**Installing and Configuring Toolchain**

**Write your first C program**

Step 1 : choose any text editor for writing your c program eg: Atom, Sublime text, vim, vi, notepad ++, etc...

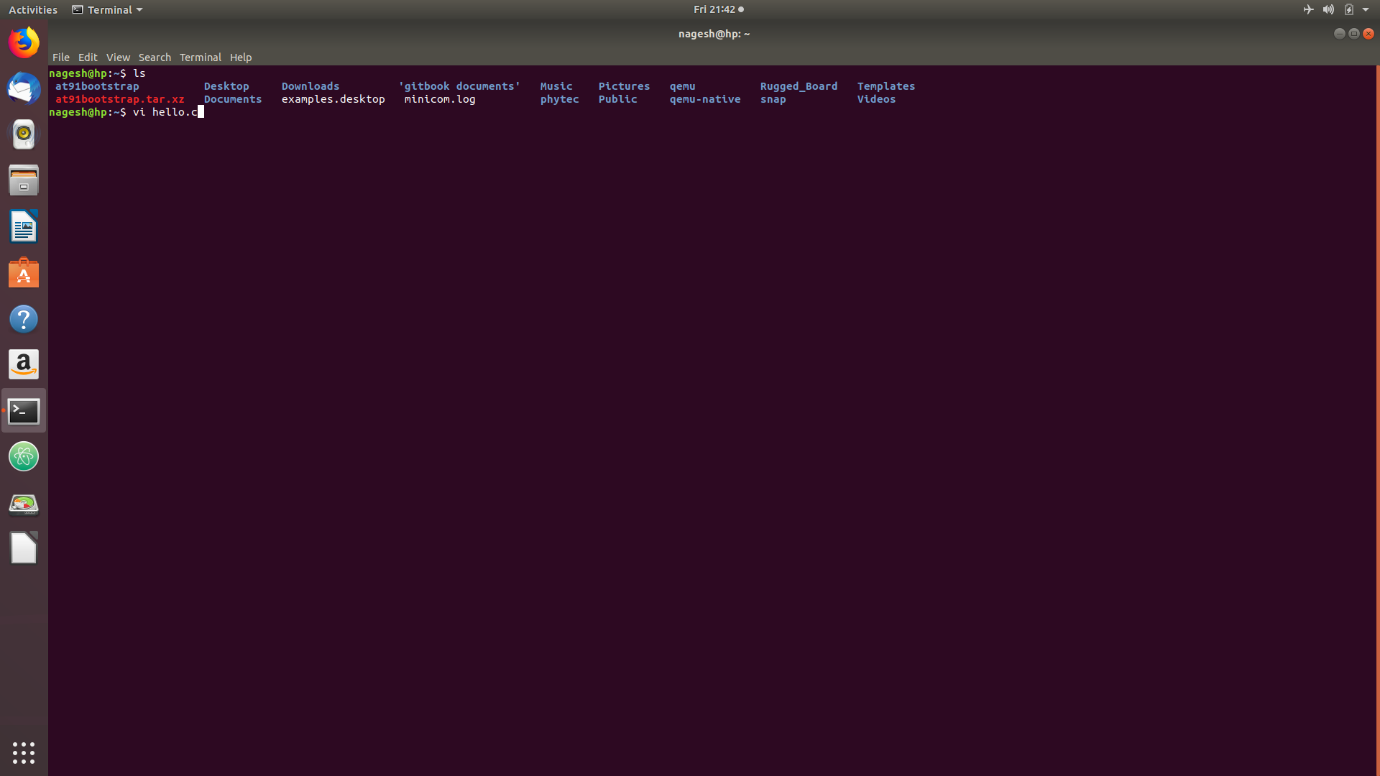


Fig: hello.c file using vi editor

**Step2**: Write a small C code on your HOST PC and save it with a .c extension also I changed the name of hello.c to mine.c in order to change the name you can give the command **mv hello.c mine.c** the name hello.c will change to mine.c . You can try our example code given below:

(You can save the code in any directory just remember the path where you have saved the code )

#include <stdio.h>

int main()

{

printf ("Hello board!!\n");

return 0;

}

**Note**: For saving in vim and vi use the key **Esc** and then type **:wq!**

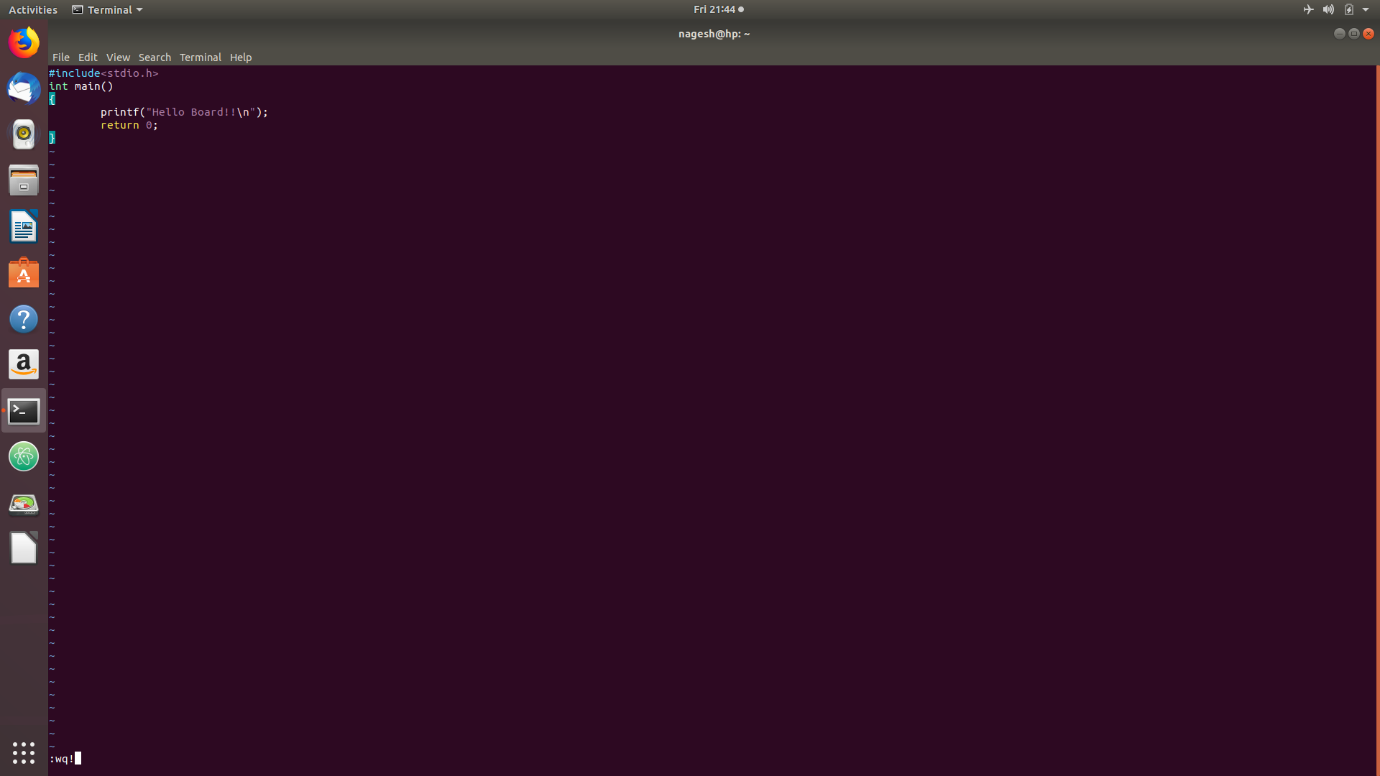
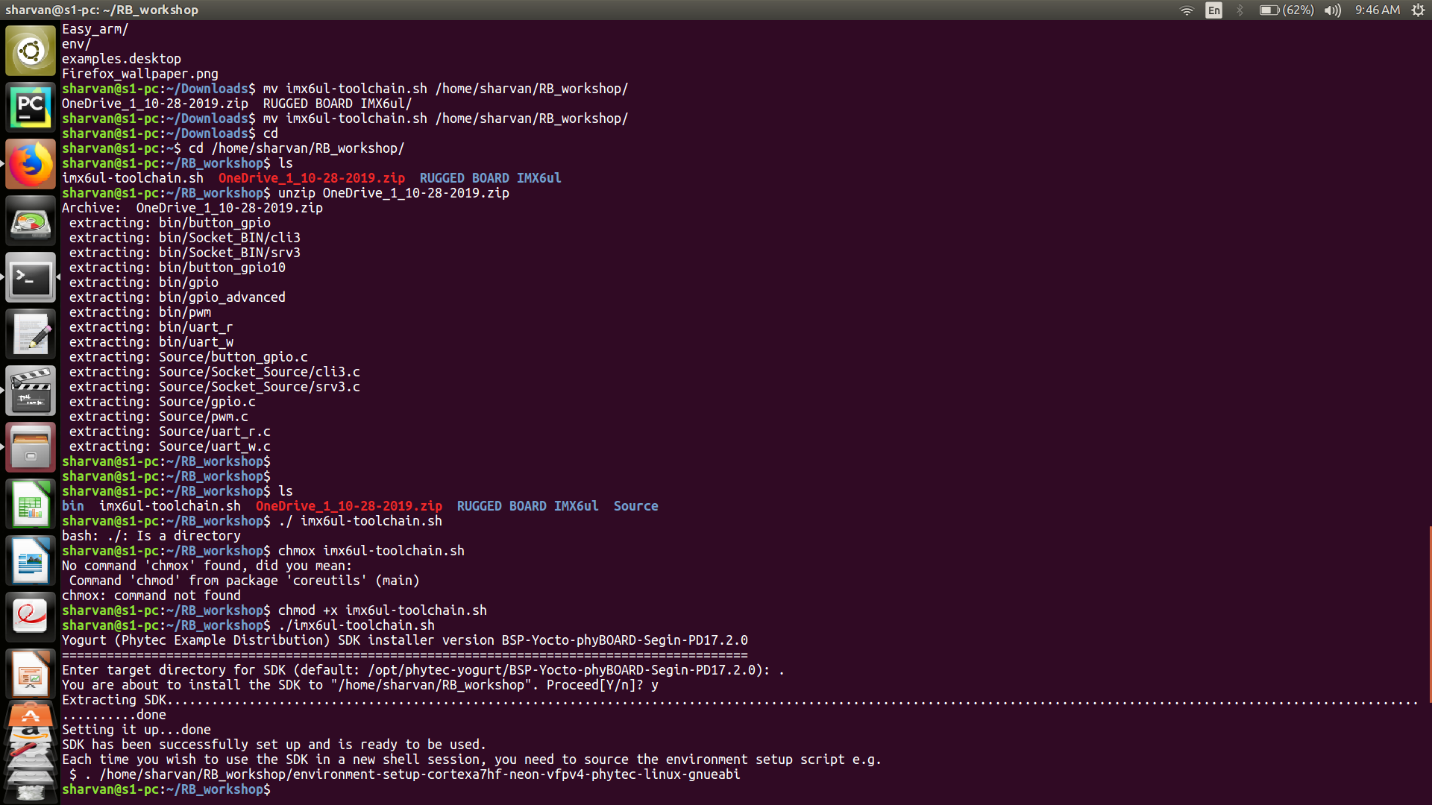


