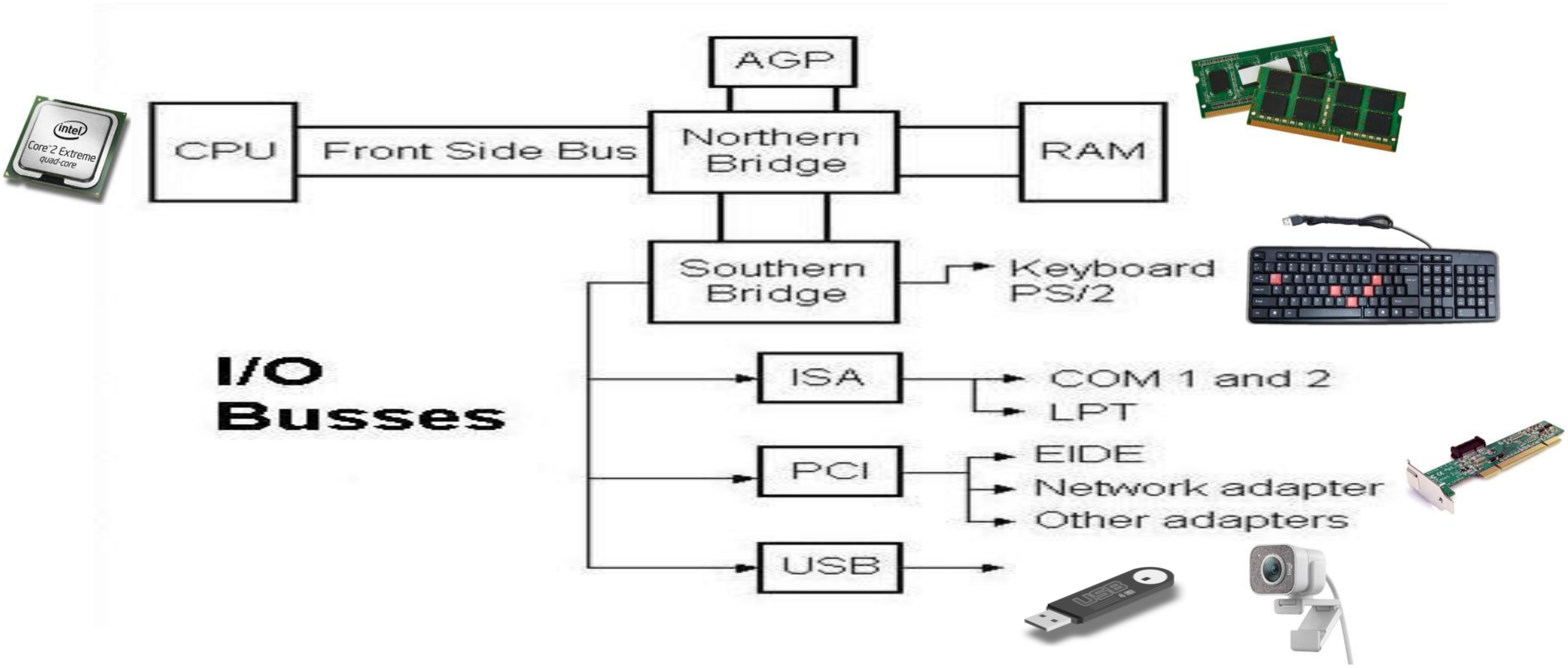
EMBEDDED OPERATING SYSTEMS

Embedded Linux on Beaglebone Black

Computer Bus Architecture



Booting and bootloaders

- Microprocessors / Microcontrollers
  - Can execute code residing in memory (RAM / ROM)
- Operating Systems (OSes)
  - Reside on large-capacity devices
    - Hard-disks, SD cards,…

- On boot, memory does not contain the OS
- Special software needed
  - Bring OS from media to memory

- Role of bootloader
  - Initialize hardware, specially the memory controller
  - Provide boot parameters to the OS
  - Start the OS

