EMBEDDED SYSTEMS I

INTRODUCTION TUTORIAL

STUDENT:
Pablo Vinícios da Silva Araújo

QUIXADÁ - CE

WARNING

This tutorial was made by just one computer engineering student, so it has never been verified by any professor or graduate computer engineer. Before you start, make sure that the tutorial may have flaws.

Welcome -

1 - Cross-Compiler installation

So before we start, I'll talk about some necessary information. Before you start configuring your environment, there are a few things you need to know. This tutorial will be more focused on Linux distributions and the board that we will be focused on is the BeagleBone Black. There are some examples on the internet with other operating systems and other cards, but they are few.

By the way, make sure your initial Linux configuration is right, when you are sure, start the following pass. The first thing we need to know is if your OS is a x86_64-linux or an aarch64-linux, you need to know about that before beginning your cross-compiler download.

Installing the cross-compiler. A cross-compiler is nothing more than a compiler for a different platform from the computer used for development. For example, a computer

with the x86 architecture using the cross-compiler ARM Cortex-A8 platform. In this tutorial we will be using the arm-none-eabi, which is the specific cross-compiler for hardware, and which has bare metal programming features.

Next stage, go to the webpage Downloads | GNU Arm Embedded Toolchain Downloads – Arm Developer, there are some ARM GNU toolchains for you to make a download. In most cases, the user will make the download of the x86_64-linux so, let's continue.

First thing is access the Linux terminal and write a command to make a directory called ‘lab’ and, after this, enter it.

```
$ mkdir lab
$ cd lab/
```

Inside this directory, make another directory, and this time, call him “toolchain”, this directory will be the place where we will extract the cross-compiler download.

```
$ mkdir toolchain
$ cd toolchain/
$tar jxvf gcc-arm-none-eabi-10.3-2021.10-x86_64-linux.tar.bz2 -C toolchain/
```

Errors in this part : make sure about the cross-compiler version, the code line don't will be executed if the name of the archive has been changed.

Other one, check the path of your directories. After -C in the code line, make sure about the path, if you put it the wrong path, Linux will not find the archive.

You did it ?, let 's go.

Now we have to go open the “.bashrc”, this document is responsible for saving the Linux environment variables. Open a new terminal and put this.

```
$ gedit .bashrc
```

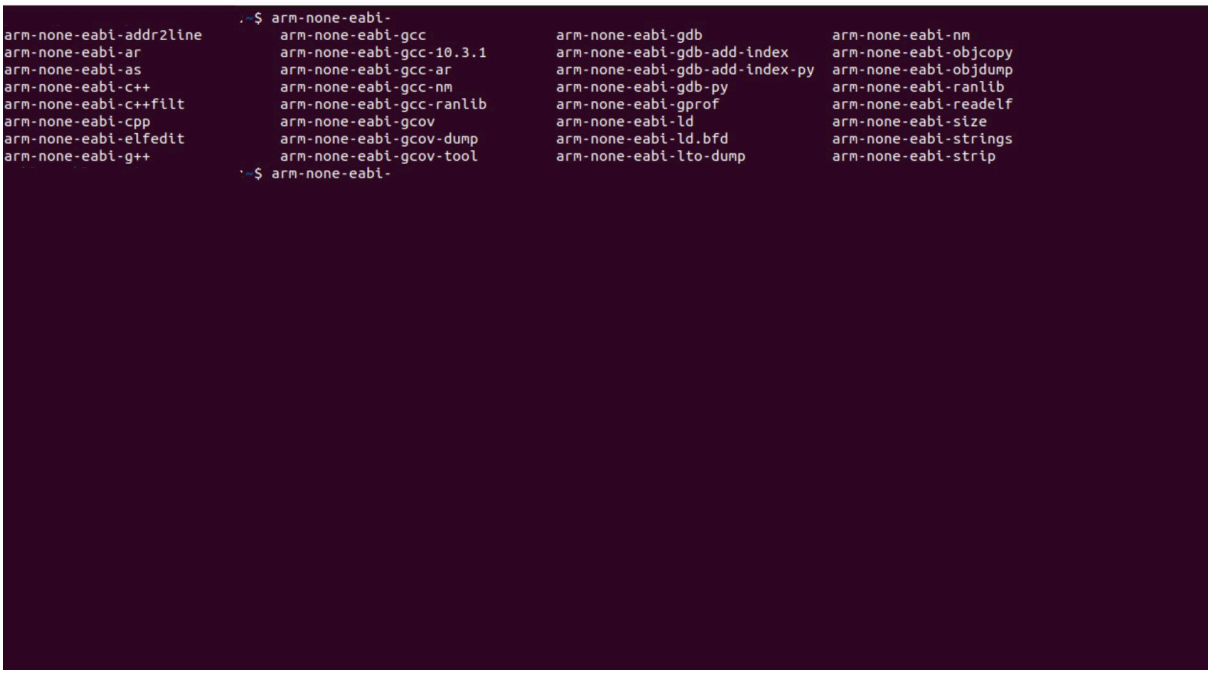
A window should appear for you, this window is the document .bashrc. Roll down the scroll mouse, in the end, put this line code and save the changes.

```
$ PATH=$PATH:~/lab/toolchain/gcc-arm-none-eabi-10.3-2021.10/bin
```