Fig : Using vi to save the code (:wq! For saving)

**Step 3:** Create a directory with the name RB\_workshop in your home directory, inside RB\_workshop directory make one more Directory with the name Toolchain which will have the imx6ul.sh file present in it as shown in the picture given below.



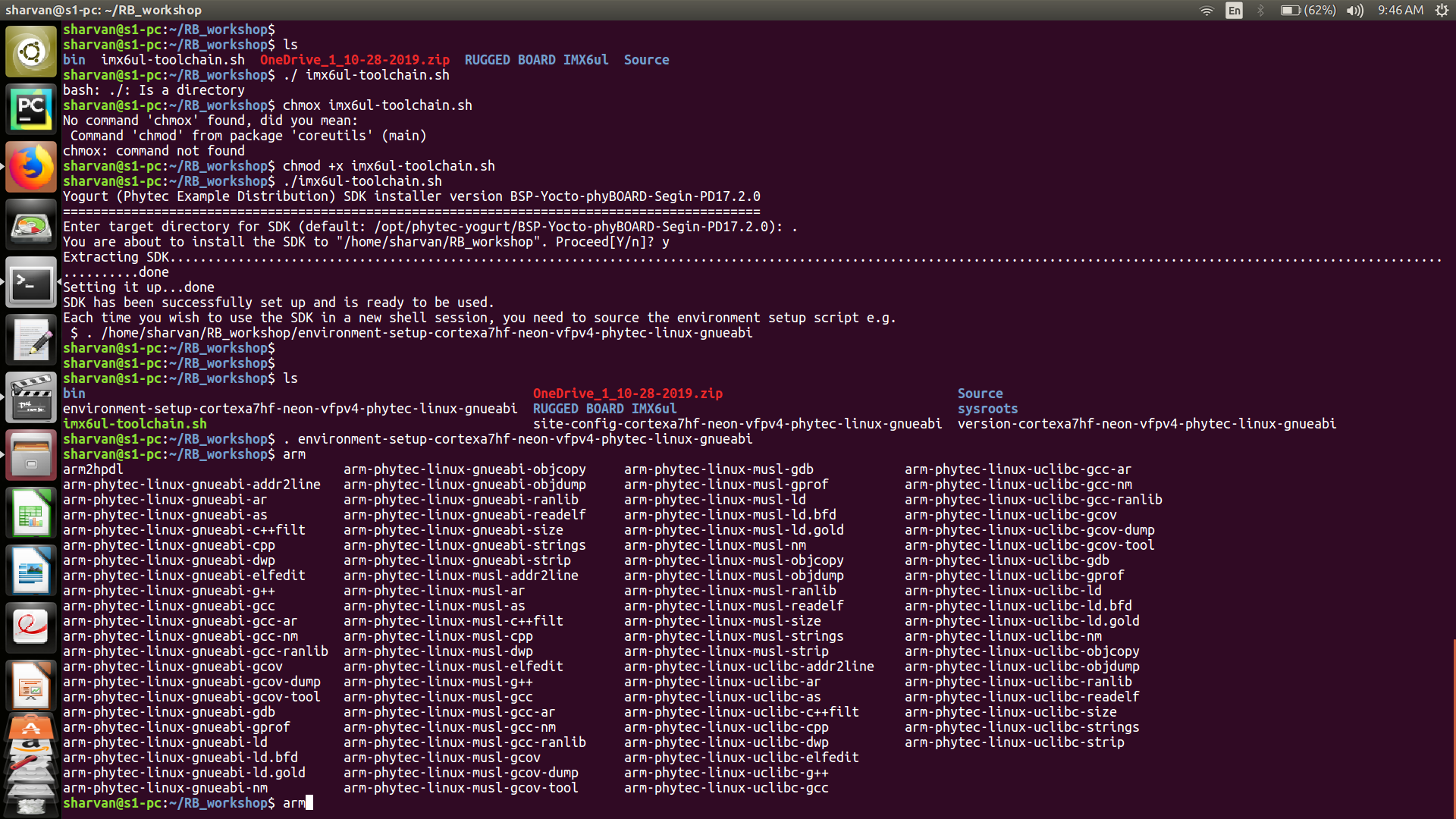
**Step 4**: As you can see the imx6ul.sh is present in the RB\_workshop directory, it is not an executable file so you should type the command **chmod +x imx6ul.sh** for executable permission in order to run this imx6ul.sh file, after giving the executable permission you should run the .sh file by typing **./ imx6ul.sh ,** it will ask you for the default location of SDK where you should give **. (which stands for the present directory where you want to extract the toolchain env file)** after this it will ask for a **y/n** permission where you have to give **y** for extraction.

