**ESE-3014 EMBEDDED SYSTEMS COMMUNICATION PROTOCOLS AND SECURITY**

**LAB 8 Report**

**GROUP No. 2**

|  |  |
| --- | --- |
| **Group Members** | **Student ID** |
| Christy Rachel Philip | C0765535 |
| Fahad Rahman | C0769871 |
| James M Chacko | C0777192 |
| Premil Presannan | C0777191 |
| Vy Nguyen | C0776242 |

**Interfacing communication – Ethernet**

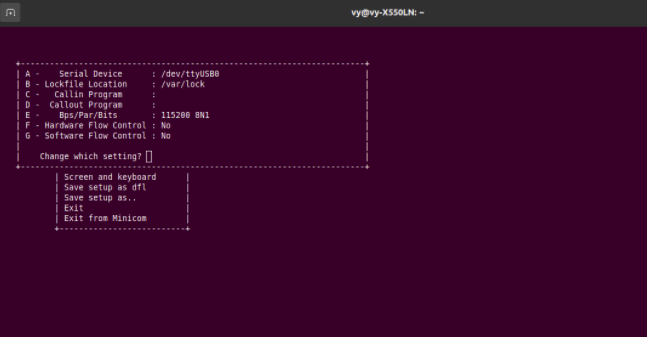
Part One: Connections

*1. Connect your serial debug (3.3-V FTDI) cable to your PC and your Beaglebone*

Connect the FTDI cable to the BeagleBone, notice that the black wire will be at the header with the white dot above it

*2. Boot your PC and launch minicom*

|  |
| --- |
| $ sudo minicom |



*3. Connect your Beaglebone to the 5-V/2-A power adapter*

*4. Connect your Beaglebone’s ethernet port to a router via an ethernet patch cable*

*ifconfig before and after Ethernet connection:*

“Ifconfig is used to configure the kernel-resident network interfaces. It is used at boot time to set up interfaces as necessary. After that, it is usually only needed when debugging or when system tuning is needed.

If no arguments are given, ifconfig displays the status of the currently active interfaces. If a single interface argument is given, it displays the status of the given interface only; if a single -a argument is given, it displays the status of all interfaces, even those that are down. Otherwise, it configures an interface.”

