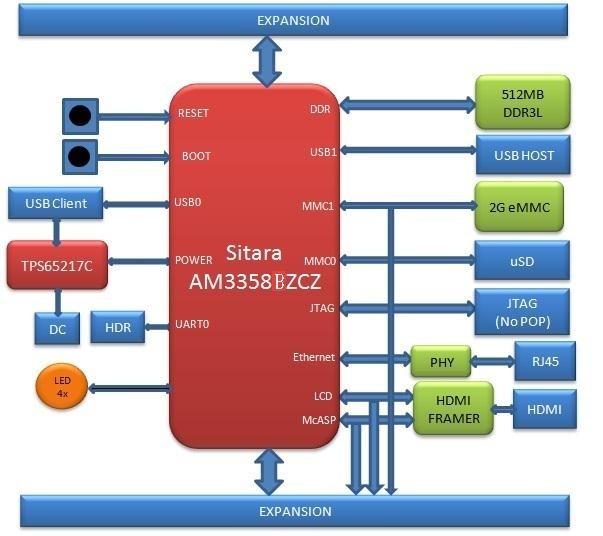
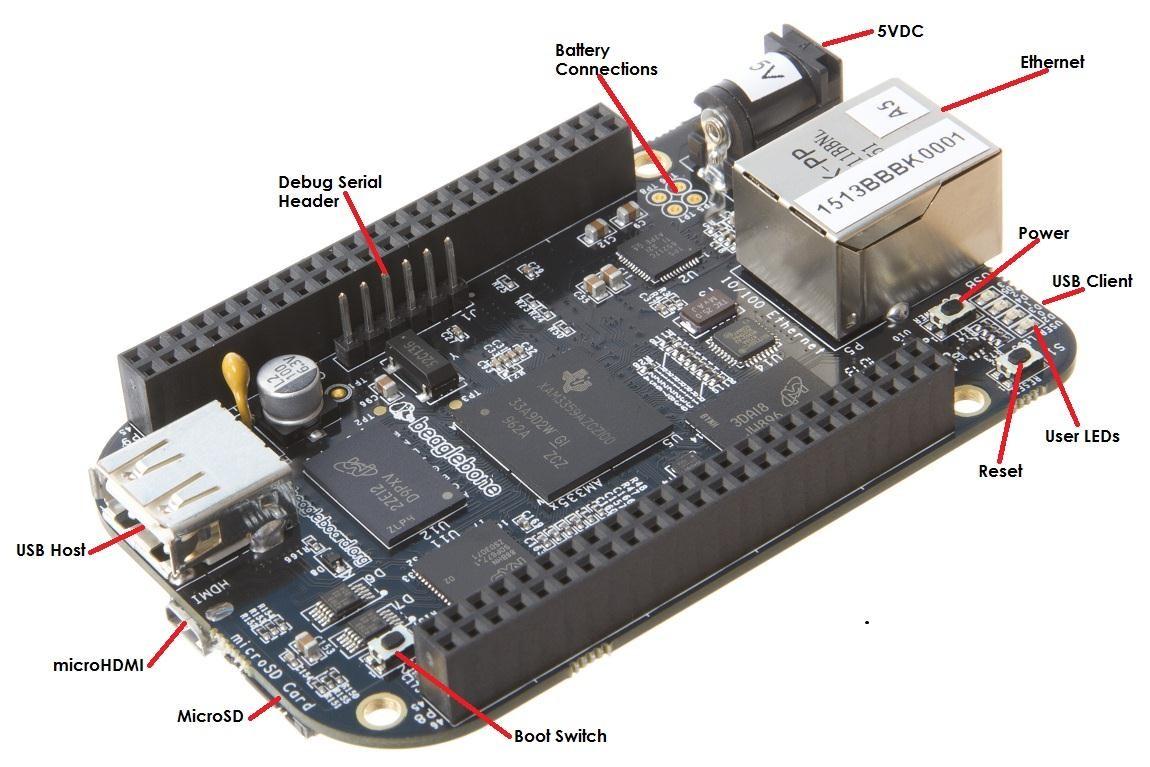
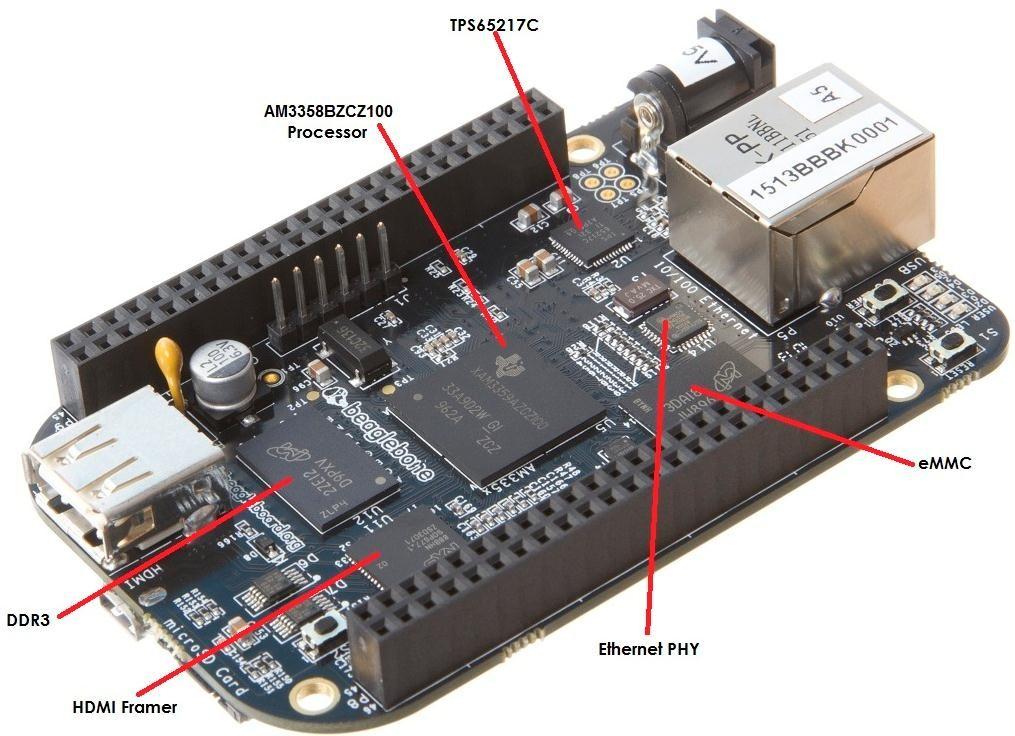
# BeagleBone Diario de Bordo

