Application of Linux Single Board Computers to Amateur Radio

Willem A Schreüder AC0KQ
willem@prinmath.com

Indian Peaks Radio Club
December 27, 2016

http://www.prinmath.com/ham/talks/

Talk Outline

- **Why Linux SBCs**
- **Getting Started**
- **BPQ Packet/RMS Gateway/APRS iGate**
- **AllStarLink Repeater**
- **Control and Monitoring**
- **SDR**
- ***Don’t freak out about the number of slides***

Single Board Computers

- **Full Linux boxes *(today's topic)***
  - **Raspberry Pi**
  - **Beaglebone**
- **Microcontrollers *(not covered)***
  - **Arduino**
  - **PICAXE**
  - **BASIC Stamp**

Why Linux SBCs?

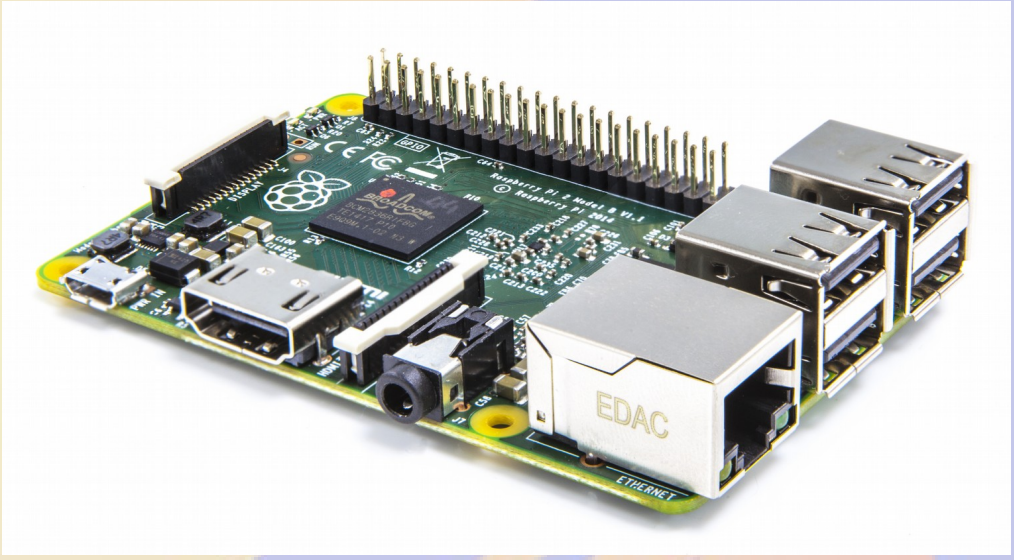
- Runs a full Linux OS
- Usable stand alone computer or server
- Built in connectivity
  - Ethernet networking
  - USB and serial
  - General purpose IO
- Low power (5V 1A)
- Expandable using daughter boards
- Inexpensive ($50 for a working system)

SBC Pros and Cons

- Pros
  - Inexpensive
  - No moving parts
  - 5V power
  - Expandable
- Cons
  - SD cards corrupted by bad power
  - SD card is not a great hard disk

Raspberry Pi

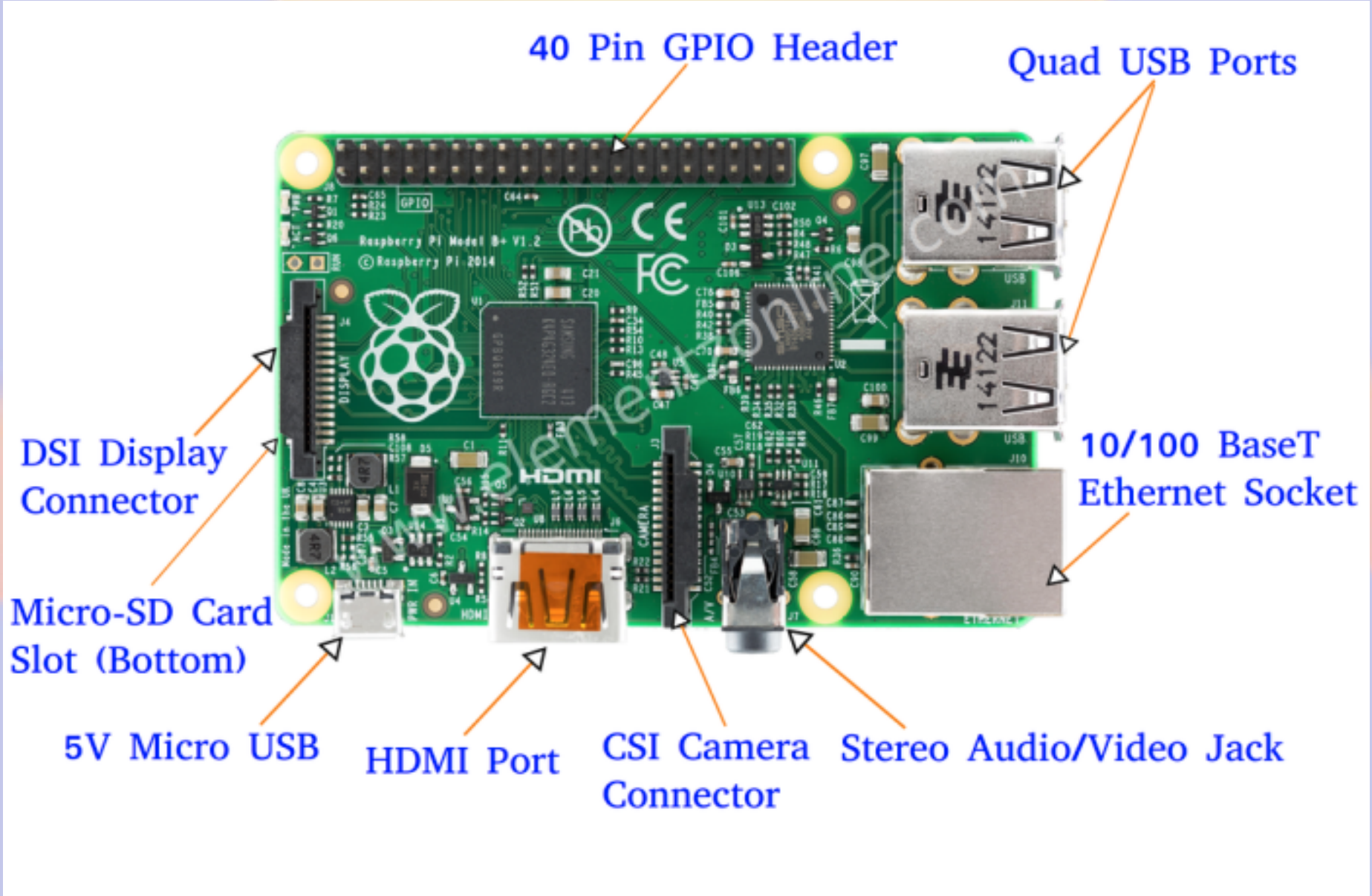
- Most Popular
- Best supported
- rPi3 most powerful
- Lots of USB ports
- Lots of daughterboards
- No analog inputs
- $35 plus SD card



Raspberry Pi models

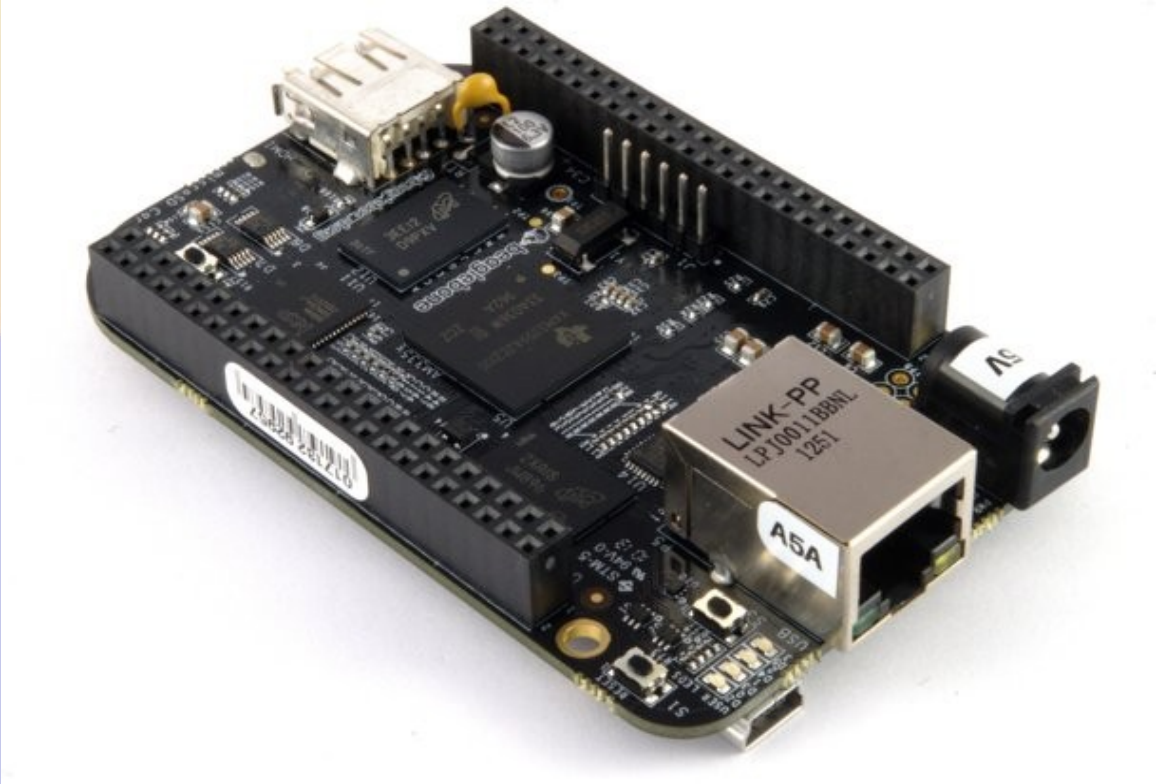
- **Raspberry Pi**
  - **A/A+ 700 MHz CPU & 256MB SDRAM, 1xUSB**
  - **B 700 MHz CPU & 512MB SDRAM, 2xUSB, Ethernet**
  - **B+ 700 MHz CPU & 512MB SDRAM, 4xUSB, Ethernet**
  - **2B 900 MHz Quad A7 & 1GB SDRAM, 4xUSB, Ethernet**
  - **3B 1.2GHz Quad 64bit & 1GB SDRAM, 4xUSB, Ethernet**
- **Compute Module**
  - **700MHz CPU & 512MB SDRAM**
- **Zero**
  - **1GHz CPU & 512MB SDRAM**

Raspberry Pi 2B



Beagle Bone

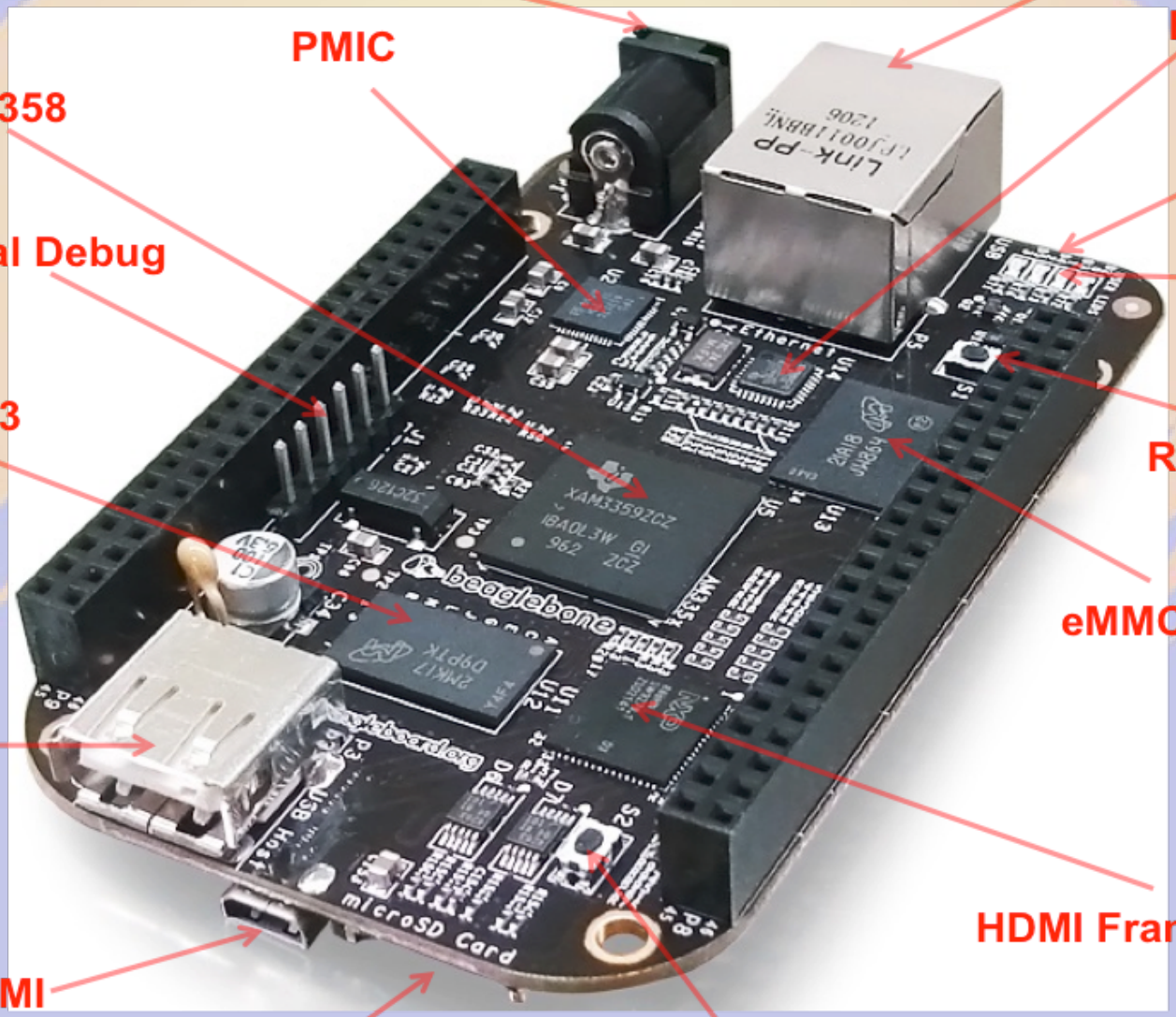
- Less well supported
- Onboard eMMC
- Power & Reset buttons
- More GPIO pins
- 8 analog inputs
- $50 street price



Beaglebone Models

- White
  - Original 720 MHz A8
- Black
  - Most Popular 1GHz A8
- Green
  - Same CPU as Black
  - No barrel power, two Grove connectors
- Industrial
  - Black with extended temperature range

Beagle Bone Black



Other Linux SBCs

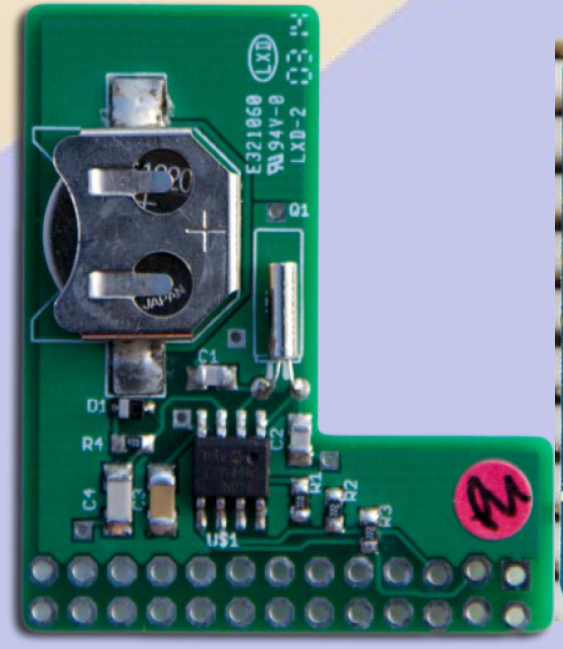
- Examples
  - Intel Edison
  - VoCore
  - Odroid
- Less well supported
- Fewer peripherals
- Sometimes better performance
- Mostly higher priced

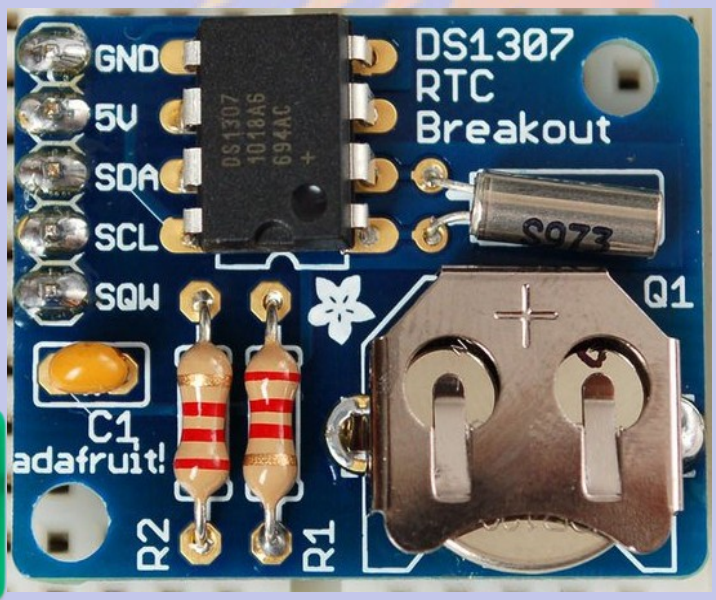
Power and Storage

- Runs on 5V DC
  - Needs clean power
  - Draws 0.5-1.0 A without daughter boards
- Micro SD card storage
  - Finite life
  - Marginal performance
  - Bad power kills SD

Must Have Accessories

- **Micro SD card**
  - **Faster is better**
    - **Class 10**
    - **UHS 1**
    - **UHS 3**
  - **At least 4GB**
    - **16GB is ample**
- **Real time clock**
  - **PiFace Shim RTC**
  - **Adafruit DS1307**
  - **Needed if no network (NTP)**



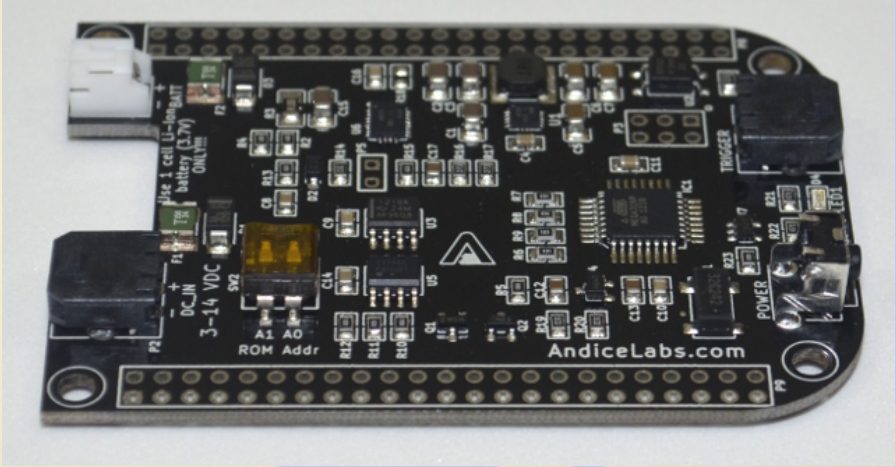


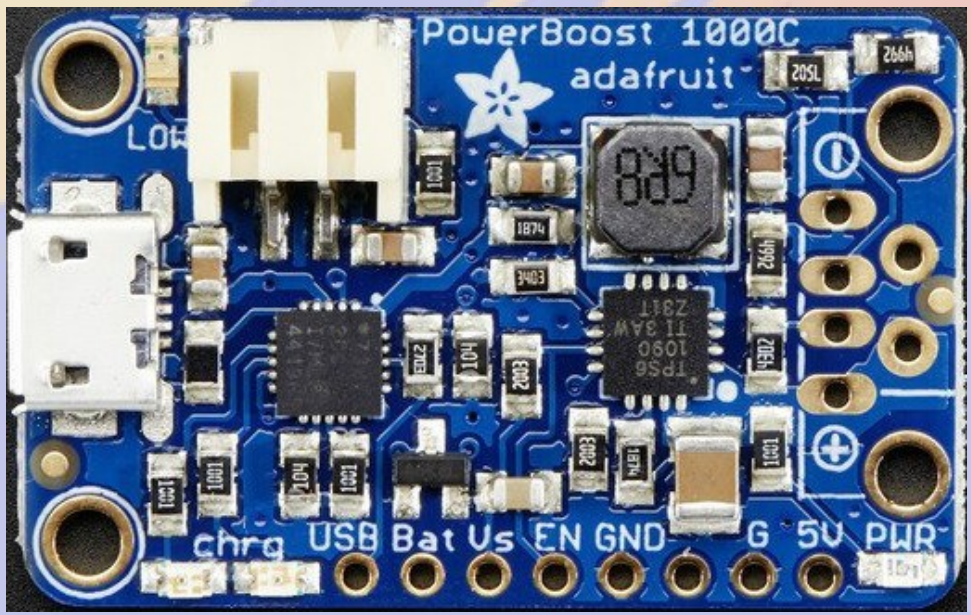
Nice to have

- **Official Raspberry 7” Touchscreen**

Power Control

- **Andice Labs Powercape**
- **Adafruit Powerboost 1000C**
- **Charges and boosts 4V from LIPO battery**



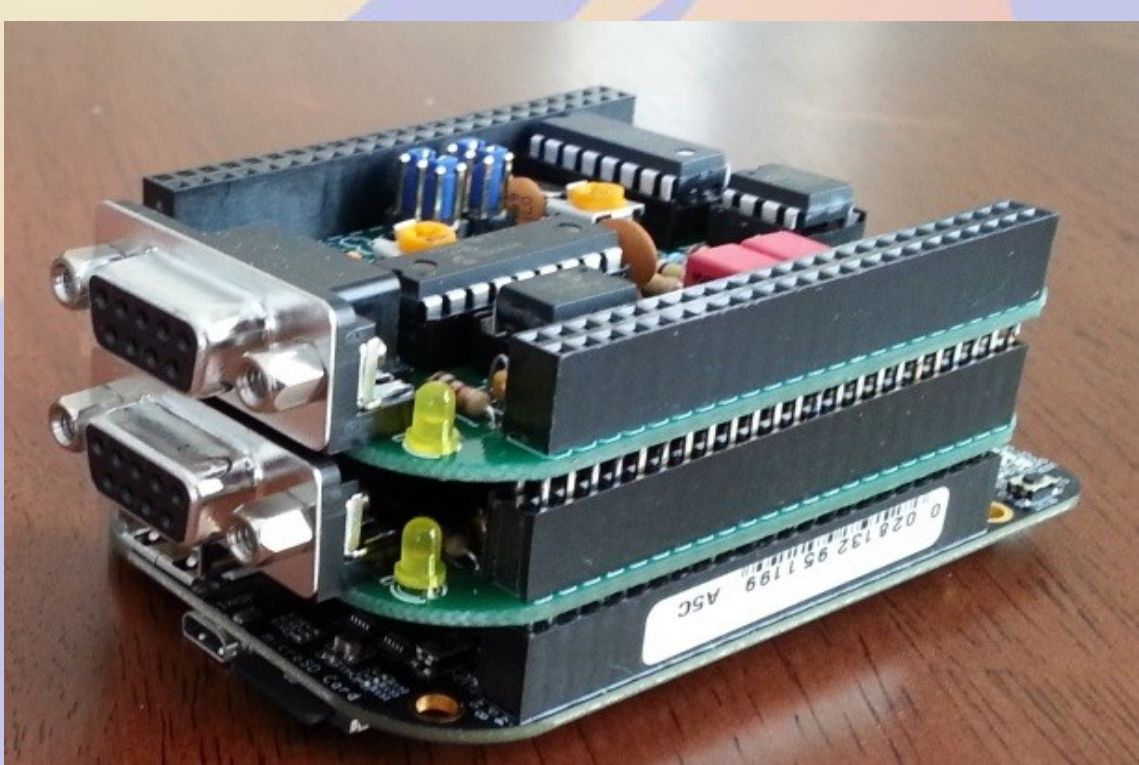


TNC-X/Pi/Black

- Designed by John Hansen W2FS
- Based on PIC Microcontroller
- MX614 Bell 202 modem chip
- KISS interface
  - Serial
  - USB
  - I2C

Why the BB/TNC-Black?

- BBB has 5 serial ports
- Mechanically stable stacked capes
- Powercape battery backup
- Lots of pins for site monitoring
- 50% more expensive



Part 1
Getting Started on the Raspberry Pi

rPi Materials

- Raspberry Pi 2B or 3B
- Micro SD card
- 5V 1A power supply
- USB A to micro USB B cable
- Ethernet cable
- Direct connection
  - Monitor or TV
  - HDMI cable
  - USB keyboard and mouse

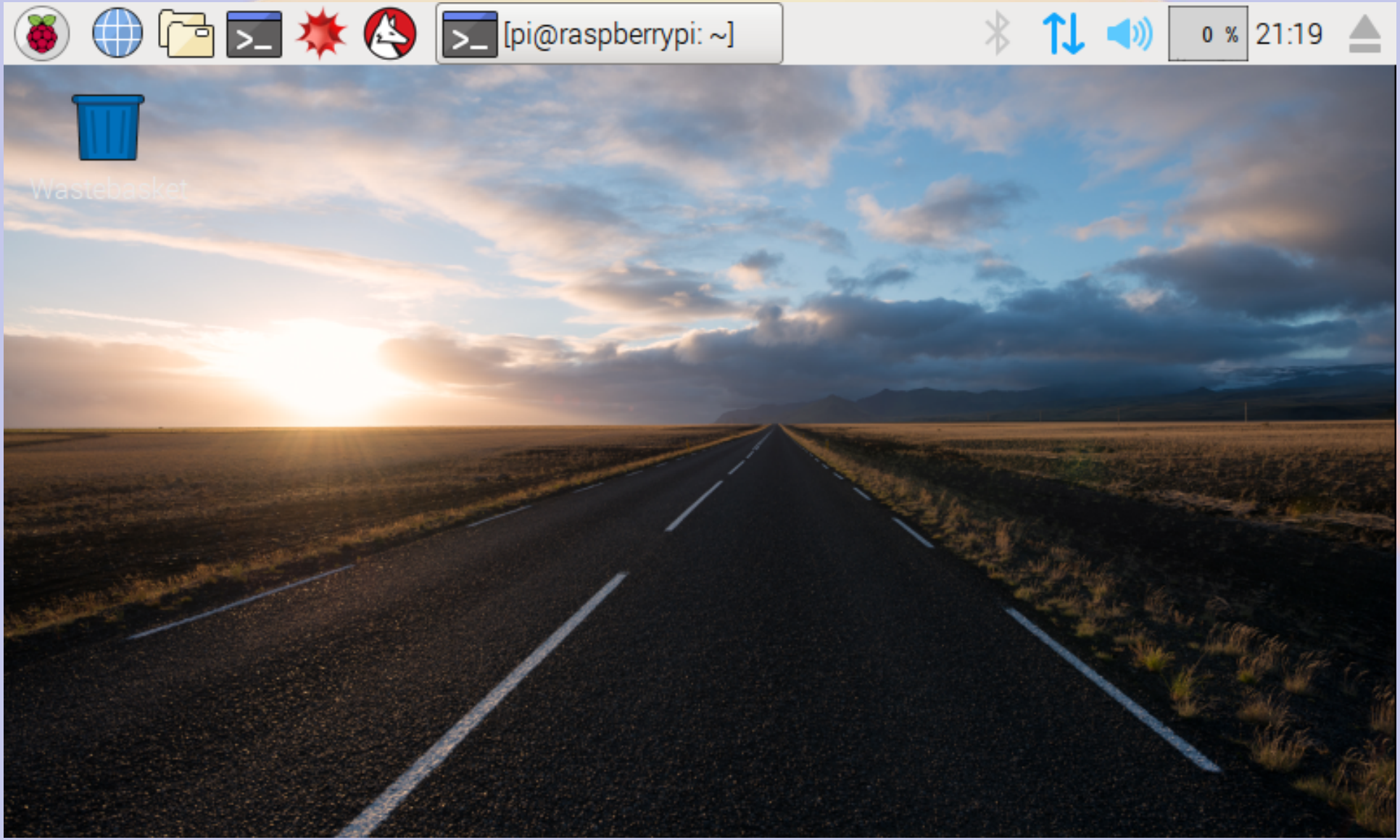
Raspberry OS Choices

- **https://www.raspberrypi.org/downloads/**
- **Debian derivatives are most popular**
  - **Raspbian (Official Supported OS)**
  - **Alternatives are**
    - **NOOBS (New Out Of the Box Software)**
    - **Ubuntu Mate (Ubuntu Desktop)**
    - **Windows 10 IOT (a.k.a. YGBSM)**
    - ***several others, some not Linux based***
- **Debian 8 (Jessie) adopts systemd**
  - **This changes how system programs are run**
    - **No more */etc/init.d/XXX* and */etc/inittab***
    - **Control programs with *systemctl***