Note : this may take up to 5 – 10 mins for extraction.

**Step 5**: Once you run the imx6ul-toolchain.sh you will get 1 environment-setup-cortexa7hf-neon-vfpv4-phytec-linux-gnueabi. Run this file by typing

**. environment-setup-cortexa7hf- neon-vfpv4-phytec-linux-gnueabi**

**Step 6**: **Make sure the toolchain is running on your PC by typing "arm" and hitting the <TAB> key twice.**

****

**if the toolchain isn't running you'll only see a very few of them.**

**NOTE: Do not close or switch your terminal as it will disable the toolchain that is running. We need to compile the C file with the enabled toolchain to get an executable that can be executed on the Board, as our PC is an X86 architecture and the board is an ARM architecture.**

**Step 7:** Now we are using libmraa (a third party library to compile our c code and provide a binary executable file.)

**Step 8**: We have provided you with a **mraa** directory please enter into that **mraa** directory and if you find a build directory inside it remove that build directory by typing: **rm -rf build/ (This will remove the build directory)**

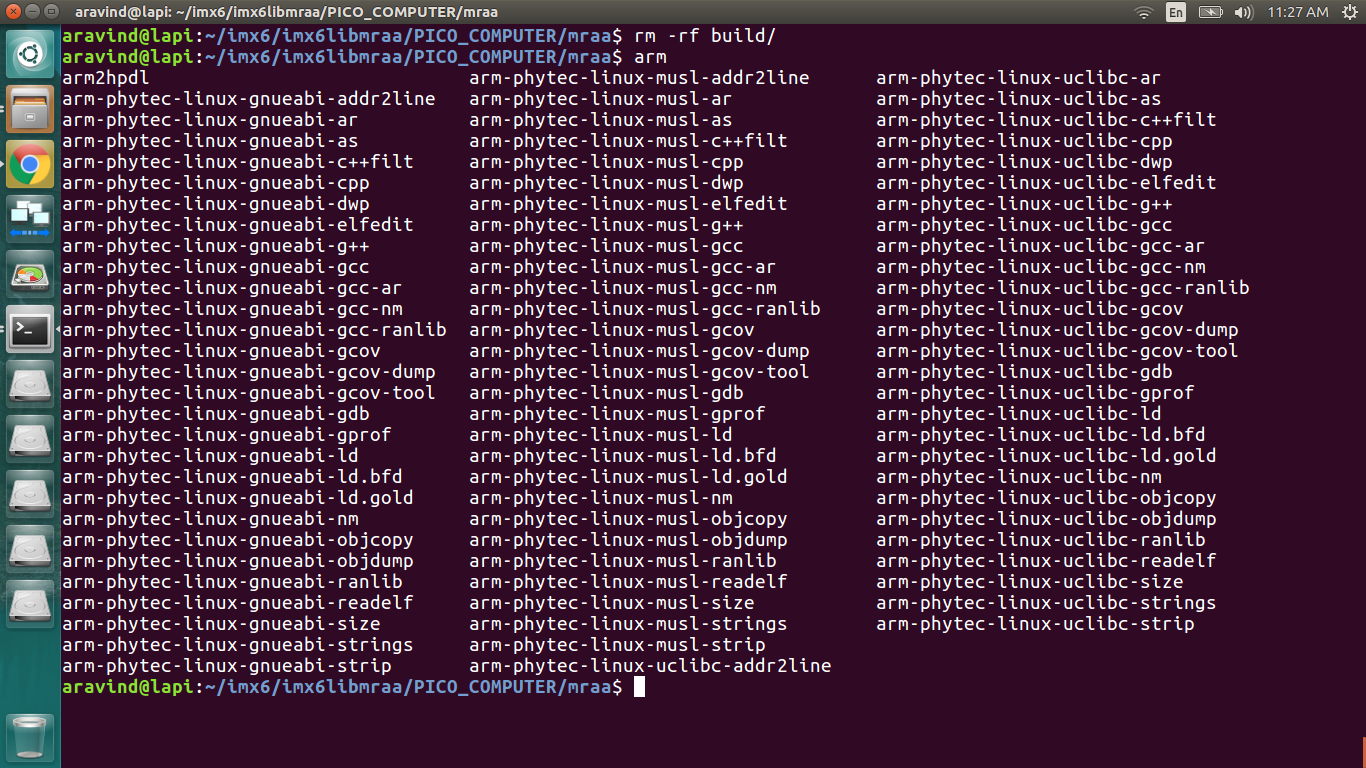


Fig : navigating to mraa directory

**Step 9**: check whether the toolchain is enabled or not by typing arm (and hit double <TAB>) you should get the above image if your toolchain is enabled.

**Step 10**: Now inside the mraa directory create a directory by typing the command **mkdir build,** then go inside examples directory and after that inside c directory as shown in the figure given below.

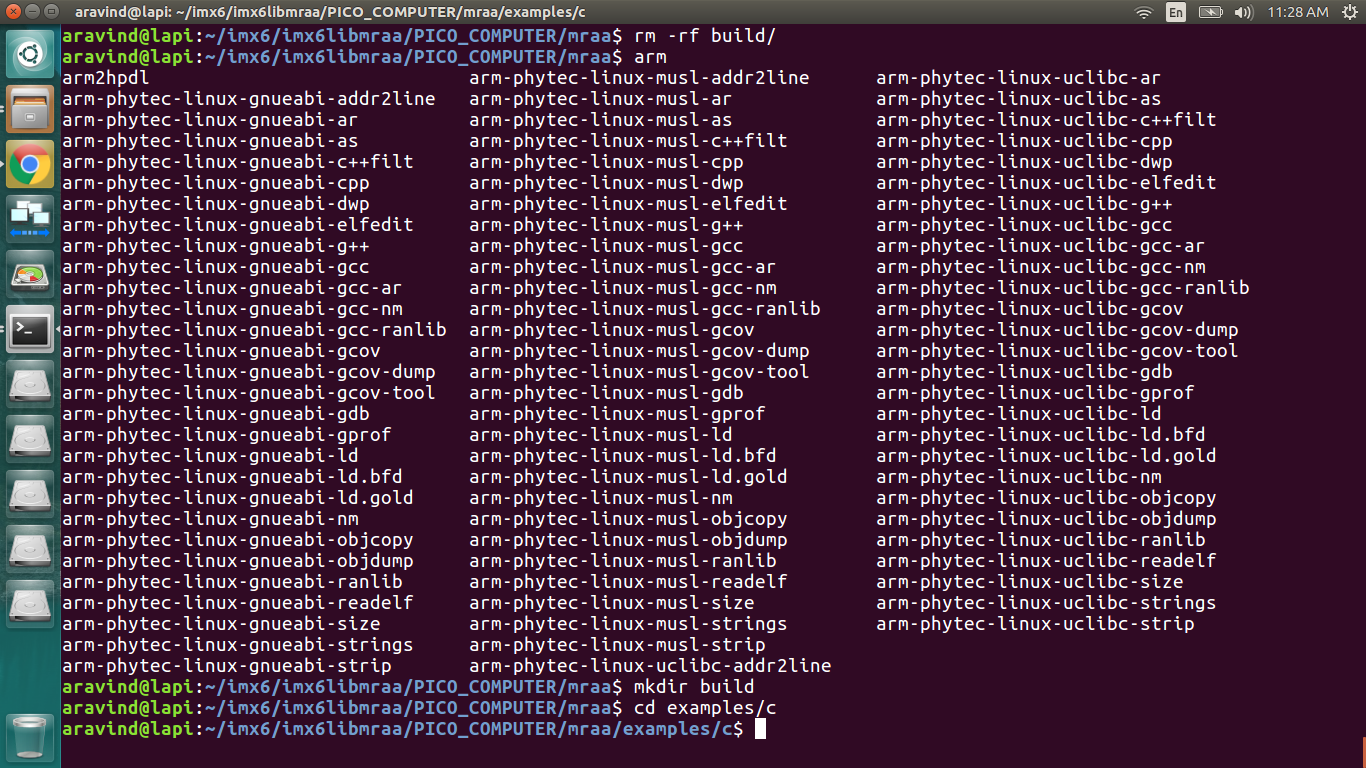


Fig : Creating a Build directory

**Step 11:** Type **ls** in c directory you will find all the C files (in the figure given below you can see various C files with **.c extension)** also one txt file with the name of **CMakeList.txt** will be present as you can see in the figure given below. Open that file with any editor in my case I have used **vi CMakeLists.txt** for opening the txt file.