|  |
| --- |
| debian@beaglebone:~$ ifconfig eth0: flags=-28669<UP,BROADCAST,MULTICAST,DYNAMIC> mtu 1500  ether 0c:b2:b7:d5:75:79 txqueuelen 1000 (Ethernet)  RX packets 30 bytes 3619 (3.5 KiB)  RX errors 0 dropped 0 overruns 0 frame 0  TX packets 76 bytes 10429 (10.1 KiB)  TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0  device interrupt 63   lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536  inet 127.0.0.1 netmask 255.0.0.0  inet6 ::1 prefixlen 128 scopeid 0x10<host>  loop txqueuelen 1000 (Local Loopback)  RX packets 320 bytes 23680 (23.1 KiB)  RX errors 0 dropped 0 overruns 0 frame 0  TX packets 320 bytes 23680 (23.1 KiB)  TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0  usb0: flags=4099<UP,BROADCAST,MULTICAST> mtu 1500  inet 192.168.7.2 netmask 255.255.255.0 broadcast 192.168.7.255  ether 0c:b2:b7:d5:75:7b txqueuelen 1000 (Ethernet)  RX packets 0 bytes 0 (0.0 B)  RX errors 0 dropped 0 overruns 0 frame 0  TX packets 0 bytes 0 (0.0 B)   TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0    usb1: flags=4099<UP,BROADCAST,MULTICAST> mtu 1500   inet 192.168.6.2 netmask 255.255.255.0 broadcast 192.168.6.255   ether 0c:b2:b7:d5:75:7f txqueuelen 1000 (Ethernet)   RX packets 0 bytes 0 (0.0 B)   RX errors 0 dropped 0 overruns 0 frame 0   TX packets 0 bytes 0 (0.0 B)   TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0    debian@beaglebone:~$ ifconfig  eth0: flags=-28605<UP,BROADCAST,RUNNING,MULTICAST,DYNAMIC> mtu 1500   inet 192.168.2.30 netmask 255.255.255.0 broadcast 192.168.2.255   inet6 fe80::eb2:b7ff:fed5:7579 prefixlen 64 scopeid 0x20<link>   ether 0c:b2:b7:d5:75:79 txqueuelen 1000 (Ethernet)   RX packets 40 bytes 4996 (4.8 KiB)   RX errors 0 dropped 0 overruns 0 frame 0   TX packets 135 bytes 19099 (18.6 KiB)   TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0   device interrupt 63    lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536   inet 127.0.0.1 netmask 255.0.0.0   inet6 ::1 prefixlen 128 scopeid 0x10<host>   loop txqueuelen 1000 (Local Loopback)   RX packets 320 bytes 23680 (23.1 KiB)   RX errors 0 dropped 0 overruns 0 frame 0   TX packets 320 bytes 23680 (23.1 KiB)   TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0    usb0: flags=4099<UP,BROADCAST,MULTICAST> mtu 1500   inet 192.168.7.2 netmask 255.255.255.0 broadcast 192.168.7.255   ether 0c:b2:b7:d5:75:7b txqueuelen 1000 (Ethernet)   RX packets 0 bytes 0 (0.0 B)   RX errors 0 dropped 0 overruns 0 frame 0   TX packets 0 bytes 0 (0.0 B)   TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0    usb1: flags=4099<UP,BROADCAST,MULTICAST> mtu 1500   inet 192.168.6.2 netmask 255.255.255.0 broadcast 192.168.6.255   ether 0c:b2:b7:d5:75:7f txqueuelen 1000 (Ethernet)   RX packets 0 bytes 0 (0.0 B)   RX errors 0 dropped 0 overruns 0 frame 0   TX packets 0 bytes 0 (0.0 B)   TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0 |

**Part Two: identifying your connections**

*5. Identify the name and IP address of your host machine’s main ethernet connection*

Similarly, running the ifconfig command on the host machine will have the result:

IP address of the host machine using ifconfig command:

|  |
| --- |
| wlp3s0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500  inet 192.168.2.11 netmask 255.255.255.0 broadcast 192.168.2.255  inet6 fe80::e0c6:3b06:30bd:a78d prefixlen 64 scopeid 0x20<link>  ether 18:cf:5e:2d:5d:88 txqueuelen 1000 (Ethernet)  RX packets 56005 bytes 60947904 (60.9 MB)  RX errors 0 dropped 0 overruns 0 frame 47364  TX packets 36606 bytes 8467224 (8.4 MB)  TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0  device interrupt 19 |

=> 192.168.2.11 is the IP address of the host machine

*6. Identify the name and IP address of your embedded system’s main ethernet port (Q: is the IP address local or global? Explain.)*

**In step 4, note that at the eth0 port:**

before connection

|  |
| --- |
| eth0: flags=-28669<UP,BROADCAST,MULTICAST,DYNAMIC> mtu 1500  ether 0c:b2:b7:d5:75:79 txqueuelen 1000 (Ethernet)  RX packets 30 bytes 3619 (3.5 KiB)  RX errors 0 dropped 0 overruns 0 frame 0  TX packets 76 bytes 10429 (10.1 KiB)  TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0  device interrupt 63 |