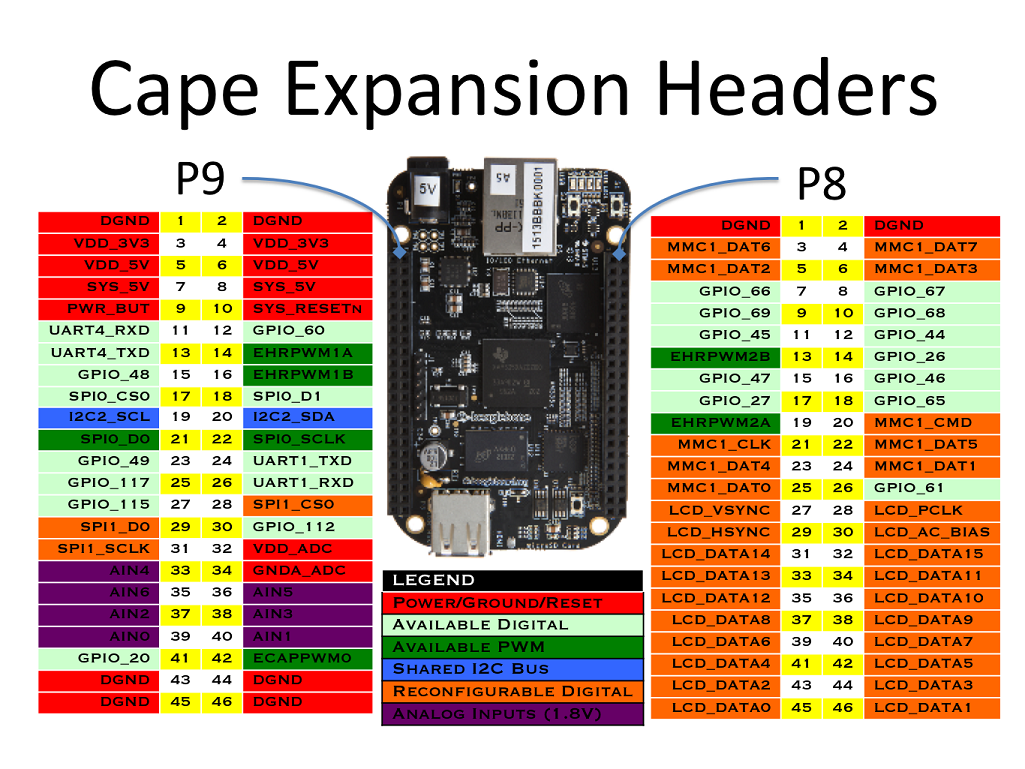
Objetivos: Primeiros passos na Beaglebone Black (BBB) e no código freemodbus.

