Real World Mobile WLAN Testing - Part 2

WLAN Pi Overview

The goal, provide Wireless LAN Professionals with a ready-to-use device capable of providing throughput measurements for assessing network performance.

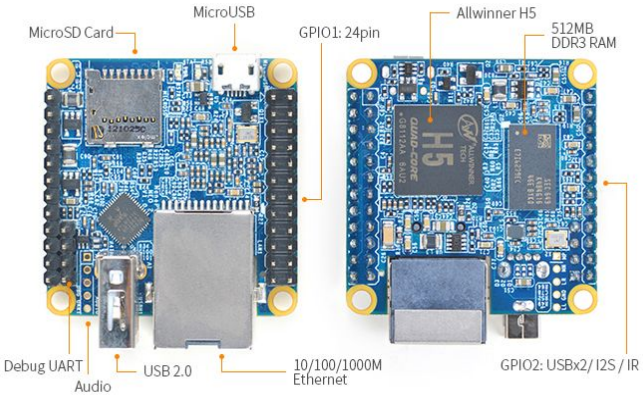
The WLAN Pi can be used to test Wired-to-Wired, Wired-to-Wireless, and even Wireless-to-Wireless. These tests can be used to assist in establishing baselines, help with troubleshooting, testing consistency, as well as measuring end-to-end network throughput.

Hardware

At the brains of the WLAN Pi is a NanoPi Neo2 single-board-computer (SBC). The Neo2 shares similarities to other SBC's like the popular Raspberry Pi. However, the Neo2 differentiates itself with its extremely portable size, low power requirements, and most importantly full 10/100/1000M Ethernet onboard, unlike the Raspberry Pi which is limited to 10/100M Ethernet port.

SBC: NanoPi NEO2

- CPU: Quad-core 64-bit Cortex A53
- Memory: either 512MB or 1GB DDR3
- Connectivity: 10/100/1000M Ethernet
- Storage: MicroSD
- USB: 1 x USB Type A
- Power input: Micro USB 5V/2A
- Dimensions: 40x40mm
- Display: OLED



Wireless Adapter

- Comfast CF-912AC (Realtek 8812au chipset)
- 1200Mbps USB Wifi Dual Band 802.11ac/a/b/g/n

External Battery

- Koral Luma 3000 Portable Charger
- 3000mAh

wirelessLAN PROFESSIONALS CONFERENCE *Real World Mobile WLAN Testing*

WLAN Pi Quick Reference Sheet

Default Logins

SSH

Username: wlanpi
Password: wlanpi

Kismet

Username: kismet
Password: wlanpi

Installed Software

Operating System: *Debian Stretch - Kernel 4.14*

Running on boot:

Application	Version	Note
Iperf3 server	3.1.3	Port 5201 (default)
Iperf2 server	2.0.9	Port 5001 (default)
ZAPd (Ruckus) daemon	1.83.18	
Ekahau eperf		Port 5202
Speed Tests (HTML5)		http://{ip address}
Kismet	Dev build	http://{ip address}:2501
WiFi Explorer Pro Sensor		Works with WiFi Explorer Pro (macOS only)

Additional Software Installed:

H.O.R.S.T - Live Wi-Fi packet analysis
TCPDump - common packet analyzer that runs under the command line
HOSTAPD - IEEE 802.11 AP and IEEE 802.1X/WPA/WPA2/EAP/RADIUS Authenticator

Button configuration:

F1 - Date & Time

F2 - IP & System Info

F3 - Shutdown Menu

Powering Your WLAN Pi

Koral Luma 3000
The included power bank will power the WLAN Pi, plug it it and it should automatically power on.

PoE Power (optional)
make to use a gigabit PoE splitter
Example: [PoE Texas - WT-GAF-MicroUSB]

**any 5v/1a USB power source could be used to power the WLAN Pi*

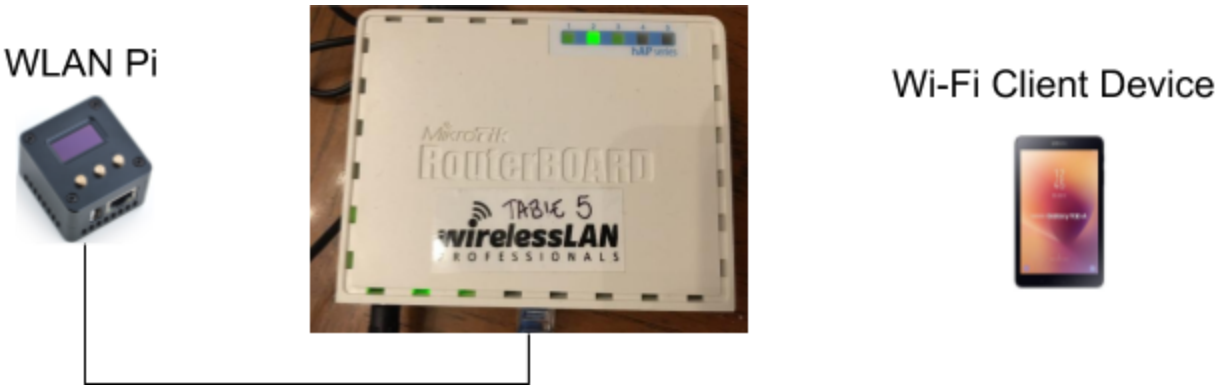
Hands on labs

Connect your Wi-Fi adapter to the USB port on the WLAN Pi

The following labs require the provided USB Wi-Fi adapter to be connected to the WLAN Pi.

Tested Wi-Fi adapters include: Comfast CF-912AC, Odroid Module 5, and Ekahau SA-1

Connect the WLAN Pi to your MikroTik Wireless router



The WLAN Pi will be acting as our network endpoint for performance testing. More details on what this box is will come tomorrow.