Right, so now, we are going to make sure about the cross compiler installation. Close your terminal and open him again. Put this code line and press TAB.

```
$ arm-none-eabi-
```

With this list of commands, it will be possible to check whether or not the cross compiler that was downloaded was in fact installed on your machine correctly.
If any type of error appears, try to redo the steps and interpret it according to the error that the Linux terminal shows you.



If your terminal starts to list a lot of cross-compilers, you did it, congratulations.

2 - Setting Environment TFTP

Trivial File Transfer Protocol (TFTP) provides a minimalist way to transfer files. It is generally used as a part of PXE boot or to update configuration or firmware on devices that have limited memory, such as routers,
IP phones and embedded systems as a whole.

So, we are going to start setting up TFTP. First, put this code line in the terminal.

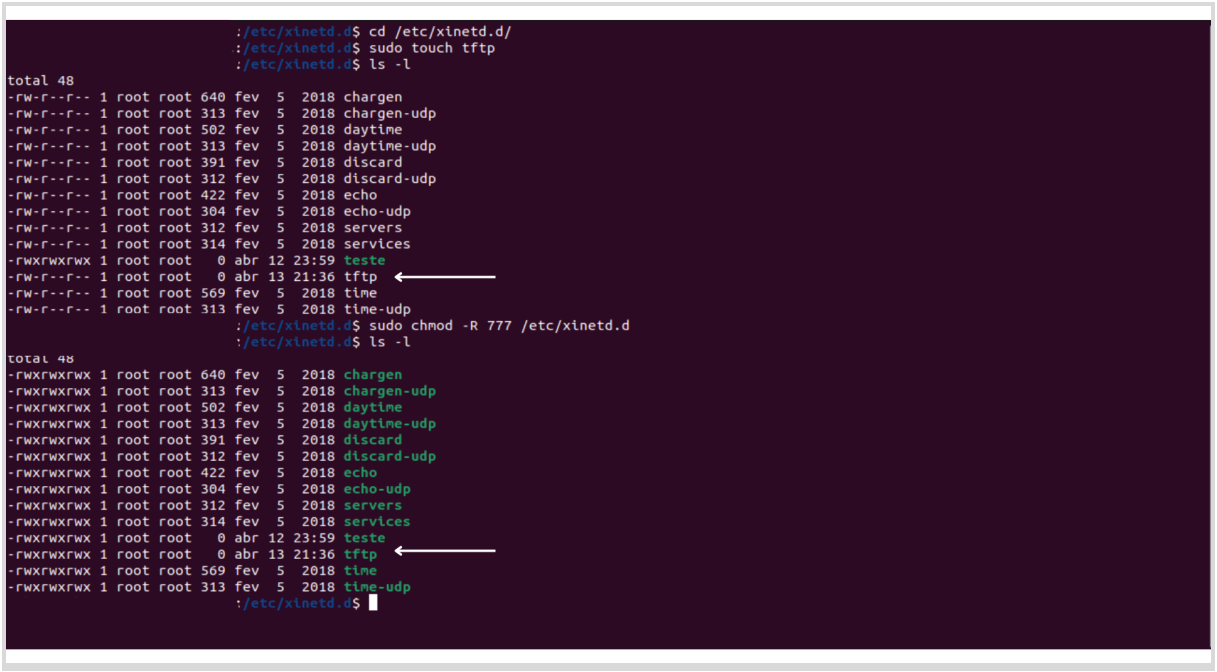
$ sudo apt-get install xinetd tftpd tftp