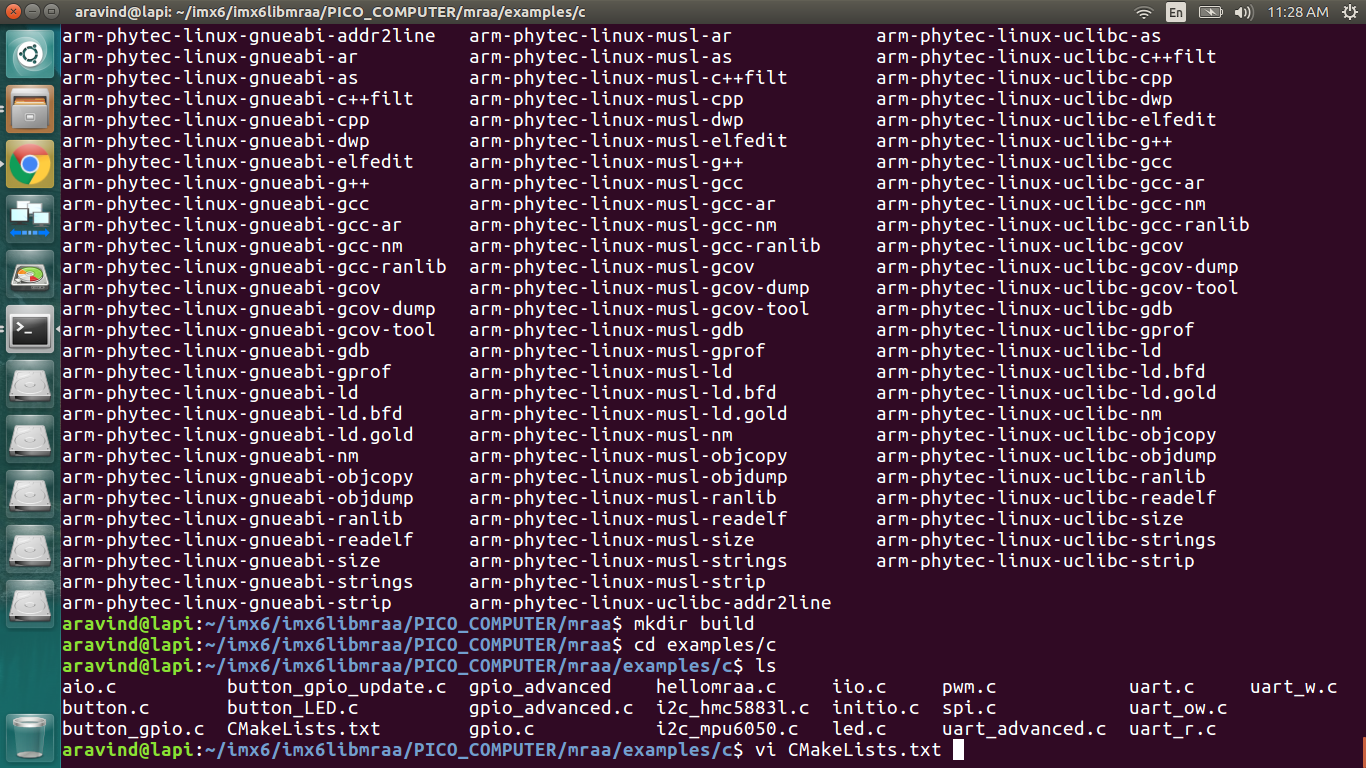


Fig: CMakeLists.txt

**Step 12**: Once you open the CMakeLists.txt you will find **add\_executable(mine mine.c)** this is the file which I have added with other files similarly add it to **target\_link\_libraries(mine mraa)**. For adding in vi type **i** to insert anything into the editor, Please check the figure given below the Note for getting a clear picture of how to add both the above things into the CMakeLists.txt

Note: **mine(name of the file)** is just an example you can give any name to your file as per your requirement. Also don’t forget to save it once you have done the above steps, for saving you have to type **ESC** key and then **:wq!**

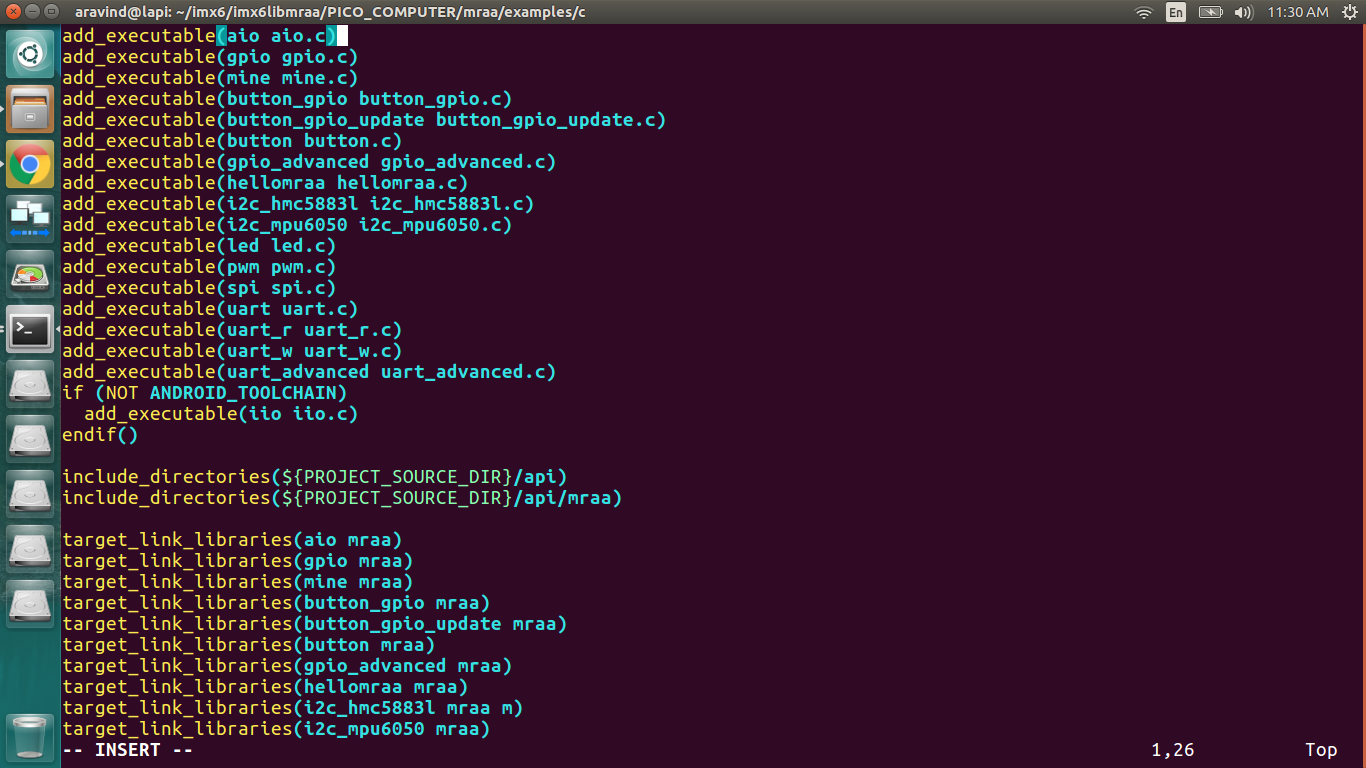


Fig: CMakeLists.txt

**Step 13**: Once you have saved your c file into the CMakeLists.txt navigate back to the build directory as shown in the figure below and give the command **cmake ..**

