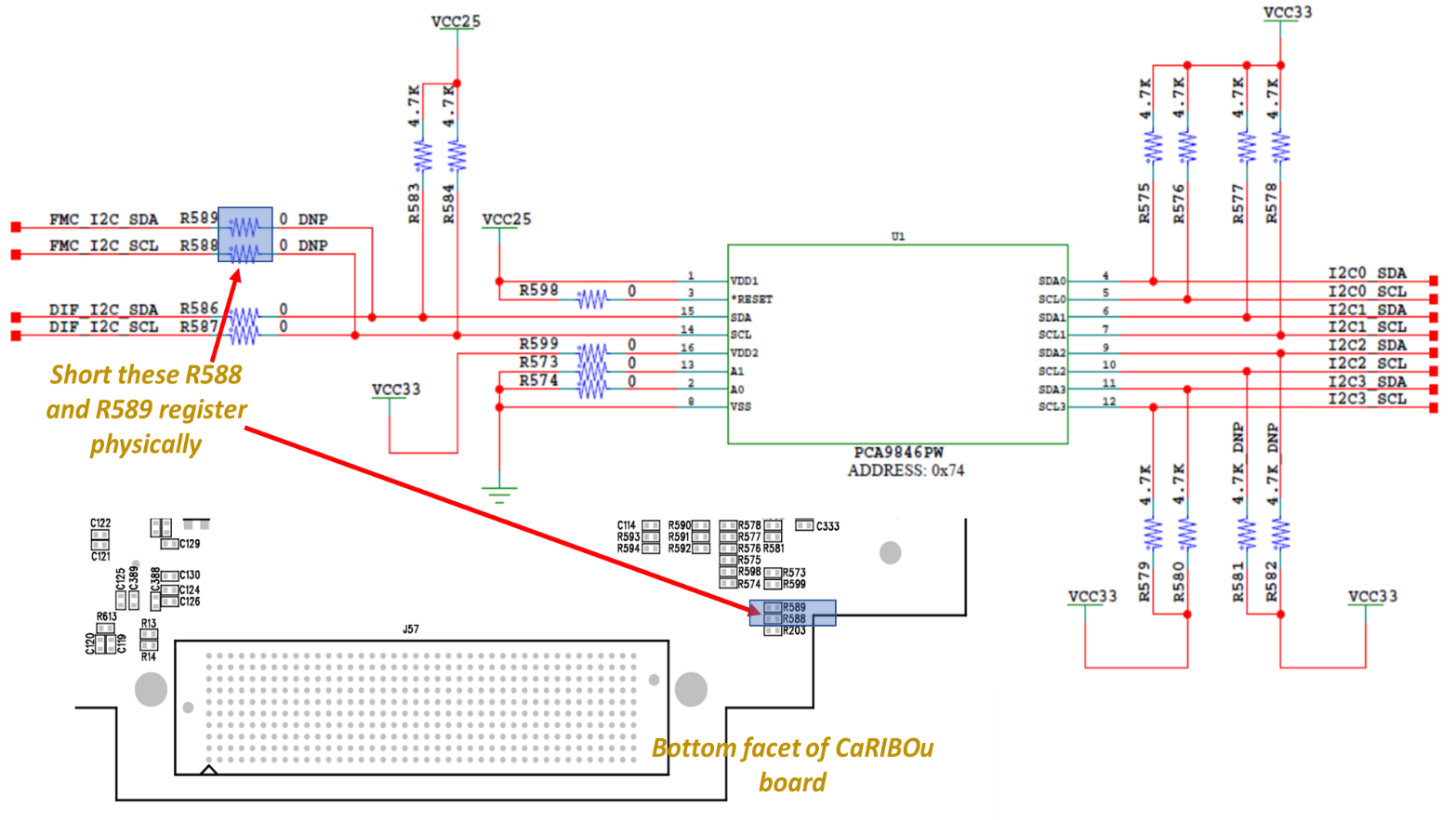


At later stage of the SD card preparation, it will ask for the absolute location of SD card. If you have already installed Gnome partition (gparted) software, you could locate the mount point of SD card.