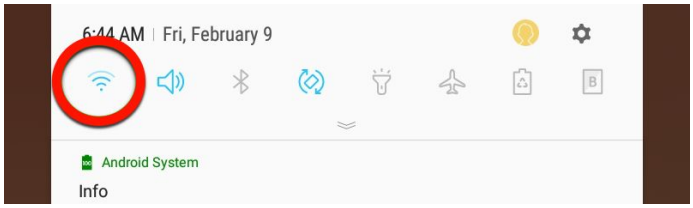
1. Gather your WLAN Pi, USB cable, Ethernet cable, and battery.
2. Connect the WLAN Pi to any ethernet port on any port **EXCEPT** **port 1**.
3. Connect the WLAN Pi to the battery pack.
   The smaller micro-USB end connects to the WLAN Pi on the side opposite the Ethernet jack.
   The larger USB-A end connects to the battery pack.
4. The unit should show internal lights and after a few moments it will boot and the display will activate and show various statistics, including the IP address.
5. Record the IP address displayed on your WLAN Pi

Switch your test devices to the Lab SSID

1. The SSIDs for your lab will be:

   Table # - 2.4GHz
   and
   Table # - 5GHz

2. Swipe down from the top of the screen to open the Quick Settings menu. **Long press** on the Wi-Fi icon to open the SSID list.



3. Select your "Table #" SSID and verify that it connects. You may see a warning that this network has no Internet access.

Lab 1 - Kismet

Kismet is a wireless network detector, sniffer, and intrusion detection system. Kismet works predominantly with Wi-Fi (IEEE 802.11) networks, but can be expanded via plug-ins to handle other network types.

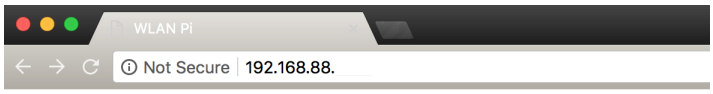
- 802.11 sniffing
- Standard PCAP logging (compatible with Wireshark, TCPDump, etc)
- Client/Server modular architecture
- Plug-in architecture to expand core features
- Multiple capture source support
- Live export of packets to other tools via tun/tap virtual interfaces
- Distributed remote sniffing via light-weight remote capture

More Info: https://www.kismetwireless.net/

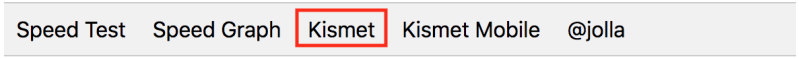
Developer: Mike Kershaw - @KismetWireless

Task 1 - Enable a source for Kismet to use

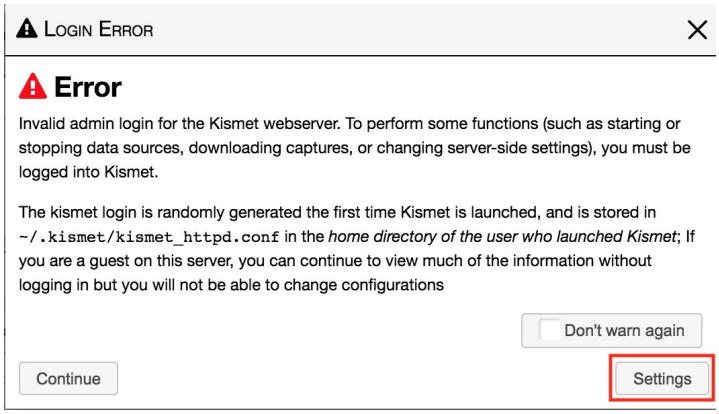
1. Open a web browser, enter in the address bar the IP address of your WLAN Pi



2. From the WLAN Pi home page select “Kismet”

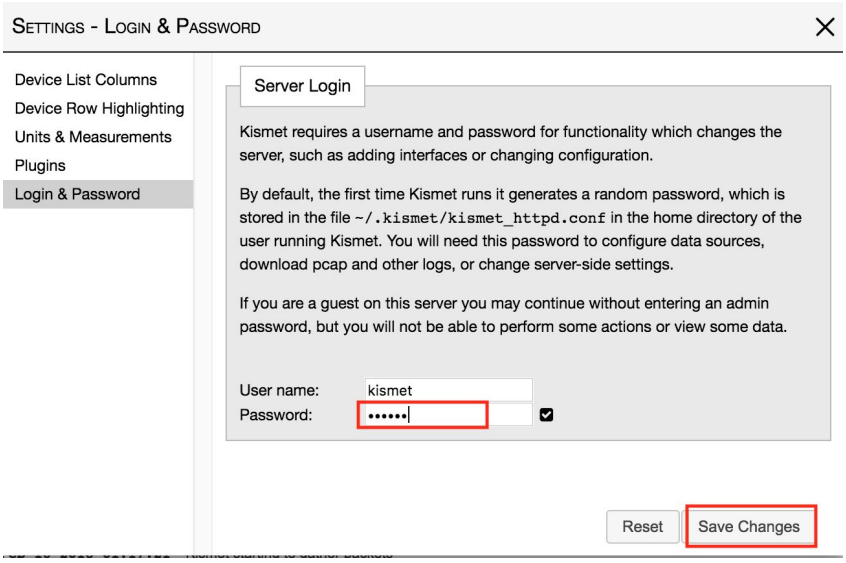


3. “Login Error” - Select **settings**



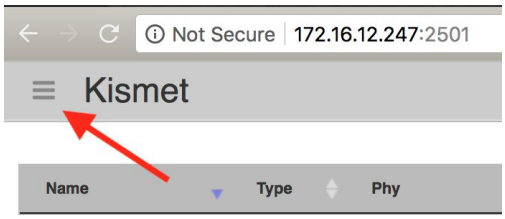
4. Login to the Kismet Server to be able to make changes.

   Username: **kismet**
   Password: **wlanpi**

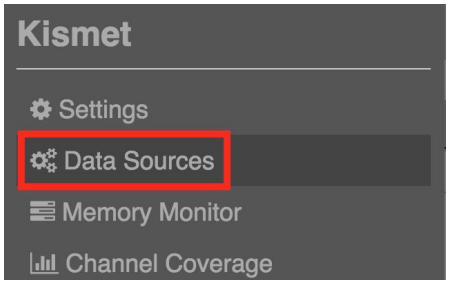


Task 1 - Enable a source for Kismet to use (cont.)