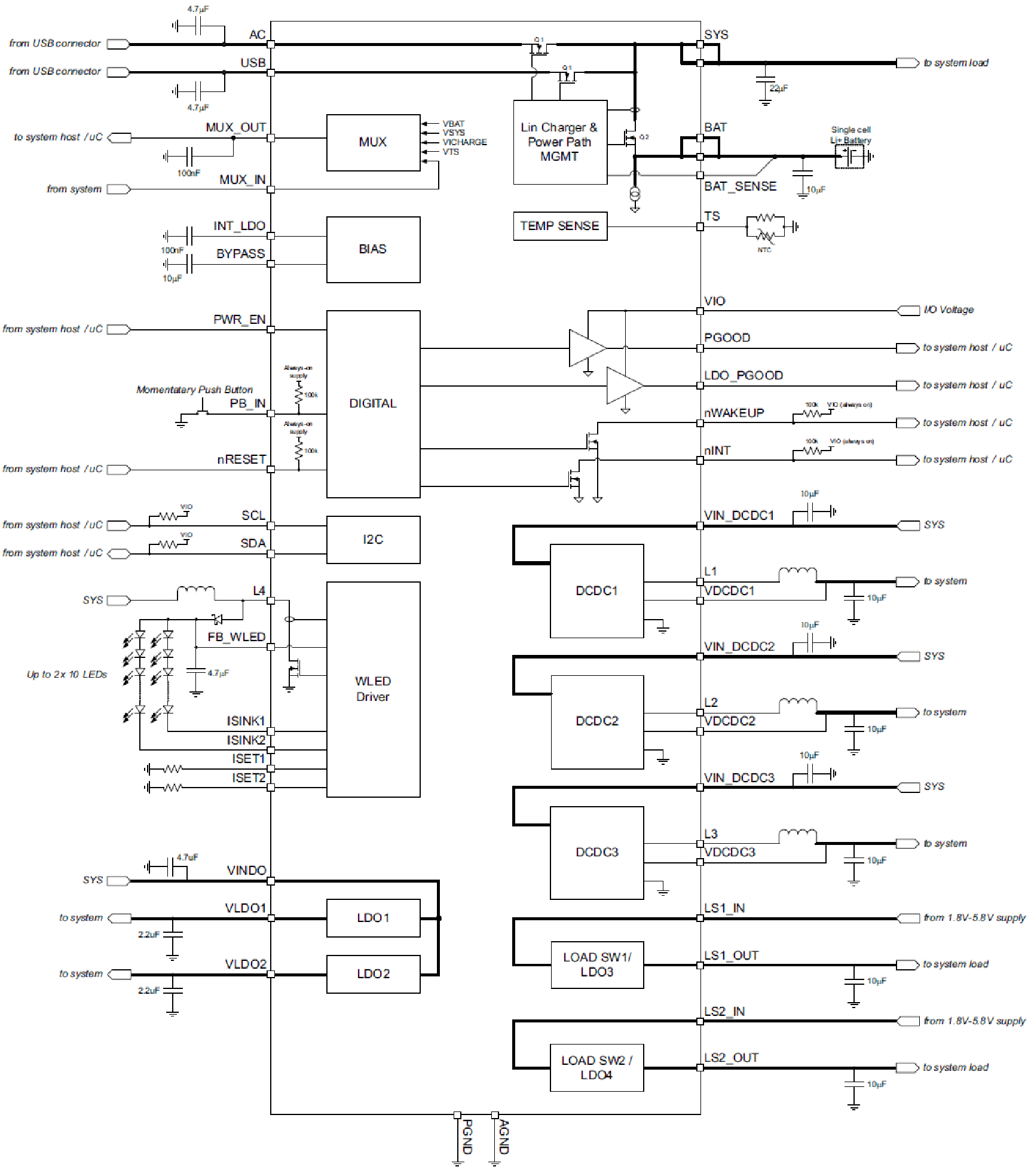
# Pinagem da BBB:











# Maquina Virtual Linux

O usuário e a senha da máquina virtual Ubuntu do fabio é :

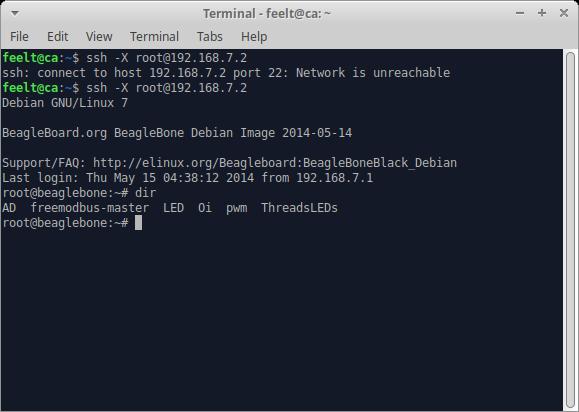
User: UFU

Senha : 1

## Conectando a BBB pelo cabo serial

quando configurado somente o cabo serial:

ufu@SE:~$> ssh -X [root@192.168.7.2](about:blank)



Aqui acabou que eu não precisei logar, mas tem vezes que precise logar. (descobrir a senha da BBB)

## Transferindo arquivos através do Filezilla

Ainda dentro da máquina virtual eu conectei também ao filezilla. Este programa servirá para descarregar o programa diretamente na BBB após ter compilado no PC (Linux, neste caso da máquina virtual). Ele serve para copiar toda a pasta ou conteúdo para dentro da beaglebone, ou fazer o contrário, ler algo da beaglebone para o computador.