After that, enter in the directory /etc/xinetd.d. Next stage, we are going to create an archive called “tftp” and we are going to change permissions.

$ cd /etc/xinetd.d/
$ sudo touch tftp
$ ls -l

You will see the permissions in the whole archives.

$ sudo chmod 777 -R /etc/xinetd.d/tftp
$ ls -l



Check the permissions after and before the fourth command, if they change, you did it. Now, we are going to open this document with the following command.

```
$ gedit tftp
```

In the document, put this:

```
service tftp
{
protocol = udp
port = 69
socket_type = dgram
wait = yes
user = nobody
server = /usr/sbin/in.tftpd
server_args = /tftpboot
disable = no
}
```

Save your changes and close the terminal.

Now we are going to create the environment TFTPBoot. Open the Linux terminal and put these code lines.

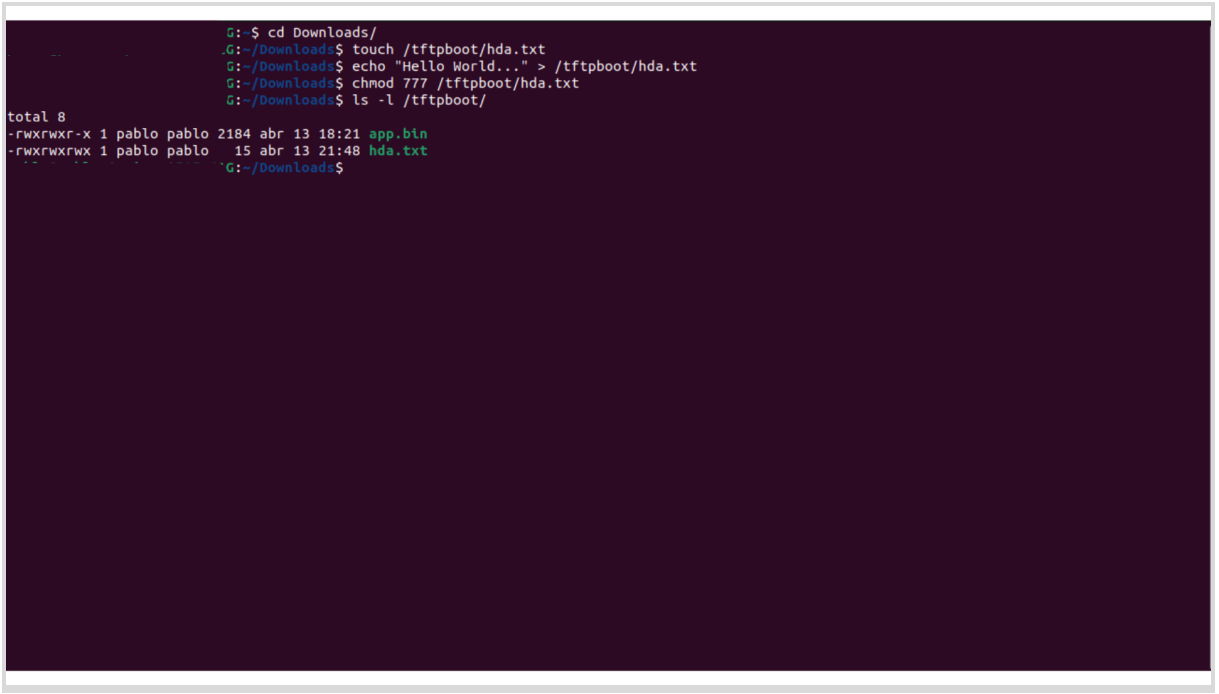
```
$ sudo mkdir /tftpboot
$ sudo chmod -R 777 /tftpboot
$ sudo chown -R nobody /tftpboot
```