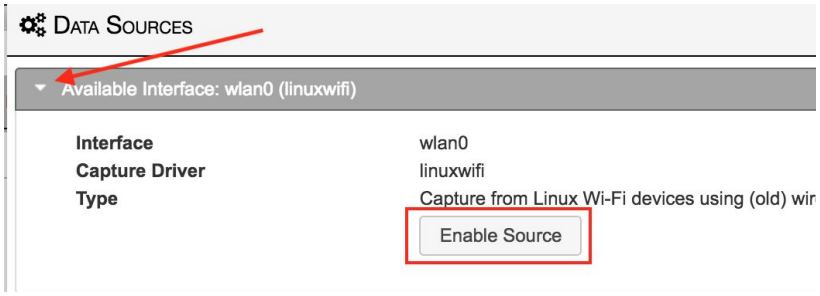
5. Click **Kismet Menu**



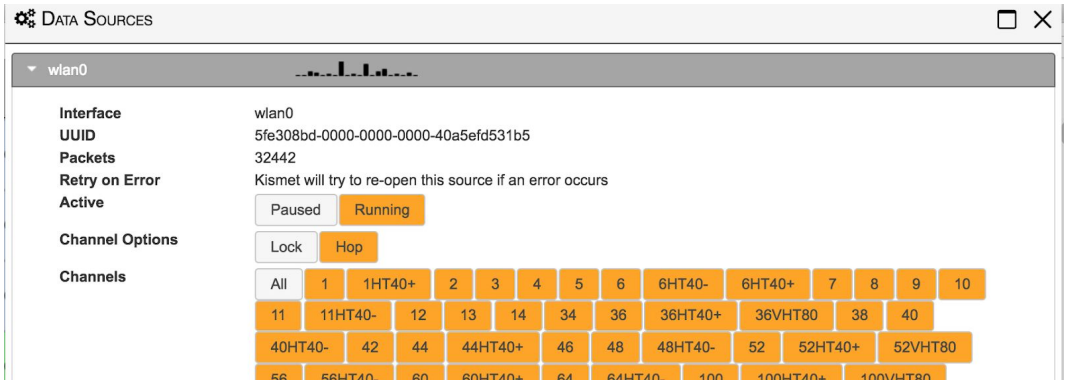
6. Click **Data Sources**



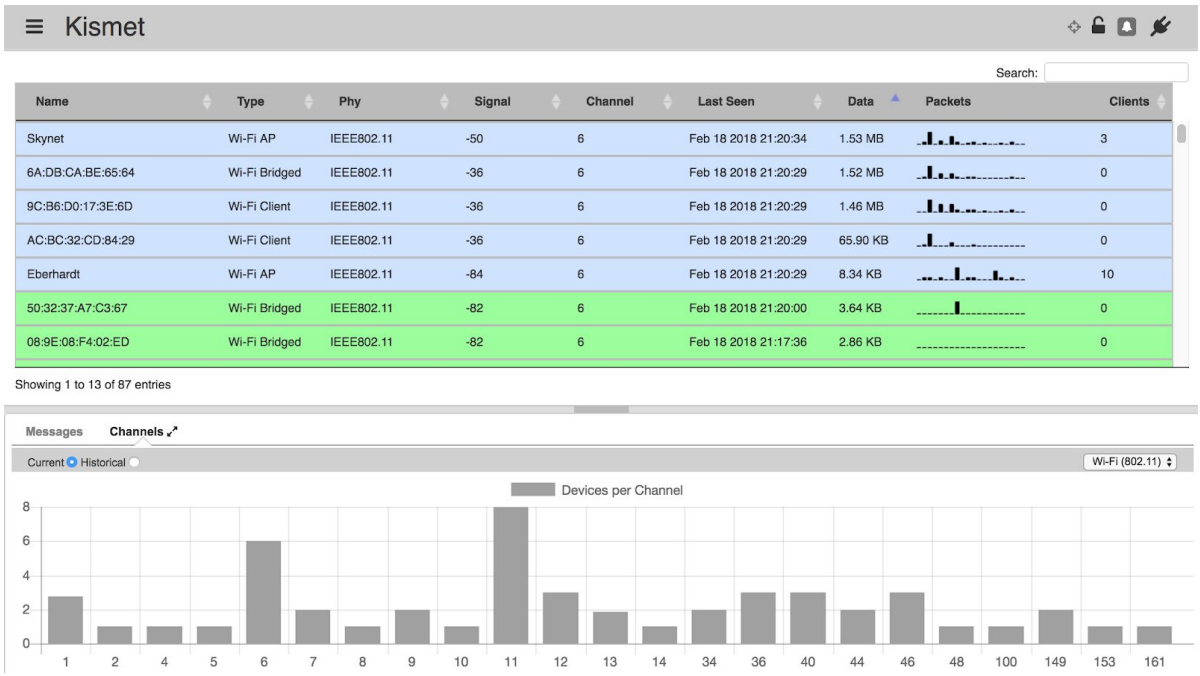
7. Click **Enable Source**



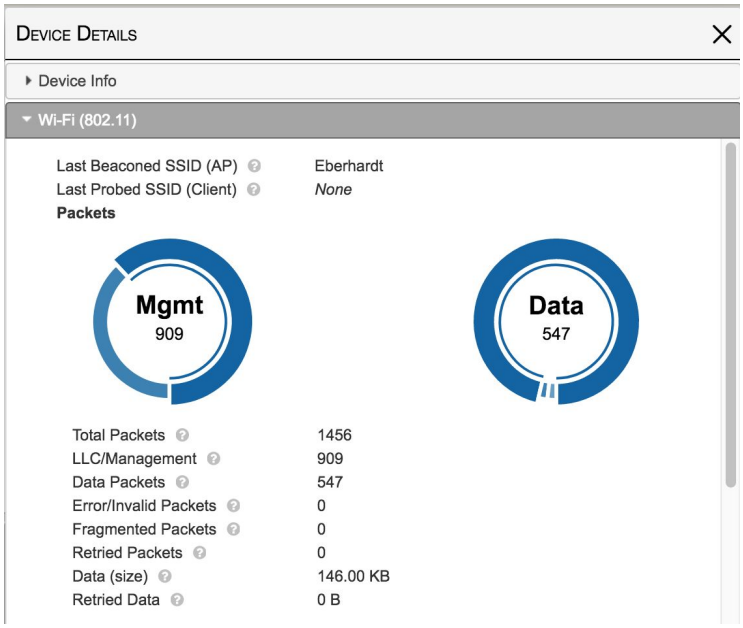
8. If successful you should see the following:



Task 2 - Explore the Kismet Dashboard

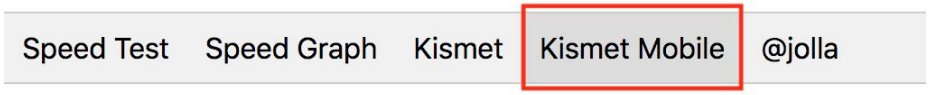


Select a Wi-Fi AP or Wi-Fi Client and explore the captured details:

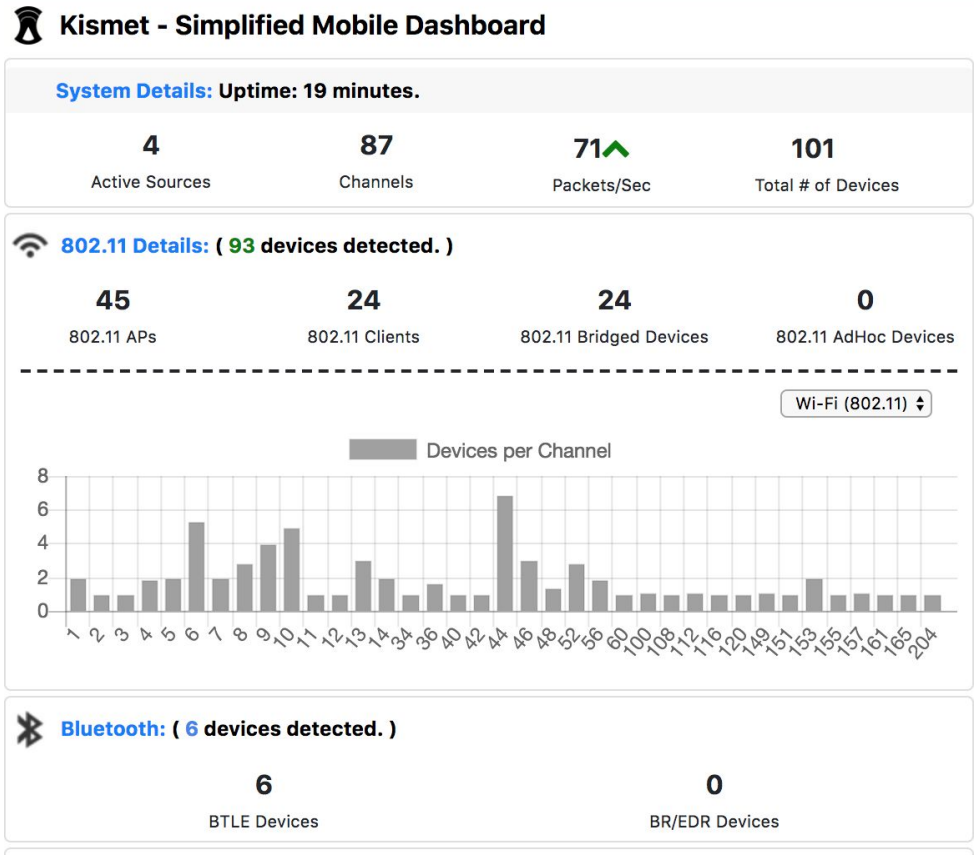


Task 3 - Use an Android or iOS device to explore the Kismet Mobile dashboard

1. Open up a web browser on your mobile device and navigate to the homepage of your WLAN Pi and select 'Kismet Mobile'



2. Explore



Lab 2 - H.O.R.S.T - (Highly Optimized Radio Scanning Tool)

"horst" is a small, lightweight IEEE802.11 wireless LAN analyzer with a text interface. Its basic function is similar to tcpdump, Wireshark or Kismet, but it's much smaller and shows different, aggregated information which is not easily available from other tools. It is mainly targeted at debugging wireless LANs with a focus on ad-hoc (IBSS) mode in larger mesh networks. It can be useful to get a quick overview of what's going on on all wireless LAN channels and to identify problems.

Features:

- Shows signal (RSSI) values per station
- Calculates channel utilization ("usage") by adding up the amount of time the packets actually occupy the medium
- "Spectrum Analyzer" shows signal levels and usage per channel
- Graphical packet history, with signal, packet type and physical rate
- Shows all stations per ESSID and the live TSF per node as it is counting
- Detects IBSS "splits" (same ESSID but different BSSID – this is a common driver problem)
- Statistics of packets/bytes per physical rate and per packet type
- Has some support for mesh protocols (OLSR and batman)
- Can filter specific packet types, source addresses or BSSIDs
- Client/server support for monitoring on remote nodes

More details: https://github.com/br101/horst

Developer: Bruno Randolf - @spiralsun69

Tasks:

1. Connect to the WLAN Pi using SSH from a Mac or PC
2. Launch HORST
3. Explore HORST interface
4. Use HORST to analyze Wi-Fi traffic
5. Analyze Traffic on different Wi-Fi channels
6. Use filters to analyze the traffic of a specific device

TASK 1 - CONNECT TO THE WLAN PI USING SSH FROM A MAC OR PC

MACOS - USING TERMINAL

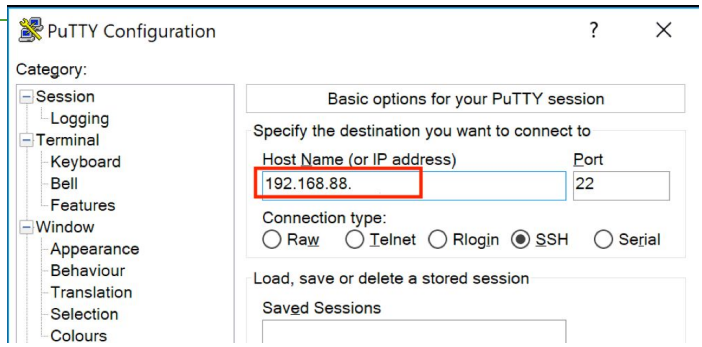
1. Open **terminal** application
2. enter the following:

   $ ssh wlanpi@x.x.x.x
   x.x.x.x = IP address of WLAN Pi

   If successful you should see something like this:

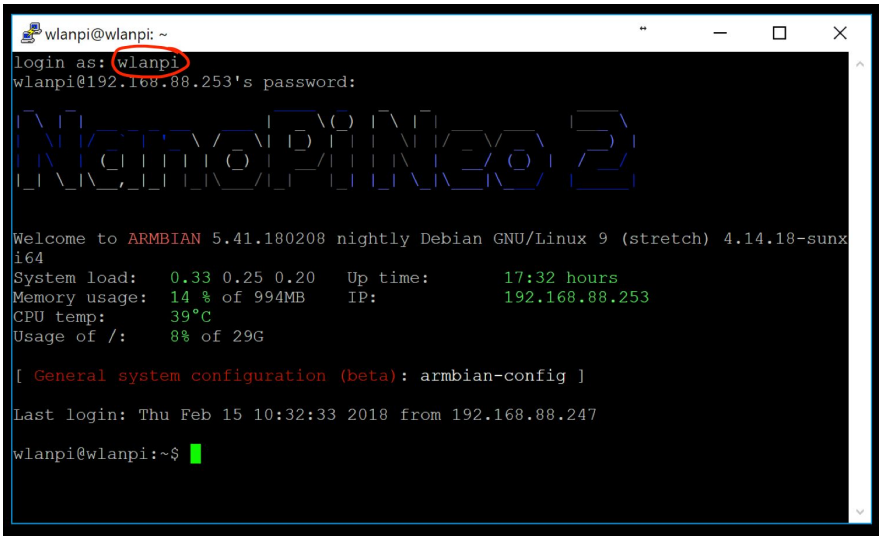
WINDOWS - USING PUTTY

1. Open Putty
2. Enter IP address of WLAN Pi
3. Click Open



4. Login as: **wlanpi**
5. Password: **wlanpi**

If successful you should see this:



Task 2 - Launch HORST

1. Kill network processes that will conflict with HORST

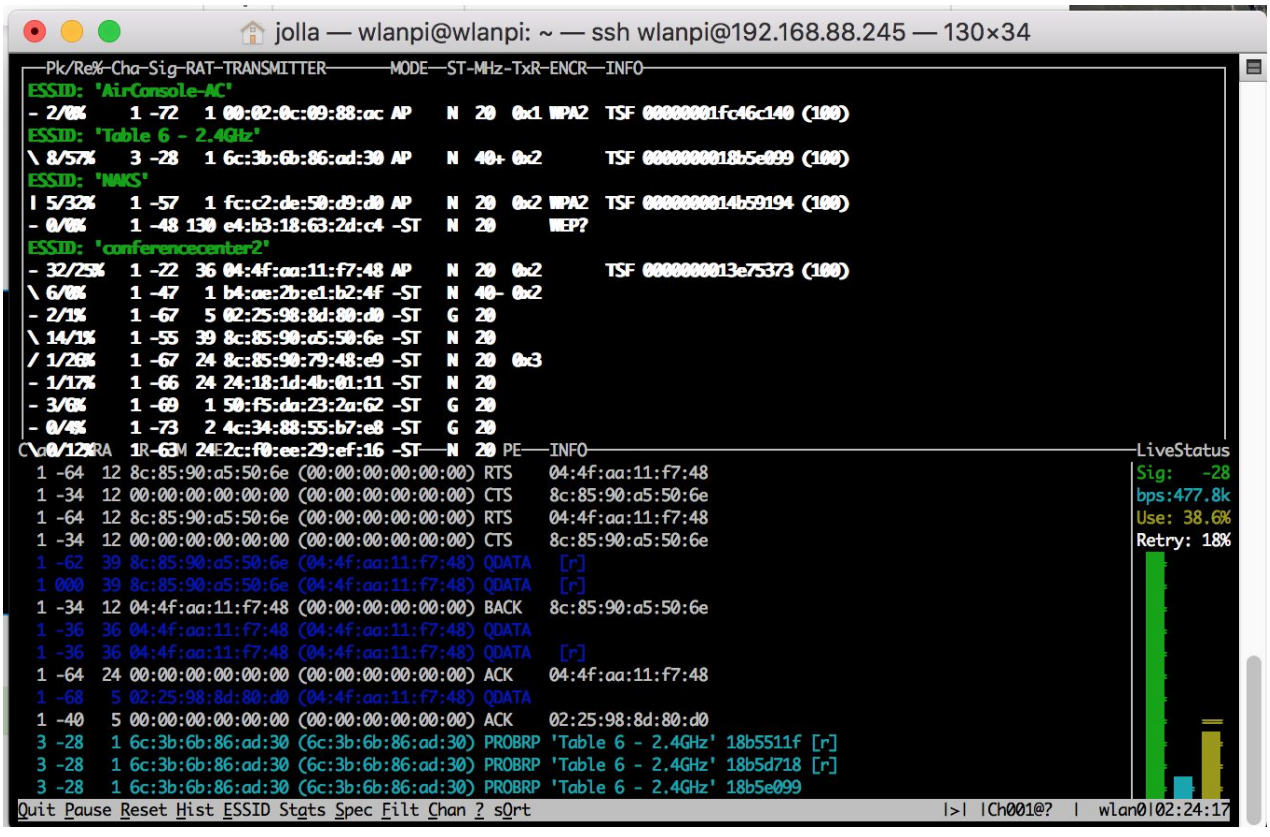
   $ sudo airmon-ng check kill
   *password = **wlanpi***

2. Launch HORST

   $ sudo horst

```
Last login: Fri Feb 16 02:19:52 2018 from 192.168.88.243

wlanpi@wlanpi:~$ sudo airmon-ng check kill
[[sudo] password for wlanpi:

Killing these processes:

  PID Name
  835 wpa_supplicant
 1082 dhclient

wlanpi@wlanpi:~$ sudo horst
```

Task 3 - Explore HORST interface



Task 3 - Explore HORST interface (cont.)