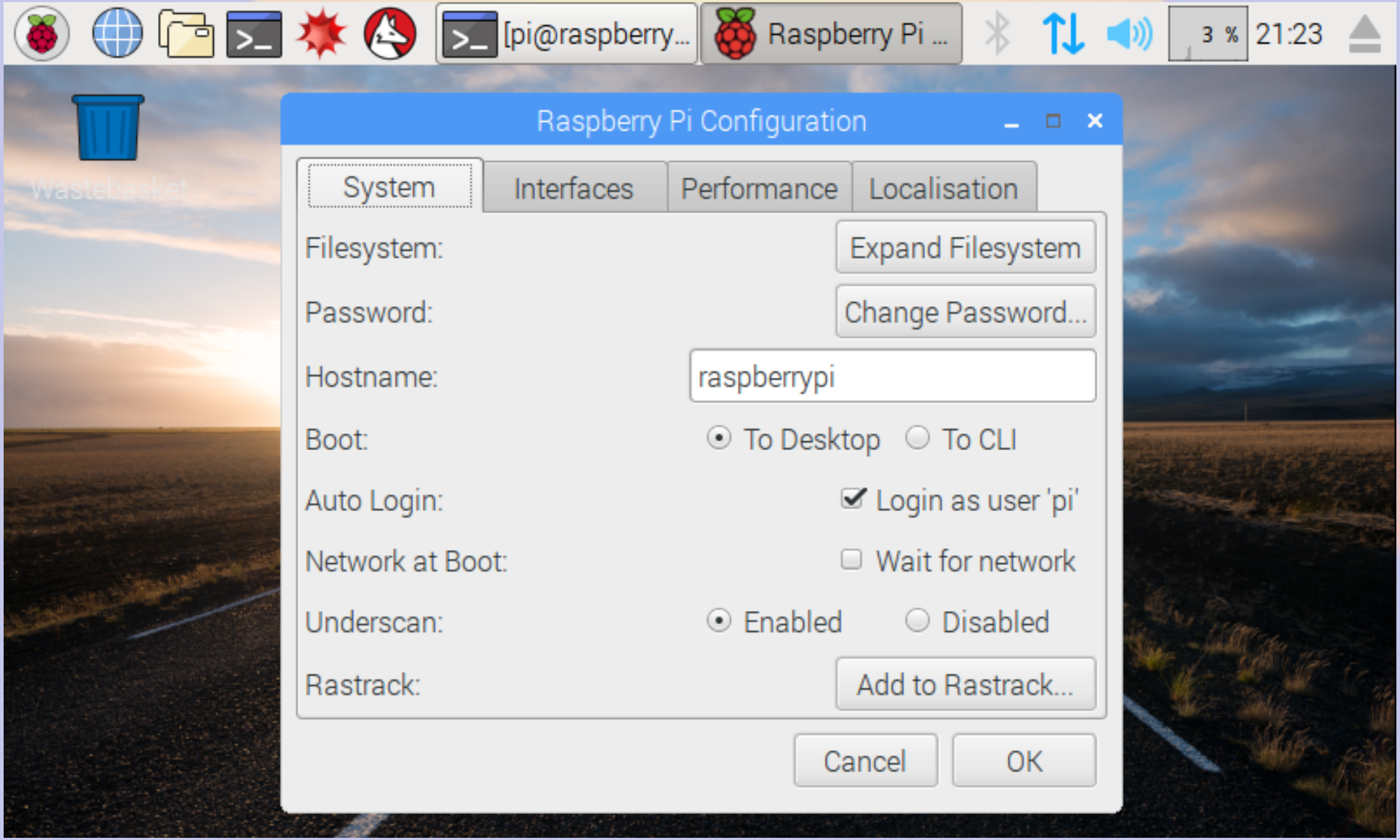
Burning the Image

- **Unzip image**
  - `2016-09-23-raspbian-jesse.img`
- **Linux or OSX command line**
  - **dd if=2016-09-23-raspbian-jesse.img of=/dev/mmcblk0**
  - **sync;sync**
- **Windows**
  - **Download Win32DiskImager**
  - **Select image file name**
  - **Select SD card drive letter**
  - **Click *Write***

First boot with Pixel



Raspberry > Preferences > Raspberry Pi Configuration



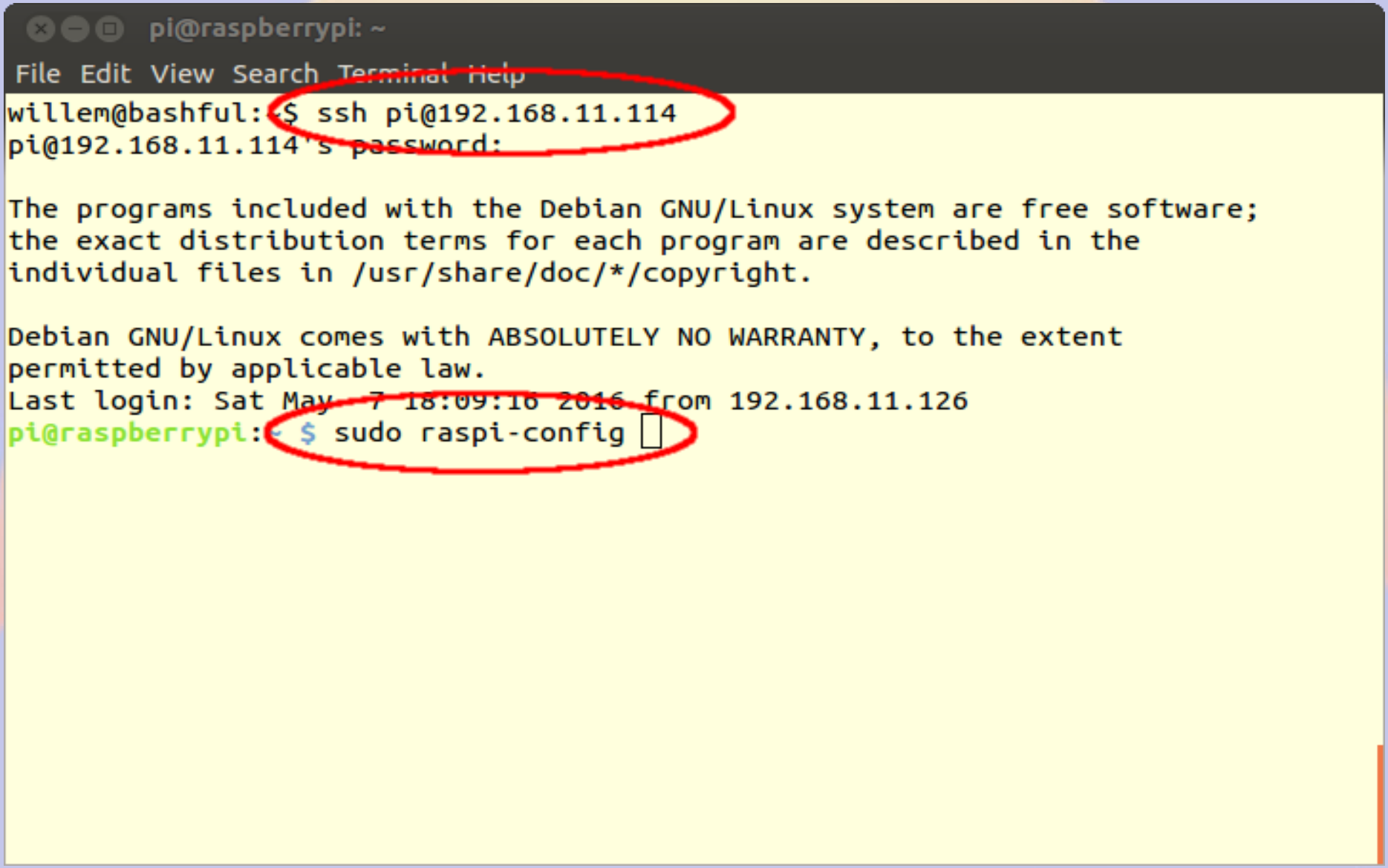
Remote Access

- Do `ifconfig` from the keyboard
- Look for hostname *raspberrypi*
  - Assign a reserved IP address and add DNS
- Advantages of using ssh
  - Can access the device from anywhere
  - Automatic logins using authorized_keys
  - Text based menus work great remotely

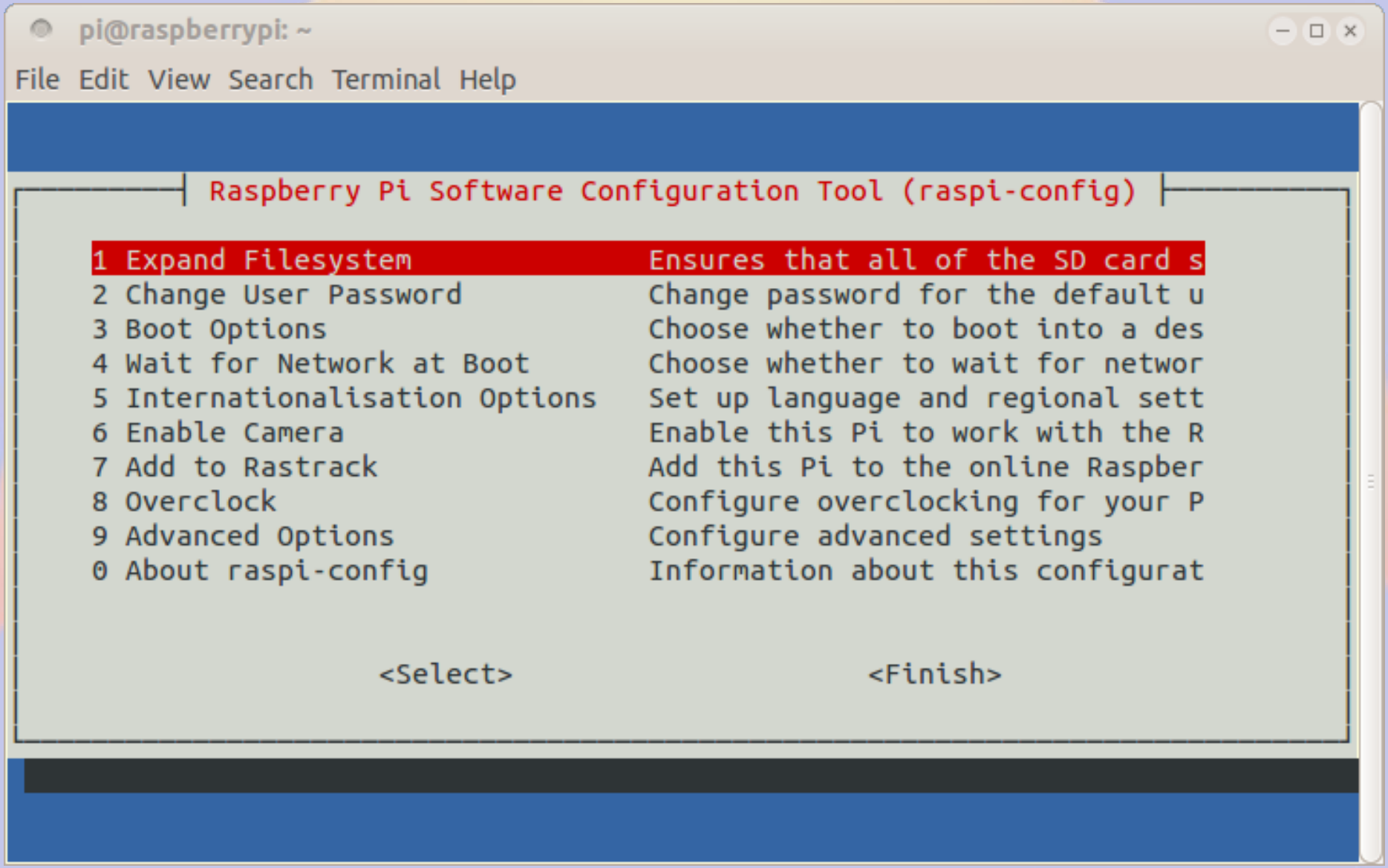
Configuring rPi

- **Plug in keyboard, mouse and screen**
  - **Menu >Preferences > rPi Configuration**
- **Plug in ethernet cable and locate the IP address**
  - **Default hostname is *raspberrypi***
  - **ssh pi@XXX.XXX.XXX.XXX**
    - **password raspberry**