Start the tftp through the xinetd.

```
$  sudo /etc/init.d/xinetd start
```

Now, test it. To test, let's do the following.

$ cd Downloads/
$ touch /tftpboot/hda.txt
$ echo "Hello World..." > /tftpboot/hda.txt
$ chmod 777 /tftpboot/hda.txt
$ ls -l /tftpboot/

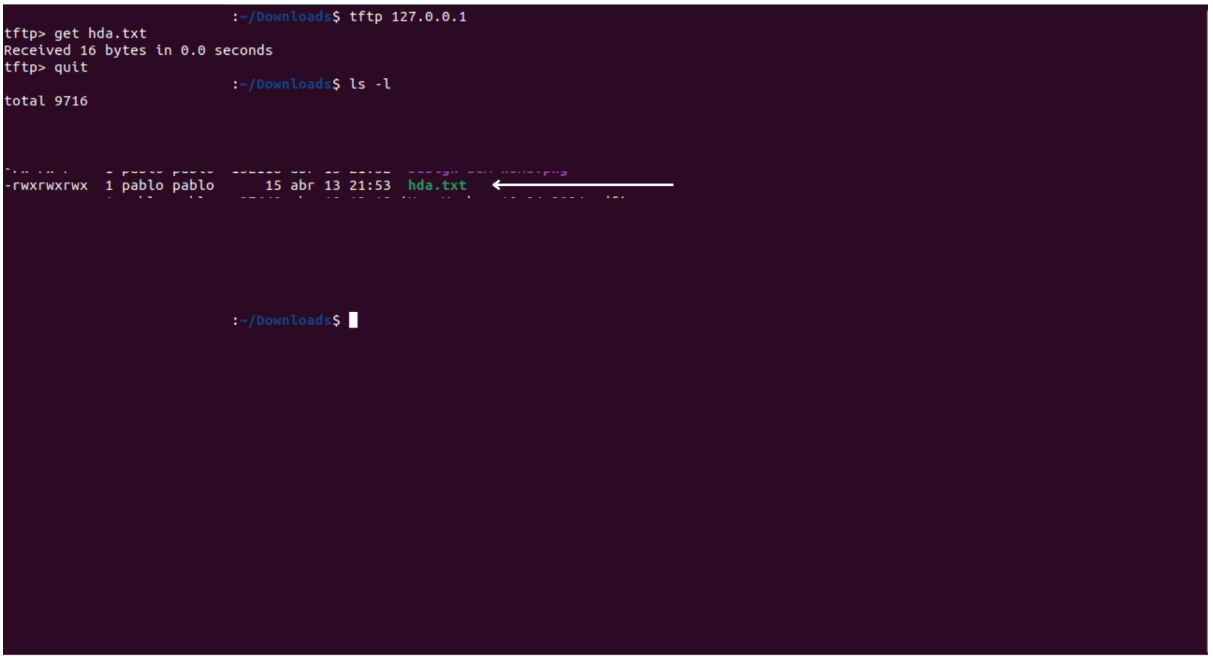


You will see the attributes, size, permissions etc…!
PS: try using sudo before setting the tftp address.

$ tftp 127.0.0.1 or $ sudo tftp 127.0.0.1
$ tftp > get hda.txt
Sent 722 bytes in 0.0 seconds - (aleatory byte size)
$ tftp> quit
$ ls -l

If you see the archive hda.txt in your Downloads folder, congratulations, your environment TFTP is working.