The initial (main) screen is split into three parts. The upper area shows a list of aggregated "node" information, the most useful information about each sender which was discovered, one per line:

/	"Spinner" shows there is activity
Pk	Percentage of this node's packets in relation to all received packets
Re%	Percentage of retried frames of all frames this node sent
Cha	Channel number
Sig	Signal value (RSSI) in dBm
RAT	Physical data rate
TRANSMITTER	MAC address of sender
MODE	Operating Mode (AP, AHD, PRB, STA, WDS), see "NAMES AND ABBREVIATIONS"
ENCR	Encryption (WPA1, WPA2, WEP)
ESSID	ESSID
INFO	Additional info like "BATMAN", IP address...

The lower area shows a scrolling list of packets as they come in:

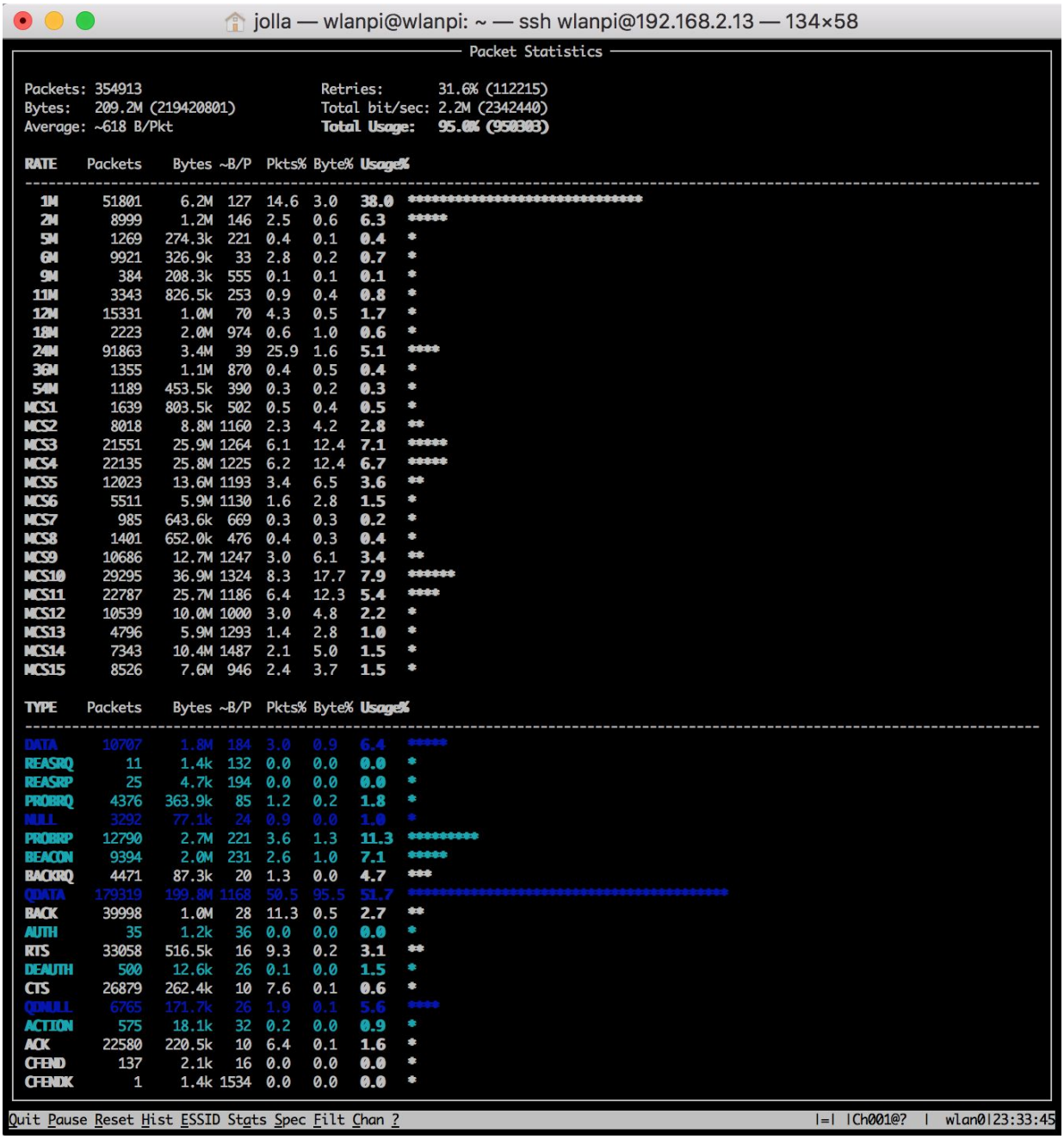
Cha	Channel number
Sig	Signal value (RSSI) in dBm
RAT	Physical data rate
TRANSMITTER	MAC address of sender
BSSID	BSSID
TYPE	Packet type, see "NAMES AND ABBREVIATIONS"
INFO	Additional info like ESSID, TFS, IP address...

HORST KEYBOARD COMMANDS

Key	Command	Description
q	Quit	Quits HORST
p or <space>	Pause	Can be used to pause/resume horst. When horst is paused it will loose packets received in the meantime.
r	Reset	Clears all history and aggregated statistical data.
h	History	The history screen scrolls from right to left and shows a bar for each packet indicating the signal level. In the line below that, the packet type is indicated by one character (See NAMES AND ABBREVIATIONS) and the rough physical data rate is indicated below that in blue.
e	ESSID	The ESSID screen groups information by ESSID and shows the mode (AP, IBSS), the MAC address of the sender, the BSSID, the TSF, the beacon interval, the channel, the signal, a "W" when encryption is used and the IP address if known.
a	Statistics	The statistics screen groups packets by physical rate and by packet type and shows other kinds of aggregated and statistical information based on packets.
s	Spectrum Analyzer	The "poor mans spectrum analyzer" screen is only really useful when horst is started with the -s option or the "Automatically change channel" option is selected in the "Chan" settings, or the config option channel_scan is set. It shows the available channels horizontally and vertical bars for each channel: Signal in green Physical rate in blue Channel usage in orange/brown By pressing the 'n' key, the display can be changed to show only the average signal level on each channel and the last 4 digits of the MAC address of the individual nodes at the level (height) they were received. This can give a quick graphical overview of the distance of nodes.
f	Filters	This configuration dialog can be used to define the active filters.
c	Channel Settings	This configuration dialog can be used to change the channel changing behaviour of horst or to change to a different channel manually.
o	Sort	Only active in the main screen, can be used to sort the node list in the upper area by Signal, Time, BSSID or Channel.