U-Boot bootloader

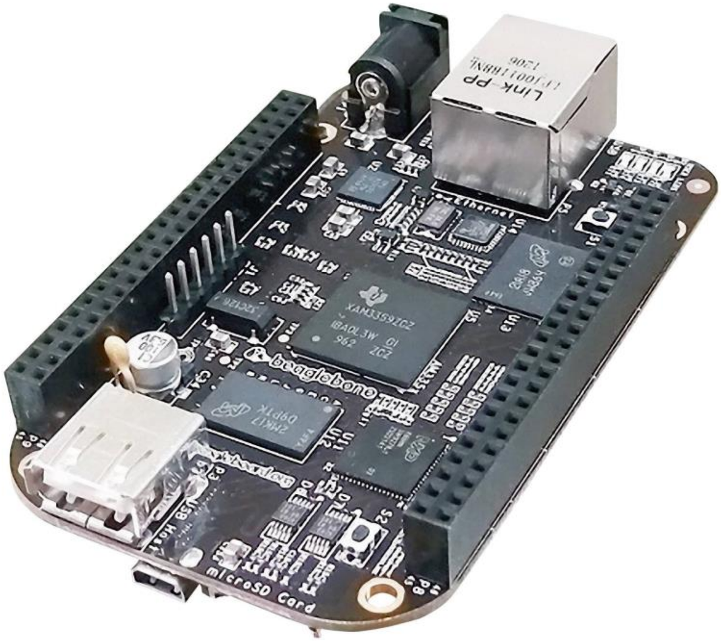
- Open-source, cross-platform bootloader

- Out-of-box support for boards and CPUs
    - PowerPC, x86, ARM, MIPS, …

- Features:
    - Customizable footprint
    - Command shell / monitor
        - With inbuilt commands
    - Variables and scripts
    - Support for Ethernet and USB

Beaglebone Black

- Low-cost, high-expansion development platform
  - Community-supported
- Hardware
  - Chipset
    - TI Sitara AM3358
    - 1GHz ARM CORTEX A8
  - Memory
    - 512MB DDR3L SDRAM
    - On-board Flash 4GB (eMMC)
    - SD/MMC connector for microSD card
  - Debug
    - 20-pin CTI JTAG, Serial header
  - Connectivity
    - USB2.0 Host, Client
    - Serial (UART0) 3.3V TTL, 10/100 Ethernet (RJ45)



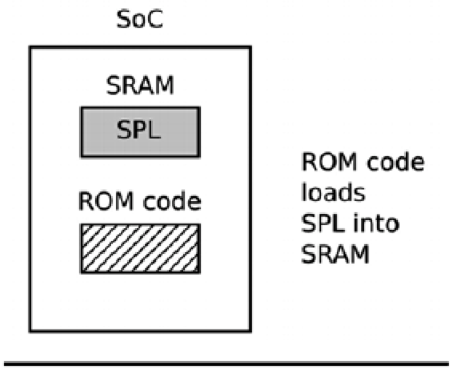
Beaglebone boot (1/5)

- Problem
  - Linux to be booted from eMMC / SD card

- Issue
  - On boot, ARM A-8 CPU
    - Does not have access to:
      - On-board SDRAM
      - Complex devices such as eMMC /SD cards
    - But does have on-chip SRAM (64KB) and ROM (256KB)
  - This leads to a 3-phase boot sequence

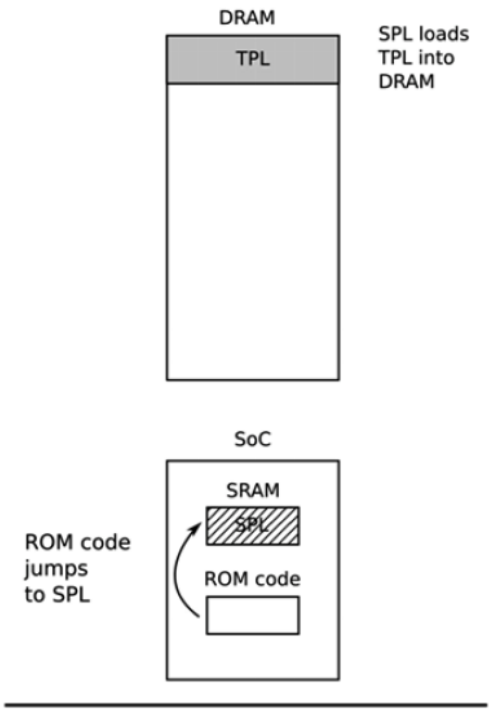
Beaglebone boot (2/5)