We can go to the next stage of the tutorial.



3 - Setting BeagleBone Black

Before starting, we must install the Minicom.

$ sudo apt install minicom

Now, take your BBB and connect the power cable to your computer, then take the TTL cable (Serial Cable), and plug the USB side in your computer and the other side, connect the pins.

Like this picture :

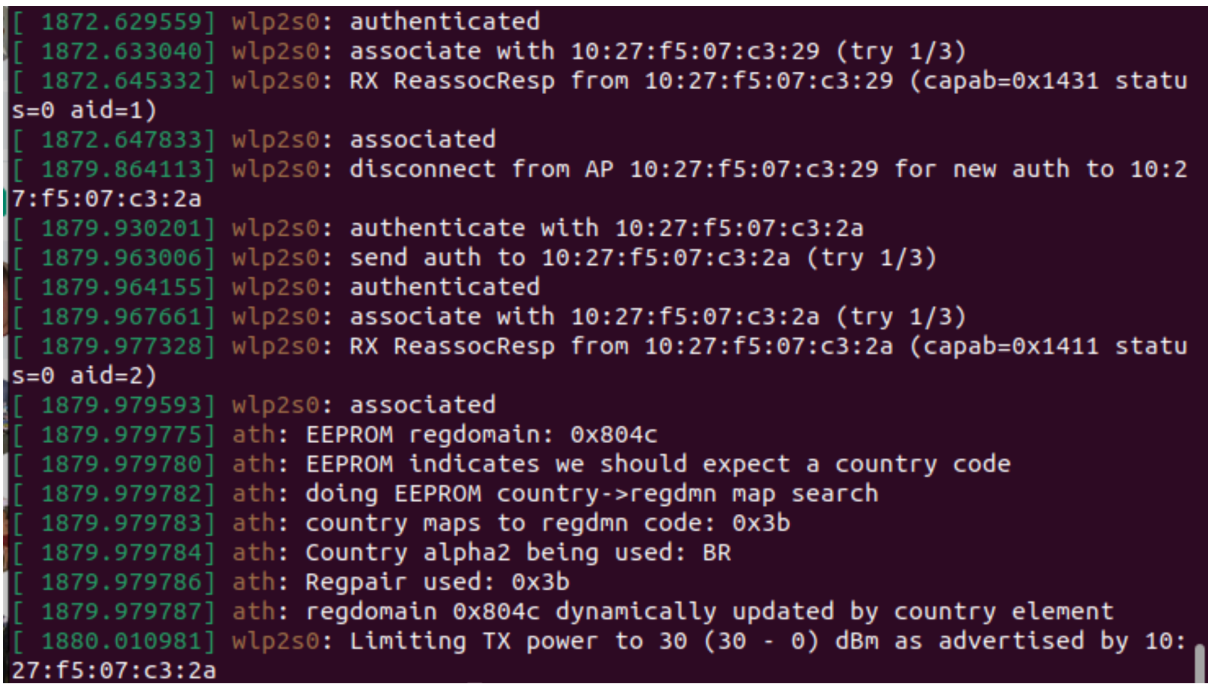
Right to left, put the black, jump two pins, put green and white.