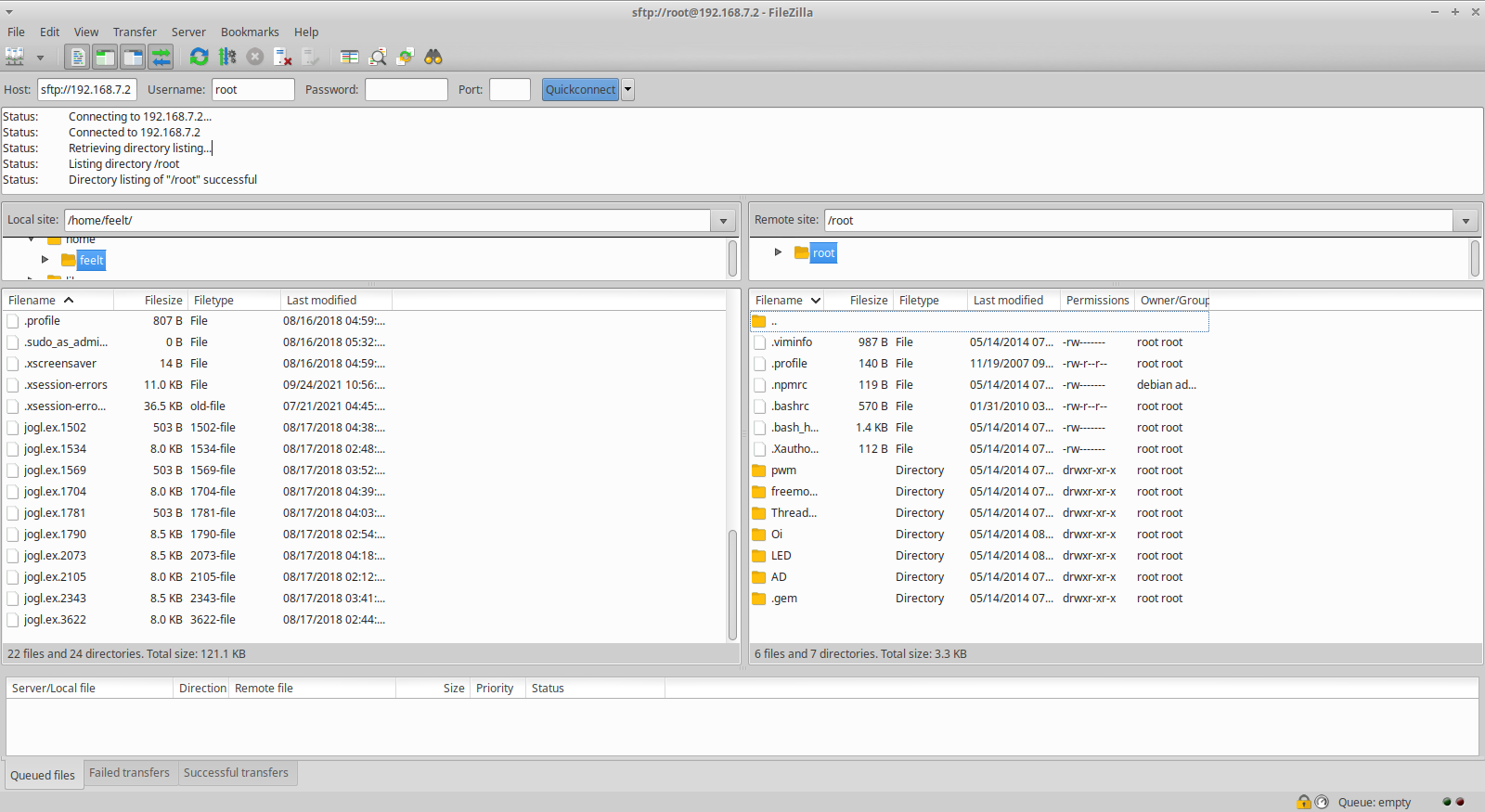
Primeiramente, coloquei o cabo USB e loguei na bbb conforme mostrado no item anterior (porta USB\SSH). Para conectar o Filezilla eu fiz a seguinte configuração:

Host : 192.168.7.2 (conexão pelo cabo serial USB)

User : root

Senha: nenhuma

Porta : 22 (que é a porta SSH).



Na Figura acima, do lado direito está o conteúdo da BBB. Do lado esquerdo o conteúdo do PC. O que eu quiser trocar entre os dois é possível, facilmente ou arrastando ou escolhendo o arquivo e fazendo upload ou download.

Dicas: quando eu vou substituir um arquivo já existente na BBB parece que as vezes da problema e até ele desconecta do Filezilla. Ideal é vc primeiro apagar na BBB (e deixar na pasta certa do lado direito), então ir no lado esquerdo (PC) e então selecionar o arquivo e fazer upload. Ele vai copiar para onde está já apontando previamente.

Outra dica eh que as vezes o filezilla, após um determinado tempo com a janela aberta fecha, e meio que trava, ficando sempre tentando reconectar. Ele não vai conseguir reconectar. Deve desligar a conexão filezilla, desconectar a conexão do terminal e tentar novamente. Para evitar isso parece que eh melhor sempre desconectar do filezilla.