- Phase-1
  - On-chip SRAM not large enough
    - Cannot hold/run full bootloader
    - Runs Secondary Program Loader (SPL)
  - Code resident in ROM (ROM code)
    - Loads SPL from pre-programmed locations into SRAM
      - eMMC / SD card / Ethernet
    - Then it jumps to beginning of that code



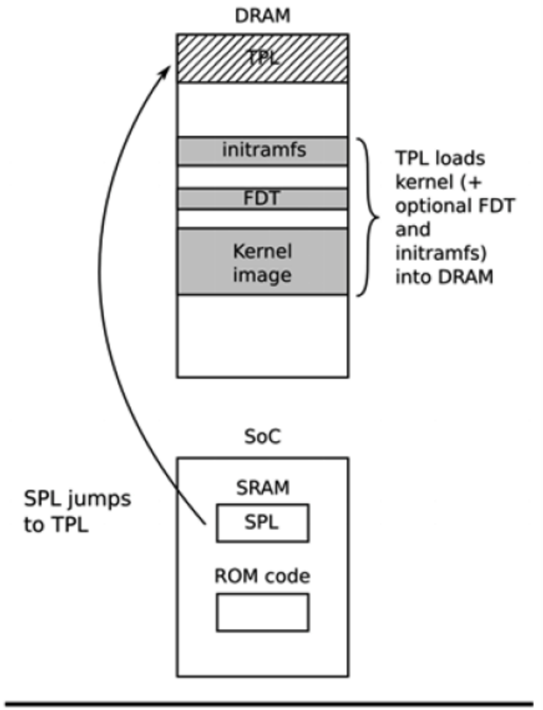
Beaglebone boot (3/5)

- Phase-2
  - SPL sets up memory controller
    - Gets access to on-board DRAM
  - Reads bootloader (*uboot.img*)
    - Called Tertiary Program Loader (TPL)
      - From partition into SDRAM



Beaglebone boot (4/5)

- Phase-3
  - TPL is a full bootloader
  - Loads kernel, devicetree and initrd
    - Into SDRAM
  - Calls the kernel's entry point
    - With boot parameters
  - Kernel takes over and boots
    - Reclaims memory used by bootloader

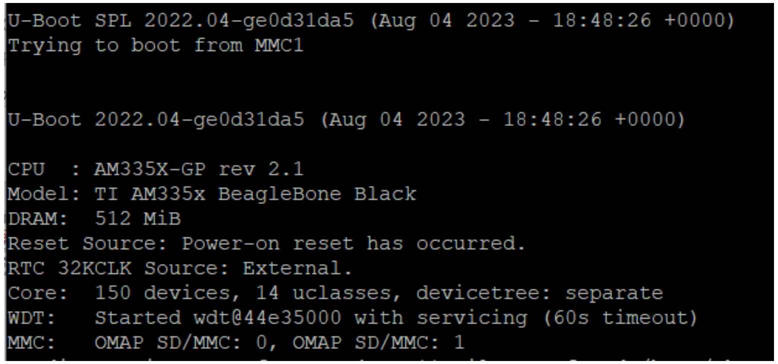


Beaglebone boot (5/5)

- U-Boot bootloader passes following info to kernel
    - Machine number of CPU
    - Basic hardware details
        - Size of DRAM detected, CPU clock speed
    - Kernel command line
        - ASCII
        - Contains the “root” filesystem
            - Path to main filesystem
        - Other behavioral parameters
    - Optionally,
        - device tree
        - initramfs (initial RAM filesystem)