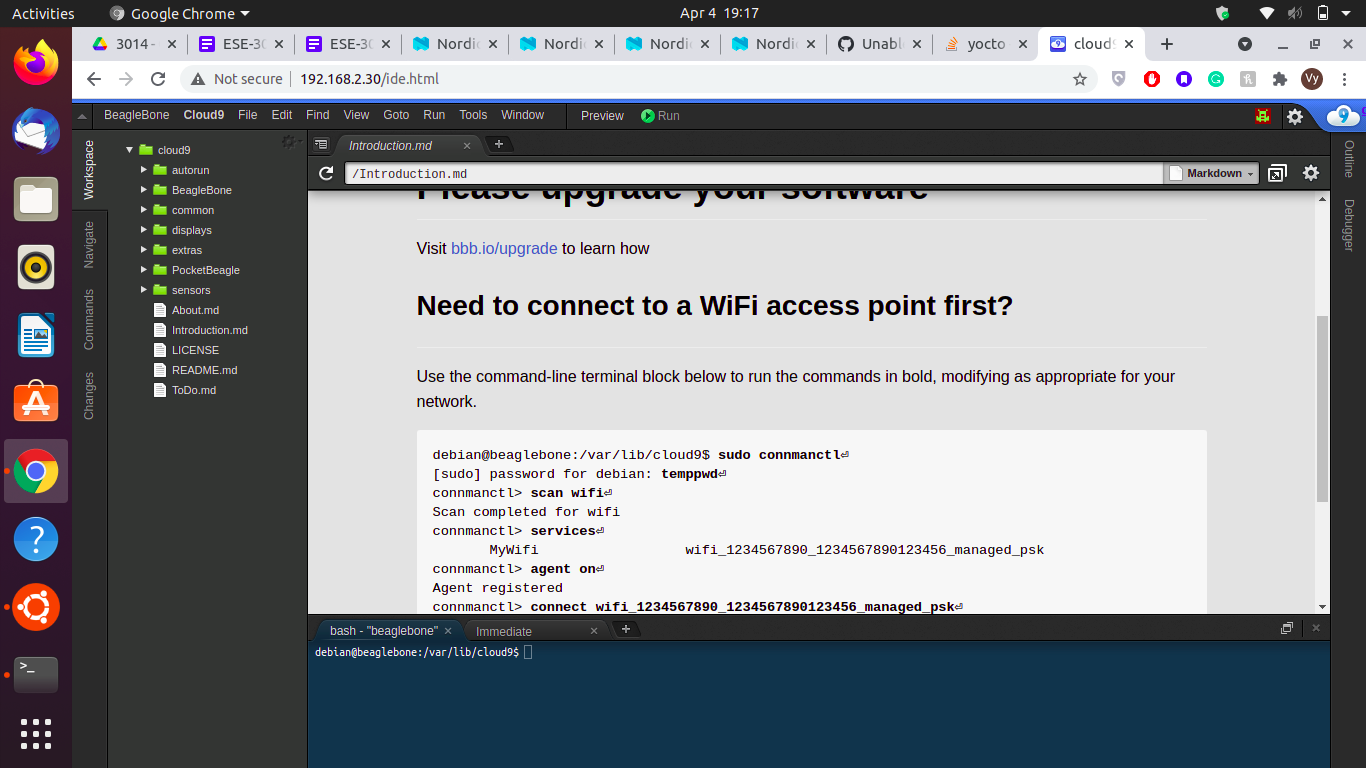
afer connection:

|  |
| --- |
| eth0: flags=-28605<UP,BROADCAST,RUNNING,MULTICAST,DYNAMIC> mtu 1500   inet 192.168.2.30 netmask 255.255.255.0 broadcast 192.168.2.255   inet6 fe80::eb2:b7ff:fed5:7579 prefixlen 64 scopeid 0x20<link>   ether 0c:b2:b7:d5:75:79 txqueuelen 1000 (Ethernet)   RX packets 40 bytes 4996 (4.8 KiB)   RX errors 0 dropped 0 overruns 0 frame 0   TX packets 135 bytes 19099 (18.6 KiB)   TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0   device interrupt 63 |

**We can see the Beaglebone Black IP address appear from the terminal is:**

**inet 192.168.2.30**

**The IP address is global** because it can be accessed via the internet (for example, accessing the IP using a browser on the host machine, it will open the clould9 ide of the specific beaglebone.