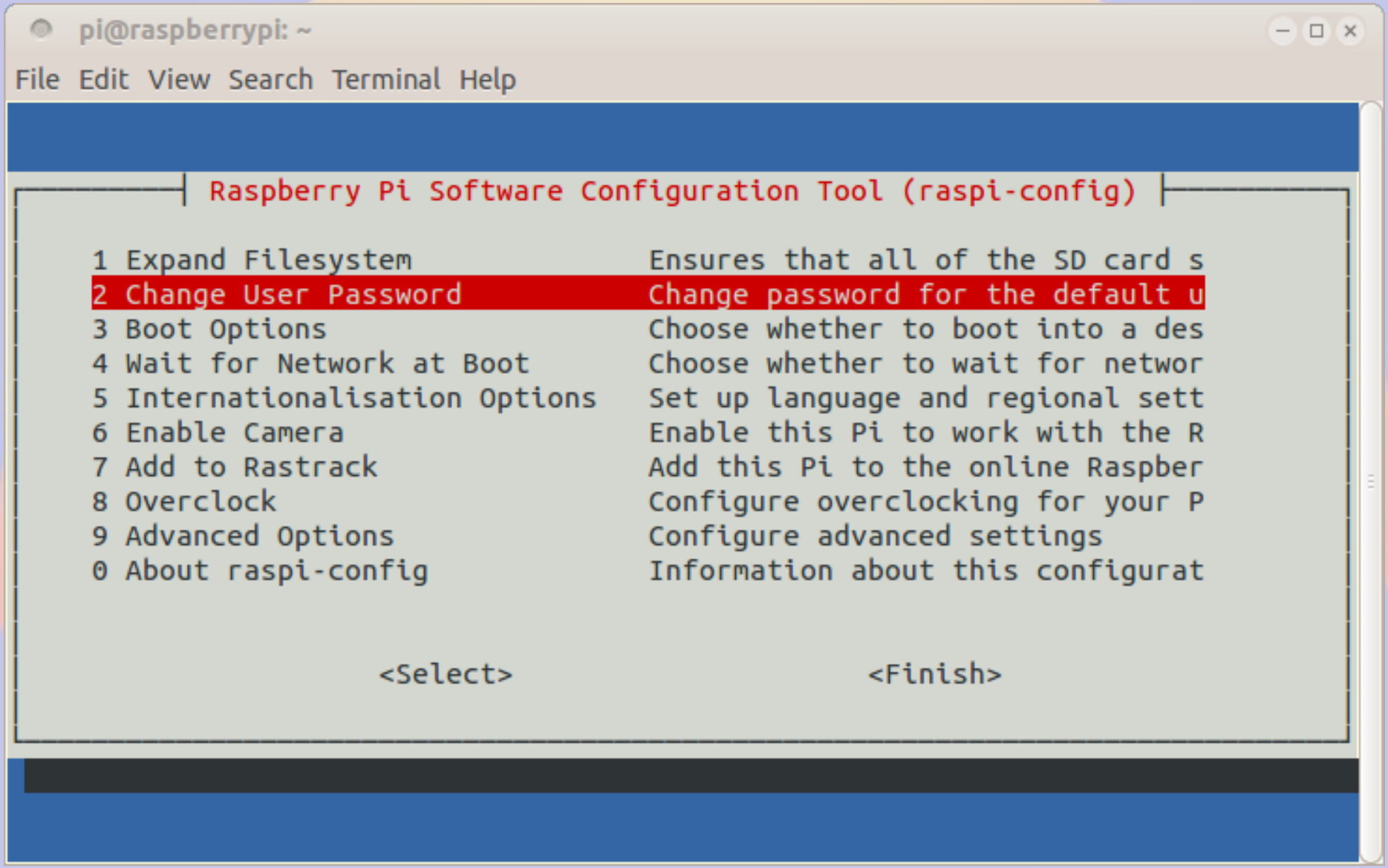
Running raspi-config



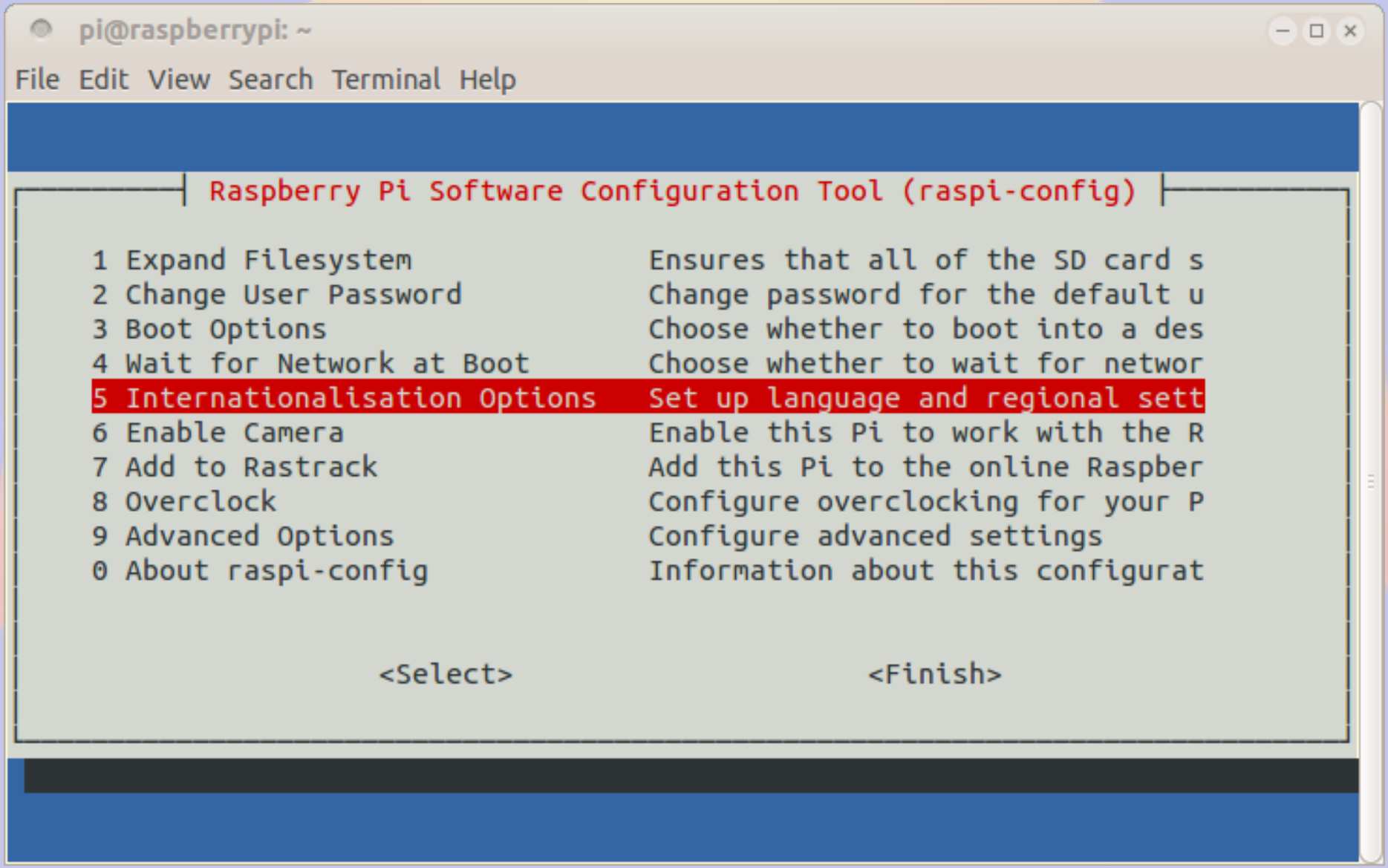
Expand file system



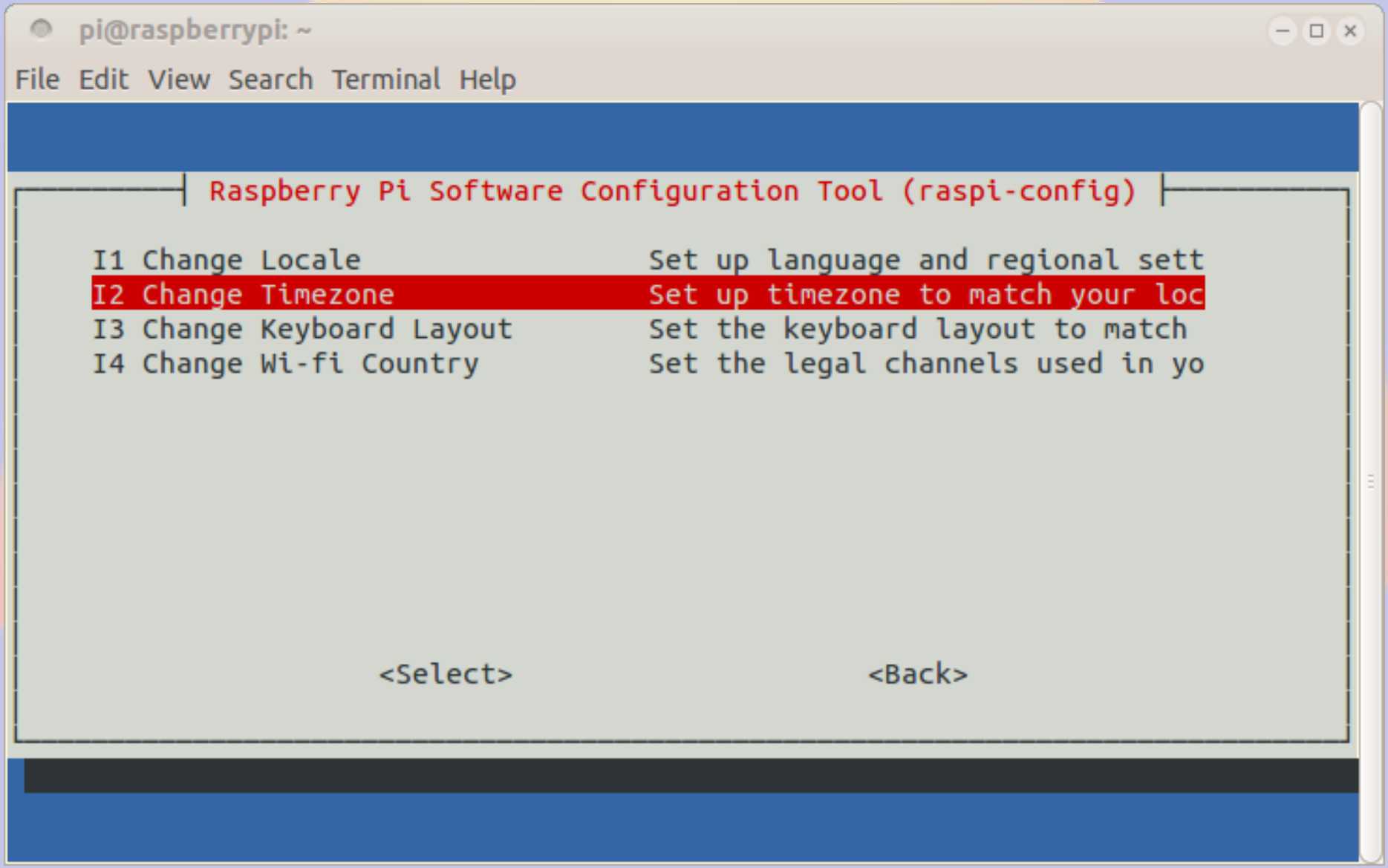
Change the Password



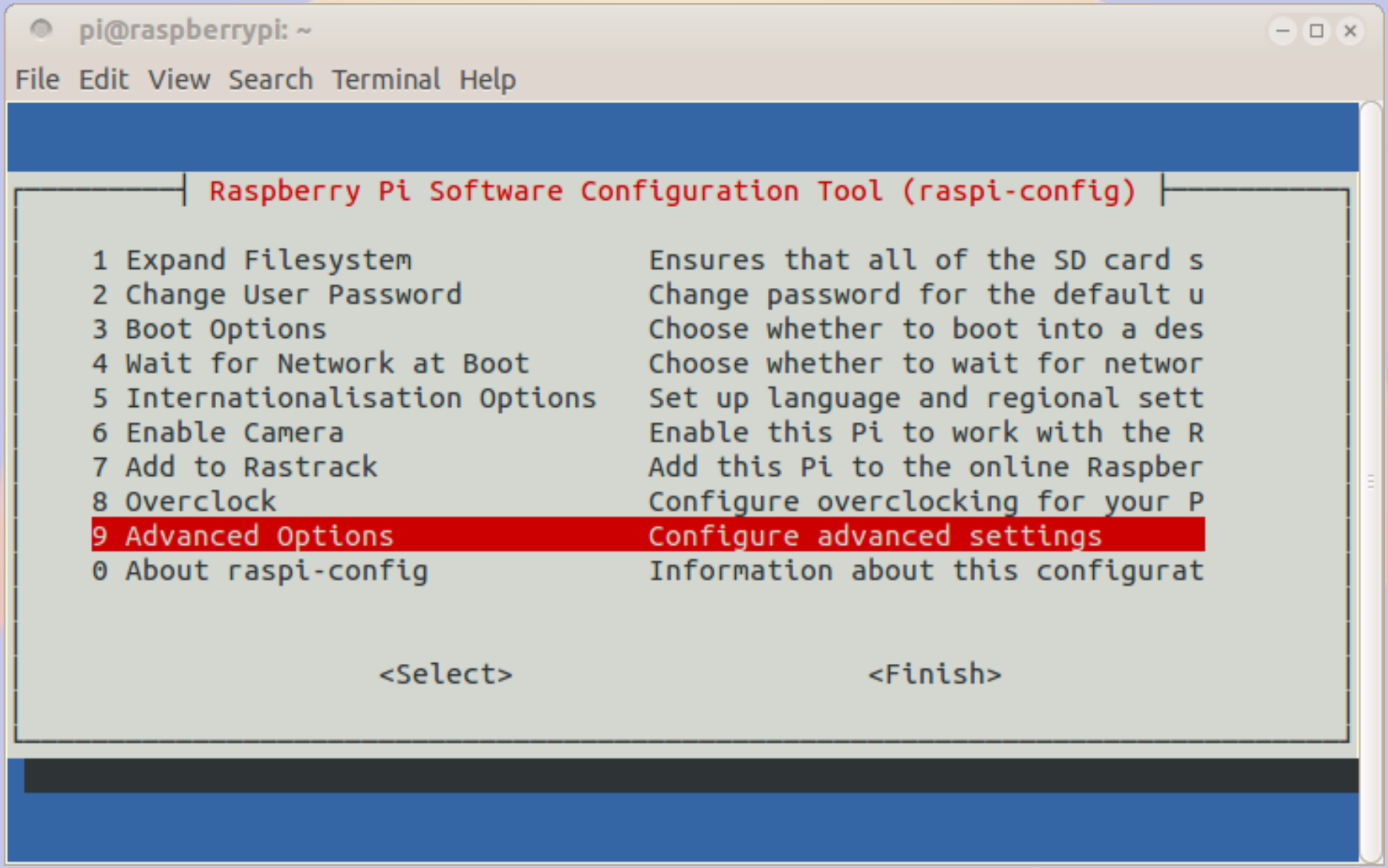
Set timezone 1



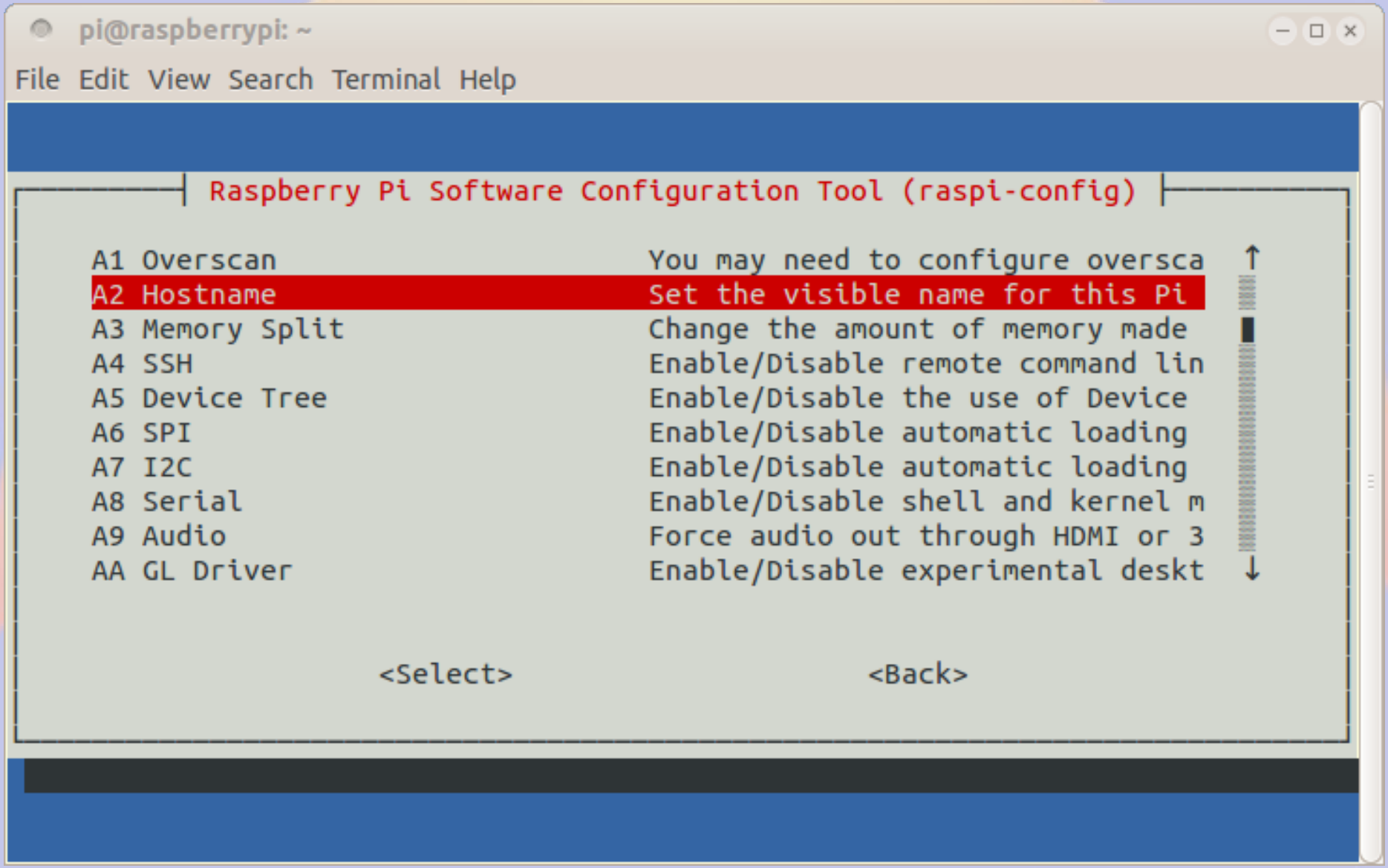
Set timezone 2



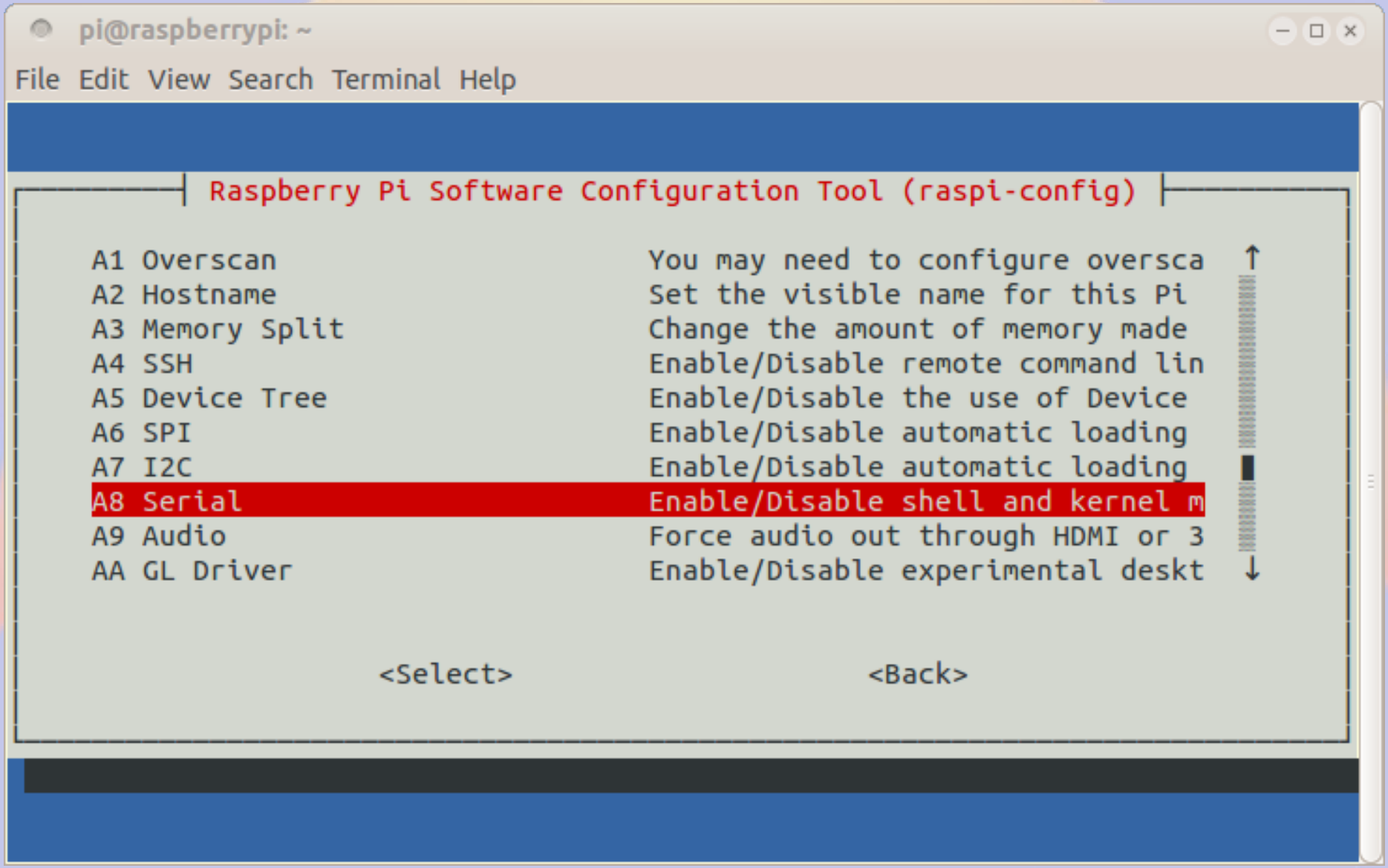
Advanced Options



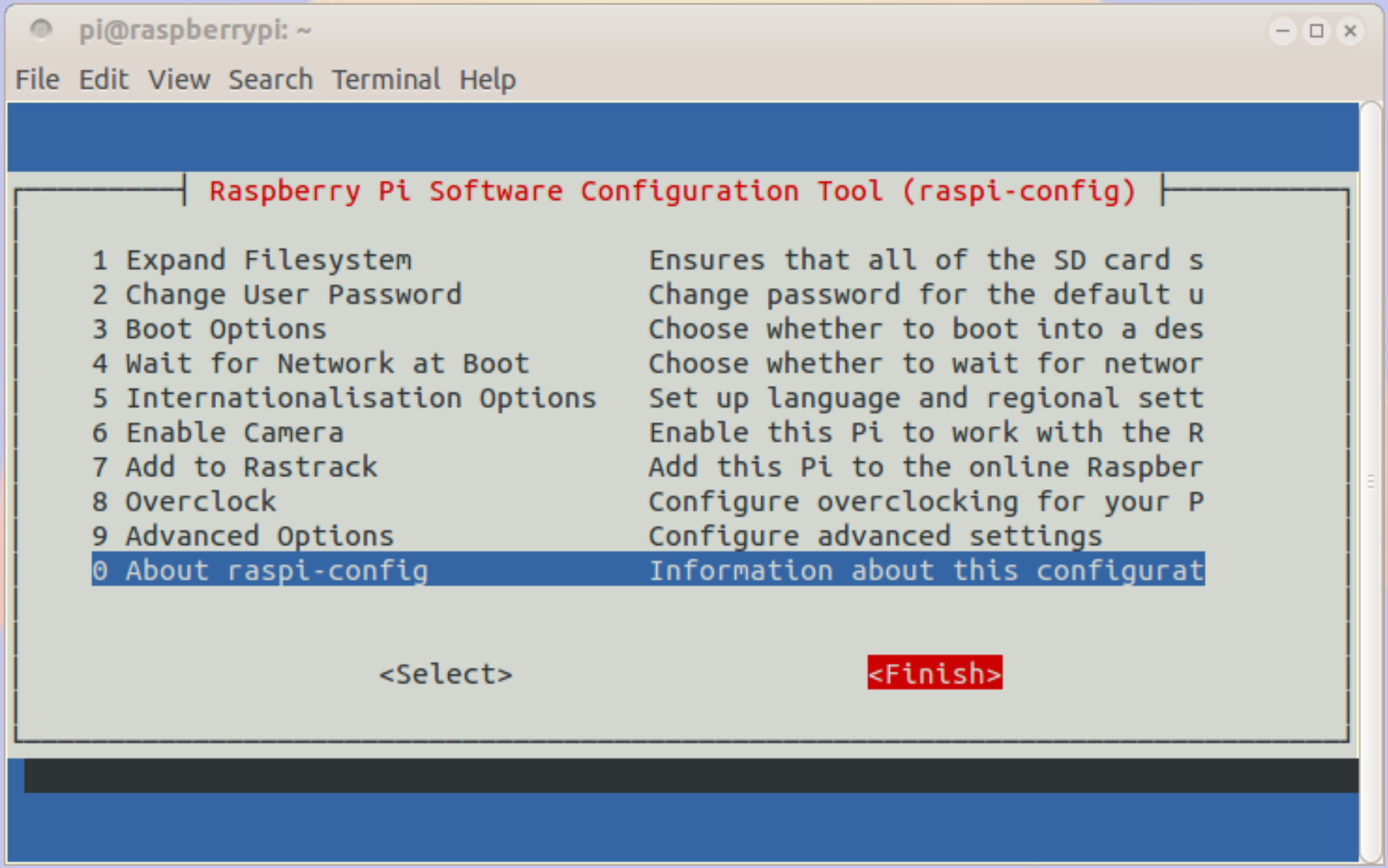
Set Hostname



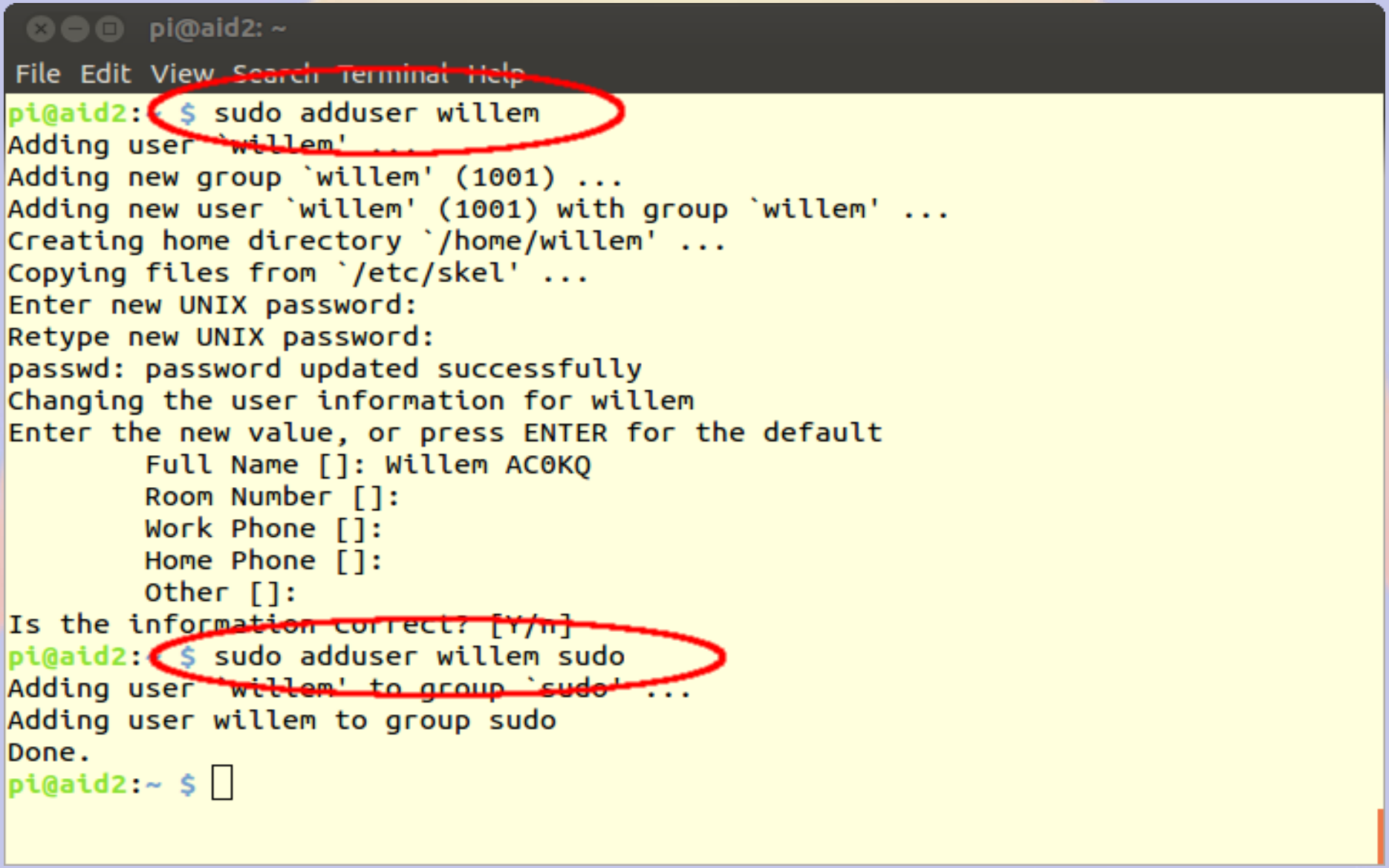
Disable serial login



Reboot and log in again



Add user willem



Part 2
BPQ BBS/RMS/iGate

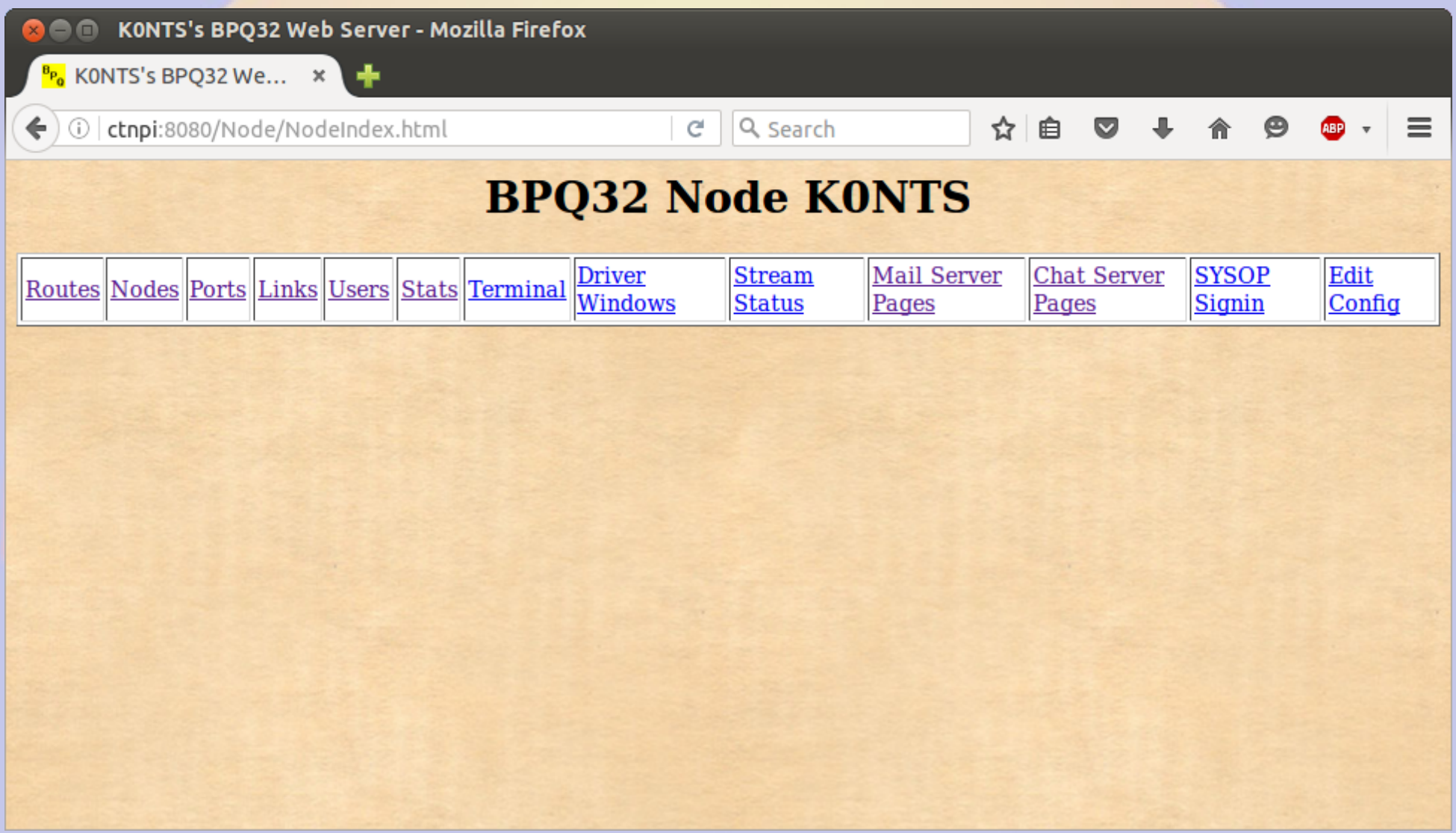
What is BPQ?

- NET/ROM compatible Packet Switch
  - Multiple ports
    - As many I2C or serial ports as you have available
  - Multiple protocols
    - Packet, Pactor, IP
  - Multiple functions
    - BBS, Chat, APRS

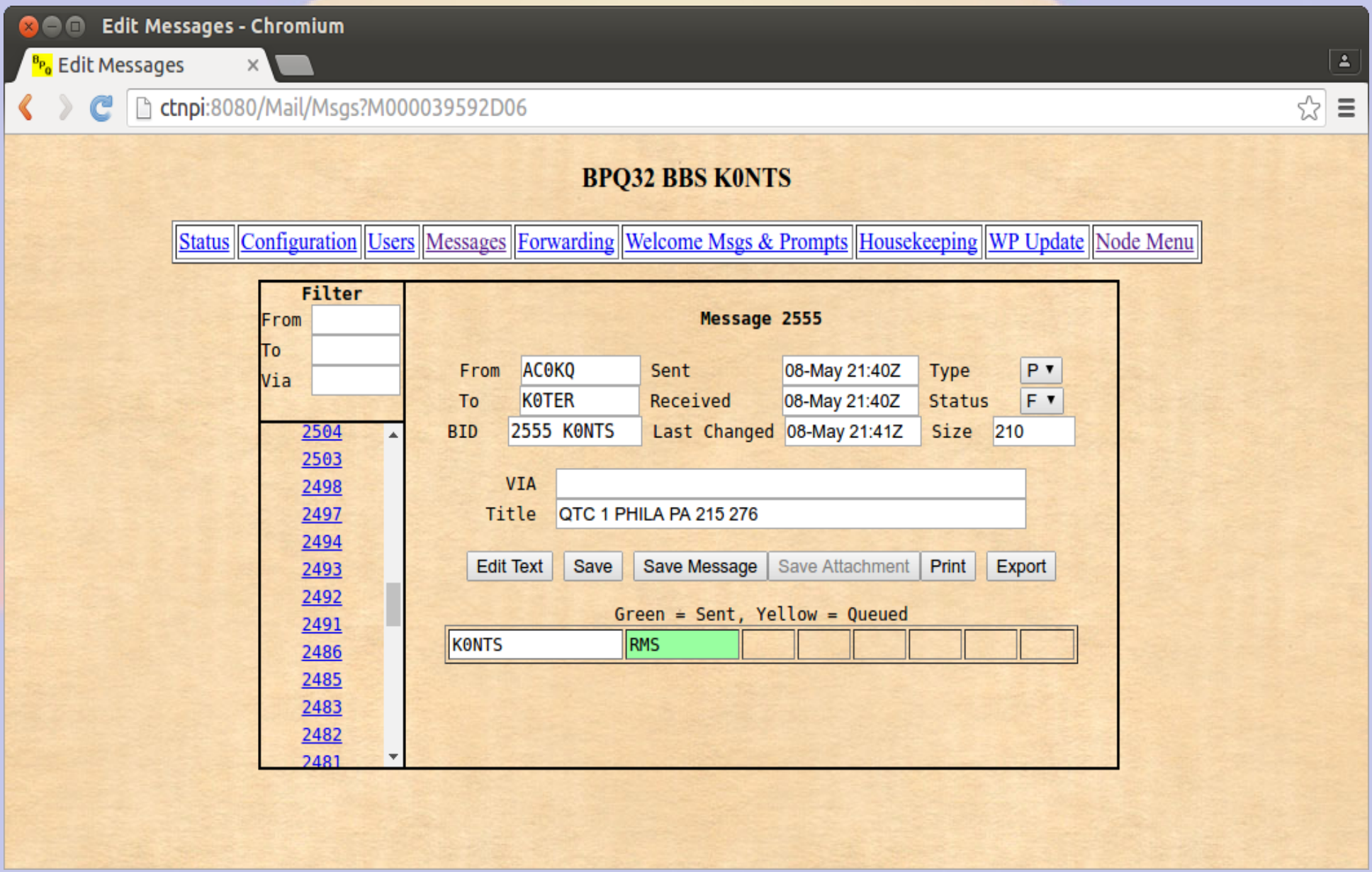
What can we use BPQ for?

- AX25 (Packet) Access point
- Bulletin Board System (BBS)
- Radio Message Server (RMS)
- APRS Internet Gateway
- Application Gateway