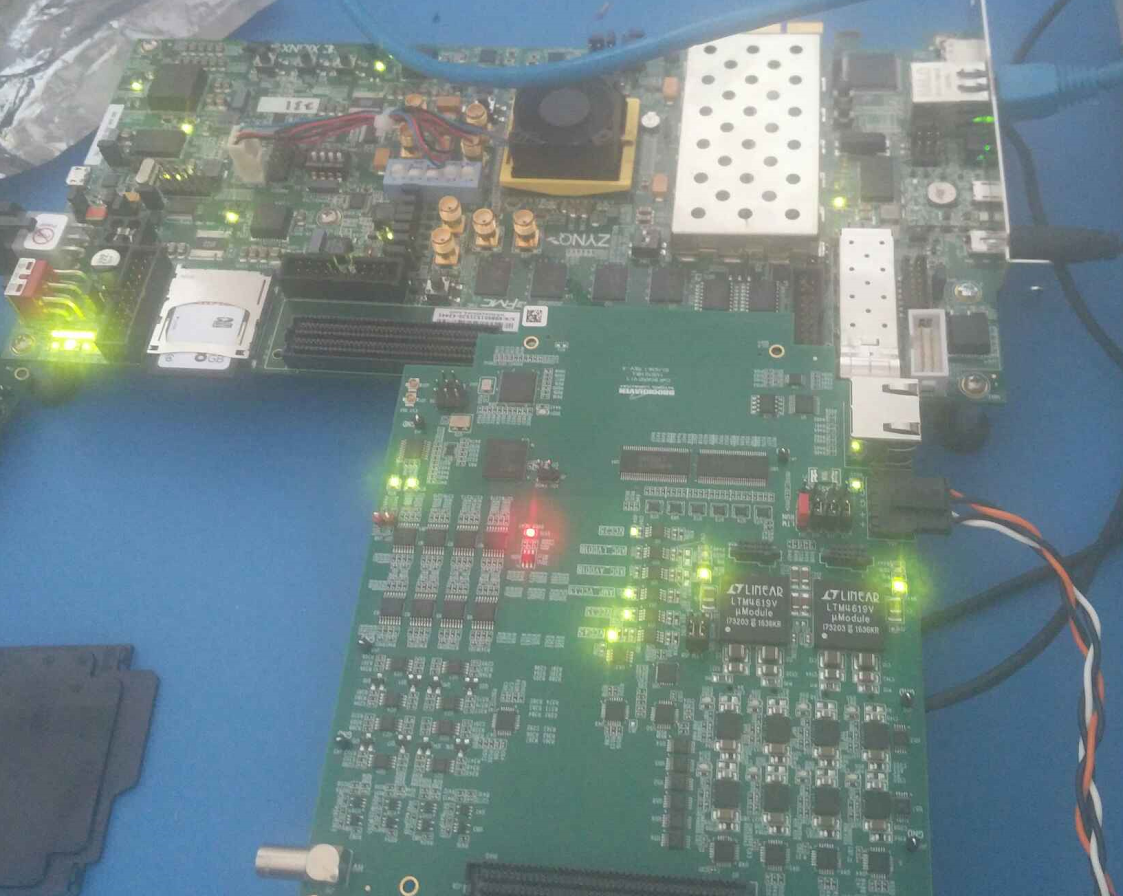


This list of devices is specifically dedicated with fixed IPs, hosted in CLIC lab. Please wait until you would see ***Done*** message appears.

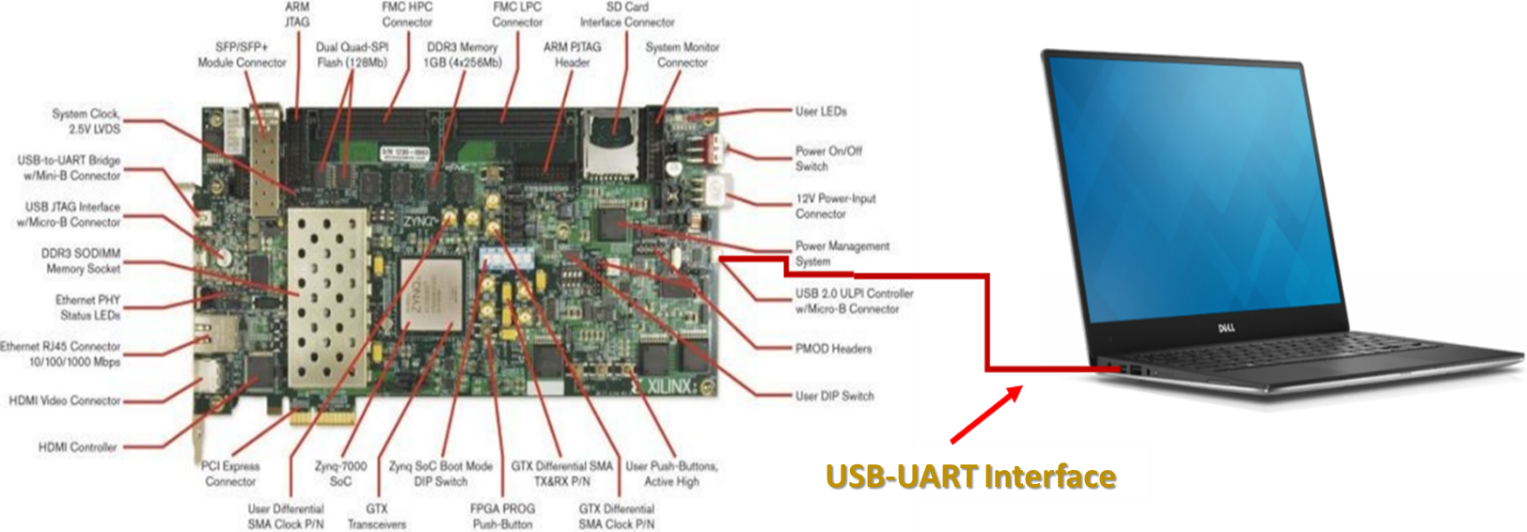
1. If the user is not using the extended FPGA Mezzanine Card (FMC) PCB, he has to short the register R588 and R589 for enabling via-communication of I2C.