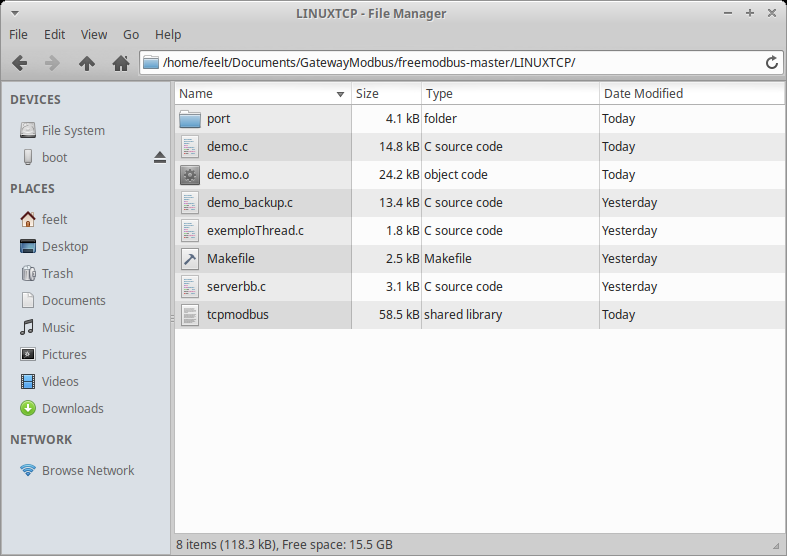
## Compilando o programa no PC

Uma opção que eu fiz foi colocar o mesmo programa na Máquina Virtual (PC) e compilar ele no PC. Para isso precisei instalar o arm-gcc no pcc.

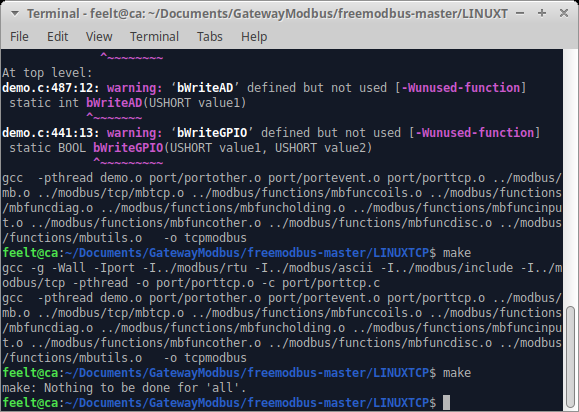
A instalação usei os passos deste link:

<https://www.ti-enxame.com/pt/linux/comando-gcc-arm-linux-gnueabi-nao-encontrado/1069884245/>

Agora o programa free-modbus compila no Linux. Para isso, basta ir na pasta do freemodbus (arquivo makefile) e então pedir para fazer um ***make***.



A janela abaixo mostra um exemplo de compilação.

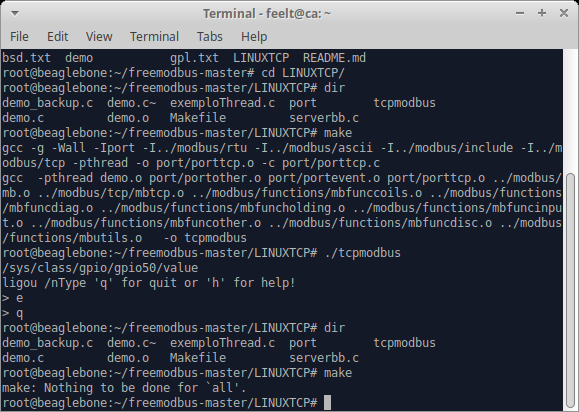


Obs: Ainda não fiz o teste de rodar o programa no PC e copiar o executável somente para a BBB.

O Marcio Cunha falou que eles utilizam para a raspberry o programa VisualCode que é mais fácil de utilizar pois ele compila e atualiza diretamente a placa.

## Compilando o programa na BBB

Após transferir os arquivos para a beaglebone, para compilar o código na própria BBB basta selecionar a pasta e executar o comando make (como foi feito no PC).



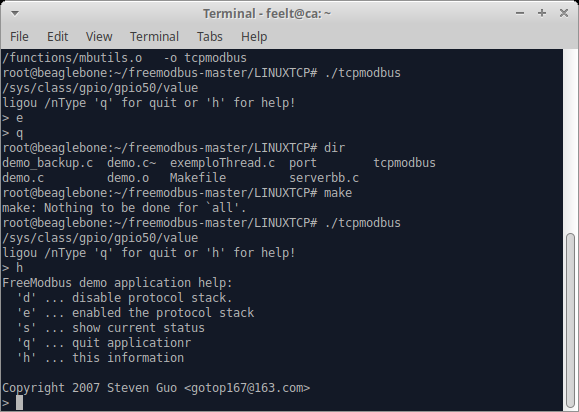
Para compilar > **make**

Para deletar a compilação anterior > **make clean**

**Make all**

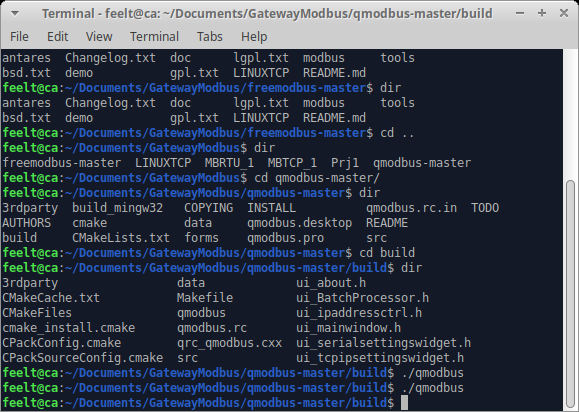
## Rodando o programa Modbus na BBB

Para rodar o programa na BBB basta executar o arquivo executável através do comando “./tcpmodbus”. Depois para rodar a tarefa modbus basta digitar “e” para executar. Existem outras opções que pode ser consultada no comando “h” para help.