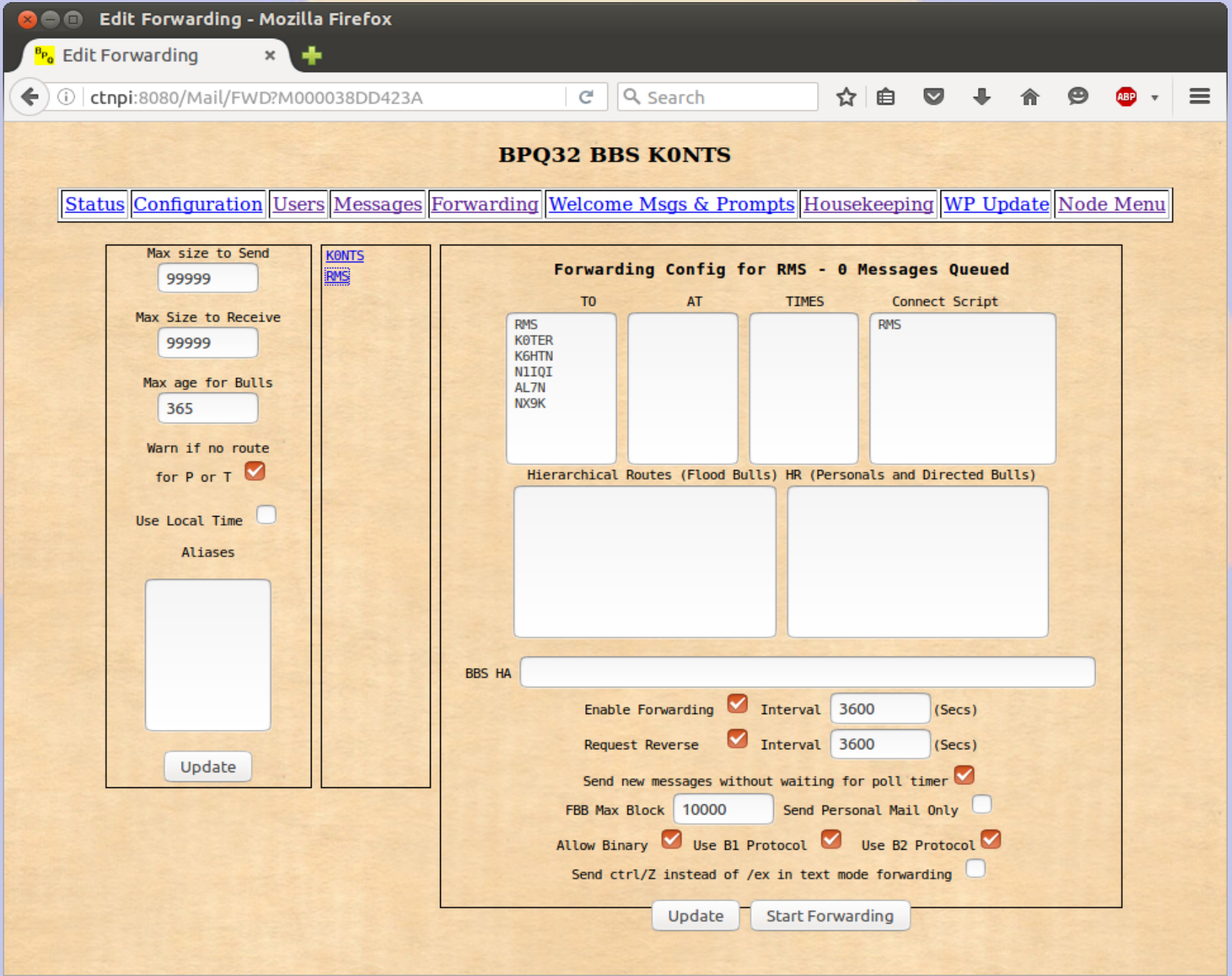
BPQ Web Configuration



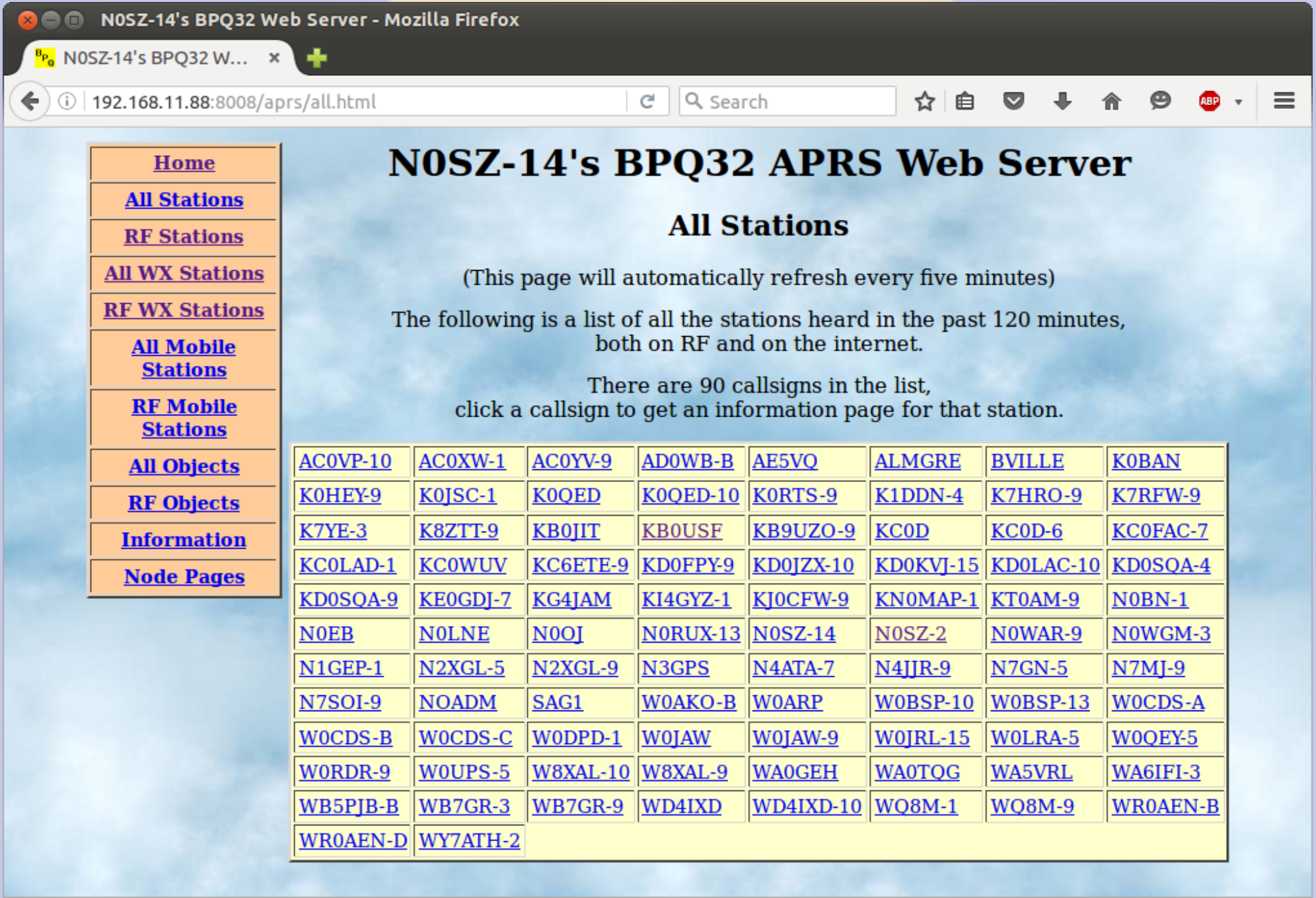
BBS Message Page



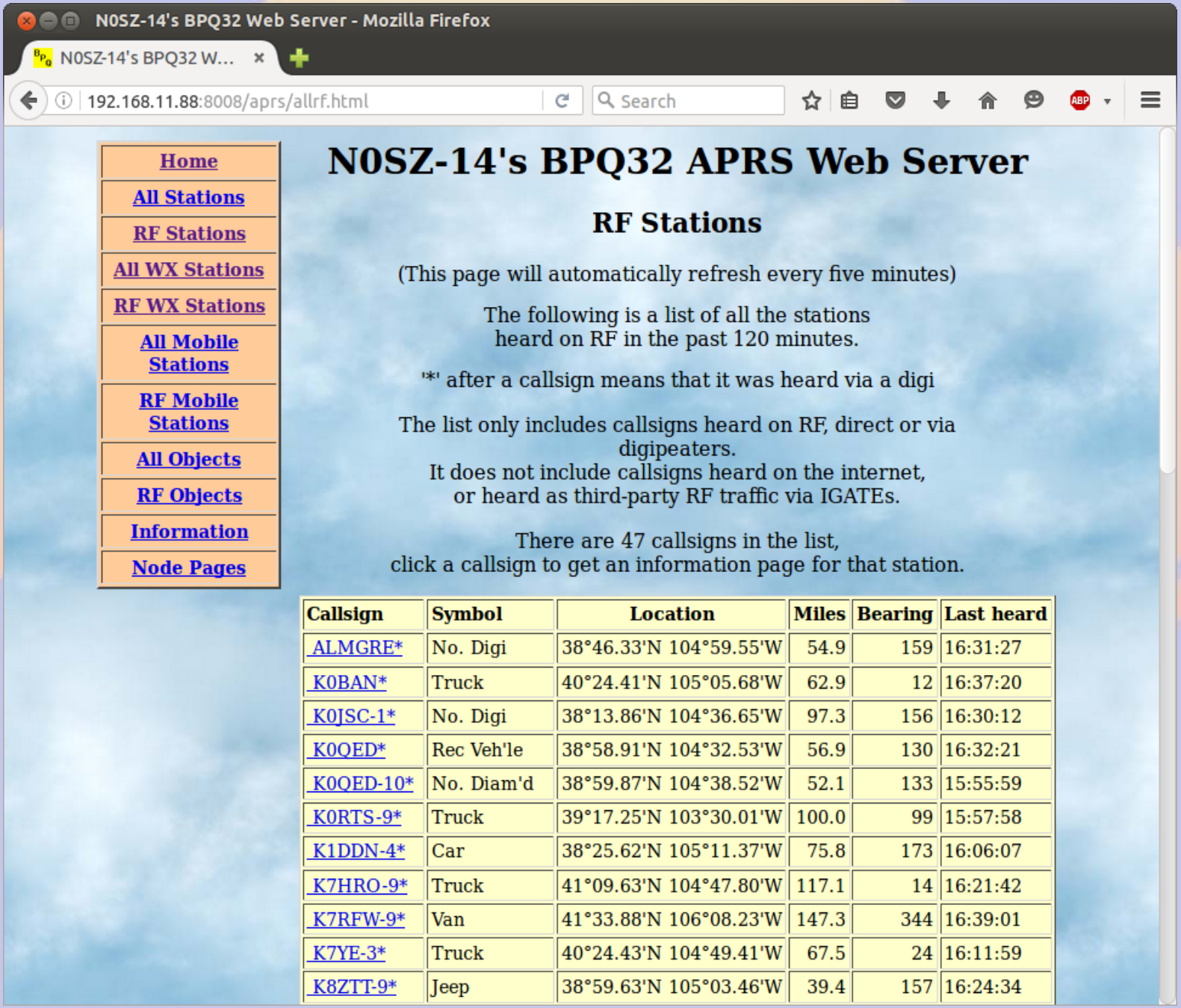
Message Forwarding



APRS Page


N0SZ-14's BPQ32 Web Server - Mozilla Firefox
N0SZ-14's BPQ32 W...
192.168.11.88:8008/aprs/all.html
Search
Home
All Stations
RF Stations
All WX Stations
RF WX Stations
All Mobile Stations
RF Mobile Stations
All Objects
RF Objects
Information
Node Pages
N0SZ-14's BPQ32 APRS Web Server
All Stations
(This page will automatically refresh every five minutes)
The following is a list of all the stations heard in the past 120 minutes, both on RF and on the internet.
There are 90 callsigns in the list, click a callsign to get an information page for that station.
AC0VP-10 AC0XW-1 AC0YV-9 AD0WB-B AE5VQ ALMGRE BVILLE K0BAN
K0HEY-9 K0JSC-1 K0QED K0QED-10 K0RTS-9 K1DDN-4 K7HRO-9 K7RFW-9
K7YE-3 K8ZTT-9 KB0JIT KB0USF KB9UZO-9 KC0D KC0D-6 KC0FAC-7
KC0LAD-1 KC0WUV KC6ETE-9 KD0FPY-9 KD0JZX-10 KD0KVJ-15 KD0LAC-10 KD0SQA-4
KD0SQA-9 KE0GDJ-7 KG4JAM KI4GYZ-1 KJ0CFW-9 KN0MAP-1 KT0AM-9 N0BN-1
N0EB N0LNE N0OJ N0RUX-13 N0SZ-14 N0SZ-2 N0WAR-9 N0WGM-3
N1GEP-1 N2XGL-5 N2XGL-9 N3GPS N4ATA-7 N4JJR-9 N7GN-5 N7MJ-9
N7SOI-9 NOADM SAG1 W0AKO-B W0ARP W0BSP-10 W0BSP-13 W0CDS-A
W0CDS-B W0CDS-C W0DPD-1 W0JAW W0JAW-9 W0JRL-15 W0LRA-5 W0QEY-5
W0RDR-9 W0UPS-5 W8XAL-10 W8XAL-9 WA0GEH WA0TQG WA5VRL WA6IFI-3
WB5PJB-B WB7GR-3 WB7GR-9 WD4IXD WD4IXD-10 WQ8M-1 WQ8M-9 WR0AEN-B
WR0AEN-D WY7ATH-2

Stations Heard on RF



Home
All Stations
RF Stations
All WX Stations
RF WX Stations
All Mobile Stations
RF Mobile Stations
All Objects
RF Objects
Information
Node Pages

N0SZ-14's BPQ32 APRS Web Server

RF Stations

(This page will automatically refresh every five minutes)

The following is a list of all the stations
heard on RF in the past 120 minutes.

'*' after a callsign means that it was heard via a digi

The list only includes callsigns heard on RF, direct or via digipeaters.
It does not include callsigns heard on the internet,
or heard as third-party RF traffic via IGATEs.

There are 47 callsigns in the list,
click a callsign to get an information page for that station.

Callsign	Symbol	Location	Miles	Bearing	Last heard
ALMGRE*	No. Digi	38°46.33'N 104°59.55'W	54.9	159	16:31:27
K0BAN*	Truck	40°24.41'N 105°05.68'W	62.9	12	16:37:20
K0JSC-1*	No. Digi	38°13.86'N 104°36.65'W	97.3	156	16:30:12
K0QED*	Rec Veh'le	38°58.91'N 104°32.53'W	56.9	130	16:32:21
K0QED-10*	No. Diam'd	38°59.87'N 104°38.52'W	52.1	133	15:55:59
K0RTS-9*	Truck	39°17.25'N 103°30.01'W	100.0	99	15:57:58
K1DDN-4*	Car	38°25.62'N 105°11.37'W	75.8	173	16:06:07
K7HRO-9*	Truck	41°09.63'N 104°47.80'W	117.1	14	16:21:42
K7RFW-9*	Van	41°33.88'N 106°08.23'W	147.3	344	16:39:01
K7YE-3*	Truck	40°24.43'N 104°49.41'W	67.5	24	16:11:59
K8ZTT-9*	Jeep	38°59.63'N 105°03.46'W	39.4	157	16:24:34

Station Map

N0SZ-14's BPQ32 Web Server - Mozilla Firefox

N0SZ-14's BPQ32 W...

192.168.11.88:8008/aprs/find.cgi?call=N0SZ-2

Search

- Home
- All Stations
- RF Stations
- All WX Stations
- RF WX Stations
- All Mobile Stations
- RF Mobile Stations
- All Objects
- RF Objects
- Information
- Node Pages

N0SZ-14's BPQ32 APRS Web Server

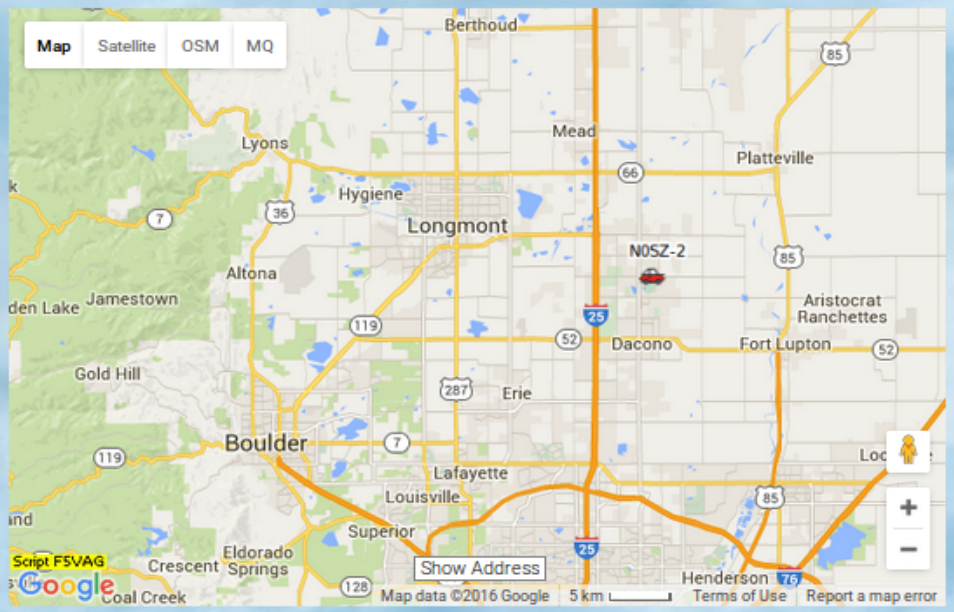
(This page will automatically refresh every five minutes)

Information about N0SZ-2

Click the callsign to look it up on qrz.com
Location: 40°07.91'N 104°55.75'W
The bearing from N0SZ-14 to N0SZ-2 is 028 degrees, the distance is 47.9 Miles

Last posit: APTT4,W0UPS-5,WIDE1,KC0D,WIDE2*

Status: /TinyTrak4 Alpha
Last heard 00:54:36 ago



How does it work?

- BPQ is a software program
  - Runs on most computers
  - Somewhat complex configuration file
- Connects to radio via Terminal Node Controller (TNC)
  - Typically serial connection
- Interconnects via IP
- Built-in BBS, iGate, Chat server, ...

rPi/BPQ vs. KPC3+ BBS

- **rPi/BPQ Pros**
  - **Lower cost ($100)**
  - **Much larger capacity (GB vs. kB)**
  - **More ports (multiple RF, serial and IP)**
  - **Sophisticated forwarding**
- **rPi/BPQ Cons**
  - **Higher current draw**
  - **Less tolerant of bad power**

Complaint: Hard to set up BPQ

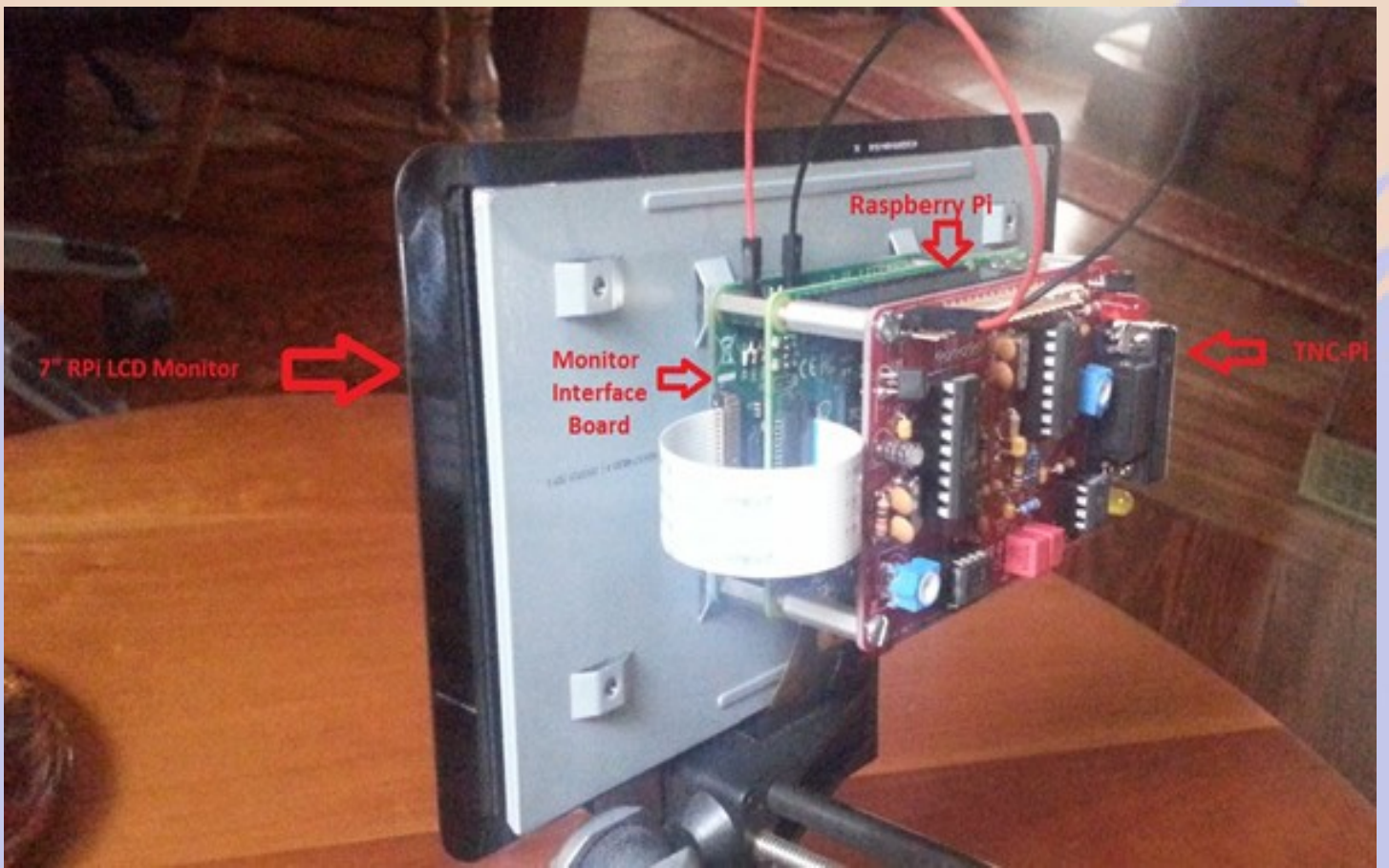
- BPQ is very sophisticated, and that necessarily adds complexity
- Solutions:
  - Use *bpq-config* to get started
  - Web interface for BBS etc.
  - Join a support group
    - Yahoo BPQ32
    - RMHAM

Why so rPi and BBB centric?

- BPQ is software – runs anywhere
  - Supported on Windows, OSX, Linux
  - Best run as a headless server
- rPi and BBB are
  - Inexpensive
  - Reliable Linux boxes
  - DC powered
  - TNC/Pi & TNC/Black daughter boards
  - All the cool kids have one

Why the rPi/TNC-Pi?

- **Extremely well supported**
- **Complete package with screen**



Brief history of BPQ

- Written by John Wiseman G8BPQ
- Originally called BPQCODE
- Became BPQ32 in late 90s
- Ported to OSX/Linux in 2000s
- Ported to Raspberry Pi/TNC-PI and Beagle Bone Black/TNC-Black

Building the TNC kit

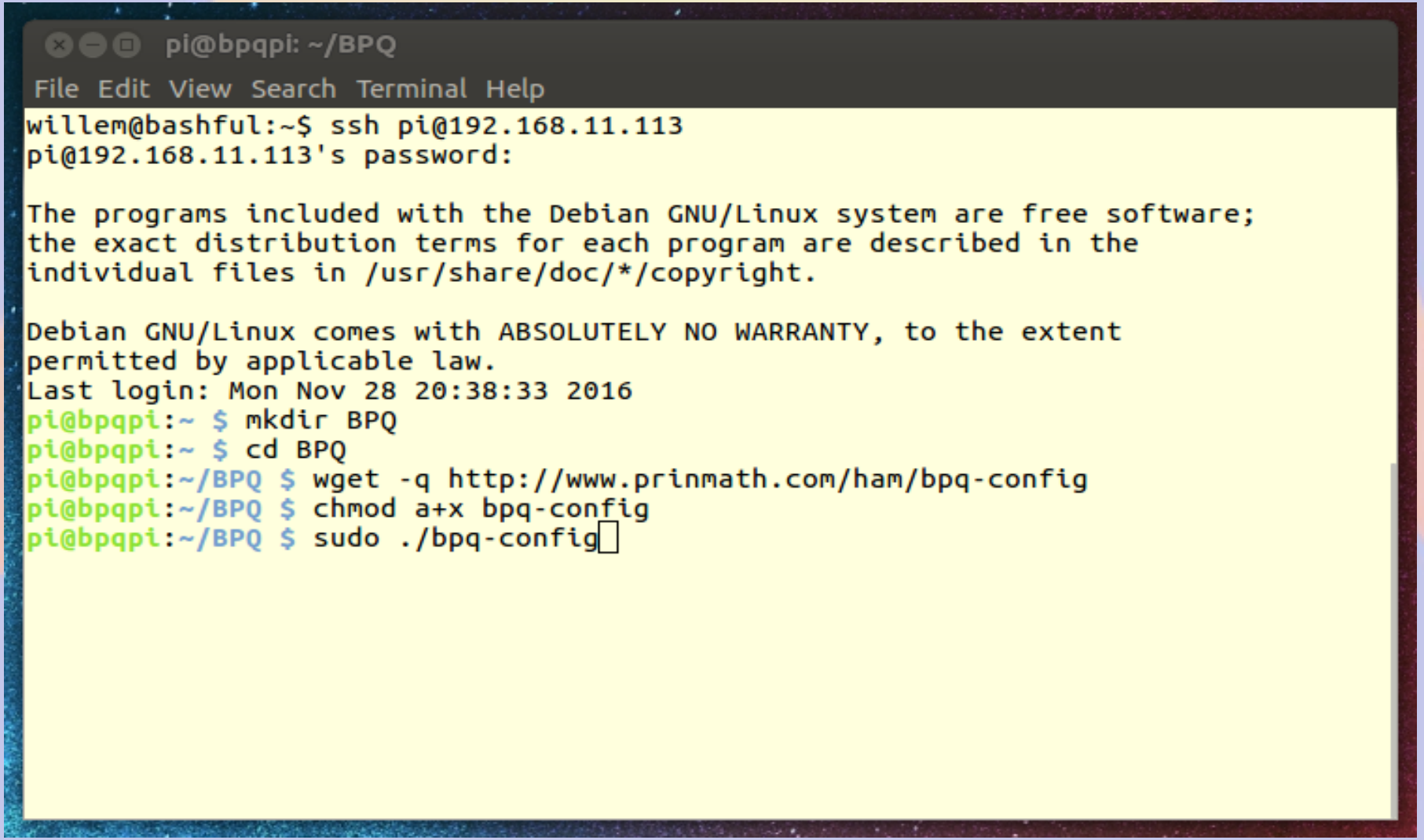
- It takes a few hours to build
  - Quality soldering iron time
  - Simple, excellent instructions
- Test it
  - Check voltages, insert ICs
  - LEDs should flash on power up
  - Configure OS and BPQ
- John W2FS provides outstanding after-sales support

Selecting a Username

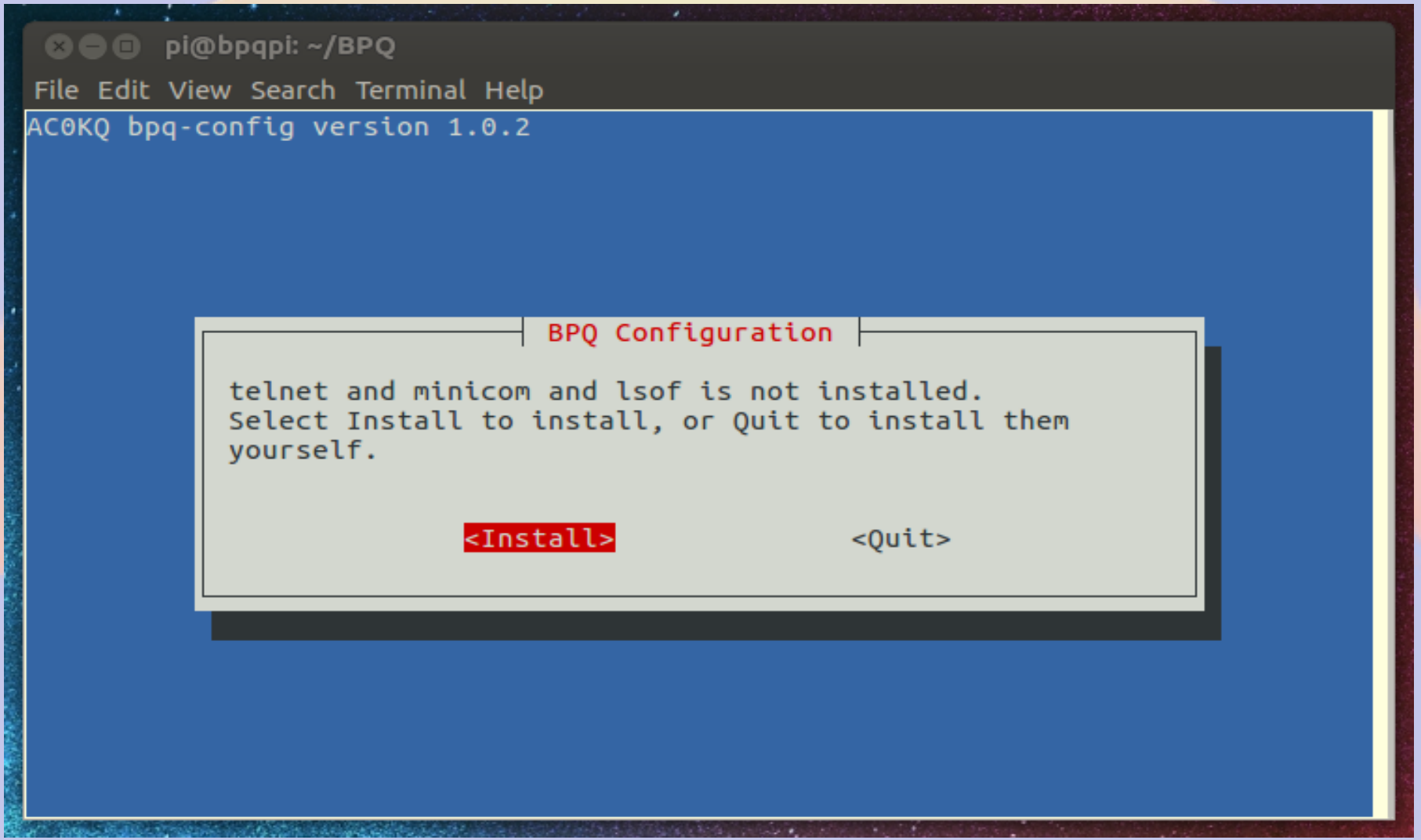
- **Default user name**
  - **Raspberry Pi = pi**
  - **Beaglebone Black = debian**
- **The default user name is good for BPQ and similar programs with multiple users**
- **Create a login for each user**
- **Create subdirectories for programs like BPQ which will clutter the home directory**

Quick Start

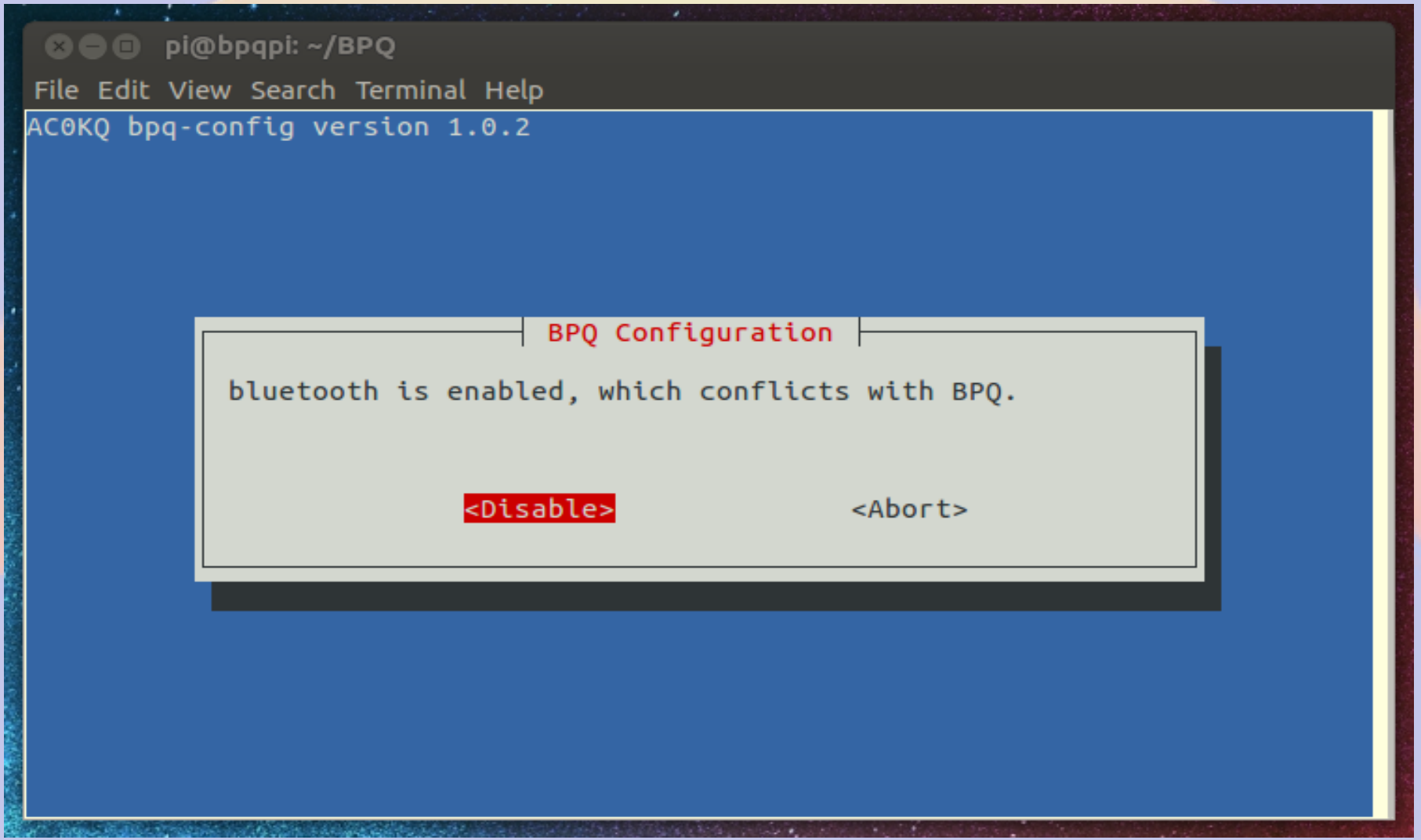
http://www.prinmath.com/ham/howto/quickstart/