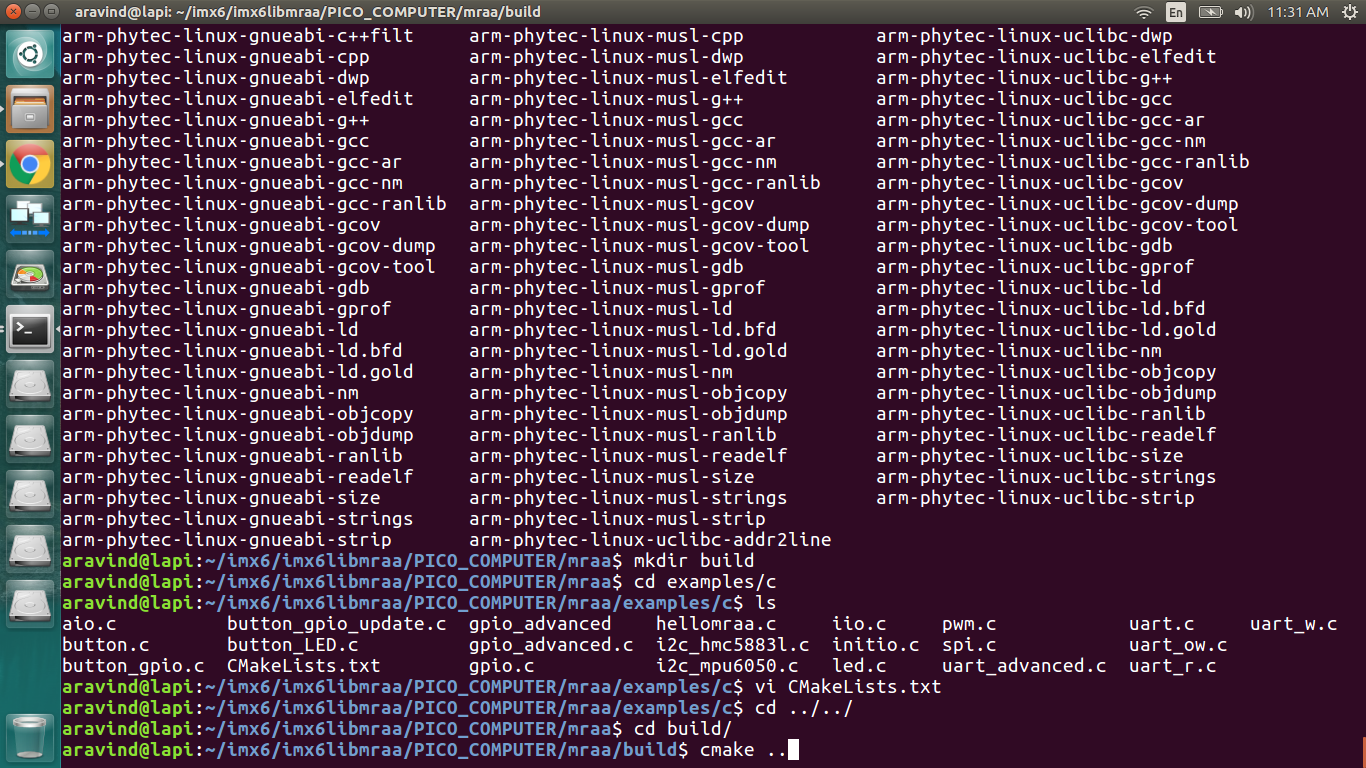


Fig : cmake.. command

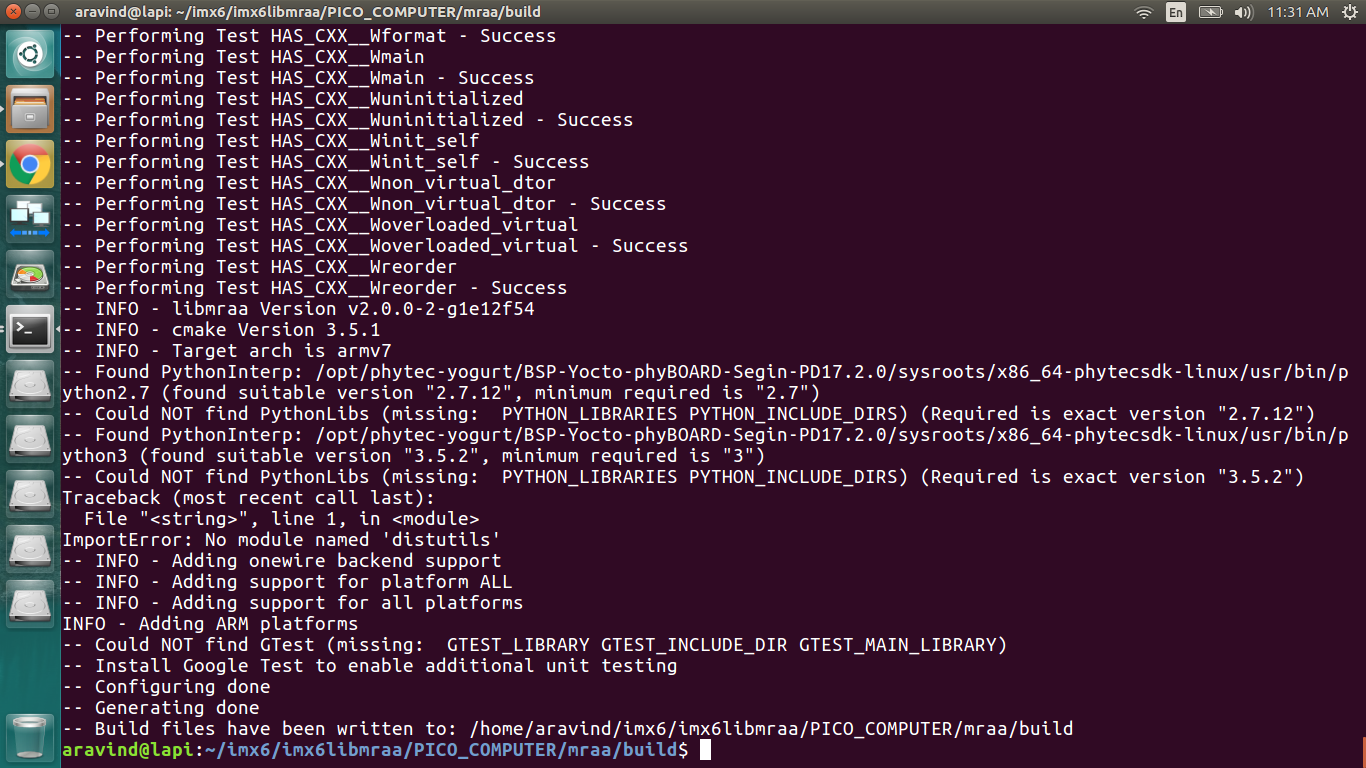


Fig : execution of cmake ..