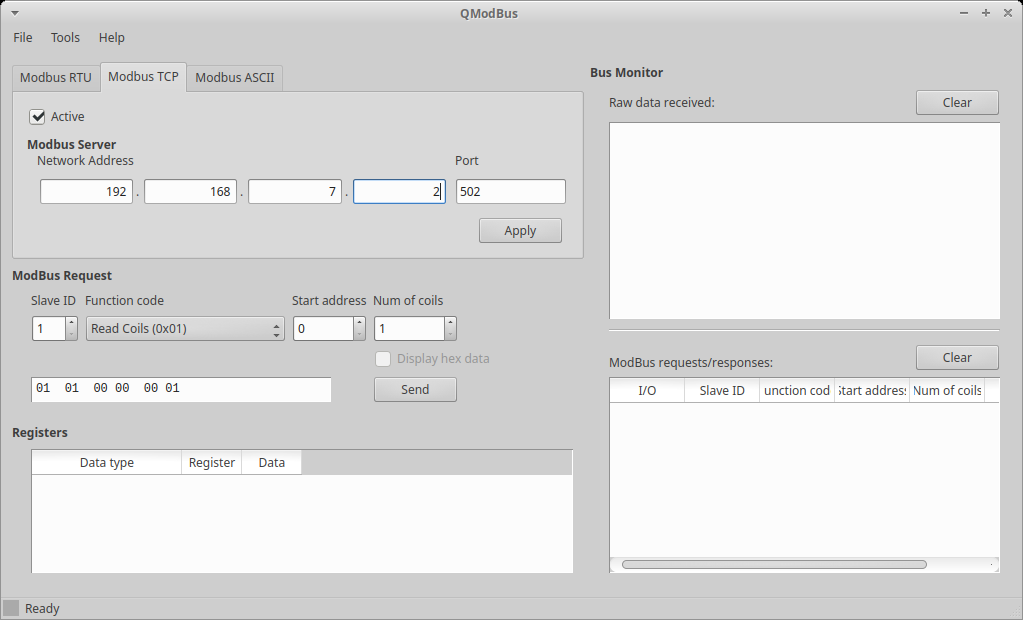


## Usando o programa qmodbus no Linux

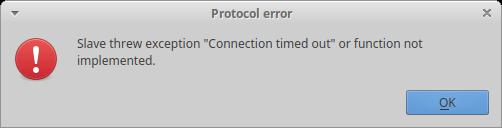
O programa qmodbus é similar ao ModScan. Ele funciona como um mestre modbus TCP ou RTU. Neste caso, para rodar o qmodbus basta entrar na pasta build e executar “./qmodbus”.



O programa eh bem intuitivo onde vc deve ativar “Active” a conexão TCP, digitar o IP selecionar os comandos e dar Apply. Caso os registros aparecerem na parte inferior da tela, significa que a leitura foi OK. Caso contrário ele mostrara uma mensagem de erro do protocolo.

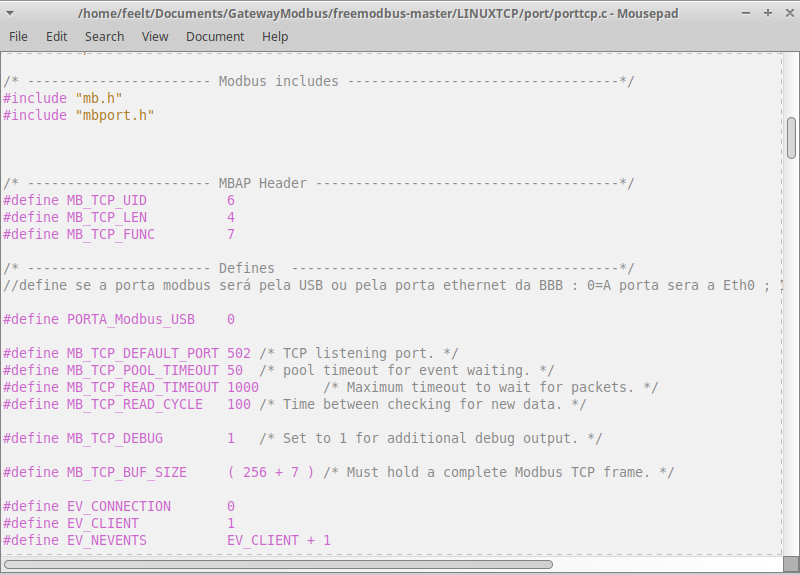


No caso, a conexão USB não estava disponível então ele vai enviar uma mensagem de erro conforme mostrado na figura abaixo