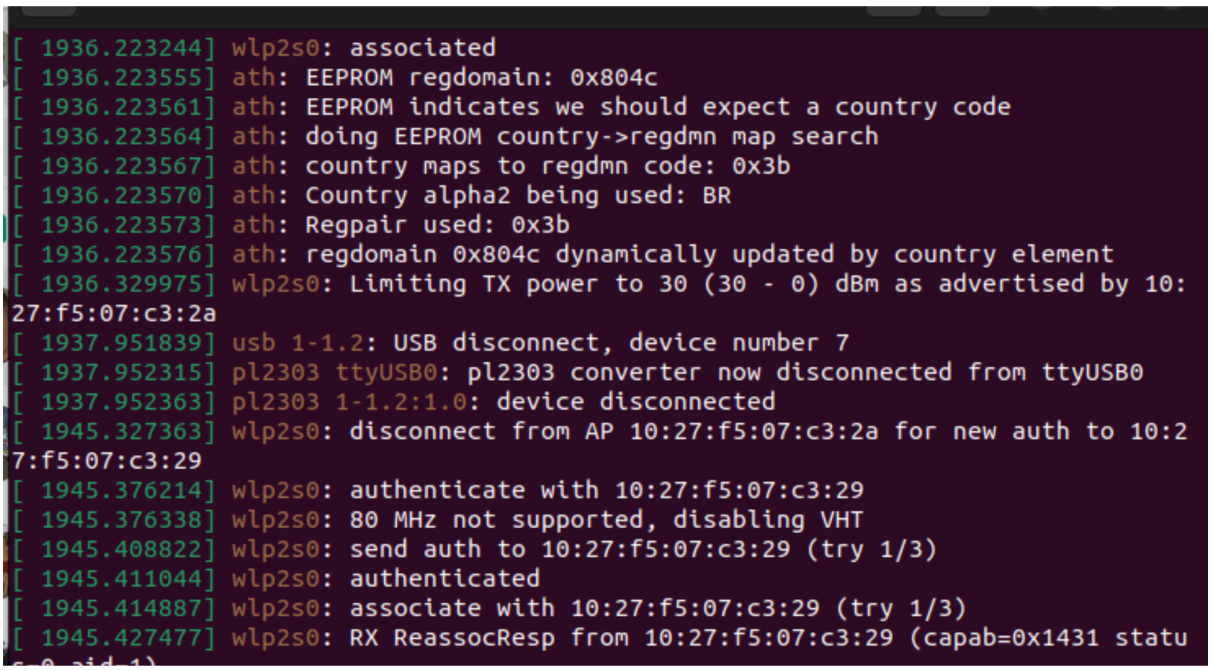
Ok, now we are going to know what our serial cable is.

$ dmesg

Roll the scroll down, and take a printscreen. Remove your TTL and repeat the command above. Compare the prints and find your model cable. Most cases, it will be “pl 2303 ttyUSB0”.

See the pictures bellow:



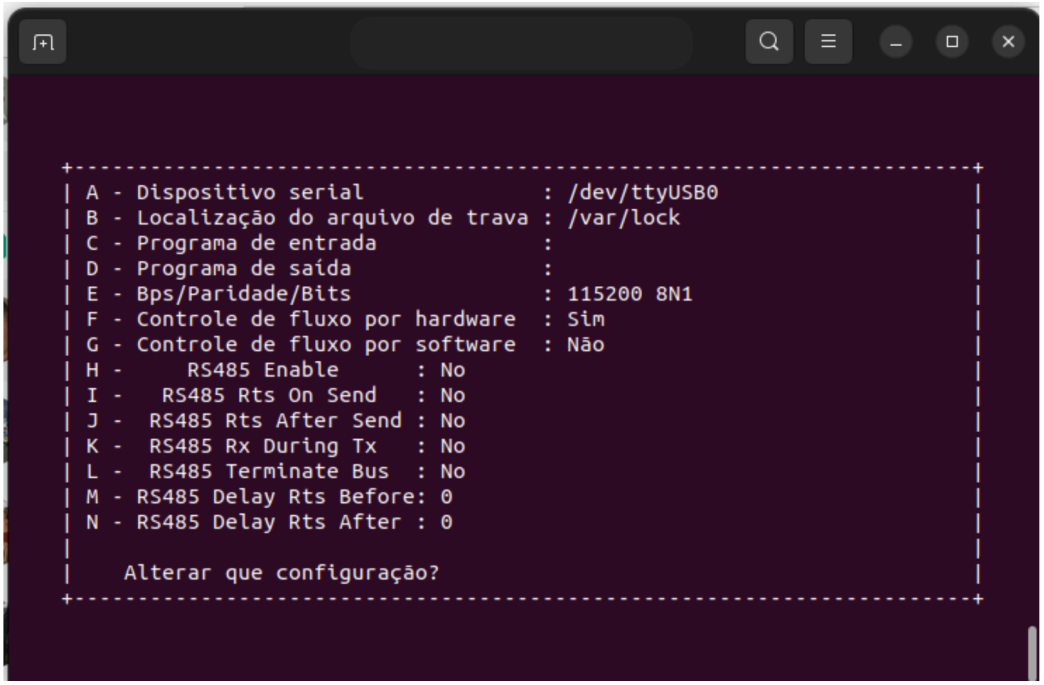


Now, open your minicom.

$ sudo minicom -s

In this window, find “Serial port configuration” and enter. Ok, press A and go to Serial Dispositiv and put /dev/”your TTL” ,

in my case I use /dev/”ttyUSB0”. Back, and save your changes in the minicom.



4 - Setting the Server Communication

Before starting, we need to know what our current network interface is. Put this code line in the Linux terminal.

```
$ ifconfig
```

The terminal will show your interface network, take your netmask and save it.
After that, take an Ethernet cable and connect it to the BBB and to your computer. Go to the Ethernet Configuration and add an new ethernet network, put a name, go to IPV4, select Manual and in Address, put something like this:

Address	Netmask	Gateway
xx.x.x.x	your Netmask	xx.x.x.x

The Address and Gateway should be the same. Save them and close your configurations.
With your new connection created, select it in the network configuration.

Now, put this code line in the terminal Linux.

```
$ sudo minicom
```

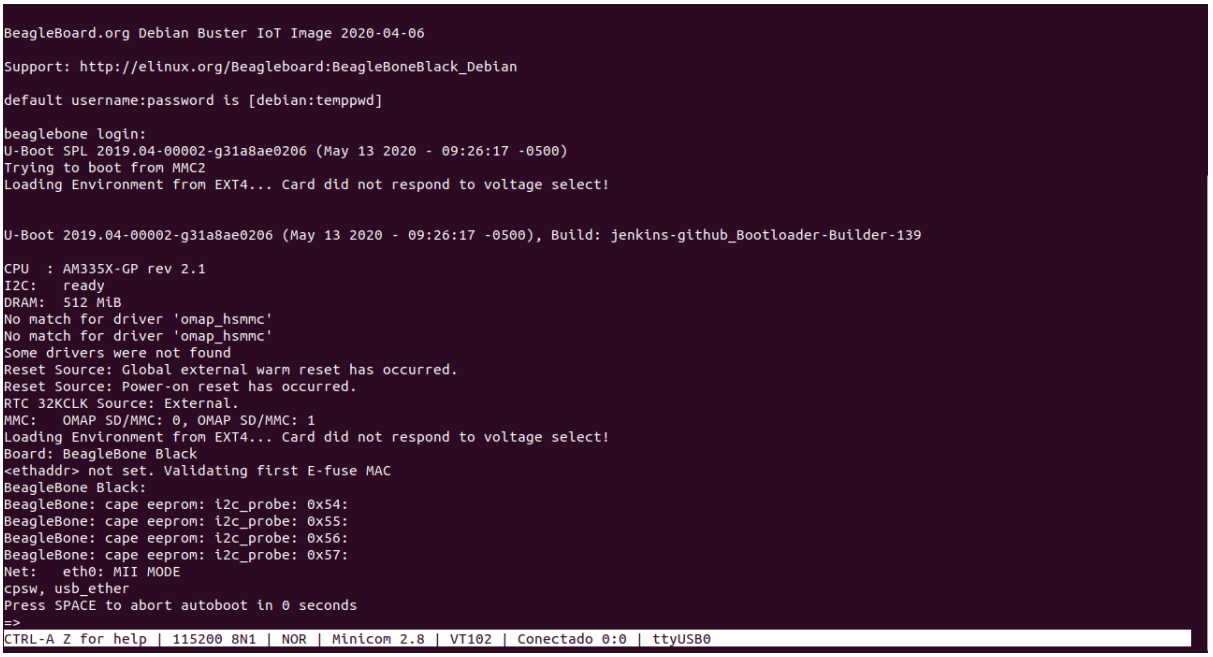
Restart your BBB and press space a few times, if you did, U-Boot will be shown to you. If you don't see U-Boot, repeat the process and restart again.
Congratulations, you stay in the U-boot of BeagleBone.