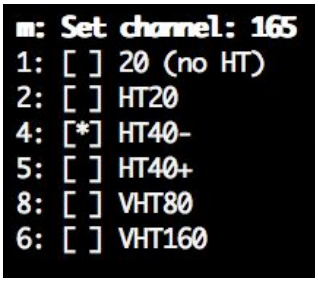
Task 4 - Analyze Wi-Fi traffic Statistics

1. Press **a** to bring up the statistics screen
2. Explore



Task 5 - Change the channel you are analyzing

1. Press **c** to bring up the channel settings
2. Press **m** to set channel
   a. Type **165**, press **enter**
3. Press **4** to change to **HT40-**
4. Press **enter** to apply
5. Press **r** to clear history



Supported Channel Configurations:

2.4 GHz Channels	Channel Width
1	HT40+
2	HT40+
3	HT40+
4	HT40+
5	HT40+-
6	HT40+-
7	HT40+-
8	HT40+-
9	HT40+-
10	HT40+-
11	HT40-
12	HT40-
13	HT40-
14	HT20

5 GHz Channels	Supported Configuration
36	VHT80+
40	VHT80+-
44	VHT80+-
48	VHT80+-
52	VHT80+-
56	VHT80+-
60	VHT80+-
64	VHT80-
100	VHT80+
104	VHT80+-
108	VHT80+-
112	VHT80+-
116	VHT80+-
120	VHT80+-

5 GHz Channels (cont.)	Supported Configuration
124	VHT80+-
128	VHT80+-
132	VHT80+-
136	VHT80+-
140	VHT80-
149	VHT80+
153	VHT80+-
157	VHT80+-
161	VHT80+-
165	HT40-

Task 6 - Use filters to show traffic of a specific device

Test 1 - Analyze a speed test using 5 GHz

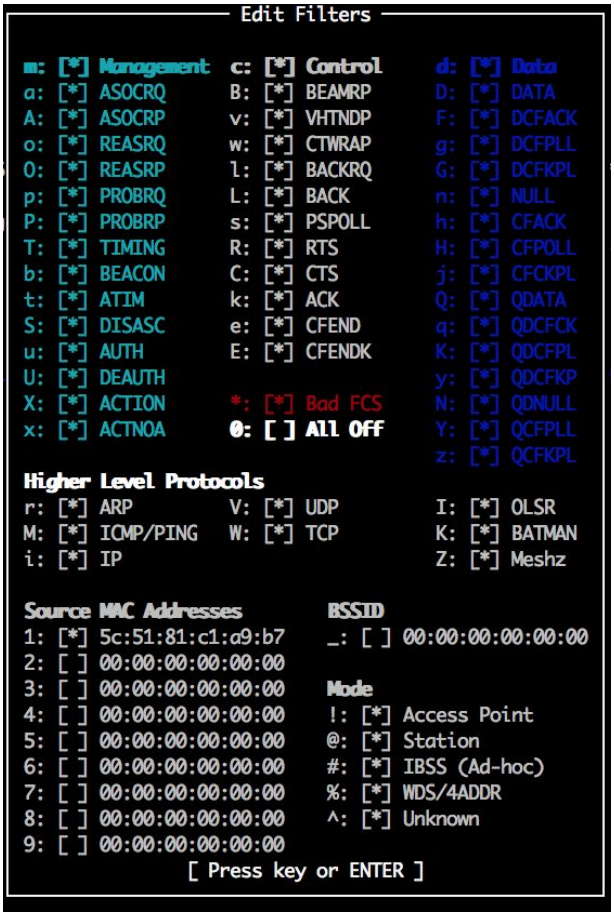
1. Press **f** to bring up the filter settings
2. Press **1** to enter a source MAC address
   a. Enter the MAC address:

      5c:51:81:c1:a9:b7

   b. press **enter** to save
3. Press **enter** to apply
4. Press **r** to clear history
5. Explore the traffic for the filtered device while running a speed test

Test 2 - Repeat using 2.4 GHz

1. Change HORST to monitor the 2.4 GHz channel of your tables 2.4 GHz SSID
2. Connect the test device to the 2.4 GHz SSID
3. Explore the traffic for the filtered device while performing a speed test



HORST NAMES AND ABBREVIATIONS

802.11 STANDARD FRAMES

MANAGEMENT FRAMES

a	ASOCRQ	Association request
A	ASOCRP	Association response
o	REASRQ	Reassociation request
O	REASRP	Reassociation response
p	PROBRQ	Probe request
P	PROBRP	Probe response
T	TIMING	Timing Advertisement
b	BEACON	Beacon
t	ATIM	ATIM
S	DISASC	Disassociation
u	AUTH	Authentication
U	DEAUTH	Deauthentication
X	ACTION	Action
x	ACTNOA	Action No Ack

DATA FRAMES

D	DATA	Data
F	DCFACK	Data + CF-Ack
g	DCFPLL	Data + CF-Poll
G	DCFKPL	Data + CF-Ack + CF-Poll
n	NULL	Null (no data)
h	CFACK	CF-Ack (no data)
H	CFPOLL	CF-Poll (no data)
j	CFCKPL	CF-Ack + CF-Poll (no data)
Q	QDATA	QoS Data
q	QDCFCK	QoS Data + CF-Ack
K	QDCFPL	QoS Data + CF-Poll
y	QDCFKP	QoS Data + CF-Ack + CF-Poll
N	QDNULL	QoS Null (no data)
Y	QCFPLL	QoS CF-Poll (no data)
z	QCFKPL	QoS CF-Ack + CF-Poll (no data)
*****	BADFCS	Bad frame checksum

CONTROL FRAMES

B	BEAMRP	Beamforming Report Poll
v	VHTNDP	VHT NDP Announcement
w	CTWRAP	Control Wrapper
l	BACKRQ	Block Ack Request
L	BACK	Block Ack
s	PSPOLL	PS-Poll
R	RTS	RTS
C	CTS	CTS
k	ACK	ACK
e	CFEND	CF-End
E	CFENDK	CF-End + CF-Ack