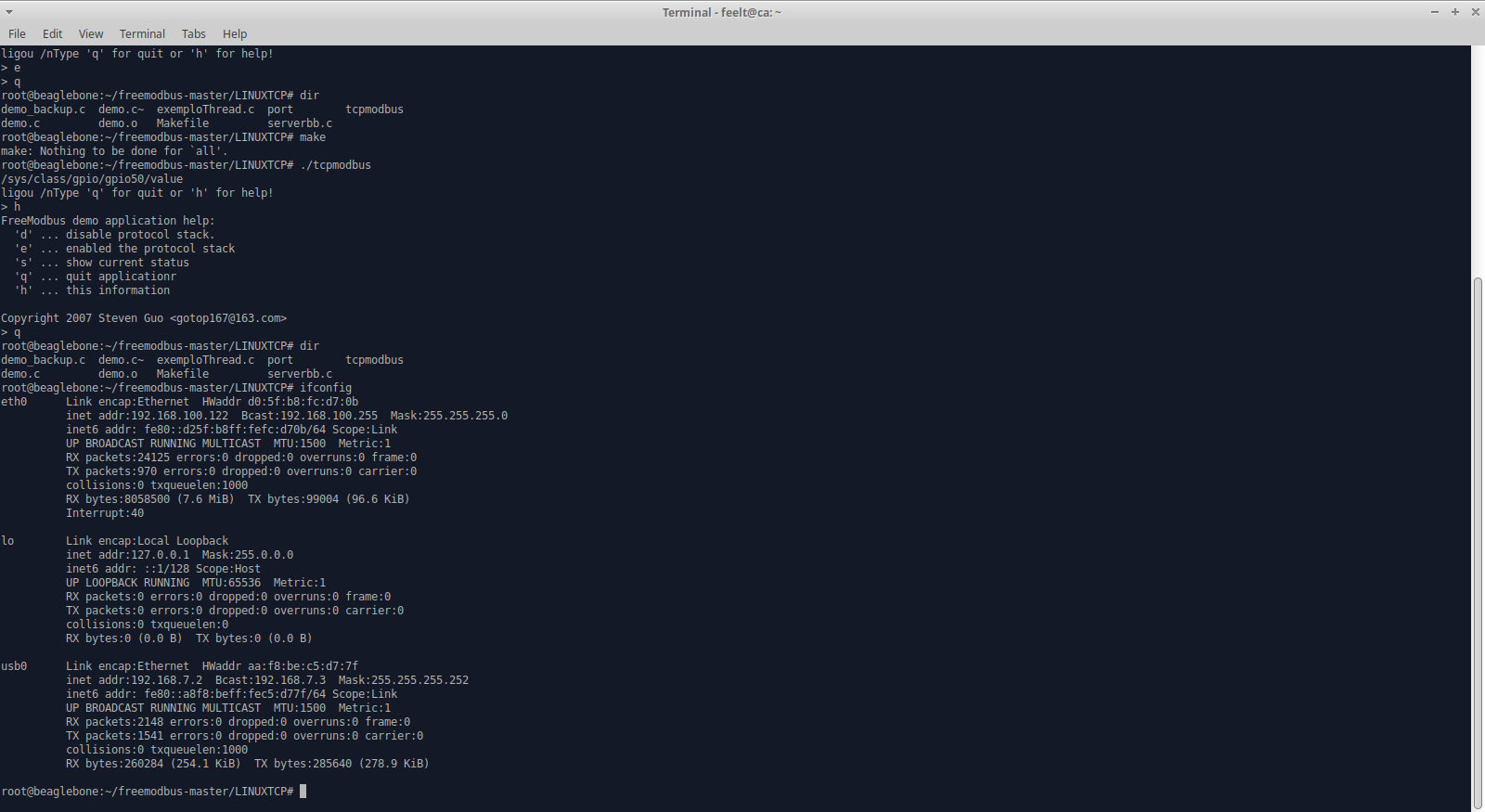
## Chaveando entre cabo ethernet ou porta USB

Dentro da BBB eh possível estabelecer conexão “Modbus TCP” tanto com o cabo serial USB quanto com a porta ethernet da BBB. Isto porque a BBB cria uma conexão TCP na porta USB. Desta forma, dentro do código eh possível escolher entre portas Ethernet ou porta USB. Isto eh feito através do define PORTA\_MODBUS\_USB no arquivo port\porttcp.c conforme mostrado na figura abaixo.



Caso for escolhido porta ethernet 0 para a BBB (PORTA\_Modbus\_USB =0), o código modbus vai responder somente na porta ethernet0. Caso o define for 1, a porta modbus vai ser a mesma da SSH (192.168.7.2 IP fixo).

No caso da ethernet0 é necessário setar o IP para o correspondente subrede do PC, como mostrado na figura abaixo:



O comando ***ifconfig*** no Linux mostra a configuração de todas as portas do device. Neste caso, é mostrado que a ***eth0*** tem ip : 192.168.100.122.

## Instalando o Eclipse Cross Compiler no Linux

O eclipse pode ser uma ferramenta mais fácil para trabalhar com o Linux e a beaglebone black.

(eu testei o Microsoft Visual Code no Windows) e ele não é compatível com a nossa beaglebone. Ele precisa ter um sistema SSH server e suporta a java que é a partir da versão 8 do Linux ou para a versão da raspberry. Desta forma parti para tentar fazer com o eclipse.