*7. Attempt to ping the connection of each machine from the other; use $ man ping for more information on the exact use of ping:*

Ping to Beaglebone Black from host machine

|  |
| --- |
| vy@vy-X550LN:~$ ping 192.168.2.30 PING 192.168.2.30 (192.168.2.30) 56(84) bytes of data. 64 bytes from 192.168.2.30: icmp\_seq=1 ttl=64 time=25.2 ms 64 bytes from 192.168.2.30: icmp\_seq=2 ttl=64 time=58.6 ms 64 bytes from 192.168.2.30: icmp\_seq=3 ttl=64 time=85.6 ms ^C --- 192.168.2.30 ping statistics --- 3 packets transmitted, 3 received, 0% packet loss, time 2003ms rtt min/avg/max/mdev = 25.166/56.459/85.569/24.707 ms |

Ping to the host machine from BBB

|  |
| --- |
| debian@beaglebone:~$ ping 192.168.2.11 PING 192.168.2.11 (192.168.2.11) 56(84) bytes of data. 64 bytes from 192.168.2.11: icmp\_seq=1 ttl=64 time=119 ms 64 bytes from 192.168.2.11: icmp\_seq=2 ttl=64 time=143 ms 64 bytes from 192.168.2.11: icmp\_seq=3 ttl=64 time=166 ms ^C --- 192.168.2.11 ping statistics --- 4 packets transmitted, 3 received, 25% packet loss, time 3003ms rtt min/avg/max/mdev = 119.591/143.170/166.723/19.244 ms debian@beaglebone:~$ |

*8. Verify that you are able to connect to Internet from your embedded system. Using a GUI Linux interface, can you open a web page?*

First, we have to install the GUI Linux interface:

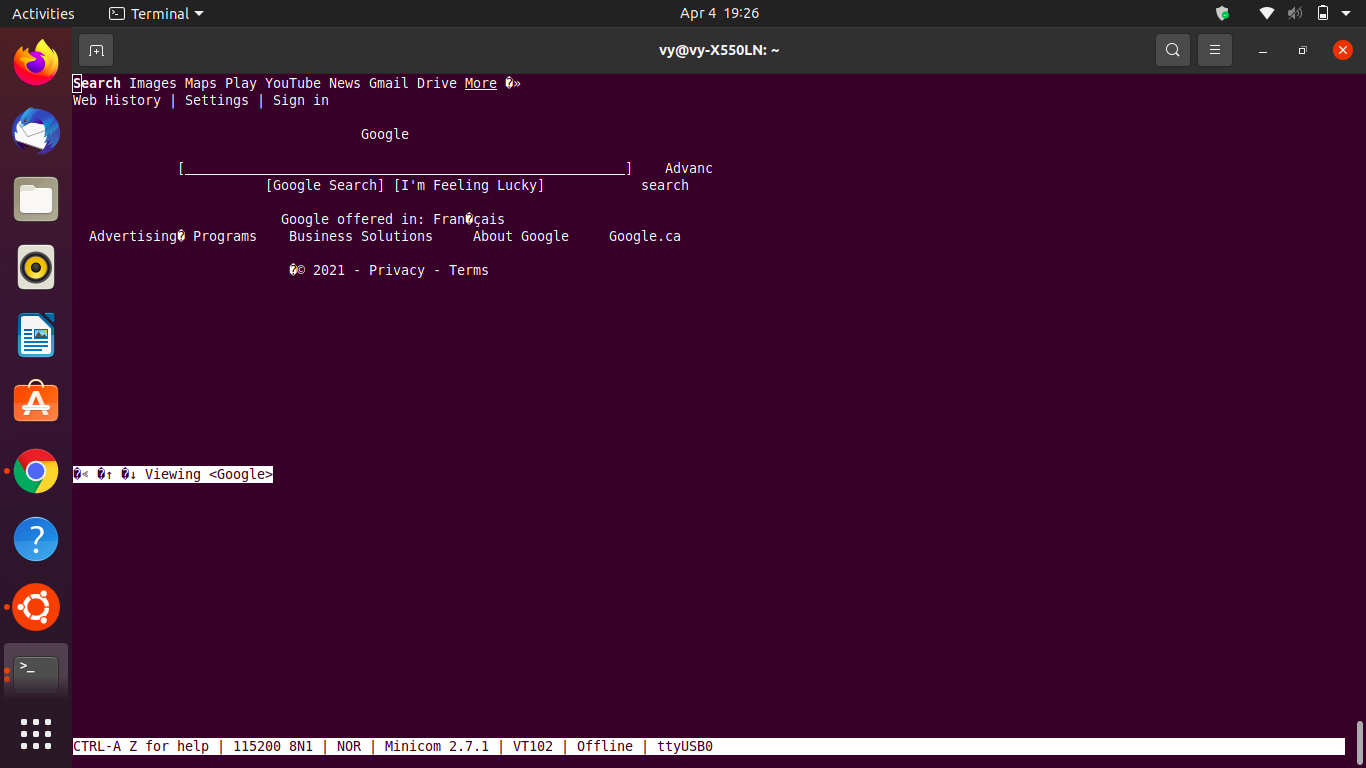
**installing w3m**

|  |
| --- |
| debian@beaglebone:~$ sudo apt-get install w3m w3m-img [sudo] password for debian:  Reading package lists... Done Building dependency tree  Reading state information... Done The following additional packages will be installed:  libgc1c2 libgif7 libid3tag0 libimlib2 libxext6 Suggested packages:  libsixel-bin xdg-utils w3m-el cmigemo xsel mpv dict dictd dict-wn The following NEW packages will be installed:  libgc1c2 libgif7 libid3tag0 libimlib2 libxext6 w3m w3m-img 0 upgraded, 7 newly installed, 0 to remove and 95 not upgraded. Need to get 1,617 kB of archives. After this operation, 3,121 kB of additional disk space will be used. Do you want to continue? [Y/n] y Get:1 http://deb.debian.org/debian stretch/main armhf libgc1c2 armhf 1:7.4.2-8 [197 kB] Get:2 http://deb.debian.org/debian stretch/main armhf libgif7 armhf 5.1.4-0.4 [40.7 kB] Get:3 http://deb.debian.org/debian stretch/main armhf libid3tag0 armhf 0.15.1b-12 [30.4 kB] Get:4 http://deb.debian.org/debian stretch/main armhf libxext6 armhf 2:1.3.3-1+b2 [48.1 kB] Get:5 http://deb.debian.org/debian stretch/main armhf libimlib2 armhf 1.4.8-1 [180 kB] Get:6 http://deb.debian.org/debian stretch/main armhf w3m armhf 0.5.3-34+deb9u1 [992 kB] Get:7 http://deb.debian.org/debian stretch/main armhf w3m-img armhf 0.5.3-34+deb9u1 [128 kB] Fetched 1,617 kB in 0s (3,472 kB/s) Selecting previously unselected package libgc1c2:armhf. (Reading database ... 39193 files and directories currently installed.) Preparing to unpack .../0-libgc1c2\_1%3a7.4.2-8\_armhf.deb ... Unpacking libgc1c2:armhf (1:7.4.2-8) ... Selecting previously unselected package libgif7:armhf. Preparing to unpack .../1-libgif7\_5.1.4-0.4\_armhf.deb ... Unpacking libgif7:armhf (5.1.4-0.4) ... Selecting previously unselected package libid3tag0:armhf. Preparing to unpack .../2-libid3tag0\_0.15.1b-12\_armhf.deb ... Unpacking libid3tag0:armhf (0.15.1b-12) ... Selecting previously unselected package libxext6:armhf. Preparing to unpack .../3-libxext6\_2%3a1.3.3-1+b2\_armhf.deb ... Unpacking libxext6:armhf (2:1.3.3-1+b2) ... Selecting previously unselected package libimlib2:armhf. Preparing to unpack .../4-libimlib2\_1.4.8-1\_armhf.deb ... Unpacking libimlib2:armhf (1.4.8-1) ... Selecting previously unselected package w3m. Preparing to unpack .../5-w3m\_0.5.3-34+deb9u1\_armhf.deb ... Unpacking w3m (0.5.3-34+deb9u1) ... Selecting previously unselected package w3m-img. Preparing to unpack .../6-w3m-img\_0.5.3-34+deb9u1\_armhf.deb ... Unpacking w3m-img (0.5.3-34+deb9u1) ... Setting up libid3tag0:armhf (0.15.1b-12) ... Setting up libxext6:armhf (2:1.3.3-1+b2) ... Processing triggers for mime-support (3.60) ... Setting up libgc1c2:armhf (1:7.4.2-8) ... Setting up libgif7:armhf (5.1.4-0.4) ... Processing triggers for libc-bin (2.24-11+deb9u4) ... Setting up w3m (0.5.3-34+deb9u1) ... Processing triggers for man-db (2.7.6.1-2) ... Setting up libimlib2:armhf (1.4.8-1) ... Setting up w3m-img (0.5.3-34+deb9u1) ... Processing triggers for libc-bin (2.24-11+deb9u4) ... |