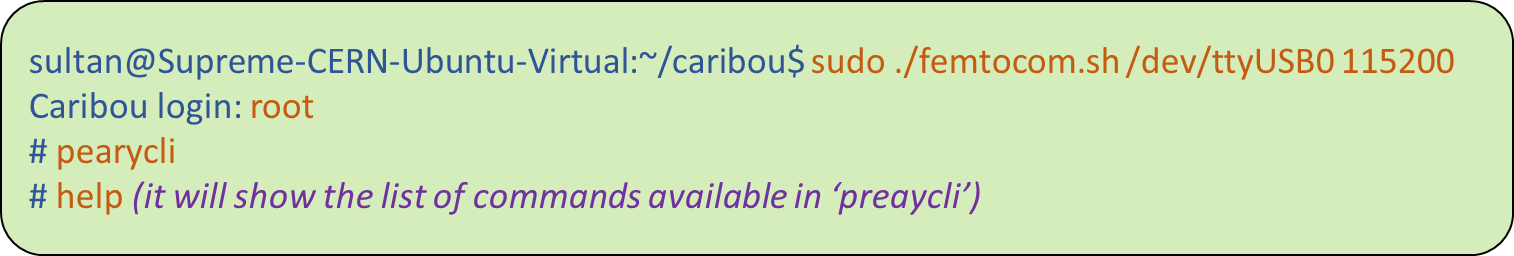


1. One may connect the CaRIBOu board directly to High Pin Connector (HPC) as following. Please put yocto-peary-image ready SD card in the FPGA board. Power On the CaRIBOu board with 6V and then switch On the ZYNQ706-FPGA board.

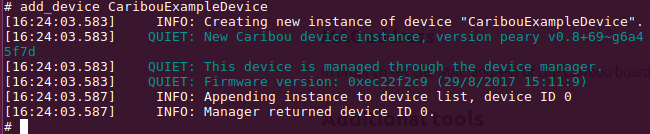


1. For the USB-UART communication (viz. connection settings below), one has to go to ***/meta-caribou/scripts*** and run the ***femtocom*** script.





1. There has been an emulated example device. One can simply try to add it for checking I2C communication as enabled.



One can also try to add ‘***ATLASPix***’ device, if one has already made the SD image ready for ATLASPix device.

# Import the ready image to VirtualBox

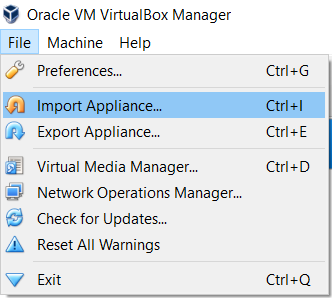
It is rather a rapid solution. This section is sharing the information of an already ready and built Ubuntu image for Peary-ATLASPIX system developers. The last stable progress for ATLASPix integration into peary system at CERN has included in the following image. One would just need to download the image copy, import it into VirtualBox interface and run the image.

1. Please follow the **first two steps** to install VirtualBox into your windows machine, as stated in section, entitled, ‘**Installing with VirtualBox**’.
2. Download the ***.ovf*** file from the following location.

[Download Here](https://drive.google.com/file/d/0B4Lk2iY46EJla1MwUDVmZERiQTA/view?usp=sharing)

For any inconvenience in accessing, please drop your note at ‘d.m.s.sultan@cern.ch’.

1. Import the VirtualBox compatible image and run the machine afterwards, once the image loading be completed. Please note that a SuperUser has already set to this image.



The credential to login the Ubuntu is:

Username: **UniGE**

Password: ***unige***

# Miscellaneous