Os dados que estão aqui eu peguei do seguinte vídeo no youtube:

<https://www.youtube.com/watch?v=_SBCES64gYM> (install eclipse no linux)

<https://www.youtube.com/watch?v=qVNow-C2UZ8&t=622s> (cross compile beaglebone)

verificar java no linux :

> javac -version

Instalar java jdk no Linux (caso o comando acima der errado):

> sudo apt-get install default-jdk

Instalar o eclipse:

# Comunicando BBB com a internet

Para conseguir instalar “coisas” na BBB é necessário que ela tenha acesso na internet.

Uma forma de fazer isso eh “pingando” a internet dentro da BBB.

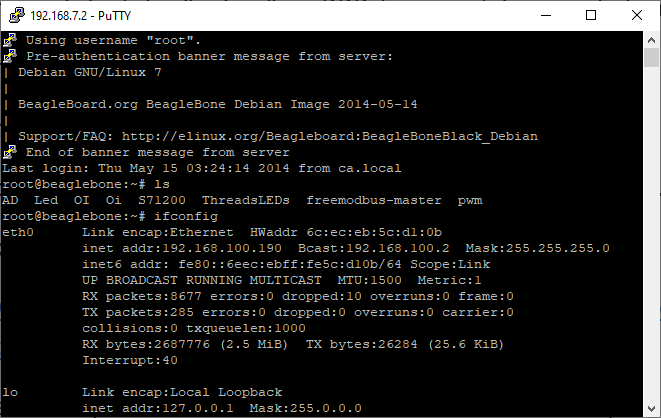
Por exemplo, ping 8.8.8.8 é pingar o servidor do google.

Dentro da maquina virtual eu não consegui fazer isso funcionar.

Este link da uma dica do que fazer

<https://www.digikey.com/en/maker/blogs/how-to-connect-a-beaglebone-black-to-the-internet-using-usb>

O que resolveu para mim foi no Windows conectar na BBB pelo Putty através da porta USB.



Dentro do putty é possível conectar a BBB e então checar um IP externo valido. Por exemplo o IP da porta eth0 era 192.168.100.190 e ela estava na mesma subnet do computador 192.168.100.99 e do roteador 192.168.100.1. Desta forma foi possível associar todos os comandos ethernet para vir para a porta do gateway com o seguinte comando:

**> sudo /sbin/route add default gw 192.168.100.1**

Após isso foi possível fazer o acesso a internet.

Bom, fiz uma conexão pela porta ethernet usando o putty (porta telnet). Ele não conseguia pingar o dns do google. Então, fiz o comando acima e consegui estabelecer a conexão com a BBB.

# Atualizando o relógio da BBB

Para verificar o relógio no Linux basta digitar o comando “date”.

Para atualizar o relógio da BBB tentei fazer varias coisas e não sei se tudo isso teve dependência do comando.

> sudo apt-get update (atualizar o Linux)

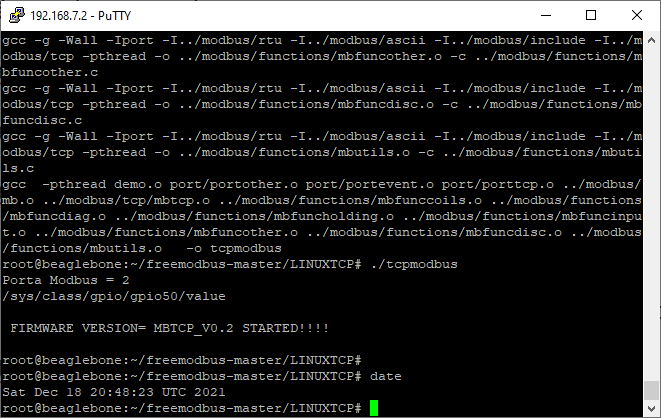
> sudo apt-get install locate (instalar o commando locate)

> sudo apt-get install ntp (instalar o servidor ntp)

Após isso foi feito o seguinte comando para atualizar o relógio do BBB:

**> ntpdate -b -s -u** [**pool.ntp.org**](http://pool.ntp.org/)

Depois pode utilizar o comando >**date**



# Configurando IP e NetMask

Antigo

ifconfig eth0 up 192.168.0.1/24

Novo

ip link set dev eth0 up && ip addr change dev eth0 192.168.0.1/24.

Ou de forma simplificada

ip l s eth0 up && ip a c dev eth0 192.168.0.1/24 netmask 255.255.255.0

**root@beaglebone:/etc# ip**

Usage: ip [ OPTIONS ] OBJECT { COMMAND | help }

ip [ -force ] -batch filename

where OBJECT := { link | addr | addrlabel | route | rule | neigh | ntable |

tunnel | tuntap | maddr | mroute | mrule | monitor | xfrm |

netns | l2tp }

OPTIONS := { -V[ersion] | -s[tatistics] | -d[etails] | -r[esolve] |

-f[amily] { inet | inet6 | ipx | dnet | link } |

-l[oops] { maximum-addr-flush-attempts } |

-o[neline] | -t[imestamp] | -b[atch] [filename] |

-rc[vbuf] [size]}

**root@beaglebone:/etc# ip route**

*default via 192.168.100.1 dev eth0*

*default via 192.168.7.1 dev usb0*

*192.168.7.0/30 dev usb0 proto kernel scope link src 192.168.7.2*

*192.168.100.0/24 dev eth0 proto kernel scope link src 192.168.100.190*