**Step 14**: after the execution of **camke ..** once again navigate to build directory and type **make**

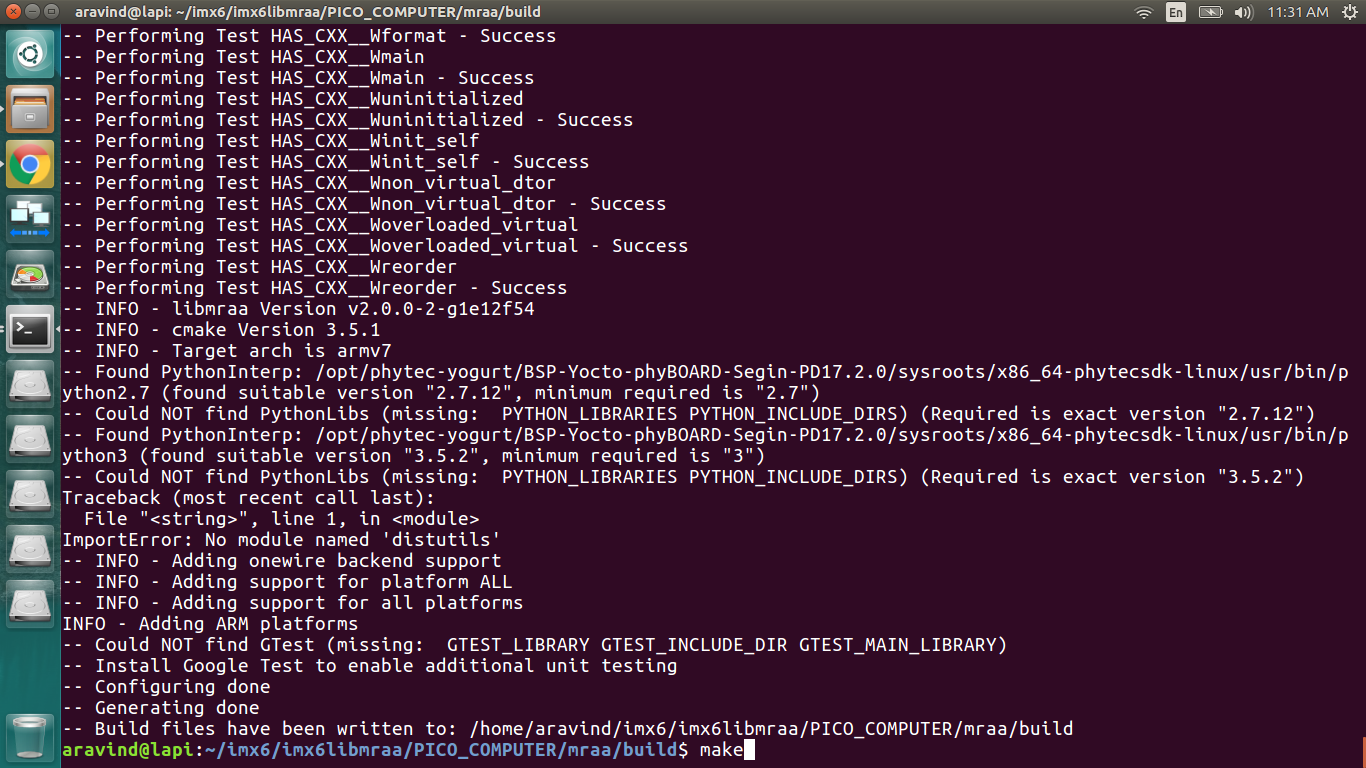


Fig : make command

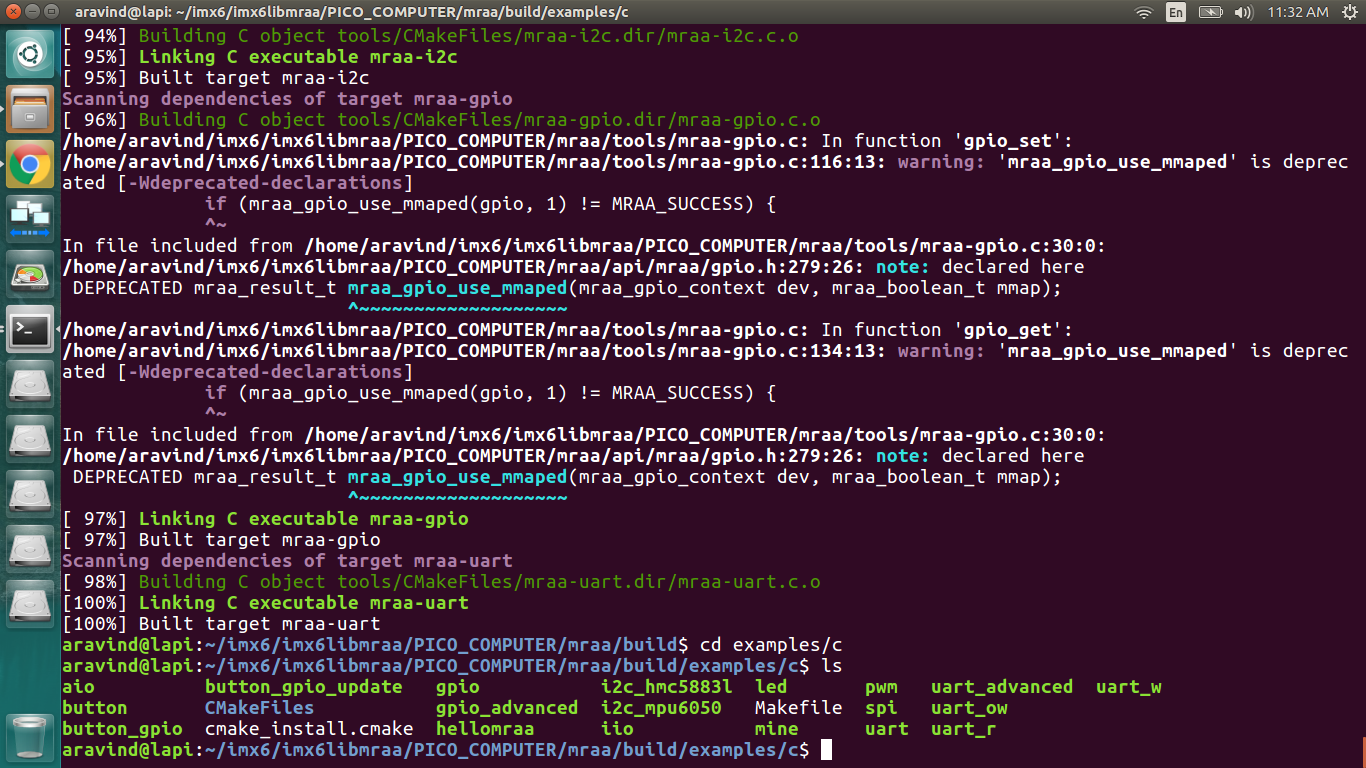


Fig : execution of make command

**Step 15:** after the execution of **make** command you will find the binary file (**mine**) inside examples directory under c. to check you can navigate to c directory and hit the command **ls** as shown in the figure above you will find your binary file(**mine** is the file name) present.