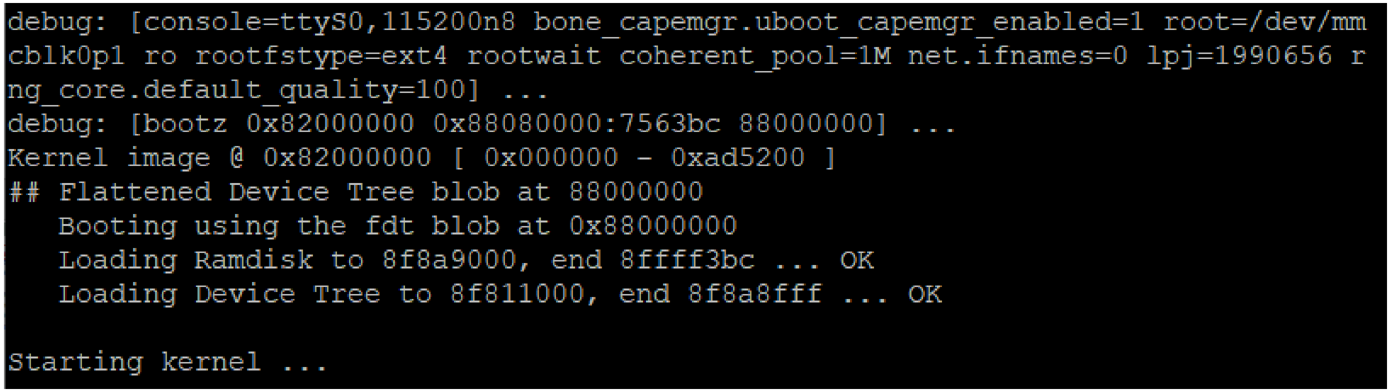
Beaglebone boot screenshots

- U-Boot SPL and TPL



```
U-Boot SPL 2022.04-ge0d31da5 (Aug 04 2023 - 18:48:26 +0000)
Trying to boot from MMC1


U-Boot 2022.04-ge0d31da5 (Aug 04 2023 - 18:48:26 +0000)

CPU  : AM335X-GP rev 2.1
Model: TI AM335x BeagleBone Black
DRAM:  512 MiB
Reset Source: Power-on reset has occurred.
RTC 32KCLK Source: External.
Core:  150 devices, 14 uclasses, devicetree: separate
WDT:   Started wdt@44e35000 with servicing (60s timeout)
MMC:   OMAP SD/MMC: 0, OMAP SD/MMC: 1
```

- U-Boot handover to Linux kernel



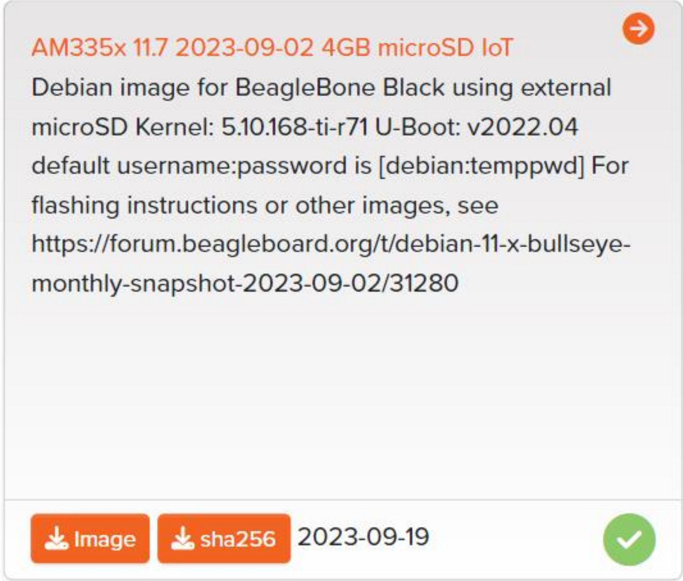
```
debug: [console=ttyS0,115200n8 bone_capemgr.uboot_capemgr_enabled=1 root=/dev/mm
cblk0p1 ro rootfstype=ext4 rootwait coherent_pool=1M net.ifnames=0 lpj=1990656 r
ng_core.default_quality=100] ...
debug: [bootz 0x82000000 0x88080000:7563bc 88000000] ...
Kernel image @ 0x82000000 [ 0x000000 - 0xad5200 ]
## Flattened Device Tree blob at 88000000
   Booting using the fdt blob at 0x88000000
   Loading Ramdisk to 8f8a9000, end 8ffff3bc ... OK
   Loading Device Tree to 8f811000, end 8f8a8fff ... OK

Starting kernel ...
```