5 - Practicing

Make a download of this binary document:

🗀 BinaryBBB

Put it in your tftpboot folder. Open the minicom and enter in BeagleBone U-Boot.



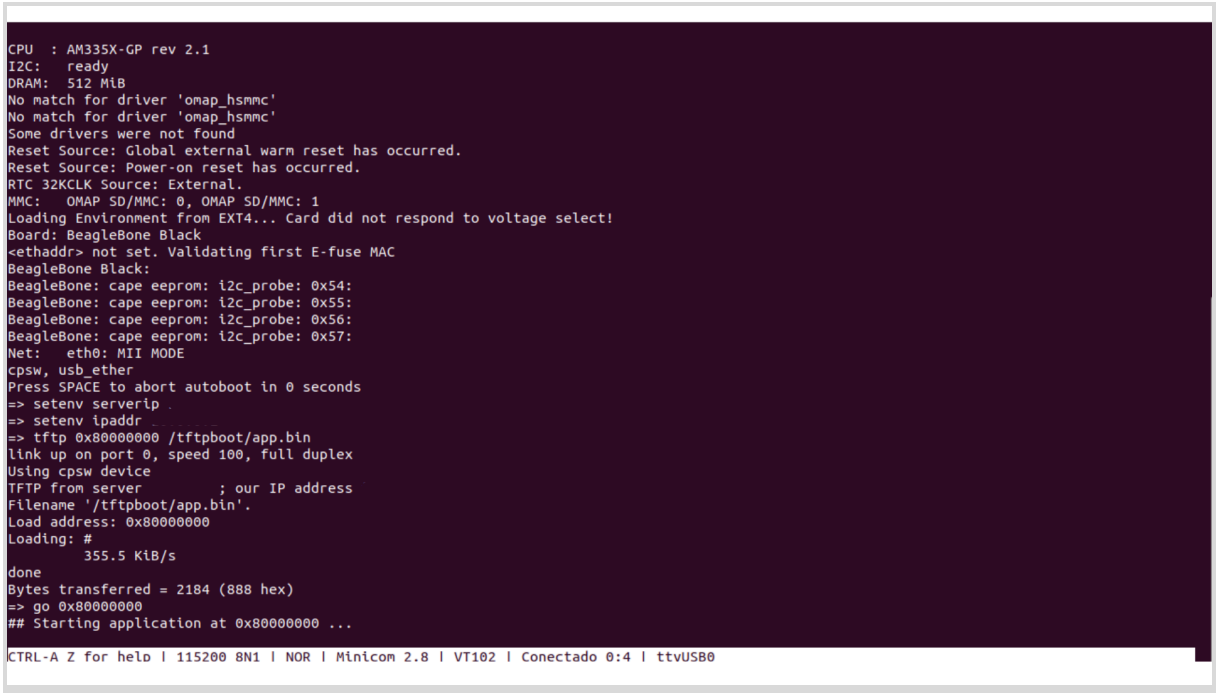
=> setenv serverip “your address”
=> setenv ipaddr “your address” - but, change the last number
=> ping “your address”

If your host “Address” is alive, continue, if not check your connection with the Ethernet cable.

=> tftp 0x80000000 /tftpboot/app.bin
=> go 0x80000000

Your BeagleBone will start to show the binary numbers 0-15 in the board leds, Congratulations, you made this.

Possible errors: you entered an instruction name incorrectly, made a mistake when placing the addresses or placing the wrong binary path that you want to pass to the board.



End.

For any questions, send an email to me:

pablovsa28@outlook.com

Thanks for reading this, bye bye.