**Step 16:** Copy the file **mine** to/var/lib/tftpboot**.** Type incp mine /var/lib/tftpboot to copy the binary file.

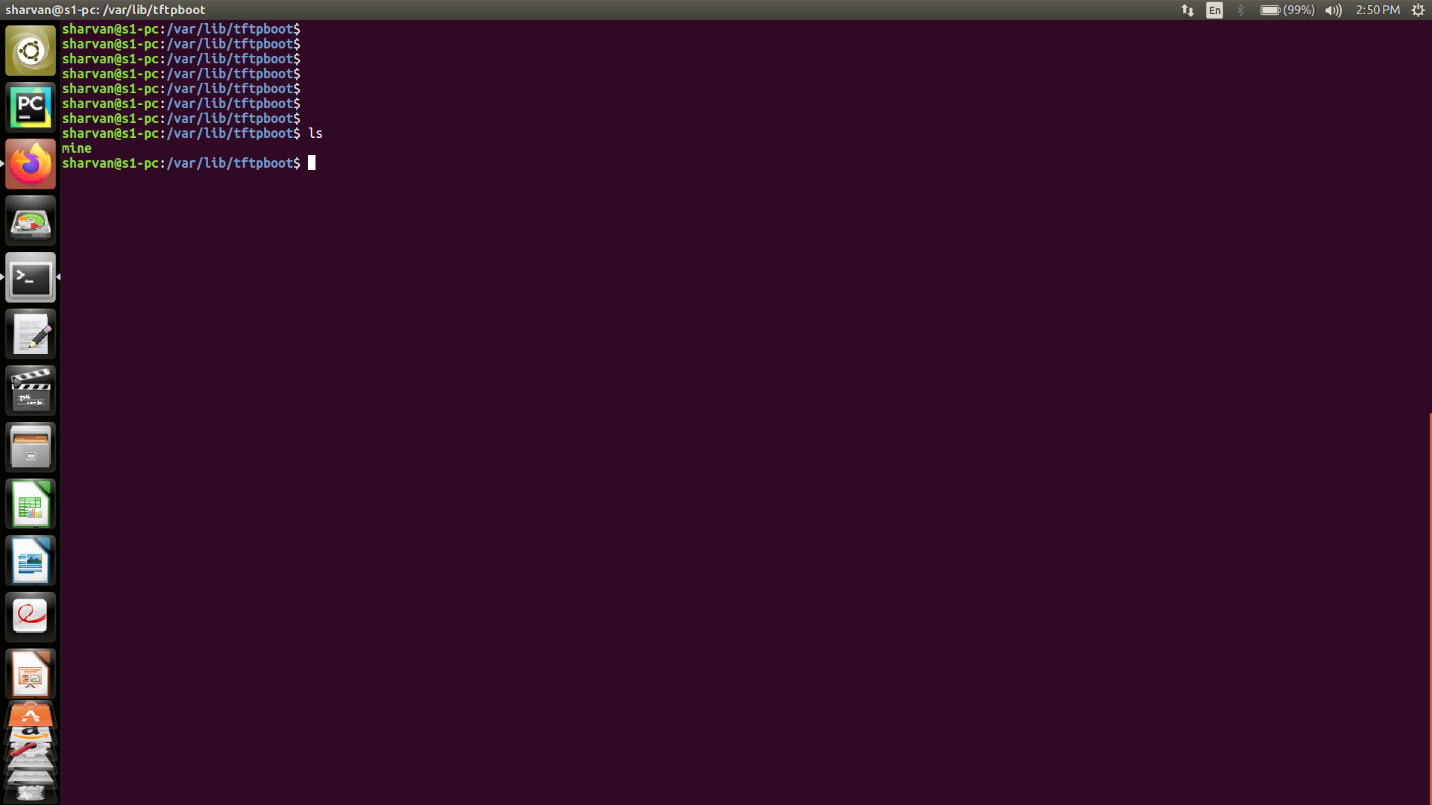


Fig : mine(binary file copied to var/lib/tftpboot)

**Step 17:** as you can see the file once you type **ls** command, it is present in /var/lib/tftpboot as shown in the figure above.

**Step 18:** Now connect your Rugged-Board with your pc and open sudo minicom and follow the steps which you have done in host setup, wait till the board boots till RFS(root file system), then type root as login.

**Step 19:** Also connect an ethernet between your board and PC, note down your board’s ip address by typing **ifconfig** command, you will see eth0 having ip address in this case it is the boards ip address as you can see in the figure given below which is **192.168.1.11 .**

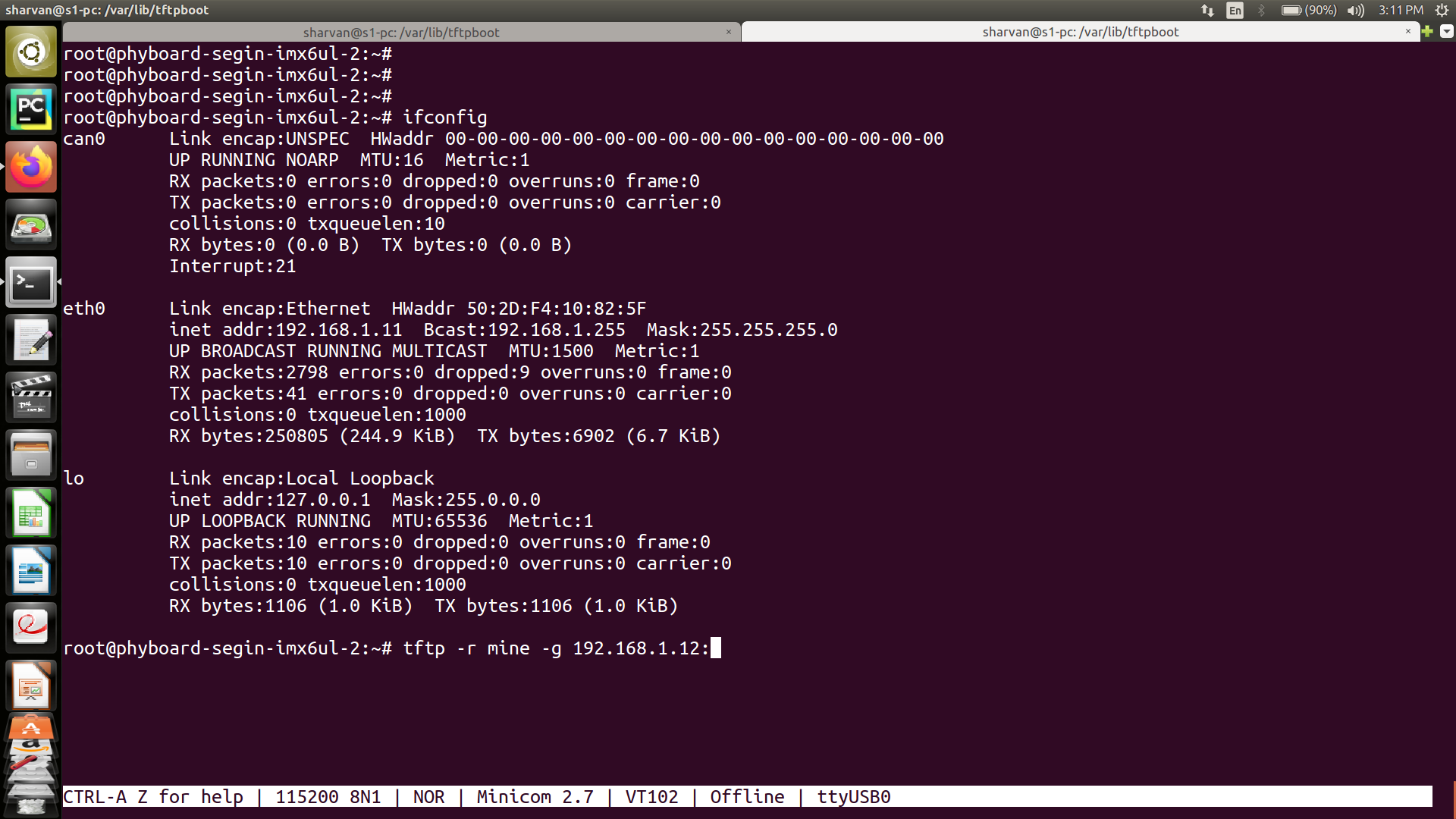
****

Fig : ip address of the board & using tftp

**Step 20:** Now you can type the following command which is tftp -r mine -g 192.168.1.12:~ (in this case the ip address **192.168.1.12** is your **host’s (pc ip address)** ip address .

**Step 21**: Now you can open your Board terminal and type ls the file mine is successfully transferred and can be seen in the root directory in the figure given below.