Course environment

- Host
  - OS: Ubuntu Linux 22.0.4.3LTS
  - Machine: Intel x86_64 based Laptop/Notebook
- Target
  - Beaglebone Black Rev C
- *Prerequisites*
  - *C programming*
  - *Make utility*
  - *Familiarity with Linux shell, VM/dual-boot usage*

Beaglebone Setup - Images

- Download latest Beaglebone distro
  - We use 4GB IOT for microSD
  - https://www.beagleboard.org/distros



AM335x 11.7 2023-09-02 4GB microSD IoT

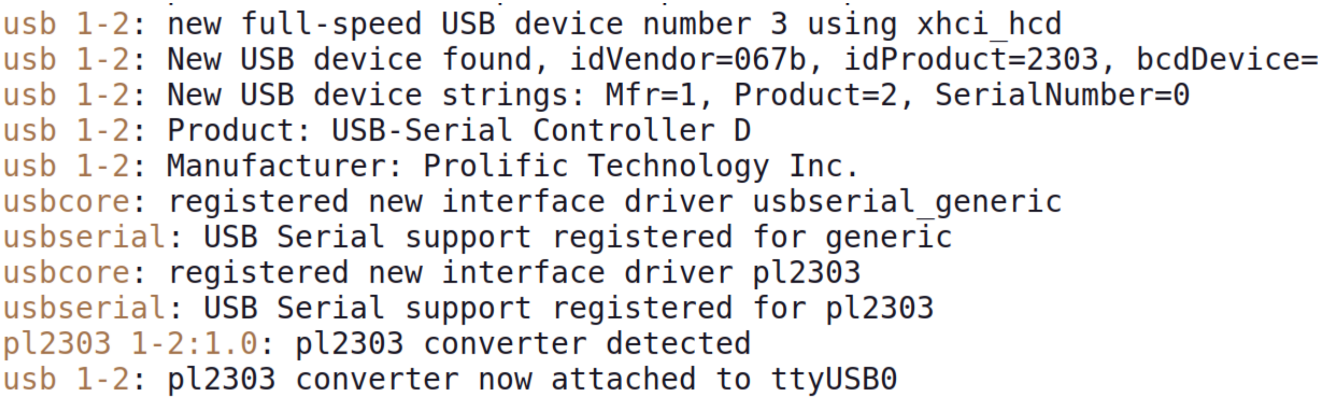
Debian image for BeagleBone Black using external microSD Kernel: 5.10.168-ti-r71 U-Boot: v2022.04 default username:password is [debian:temppwd] For flashing instructions or other images, see https://forum.beagleboard.org/t/debian-11-x-bullseye-monthly-snapshot-2023-09-02/31280

2023-09-19

- Use Balena Etcher
  - For flashing microSD card on Windows / Ubuntu
  - https://etcher.balena.io/#download-etcher

Beaglebone Setup - Serial (1/2)