Running the w3m tool:

|  |
| --- |
| debian@beaglebone:~$ w3m www.google.com |

Result:



*9. Using SSH, login to your embedded system from your host machine using the embedded platform’s IP address*

On the host machine, using this command:

|  |
| --- |
| vy@vy-X550LN:~$ ssh debian@192.168.2.30 The authenticity of host '192.168.2.30 (192.168.2.30)' can't be established. ECDSA key fingerprint is SHA256:B3WNqFUyi7yB9yYy8aoj1Yr9x5zVf9biyHOpIOEwuQY. Are you sure you want to continue connecting (yes/no/[fingerprint])? yes Warning: Permanently added '192.168.2.30' (ECDSA) to the list of known hosts. Debian GNU/Linux 9  BeagleBoard.org Debian Stretch imgtec Image 2020-04-06  Support: http://elinux.org/Beagleboard:BeagleBoneBlack\_Debian  default username:password is [debian:temppwd]  debian@192.168.2.30'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: Wed Mar 10 06:32:31 2021 debian@beaglebone:~$ |

**Part Three: “crossover connection”**

BONUS POINTS

It should be possible to establish a direct connection from your Linux host machine to your Beaglebone using an ethernet patch cable; in older systems, the ethernet cable had to be a “crossover” type to allow a direct link, however, modern systems can work with a conventional patch cable.

10. Investigate this type of connection, and see if you are able to do it!

11. Name your sources in your discussion.

This part is strictly for bonus points

*Answer:*

The other “crossover” type of connection is via USB cable ( that has been discussed in Lab 1 of this subject)

The steps are:

1, identify the ip address of the connection between the host machine and beaglebone black using ifconfig

2. identify the internet address on the host machine (can be ethernet cable or wireless).

3. configure the host to forward packets

4. Check that ufw (simple firewall utility on Ubuntu) is disabled:

5. Flush old rules from iptables (e.g. firewall):

6. Add iptables firewall rules to forward traffic from the USB "Ethernet" interface (which is connected to the BeagleBone) to the WiFi interface:

On beaglebone black:

1.configure the interface using ifconfig

2. Add default route so that the BeagleBone tries to connect to the Internet through the laptop

3. verify the internet by sending ping signal to google server.

For more information, check out the resource below:

Link of the resource:

<https://gist.github.com/pdp7/d2711b5ff1fbb000240bd8337b859412>

Reference:

<https://gist.github.com/pdp7/d2711b5ff1fbb000240bd8337b859412>