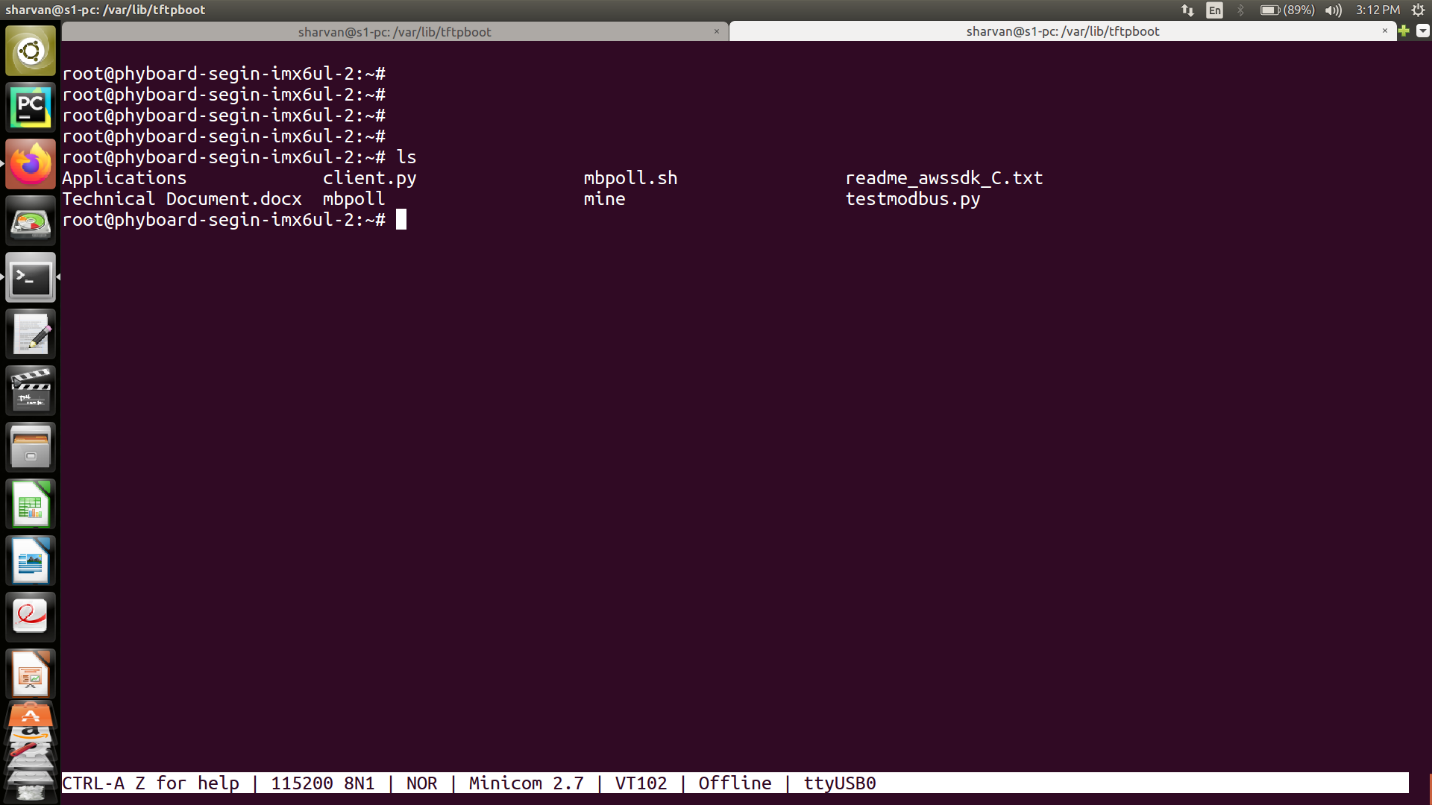
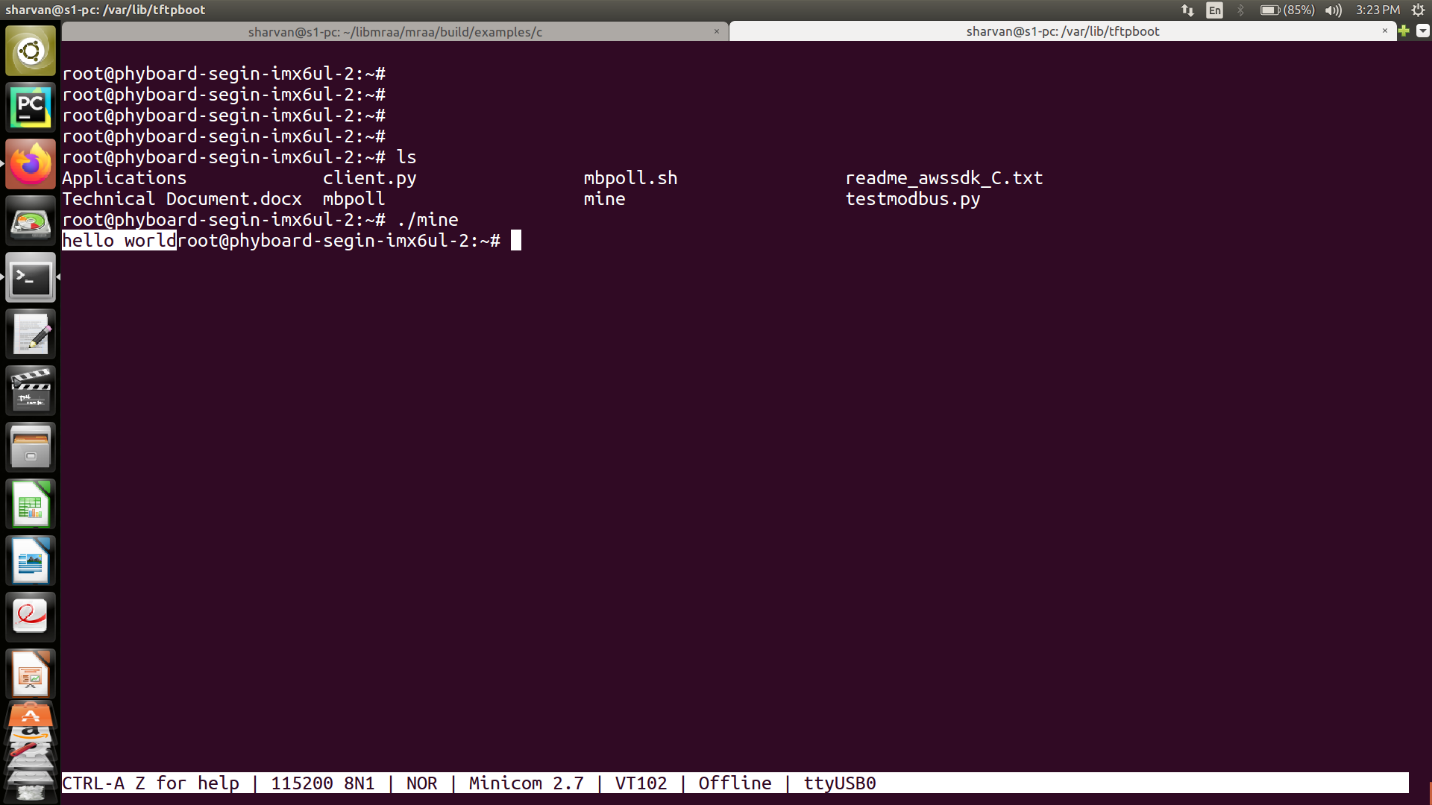
****

Fig **:** mine binary file is successfully transferred to board

**Step 22:** All that is left now is to run the binary file (mine) , for running the binary file you can type **./mine** command as shown in the figure given below.

Fig : execution of file & showing the result of the code

Note: If the file is not executing and giving you permission denied error you can change the permission by typing the command **chmod +x mine.**

**------------------Congratulations on running your first C program on the Board!!--------------------**