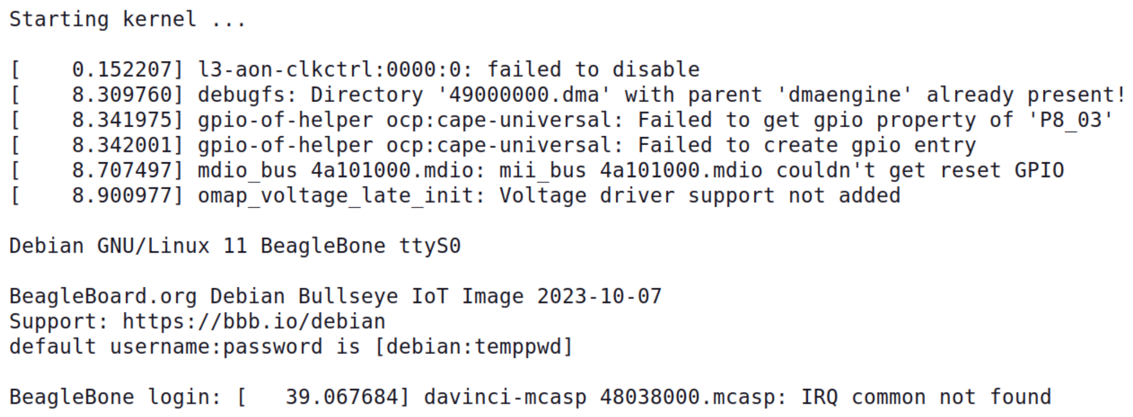
- Boot a system into Ubuntu and get a terminal
- Connect the serial cable pins as per this link
- Insert the cable into the USB port
- Type ***sudo dmesg*** to see Ubuntu detecting the cable *(usually **/dev/ttyUSB0**)*



```
usb 1-2: new full-speed USB device number 3 using xhci_hcd
usb 1-2: New USB device found, idVendor=067b, idProduct=2303, bcdDevice=
usb 1-2: New USB device strings: Mfr=1, Product=2, SerialNumber=0
usb 1-2: Product: USB-Serial Controller D
usb 1-2: Manufacturer: Prolific Technology Inc.
usbcore: registered new interface driver usbserial_generic
usbserial: USB Serial support registered for generic
usbcore: registered new interface driver pl2303
usbserial: USB Serial support registered for pl2303
pl2303 1-2:1.0: pl2303 converter detected
usb 1-2: pl2303 converter now attached to ttyUSB0
```

Beaglebone Setup: Serial (2/2)

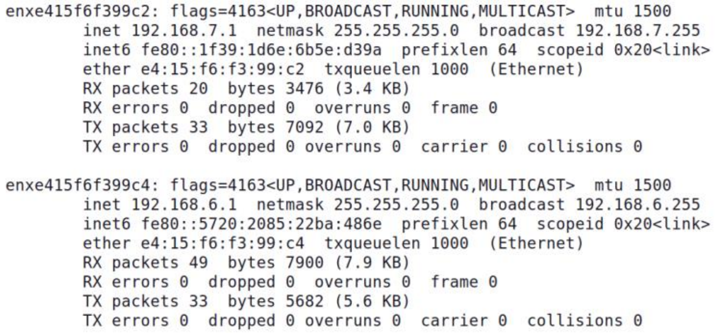
- Run ***sudo minicom –D /dev/ttyUSB0*** to get serial on Ubuntu
- Power the board, **use microSD boot**
  - Press SW S2 near microSD card before powering on
  - You should see the board running U-Boot and Linux
  - A fully booted board will look thus:



```
Starting kernel ...

[    0.152207] l3-aon-clkctrl:0000:0: failed to disable
[    8.309760] debugfs: Directory '49000000.dma' with parent 'dmaengine' already present!
[    8.341975] gpio-of-helper ocp:cape-universal: Failed to get gpio property of 'P8_03'
[    8.342001] gpio-of-helper ocp:cape-universal: Failed to create gpio entry
[    8.707497] mdio_bus 4a101000.mdio: mii_bus 4a101000.mdio couldn't get reset GPIO
[    8.900977] omap_voltage_late_init: Voltage driver support not added

Debian GNU/Linux 11 BeagleBone ttyS0

BeagleBoard.org Debian Bullseye IoT Image 2023-10-07
Support: https://bbb.io/debian
default username:password is [debian:temppwd]

BeagleBone login: [   39.067684] davinci-mcasp 48038000.mcasp: IRQ common not found
```

Beaglebone Setup: USB n/w

- Beaglebone image sets up a private network on USB
  - Wait few minutes while it comes up
- Run ***ifconfig*** on Ubuntu to confirm it has come up