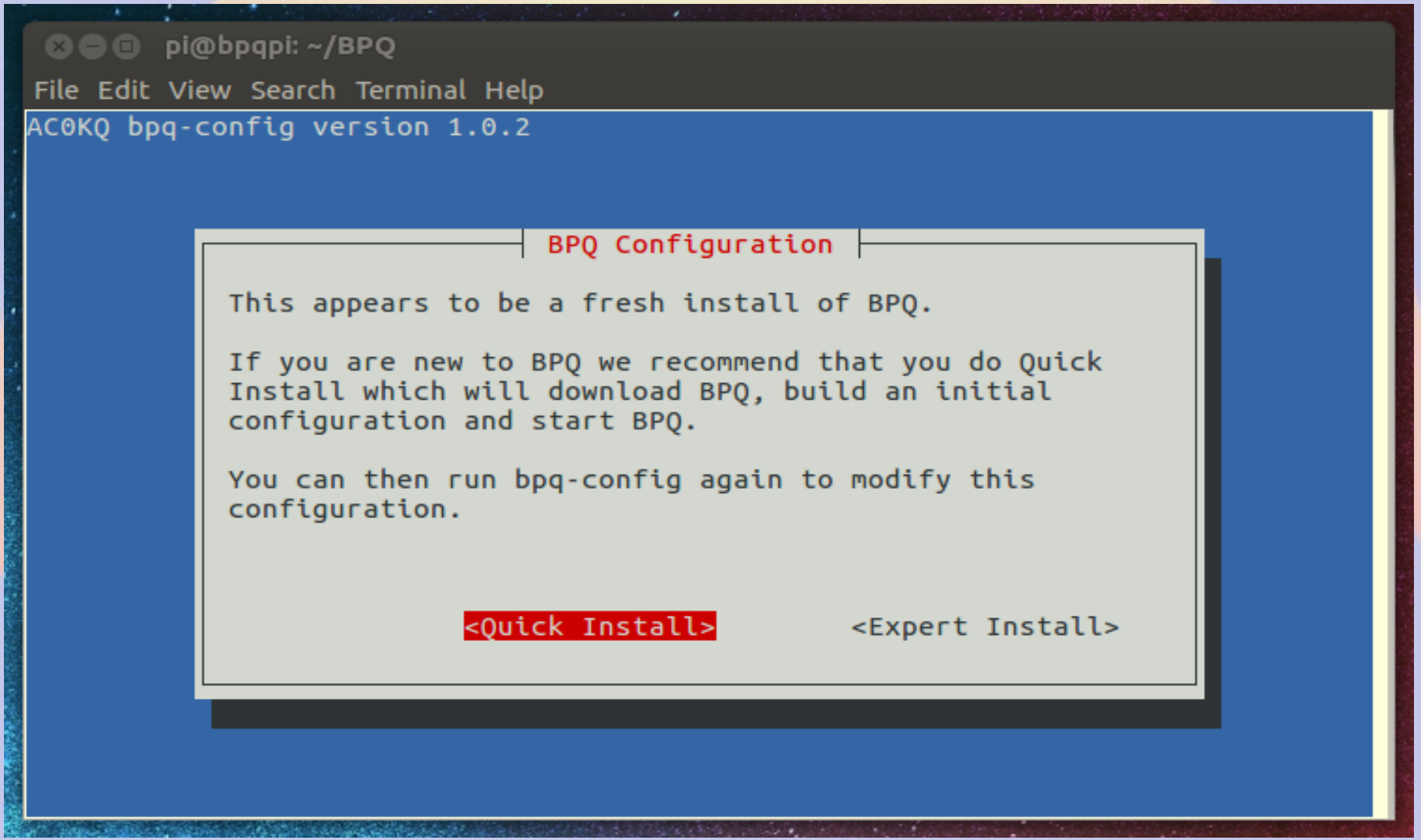
bpq-config installs programs



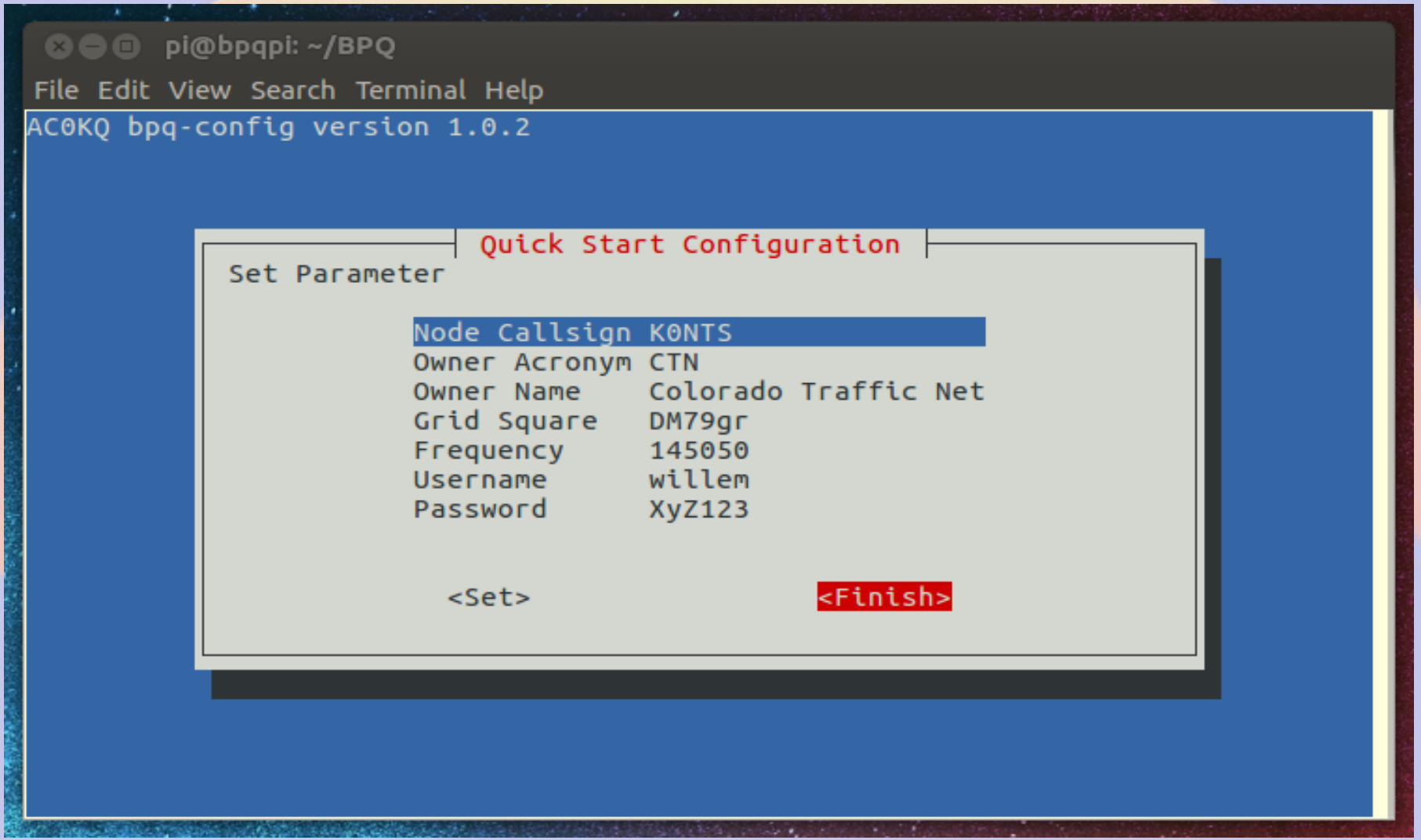
bpq-config resolves conflicts



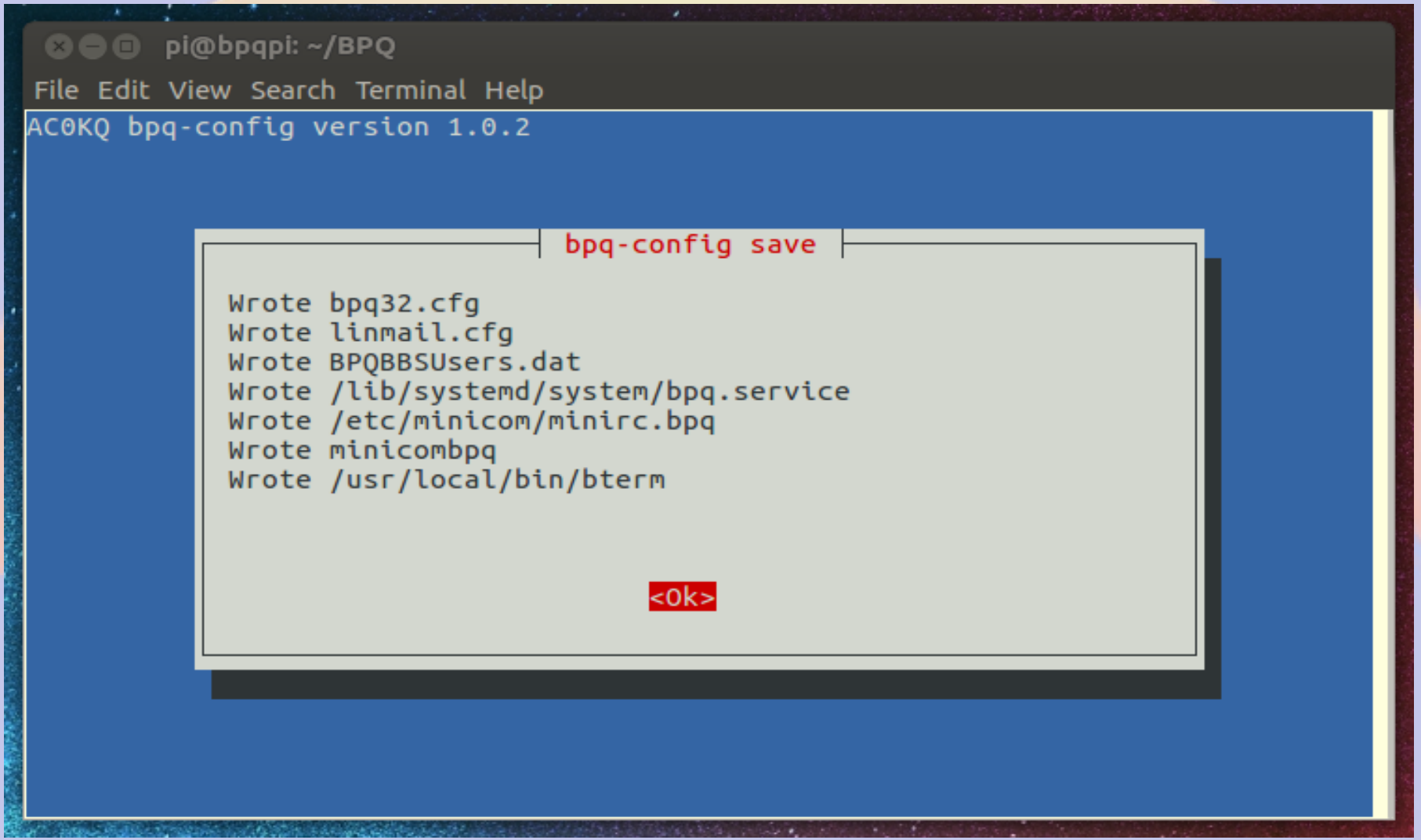
Quick Start


pi@bpqpi: ~/BPQ
File Edit View Search Terminal Help
AC0KQ bpq-config version 1.0.2
BPQ Configuration
This appears to be a fresh install of BPQ.
If you are new to BPQ we recommend that you do Quick Install which will download BPQ, build an initial configuration and start BPQ.
You can then run bpq-config again to modify this configuration.
<Quick Install>
<Expert Install>

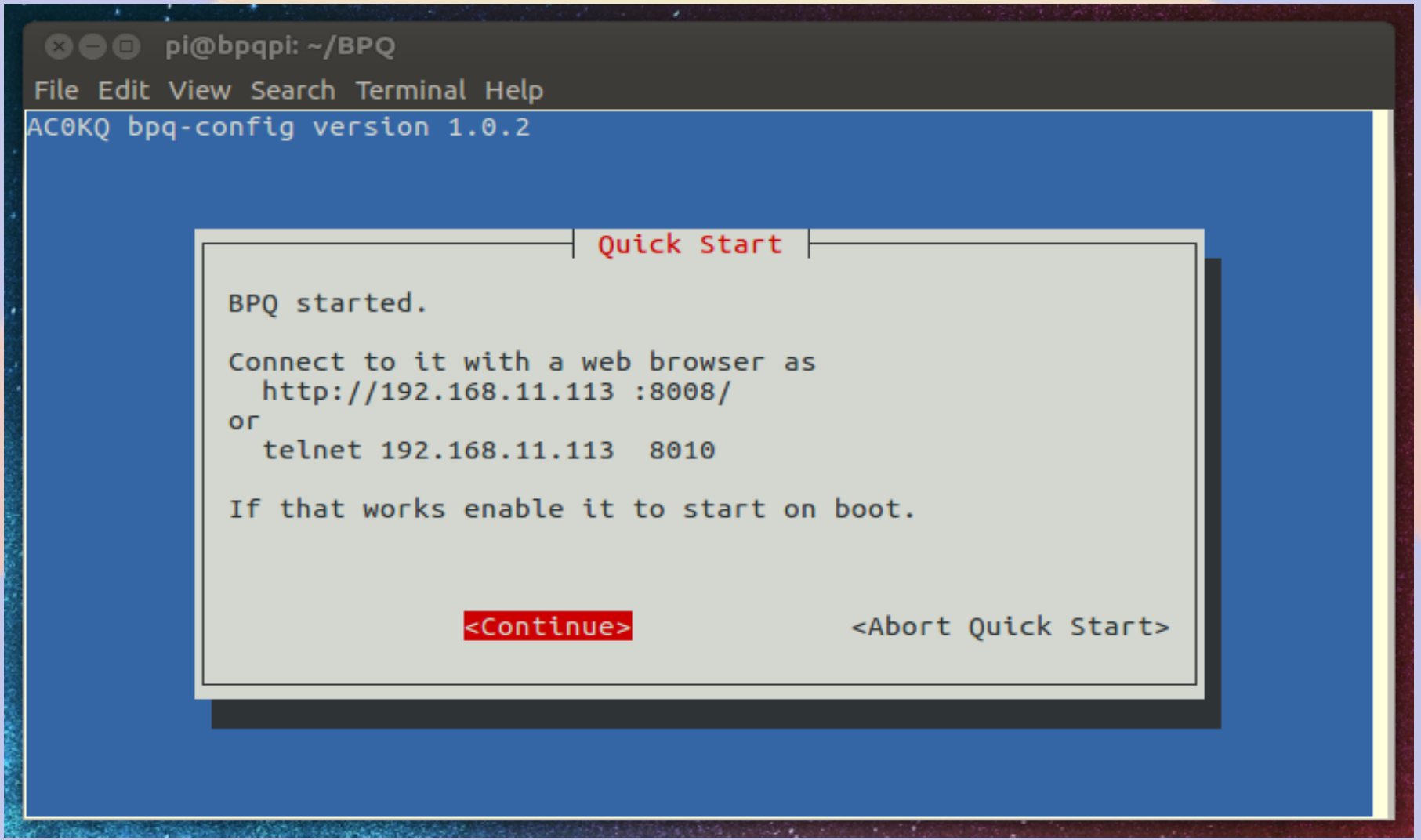
Fill the required fields



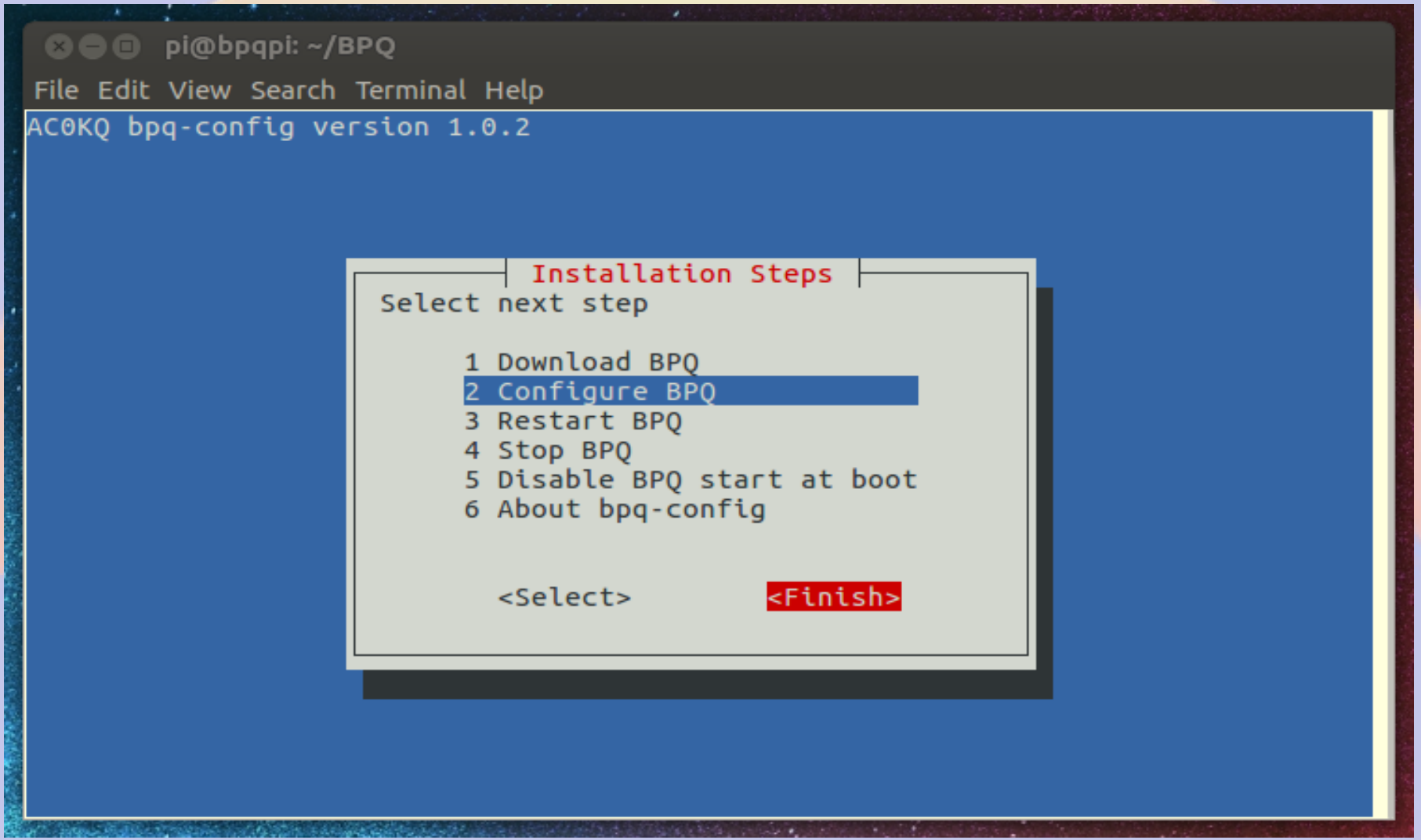
Write Configuration



Start BPQ


pi@bpqpi: ~/BPQ
File Edit View Search Terminal Help
AC0KQ bpq-config version 1.0.2
Quick Start
BPQ started.
Connect to it with a web browser as
http://192.168.11.113 :8008/
or
telnet 192.168.11.113 8010
If that works enable it to start on boot.
<Continue>
<Abort Quick Start>

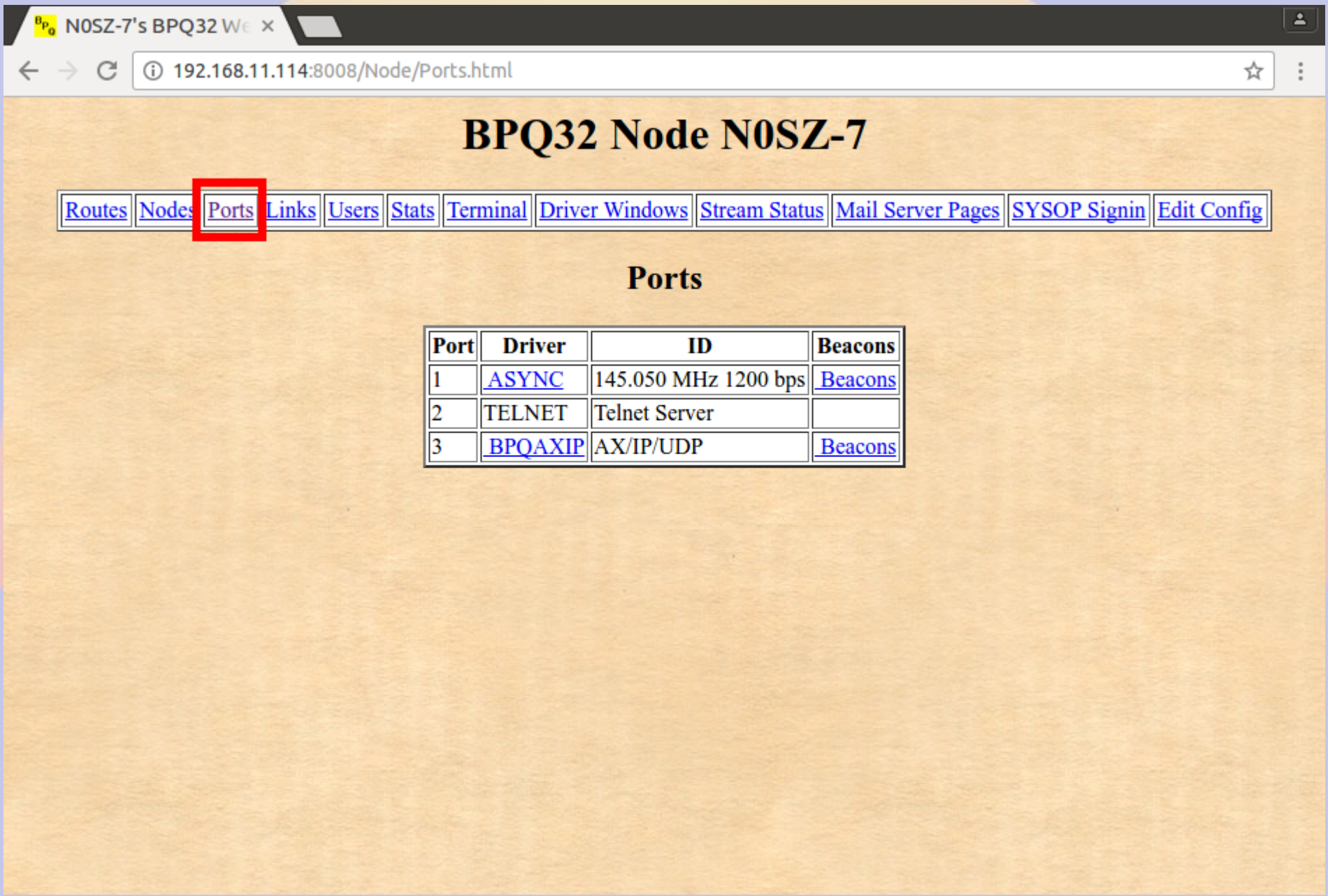
Options after Startig BPQ



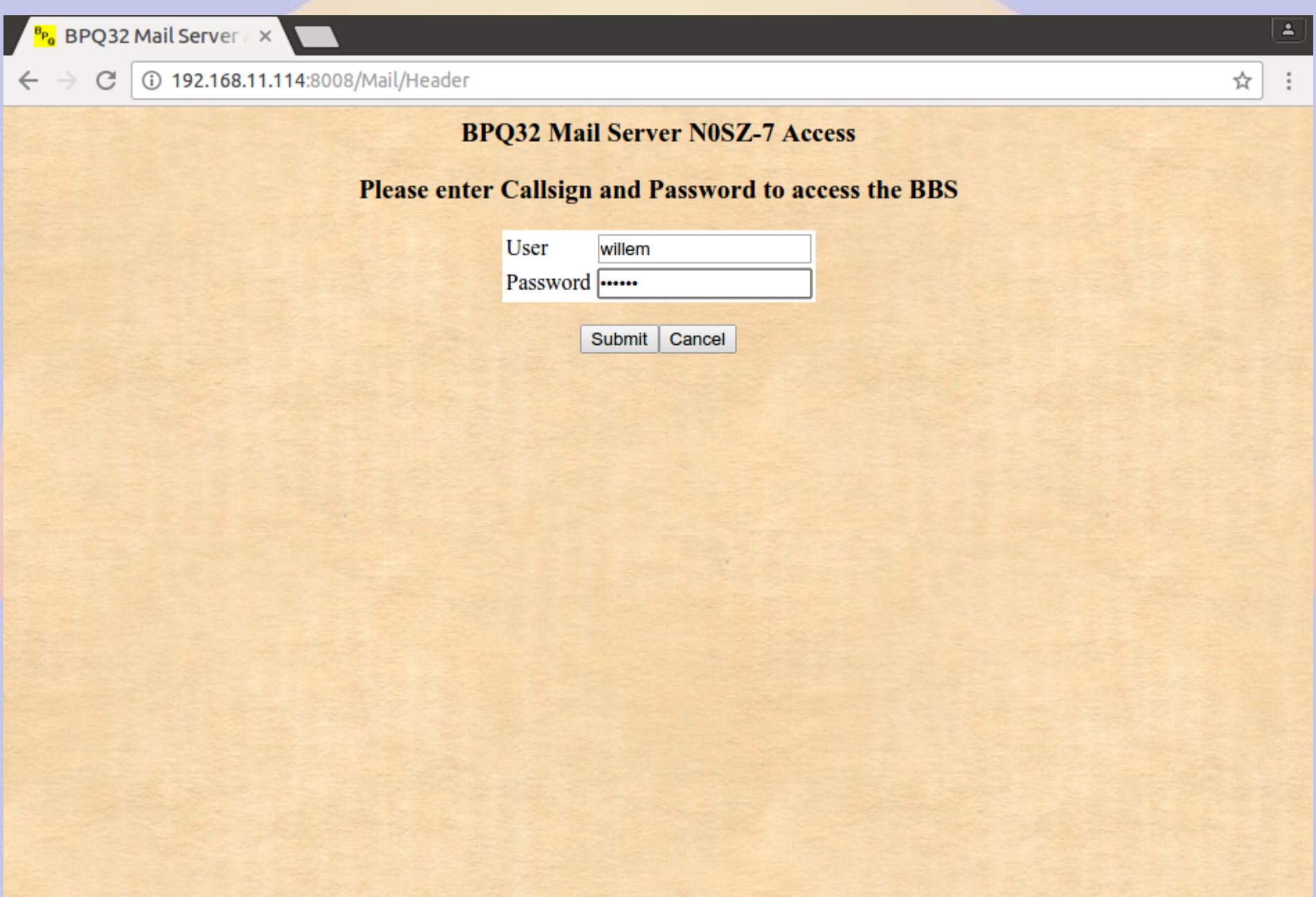
Browse to BPQ node port 8008

(if you configured a different port, use it instead)

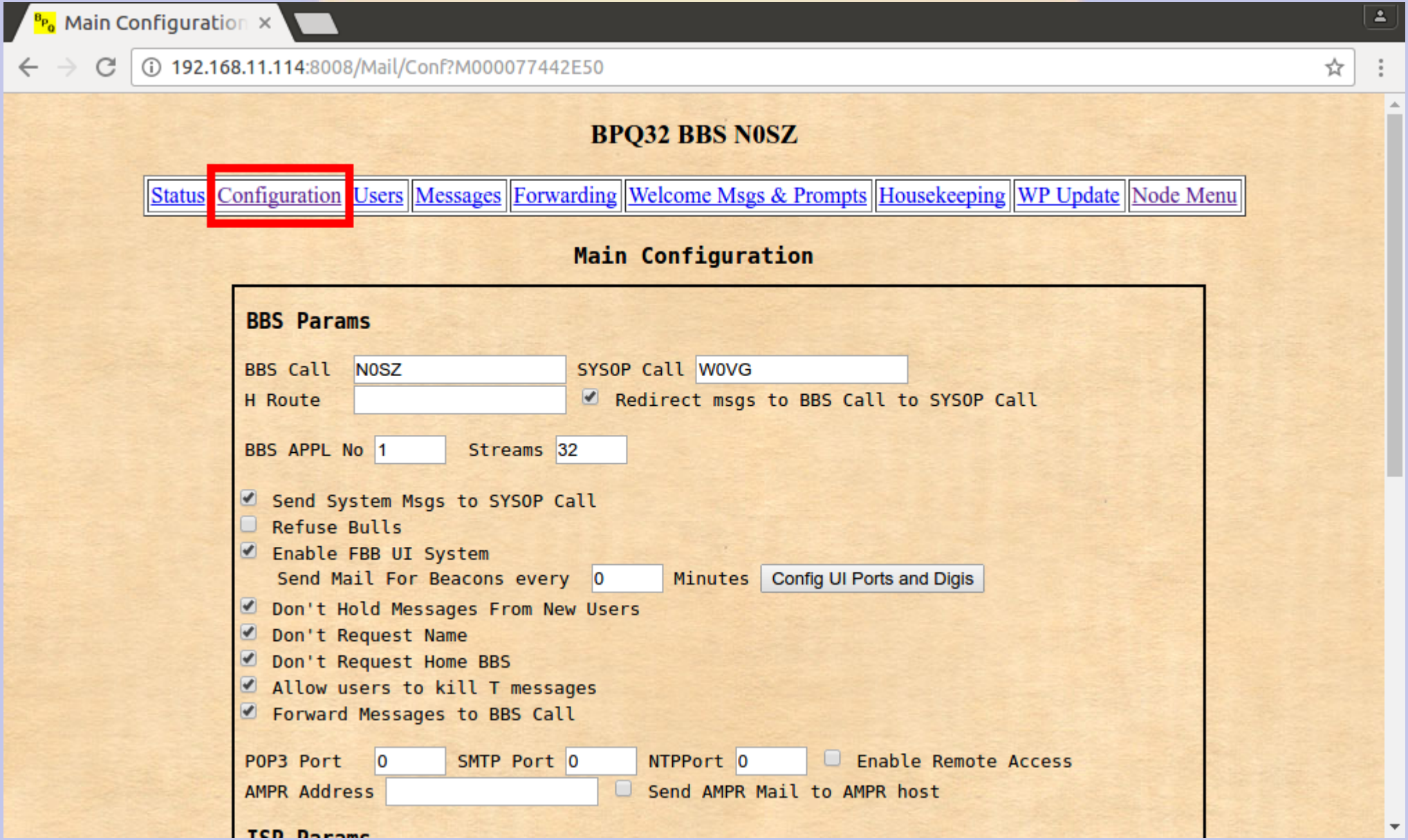
BPQ Ports


N0SZ-7's BPQ32 We
192.168.11.114:8008/Node/Ports.html
BPQ32 Node N0SZ-7
Routes
Nodes
Ports
Links
Users
Stats
Terminal
Driver Windows
Stream Status
Mail Server Pages
SYSOP Signin
Edit Config
Ports
Port
Driver
ID
Beacons
1
ASYNC
145.050 MHz 1200 bps
Beacons
2
TELNET
Telnet Server
3
BPQAXIP
AX/IP/UDP
Beacons