Packet types

Similar to 802.11 frames above but higher level and as a bit field (types can overlap, e.g. DATA + IP) and including more information, like IP, ARP, BATMAN, OLSR…

CTRL	0x000001	WLAN Control frame
MGMT	0x000002	WLAN Management frame
DATA	0x000004	WLAN Data frame
BADFCS	0x000008	WLAN frame checksum (FCS) bad
BEACON	0x000010	WLAN beacon frame
PROBE	0x000020	WLAN probe request or response
ASSOC	0x000040	WLAN association request/response frame
AUTH	0x000080	WLAN authentication frame
RTSCTS	0x000100	WLAN RTS or CTS
ACK	0x000200	WLAN ACK or BlockACK
NULL	0x000400	WLAN NULL Data frame
QDATA	0x000800	WLAN QoS Data frame (WME/WMM)
ARP	0x001000	ARP packet
IP	0x002000	IP packet
ICMP	0x004000	IP ICMP packet
UDP	0x008000	IP UDP
TCP	0x010000	IP TCP
OLSR	0x020000	OLSR protocol
BATMAN	0x040000	BATMAND Layer3 or BATMAN-ADV Layer 2 frame
MESHZ	0x080000	MeshCruzer protocol

Operating modes

Bit field of operating mode type which is inferred from received packets. Modes may overlap, i.e. it is common to see STA and PRB at the same time.

AP	0x01	Access Point (AP)
ADH	0x02	Ad-hoc node
STA	0x04	Station (AP client)
PRB	0x08	Sent PROBE requests
WDS	0x10	WDS or 4 Address frames
UNKNOWN	0x20	Unknown e.g. RTS/CTS or ACK

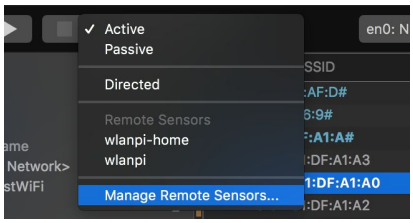
Lab 3 - WiFi Explorer Pro Remote Sensor (Mac only)

WiFi Explorer Pro allows you to connect to a remote platform and perform a passive Wi-Fi scan using a capable Wi-Fi adapter. When a remote sensor is used, the scan results are sent back to WiFi Explorer Pro for its visualization.

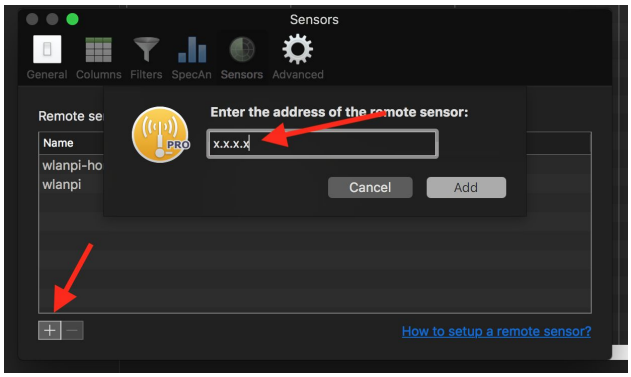
Task - Add WLAN Pi as a remote sensor

1. Install and Launch WiFi Explorer Pro

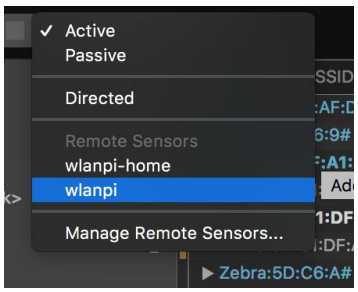
2. Go to Manage Remote Sensor



3. Click '+' to add the WLAN Pi as a new sensor by its IP address.



4. Select the new sensor as the source



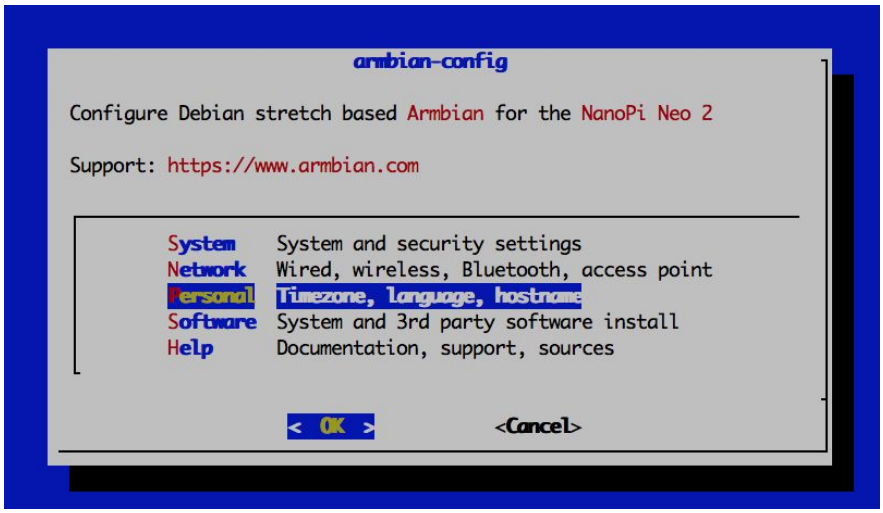
5. Explore results

SOME USEFUL LINUX COMMANDS

CHANGE YOUR LOGIN PASSWORD: **$** passwd

```
[wlanpi@wlanpi:~$ passwd
 Changing password for wlanpi.
[(current) UNIX password:
[Enter new UNIX password:
[Retype new UNIX password:
 passwd: password updated successfully
```

CHANGE HOSTNAME, TIMEZONE, LANGUAGE: **$** sudo armbian-config



UPDATE ALL INSTALLED PACKAGES: **$** sudo apt update && sudo apt upgrade && sudo apt dist-upgrade

INSTALL ADDITIONAL LINUX PACKAGES: **$** sudo apt install [name of package]

CHANGE WHAT APPLICATIONS START ON BOOT: **$** crontab -e