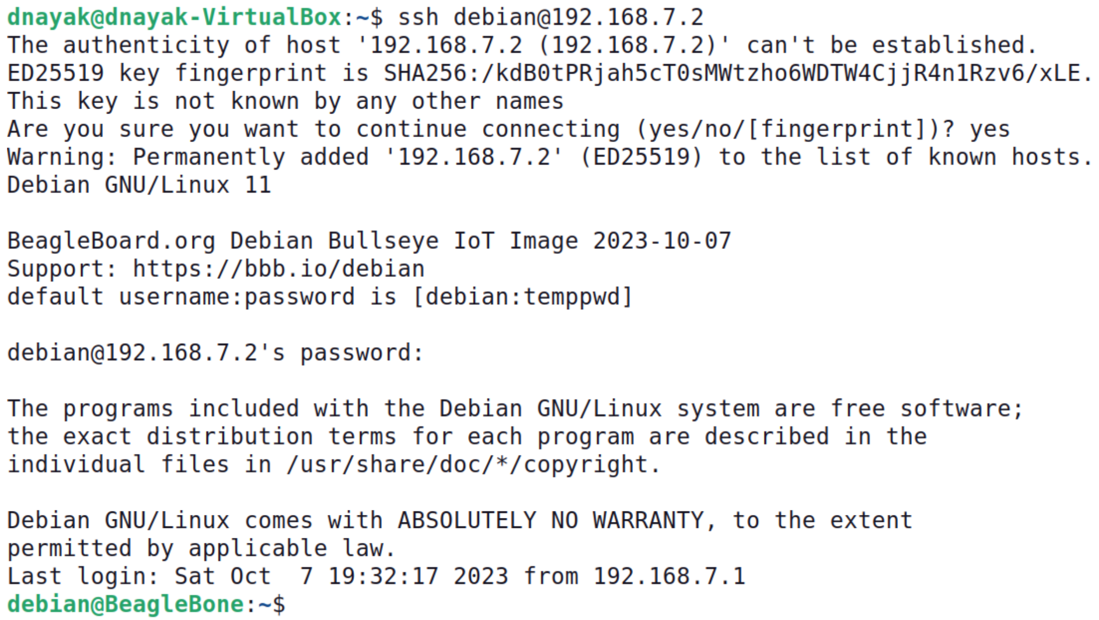


```
enxe415f6f399c2: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.7.1  netmask 255.255.255.0  broadcast 192.168.7.255
        inet6 fe80::1f39:1d6e:6b5e:d39a  prefixlen 64  scopeid 0x20<link>
        ether e4:15:f6:f3:99:c2  txqueuelen 1000  (Ethernet)
        RX packets 20  bytes 3476 (3.4 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 33  bytes 7092 (7.0 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

enxe415f6f399c4: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.6.1  netmask 255.255.255.0  broadcast 192.168.6.255
        inet6 fe80::5720:2085:22ba:486e  prefixlen 64  scopeid 0x20<link>
        ether e4:15:f6:f3:99:c4  txqueuelen 1000  (Ethernet)
        RX packets 49  bytes 7900 (7.9 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 33  bytes 5682 (5.6 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

- Connecting via SSH
  - IP addresses
    - Ubuntu gets **192.168.7.1**
    - Board gets **192.168.7.2**

Beaglebone Setup: SSH Login

- Log on to the board
  - From Ubuntu
  - Using its IP address (**192.168.7.2**)
  - *login:password is **debian:temppwd***



```
dnayak@dnayak-VirtualBox:~$ ssh debian@192.168.7.2
The authenticity of host '192.168.7.2 (192.168.7.2)' can't be established.
ED25519 key fingerprint is SHA256:/kdB0tPRjah5cT0sMWtzho6WDTW4CjjR4n1Rzv6/xLE.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '192.168.7.2' (ED25519) to the list of known hosts.
Debian GNU/Linux 11

BeagleBoard.org Debian Bullseye IoT Image 2023-10-07
Support: https://bbb.io/debian
default username:password is [debian:temppwd]

debian@192.168.7.2's password:

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Sat Oct  7 19:32:17 2023 from 192.168.7.1
debian@BeagleBone:~$
```

THANK YOU!