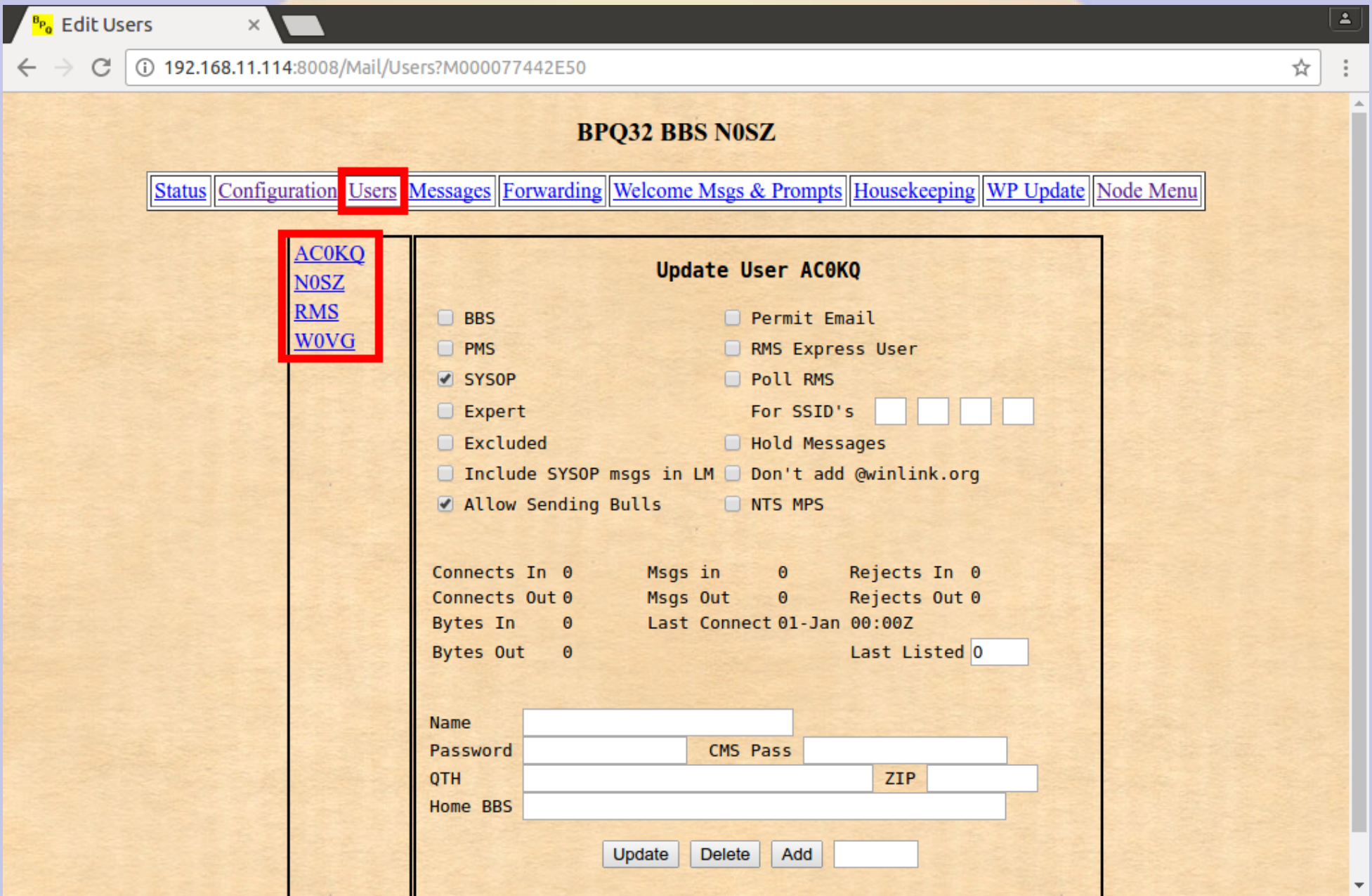
Click *Mail Server Pages*



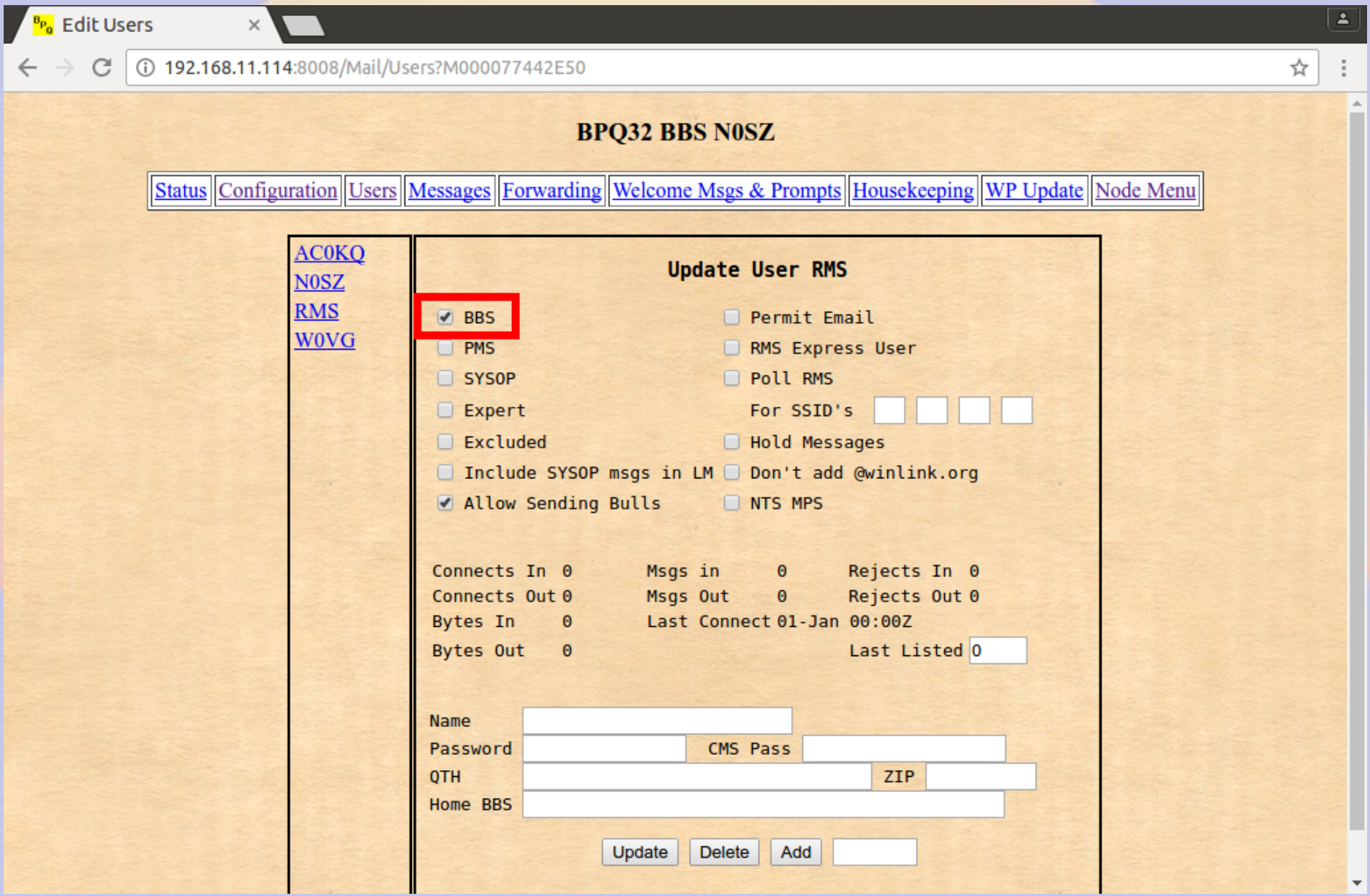
BBS Configuration
(bpq-config set most of these in linmail.cfg)



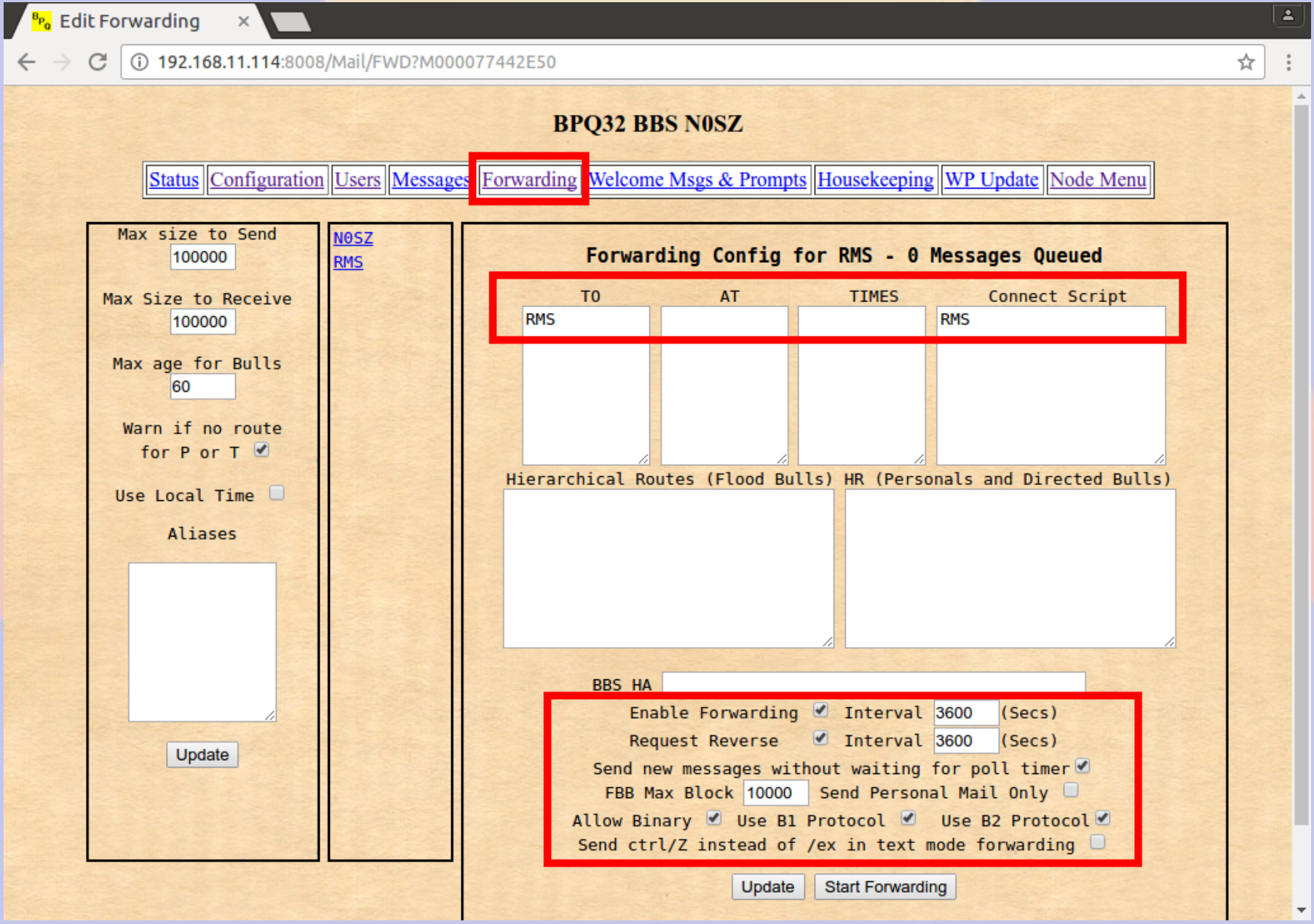
BBS Users
(bpq-config added RMS and telnet users)



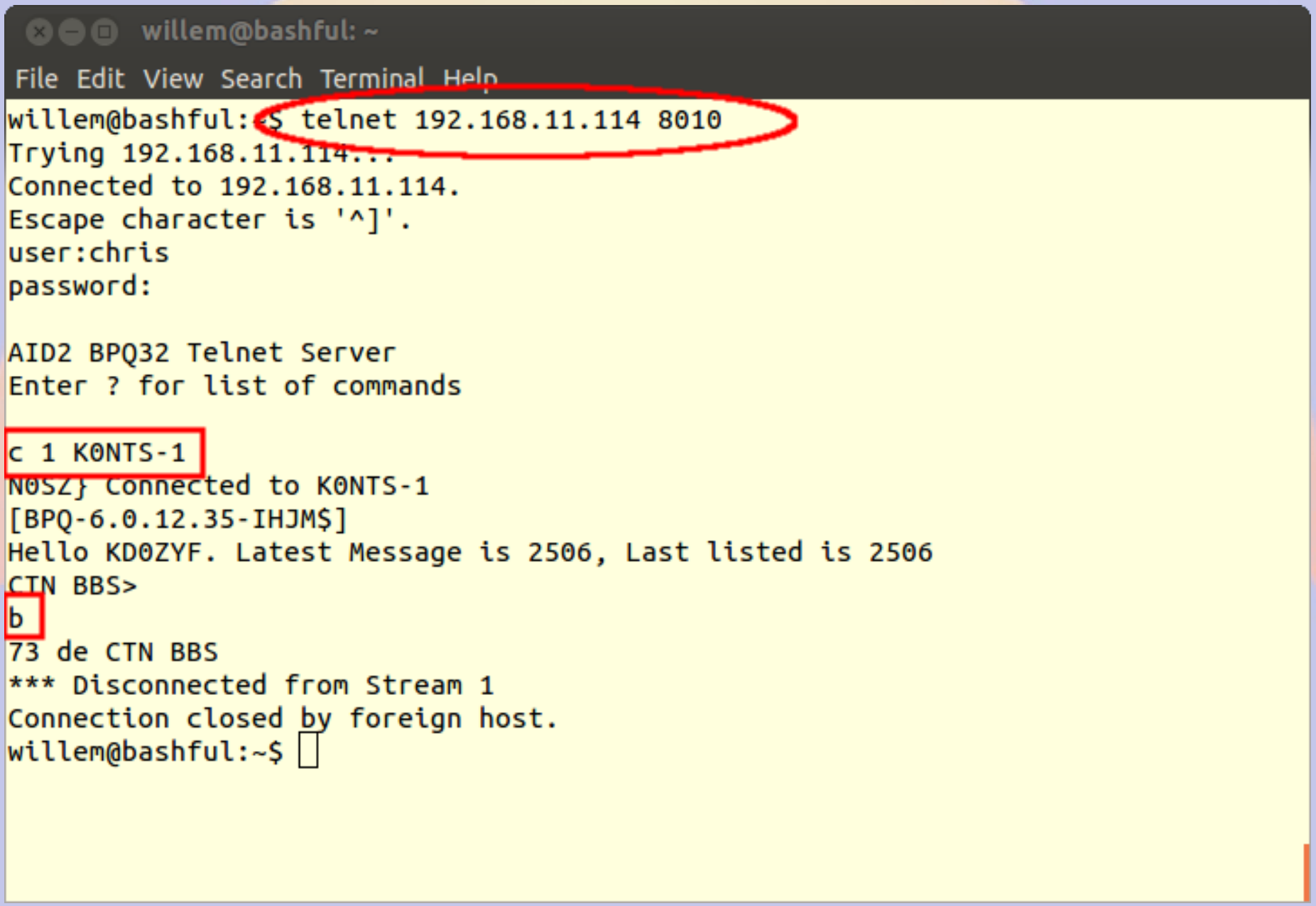
User RMS is WinLink2000



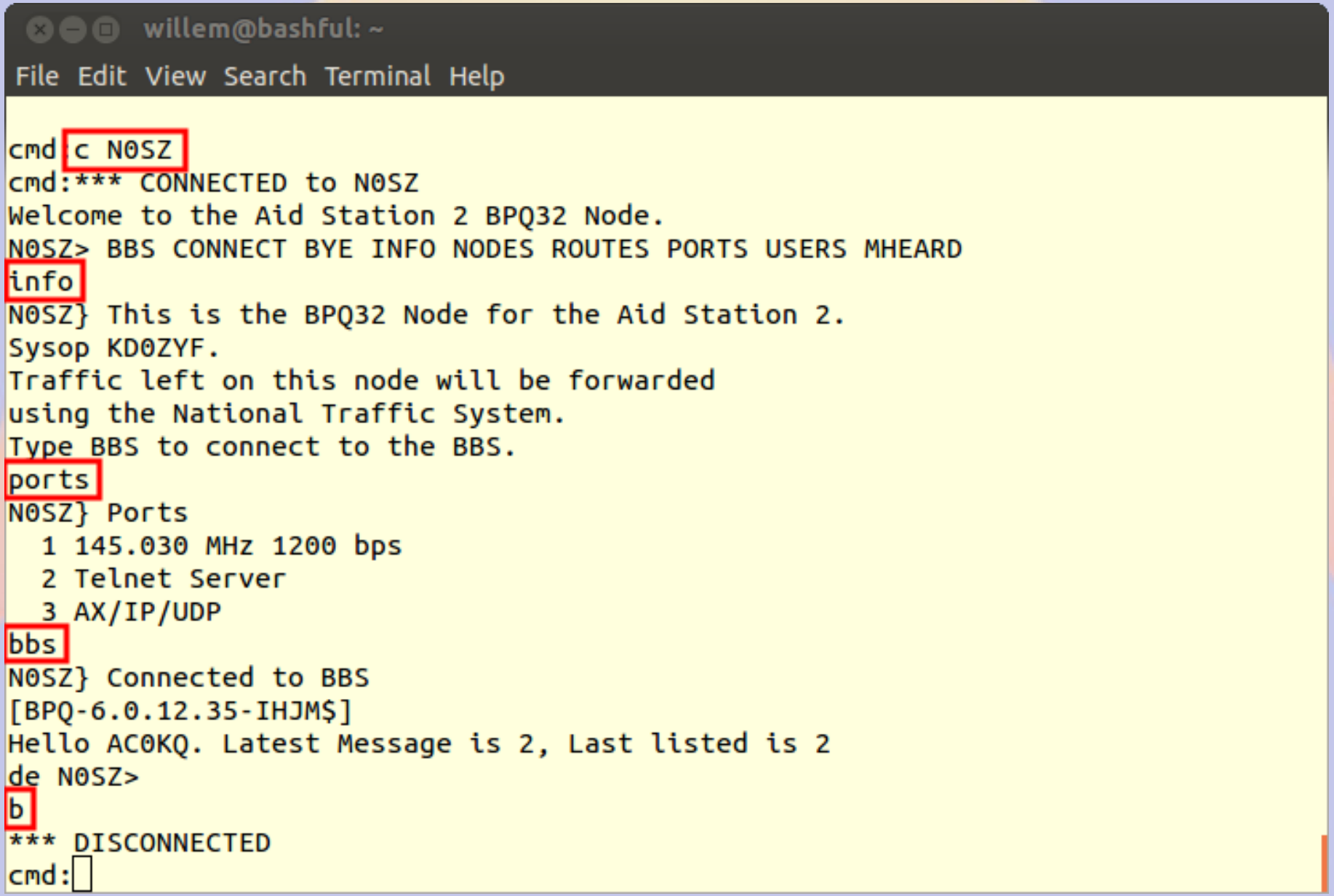
Forwarding to Winlink is Enabled



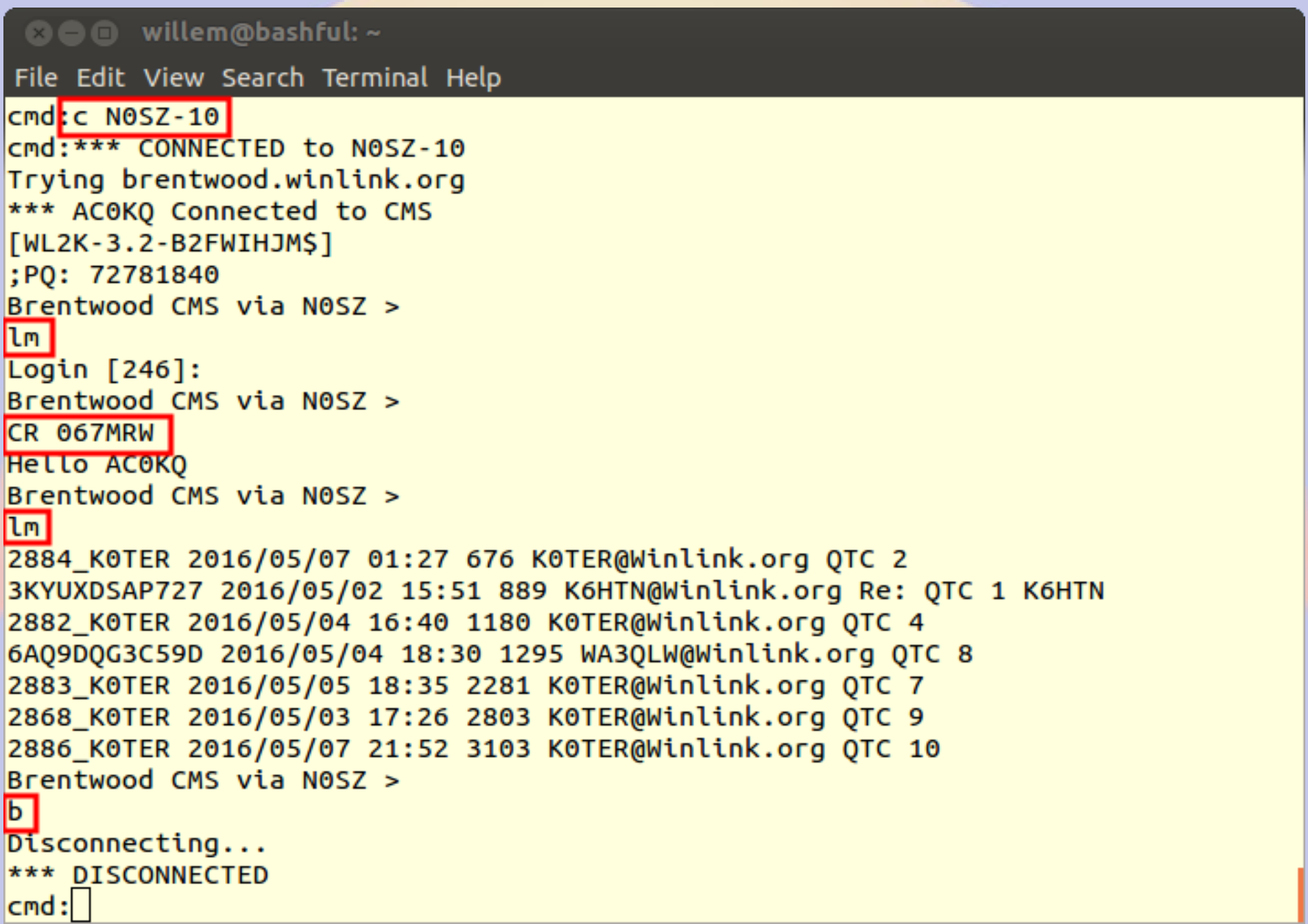
Connect out via RF



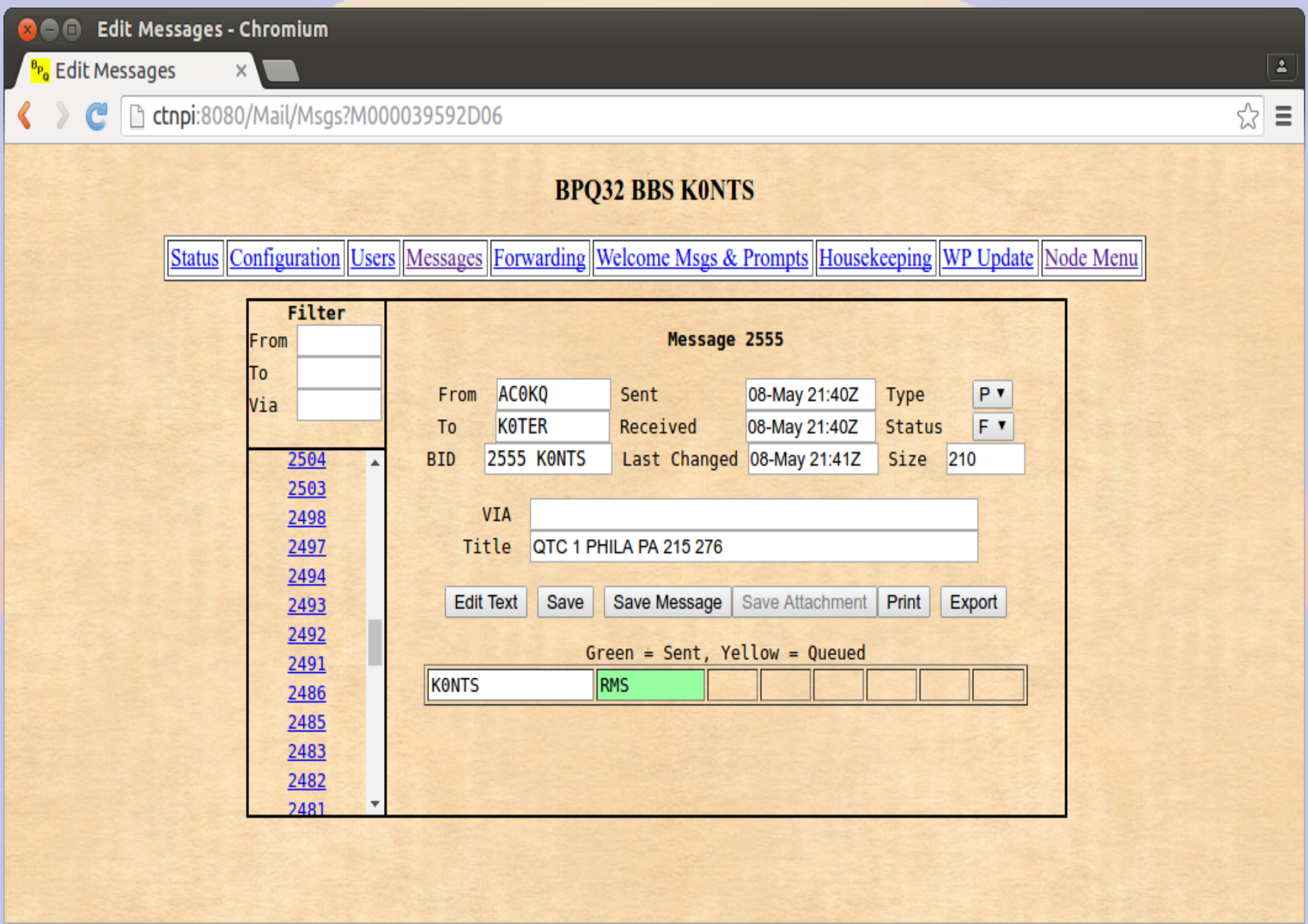
Connect in via RF (as AC0KQ)



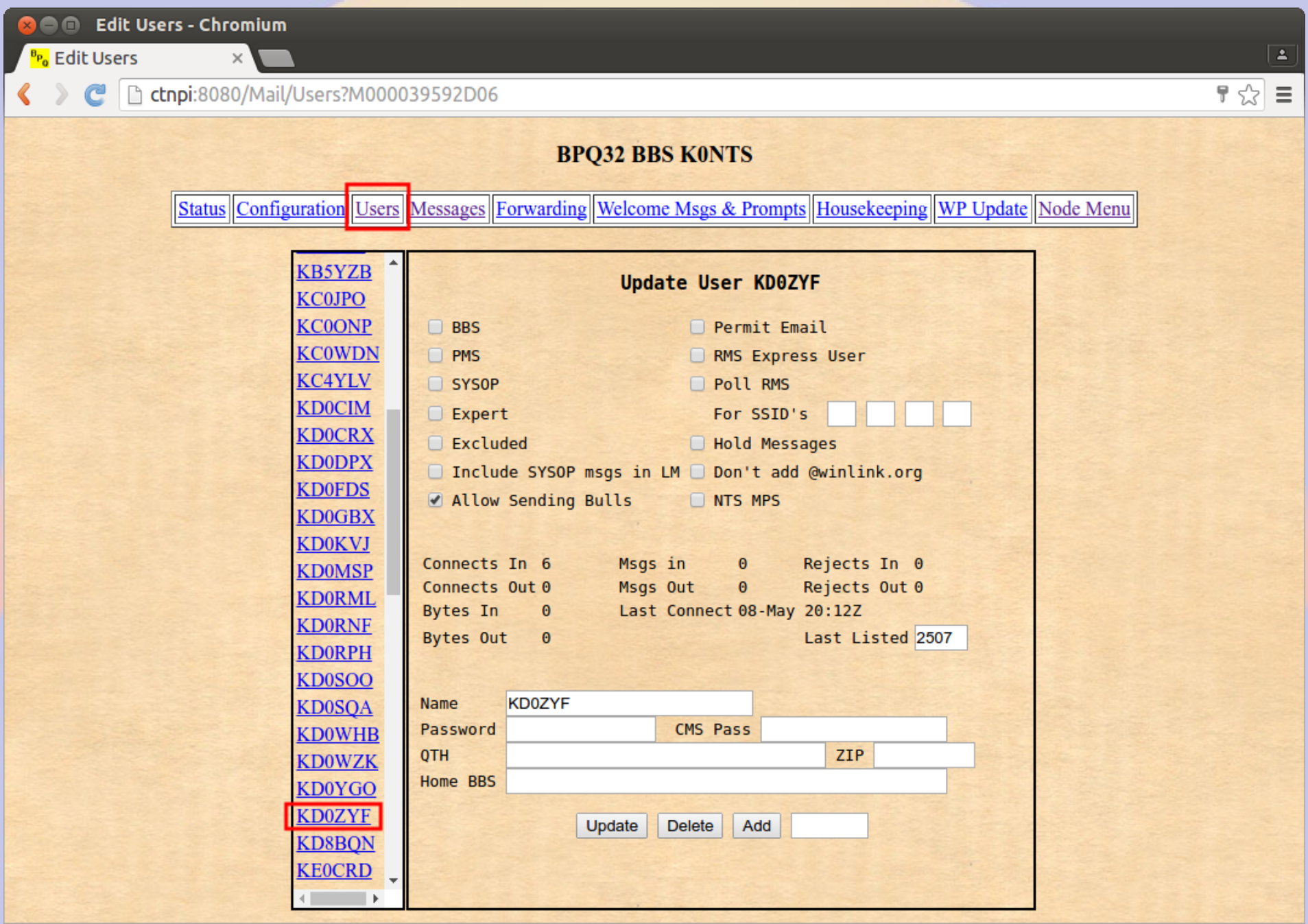
Connect via RF to WinLink



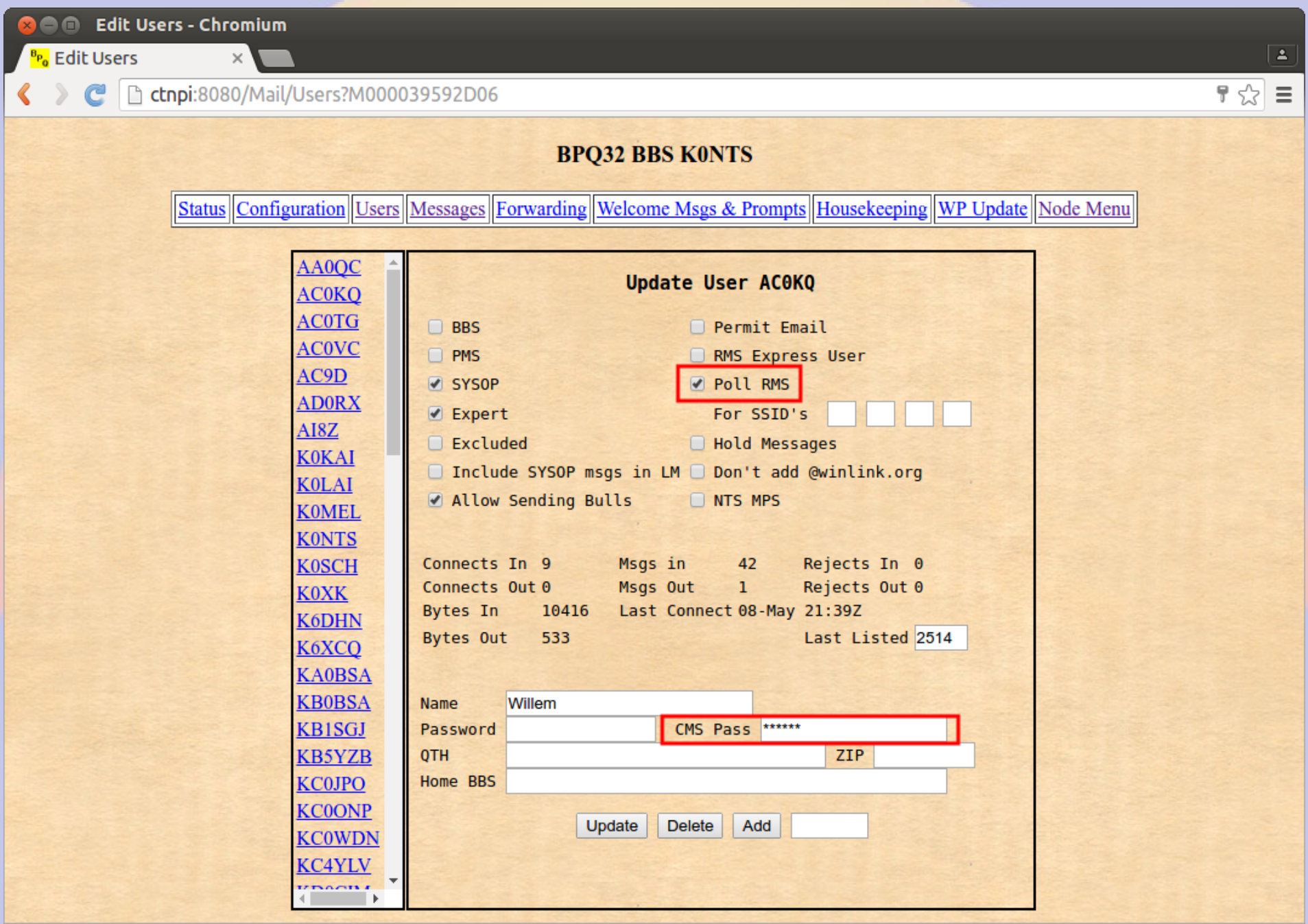
BBS Messages



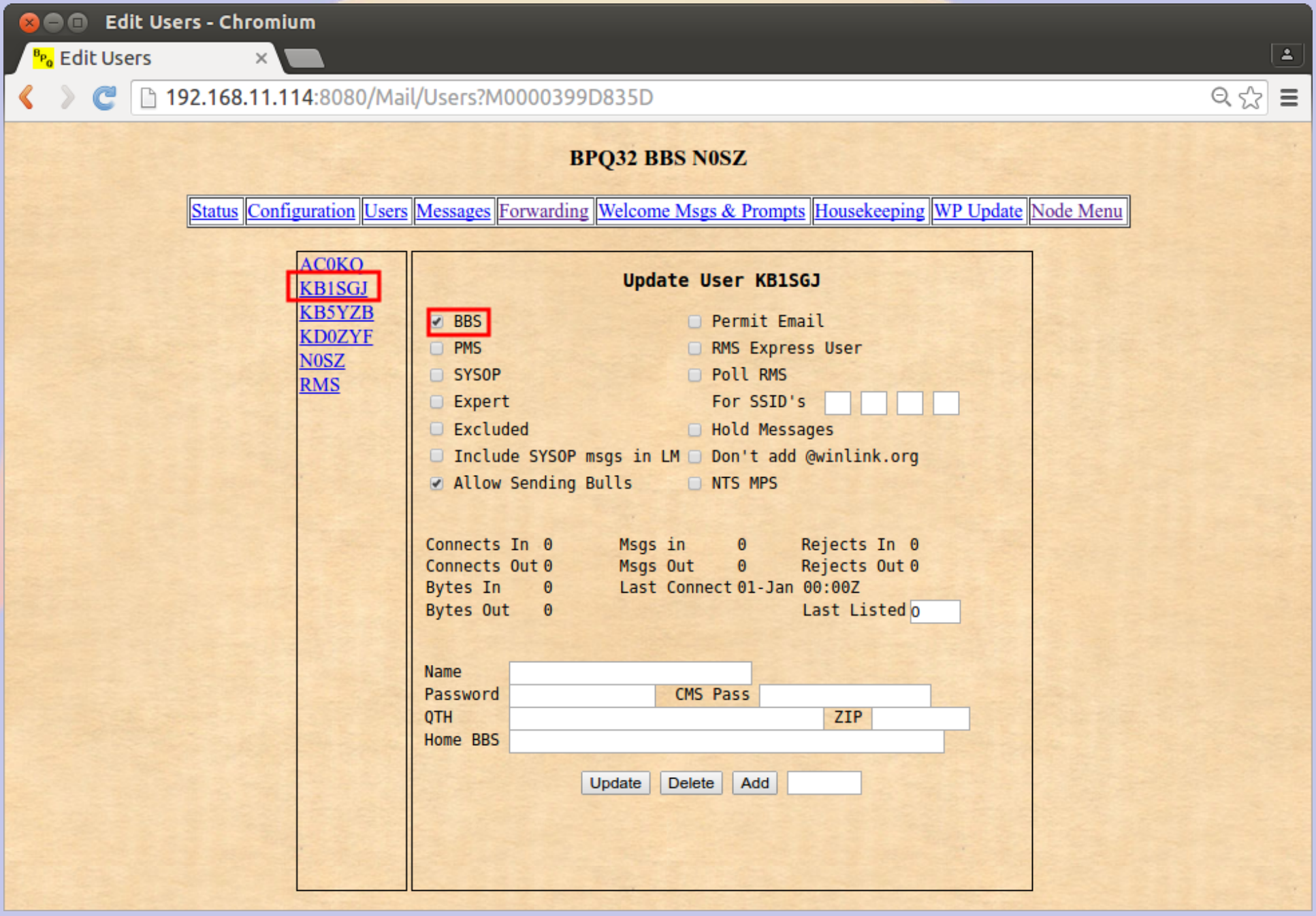
General BBS Users



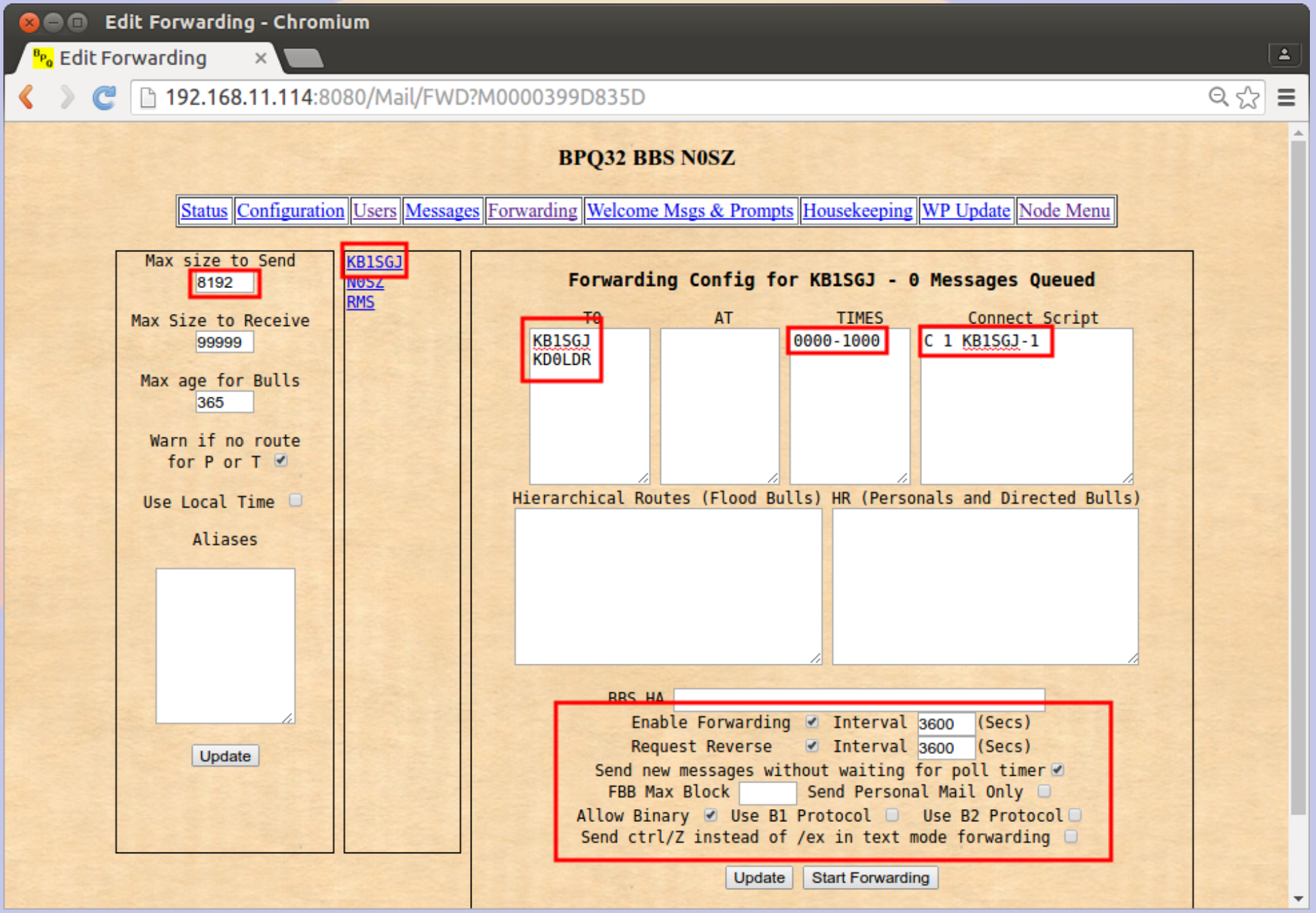
WinLink User Download



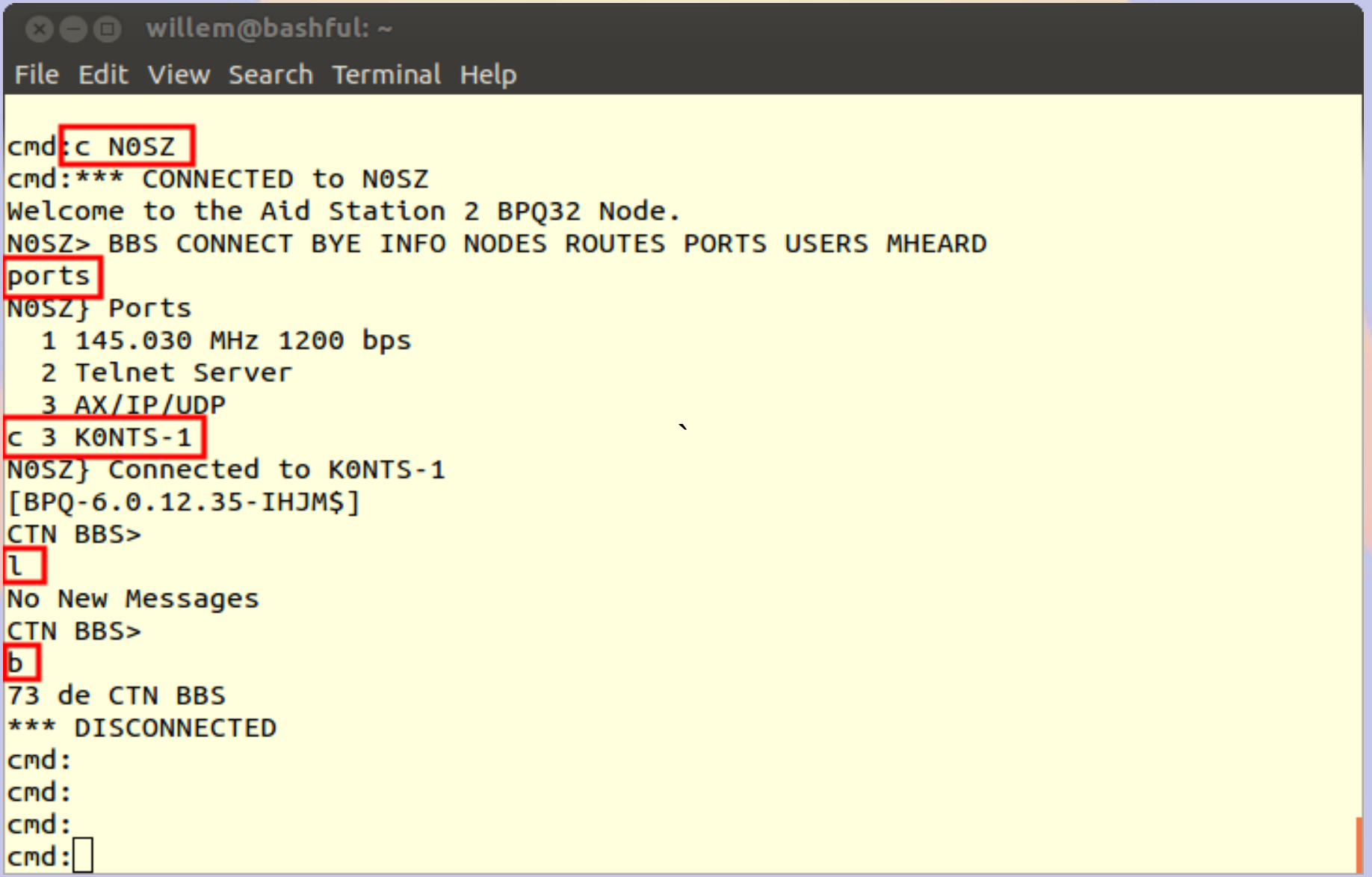
Forwarding BBS



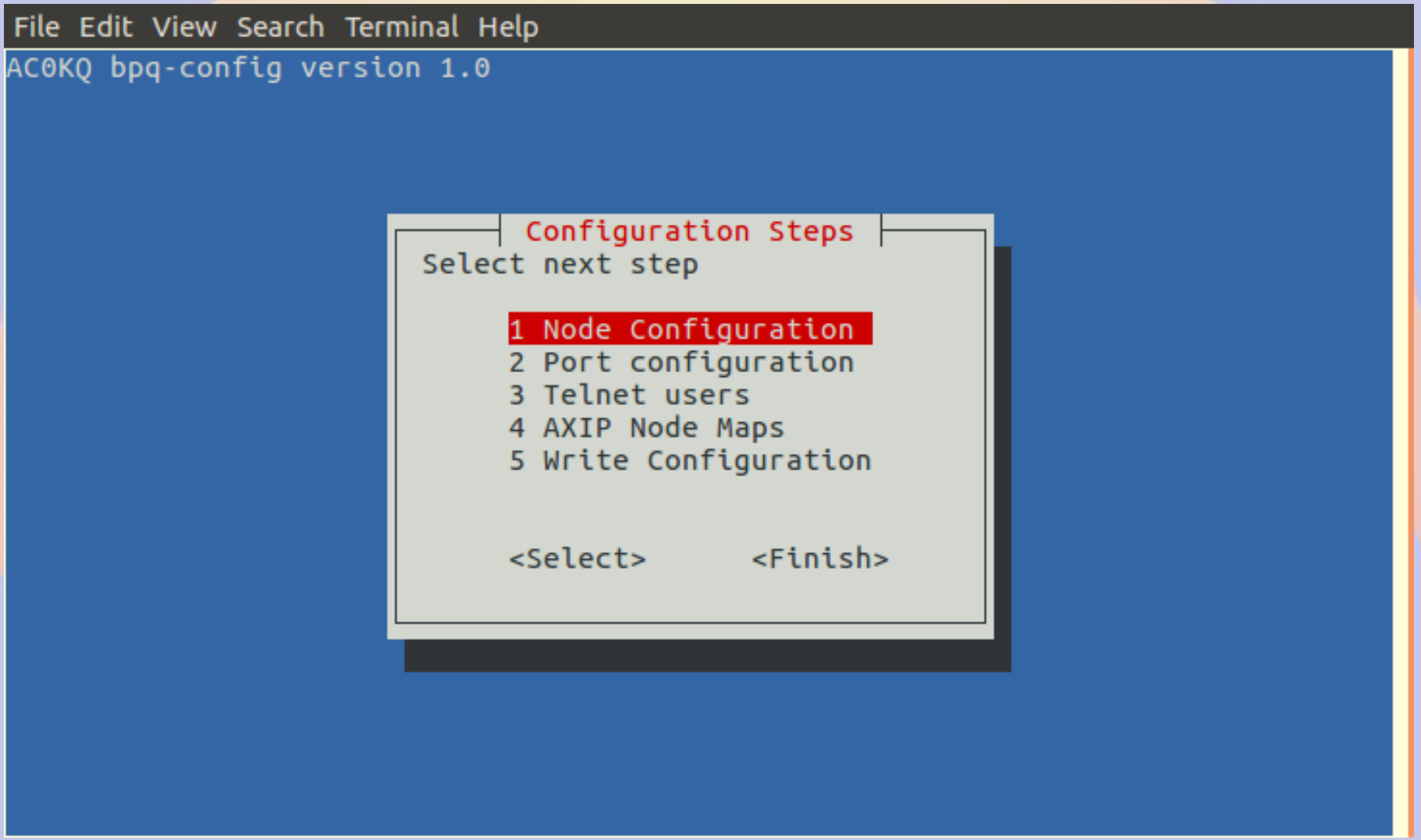
Forwarded Users



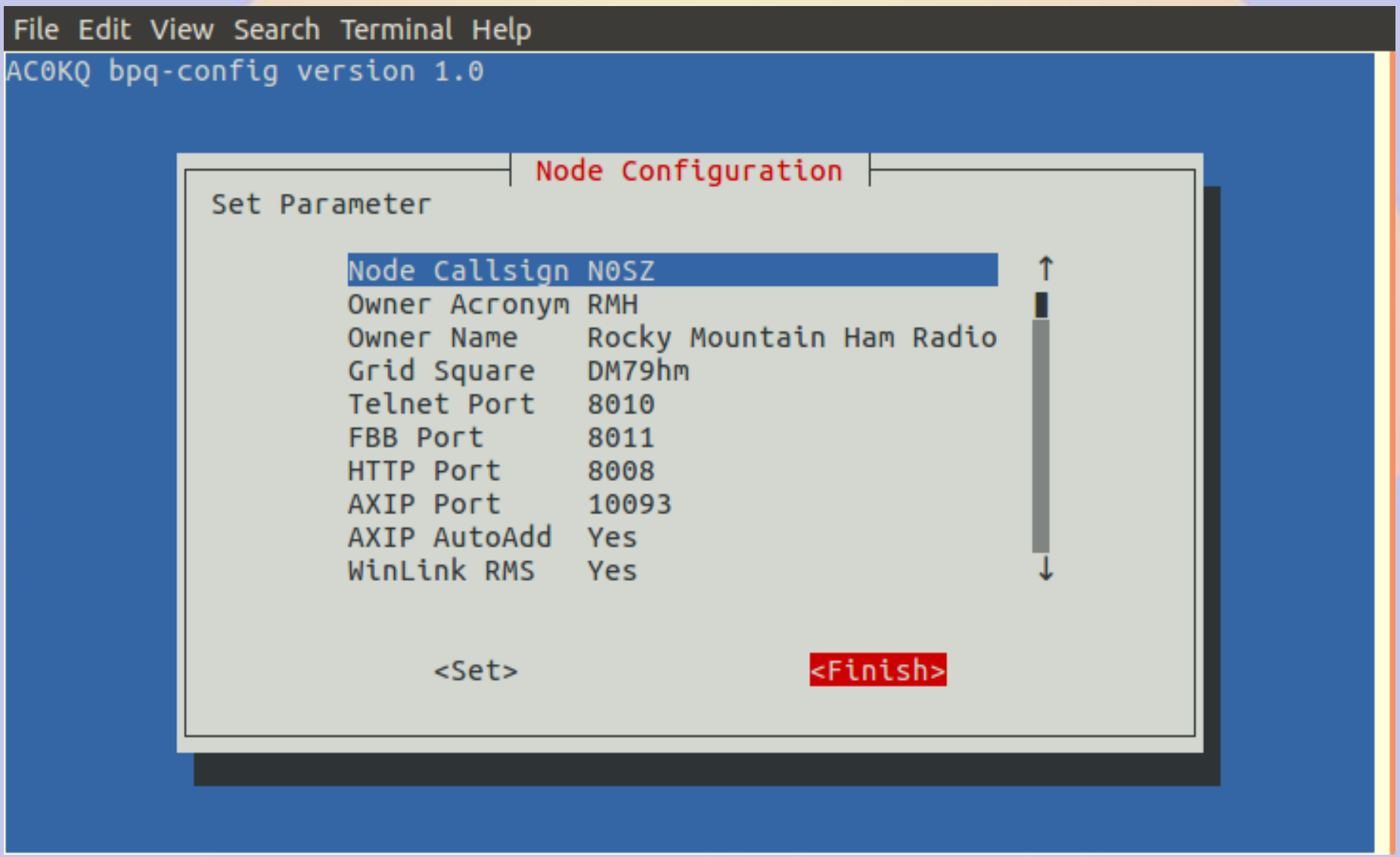
RF > N0SZ & AXIP > K0NTS-1



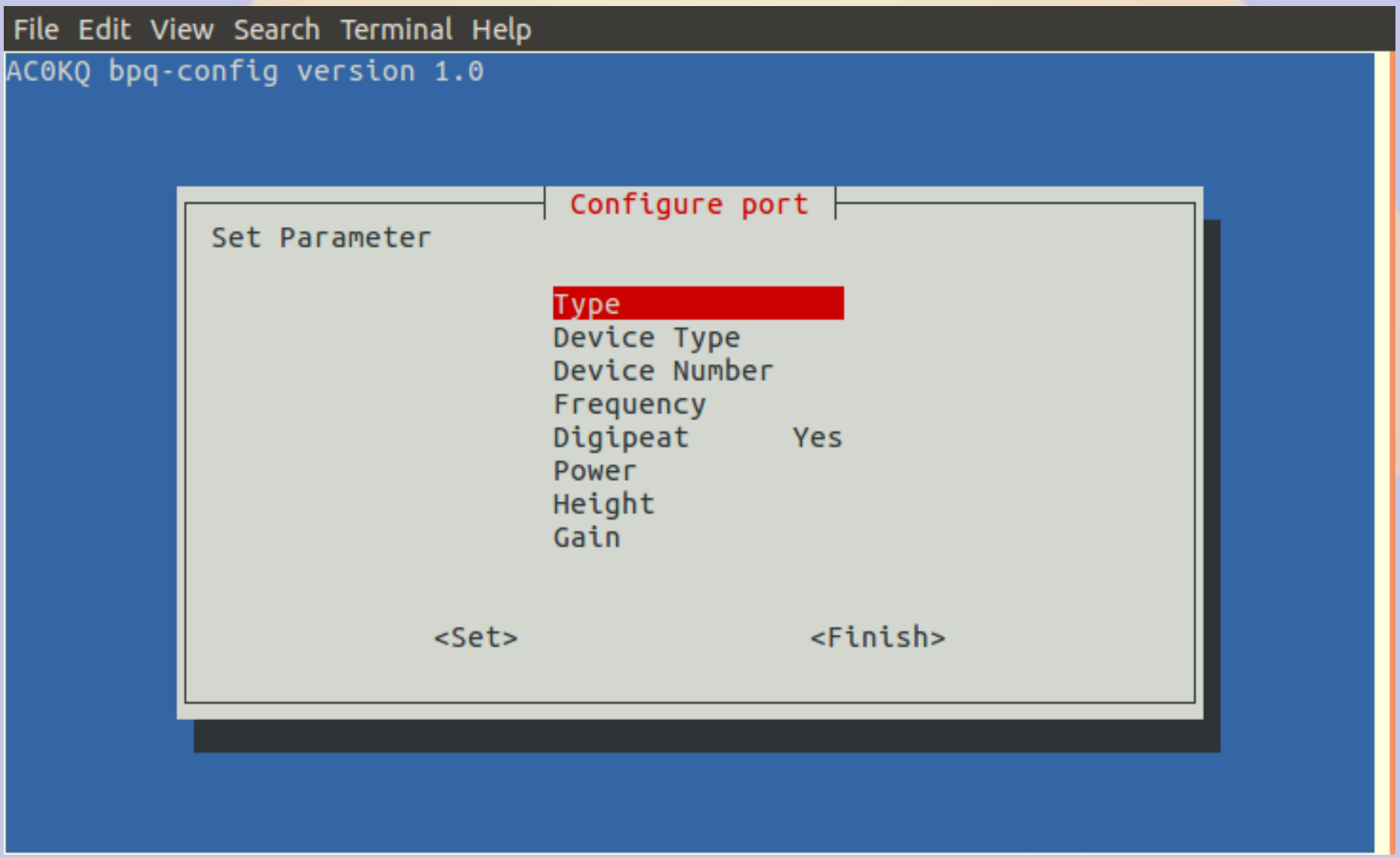
Manual Configuration Steps



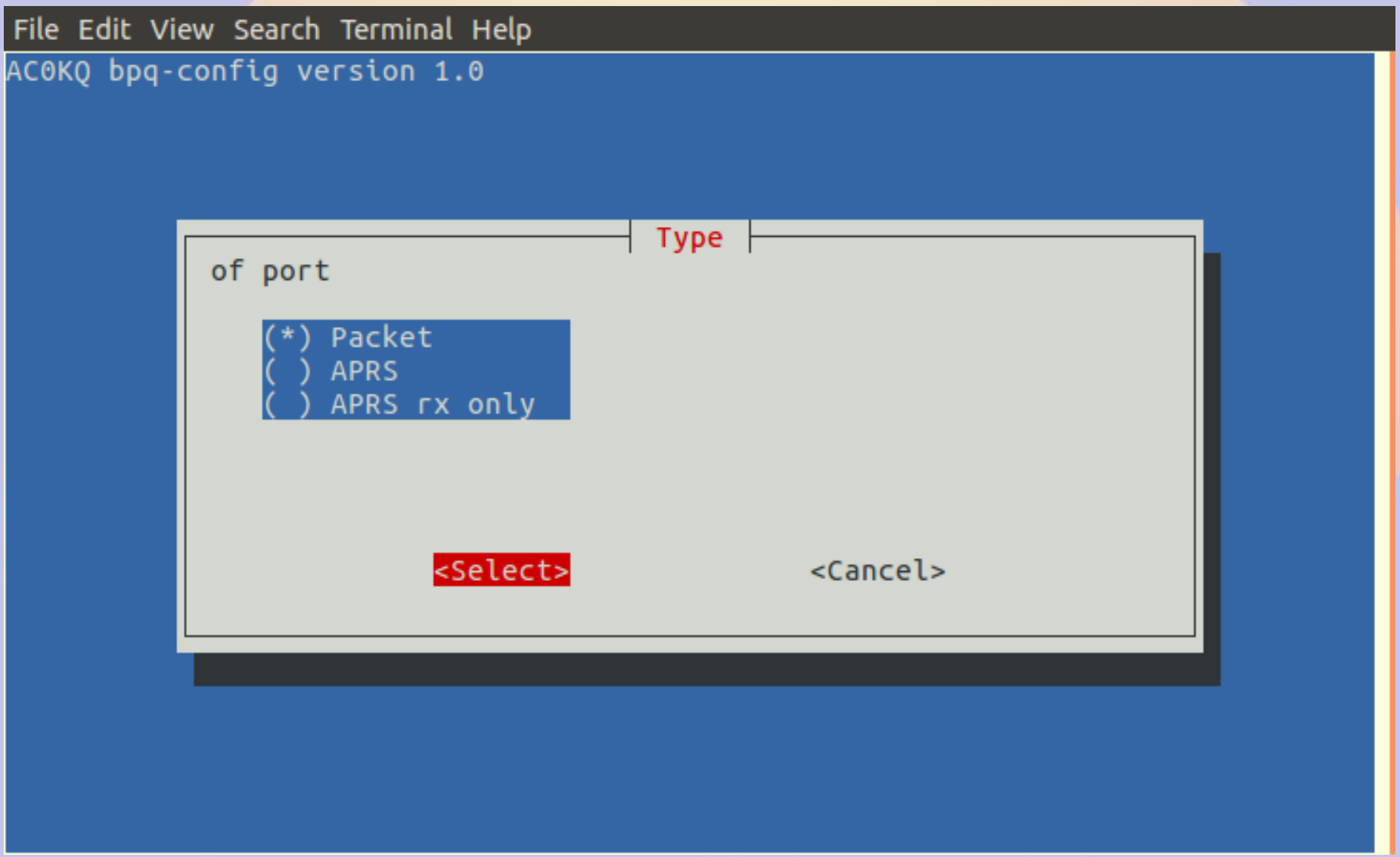
Node Configuration



Port Configuration

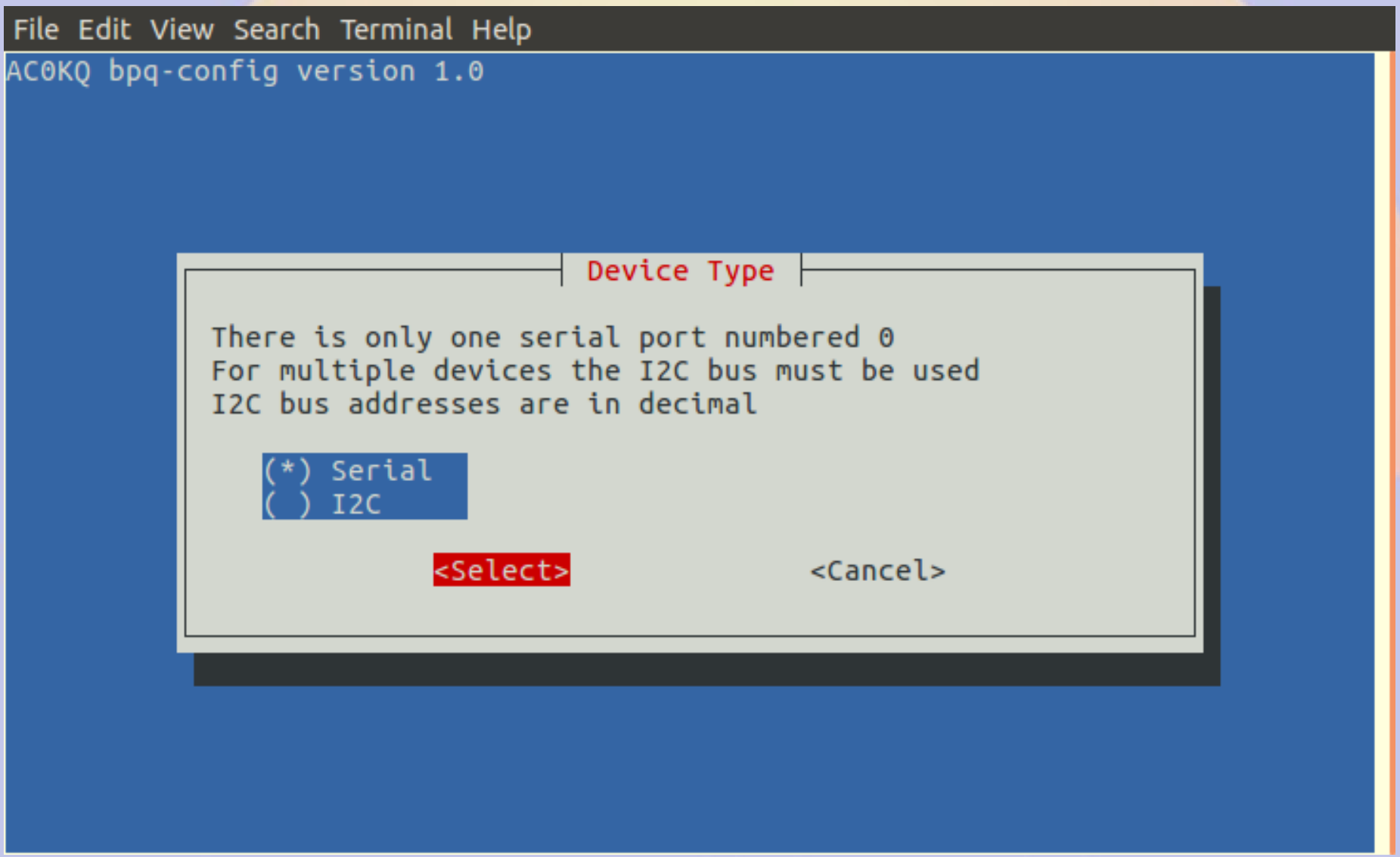


Port Type

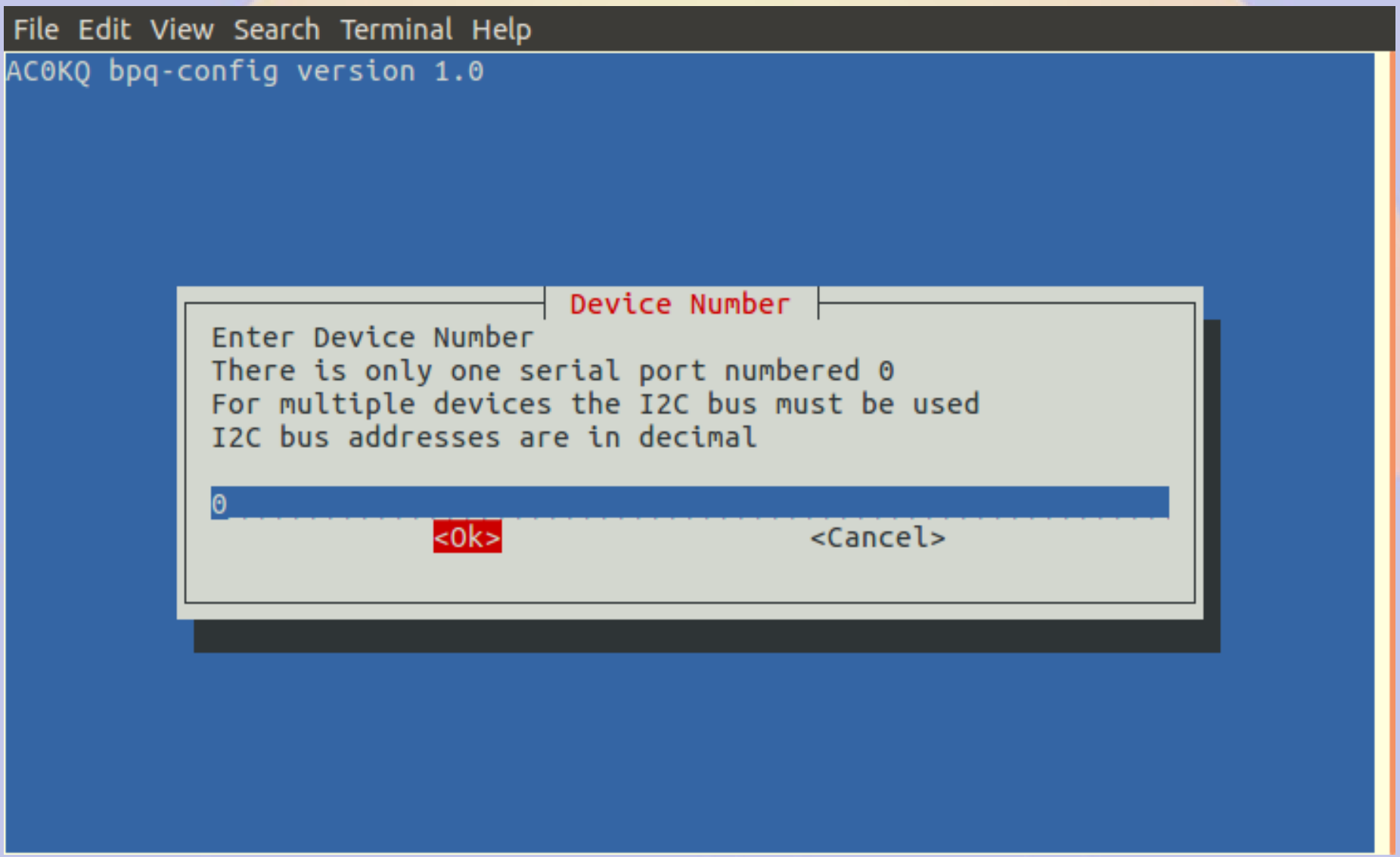

File Edit View Search Terminal Help
AC0KQ bpq-config version 1.0
Type
of port
(*) Packet
( ) APRS
( ) APRS rx only
<Select>
<Cancel>

Device Type

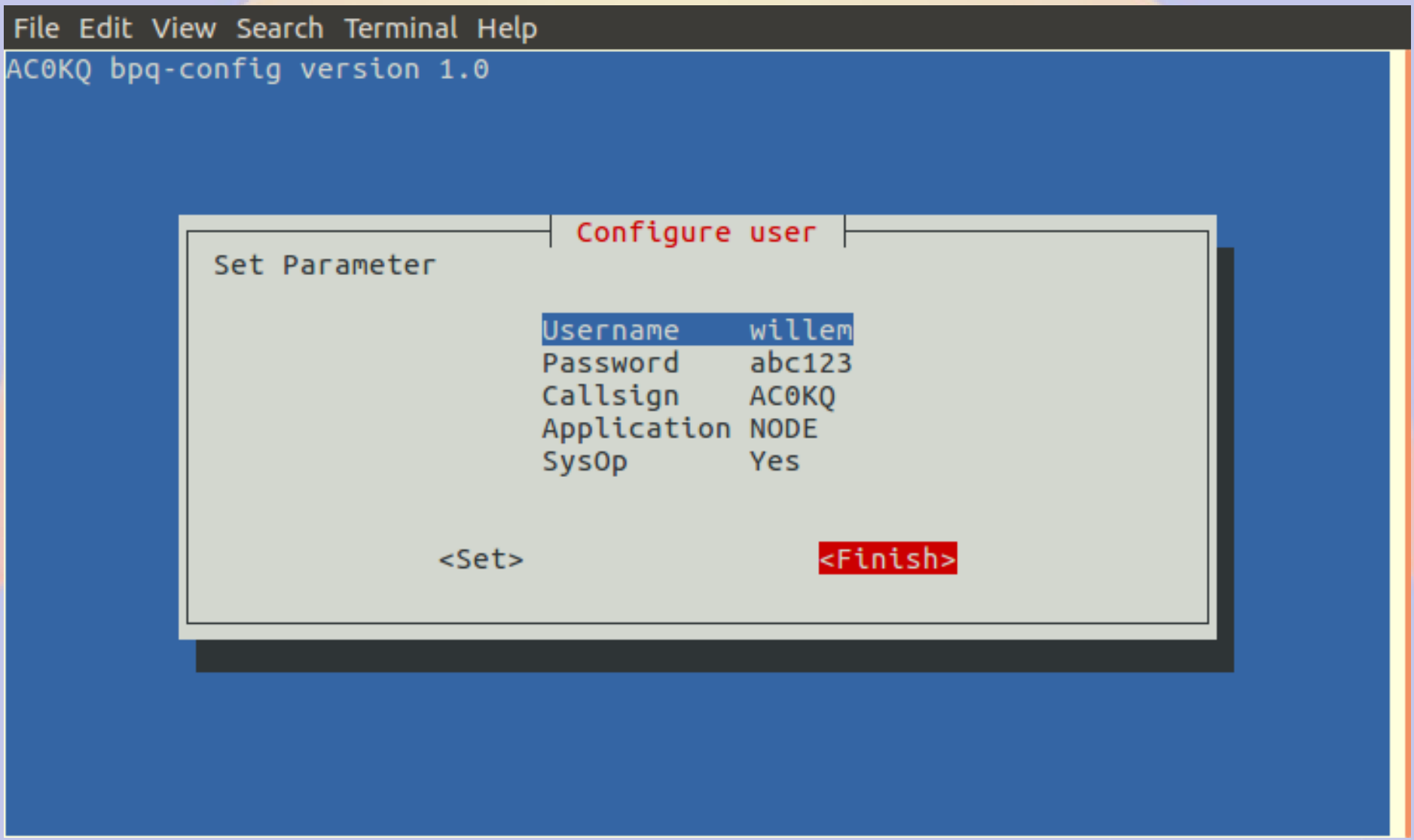
(Devices description is board specific)



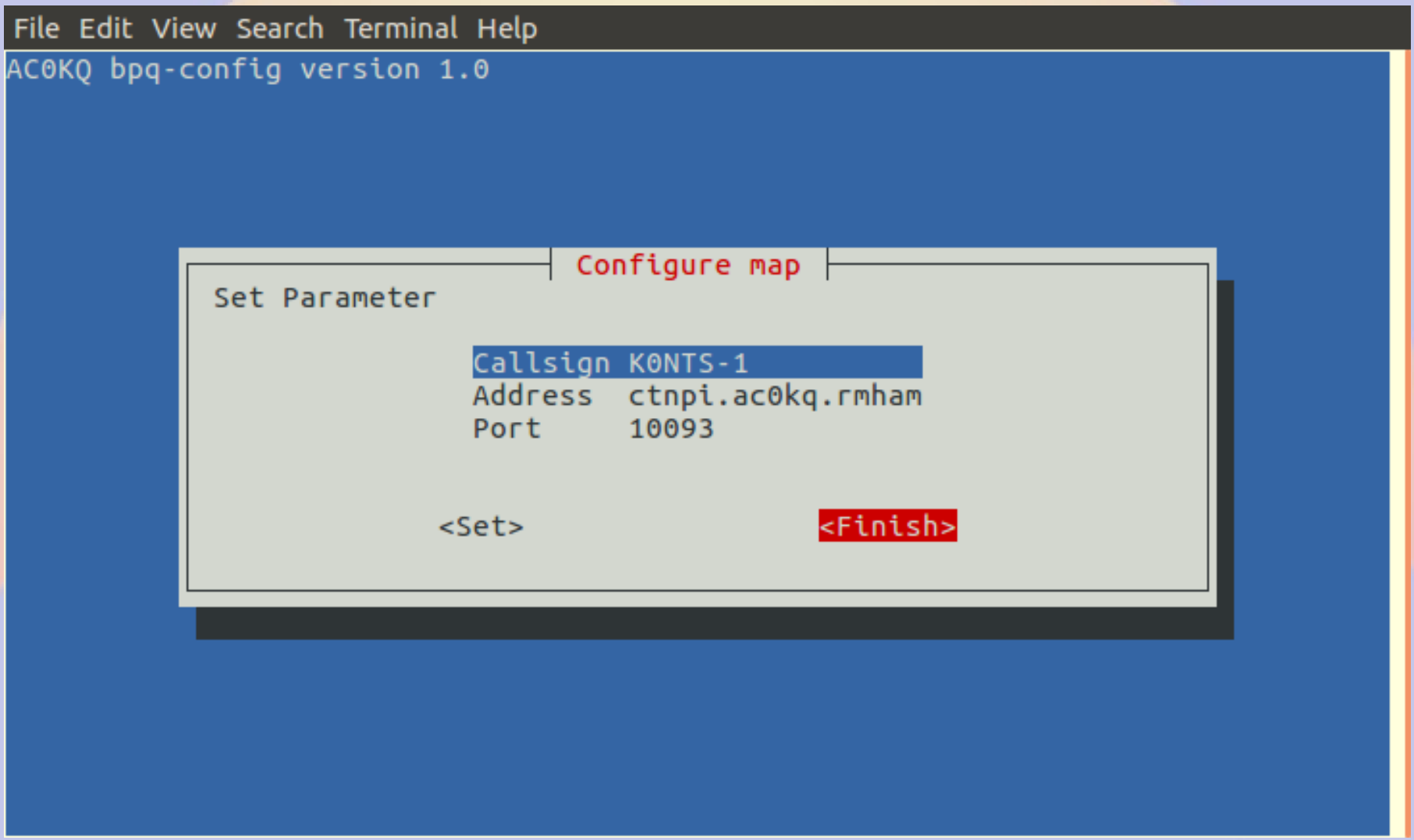
Device Number



User Configuration



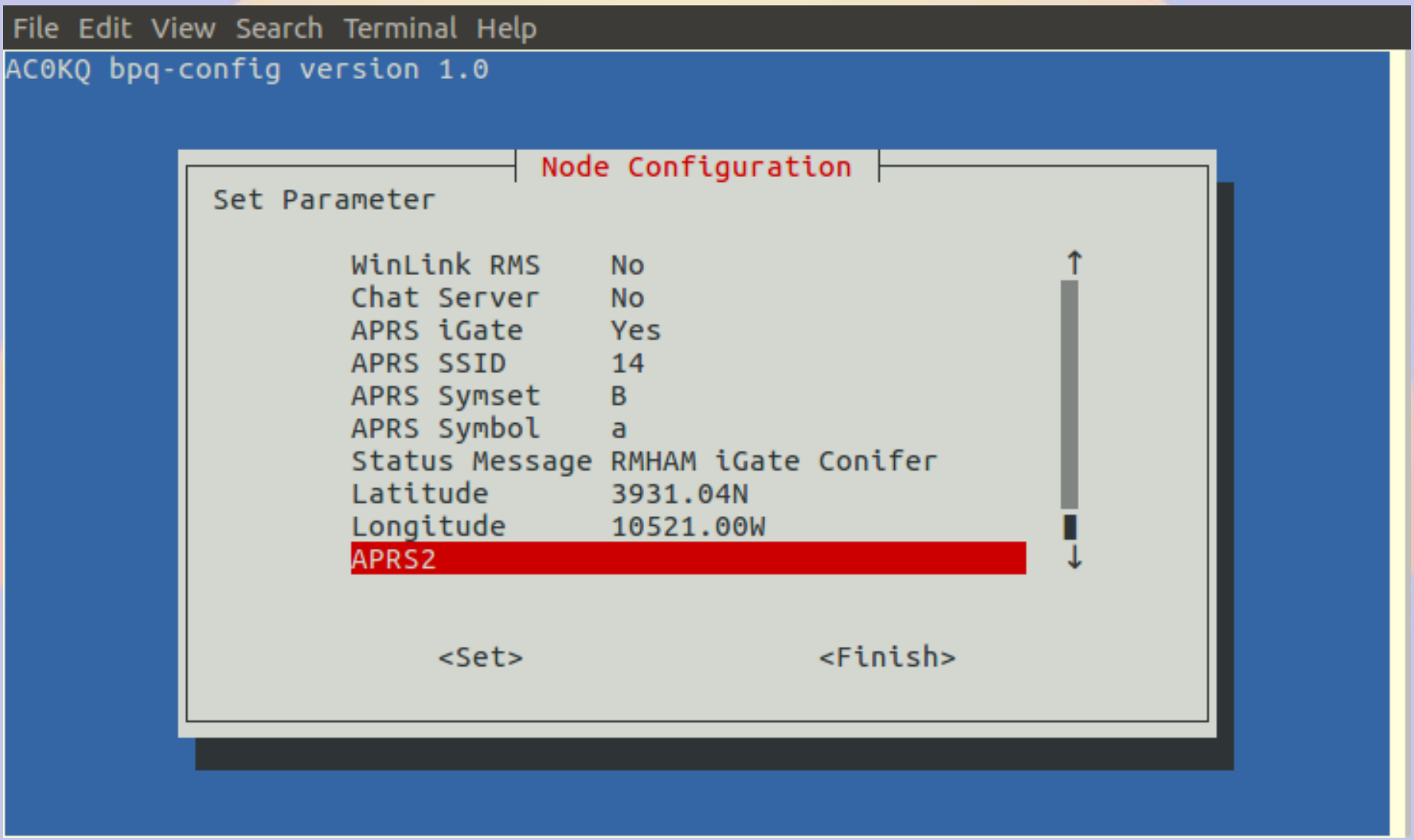
AXIP Map


File Edit View Search Terminal Help
AC0KQ bpq-config version 1.0
Configure map
Set Parameter
Callsign K0NTS-1
Address ctnpi.ac0kq.rmham
Port 10093
<Set>
<Finish>

Setting up an iGate

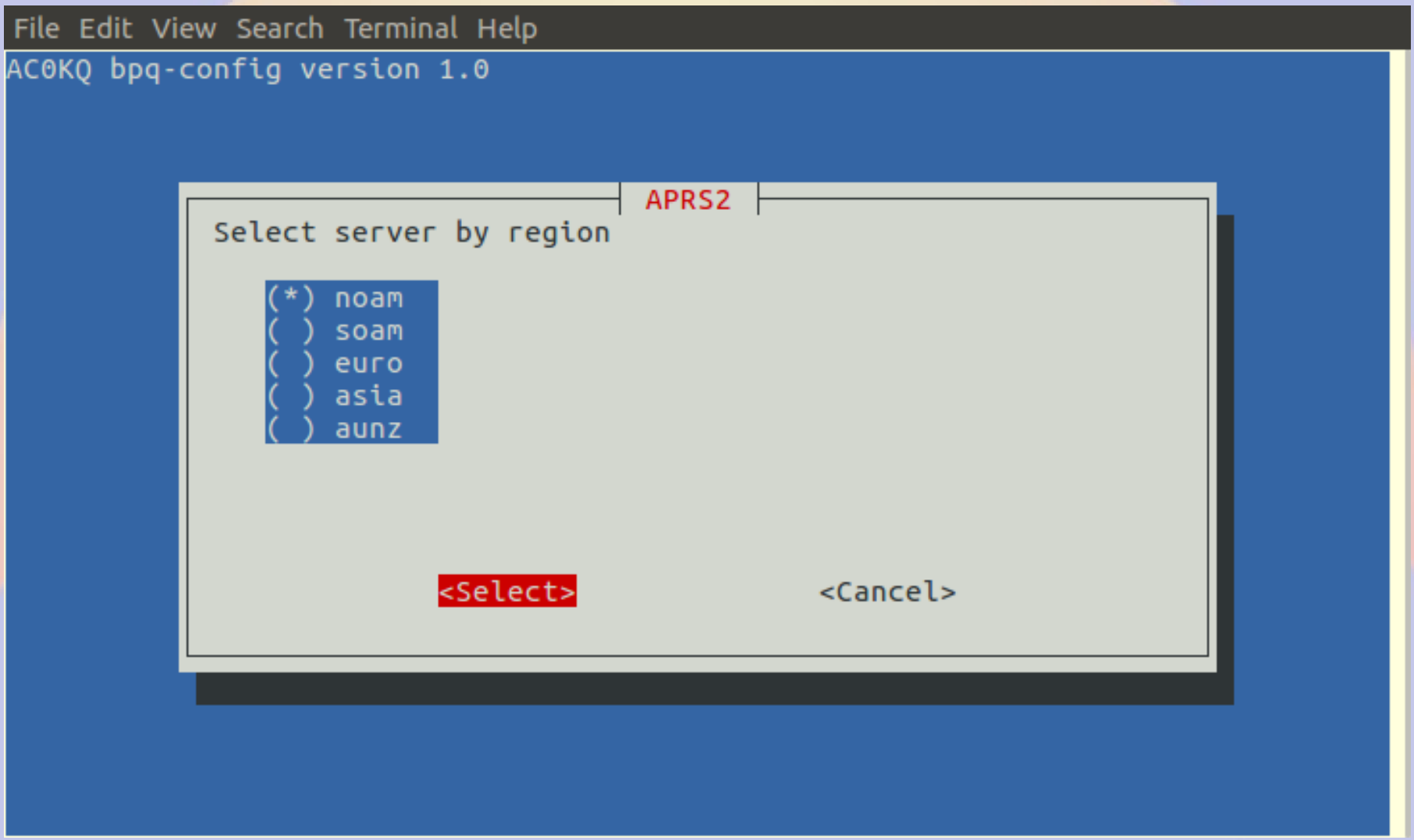
- This iGate setup is on a BeagleBone Black
  - The only difference with an rPi is the serial port names in the Port section
- The Node setup is the same as what was done previously
  - Some but not all the parameters are relevant
- Ports are mapped as Packet or APRS in port configuration
  - You can have both Packet and APRS ports on the same BPQ node

Emable iGate and set SSID, Symset, Symbol, Lat/Long

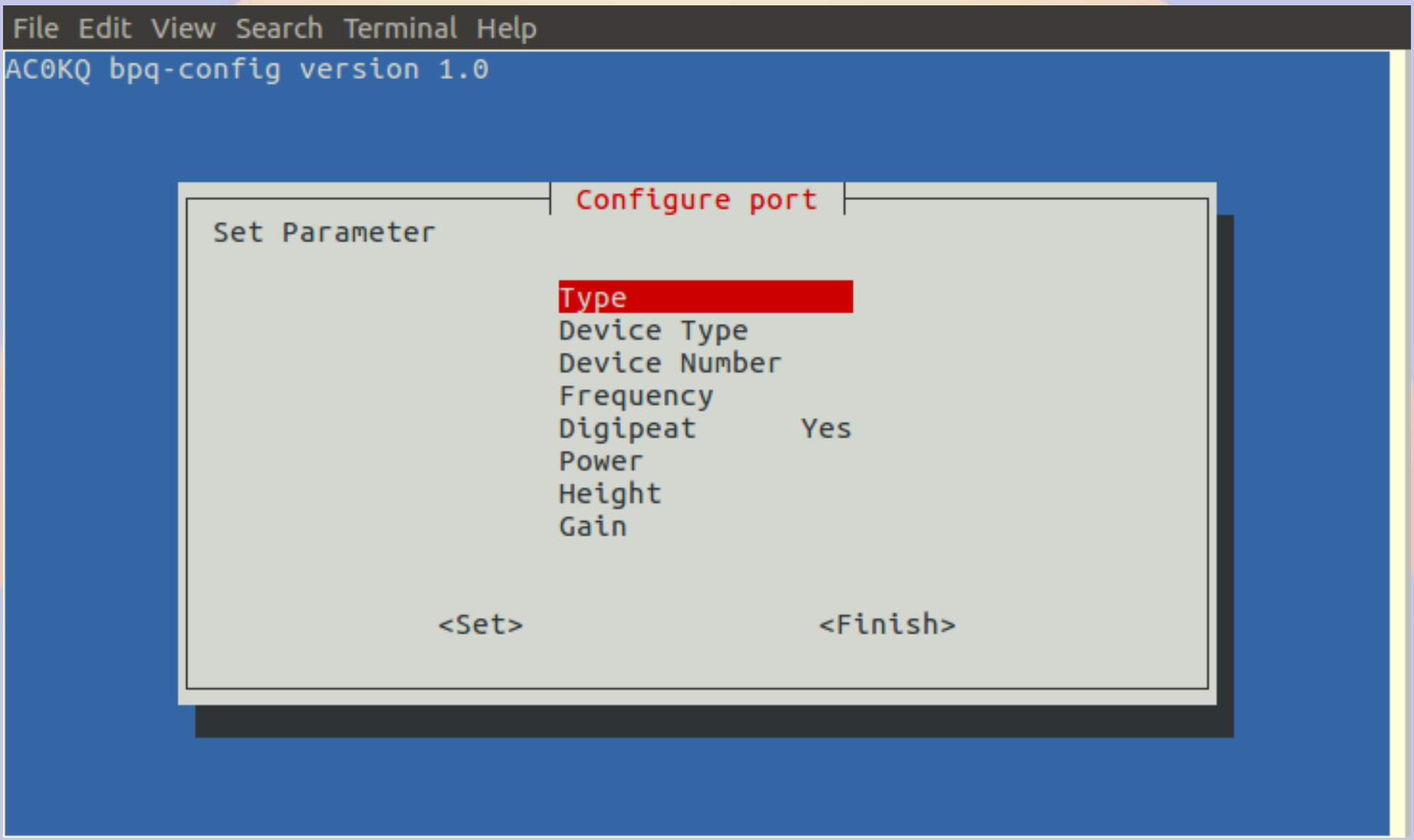


Select APRS2 Gateway

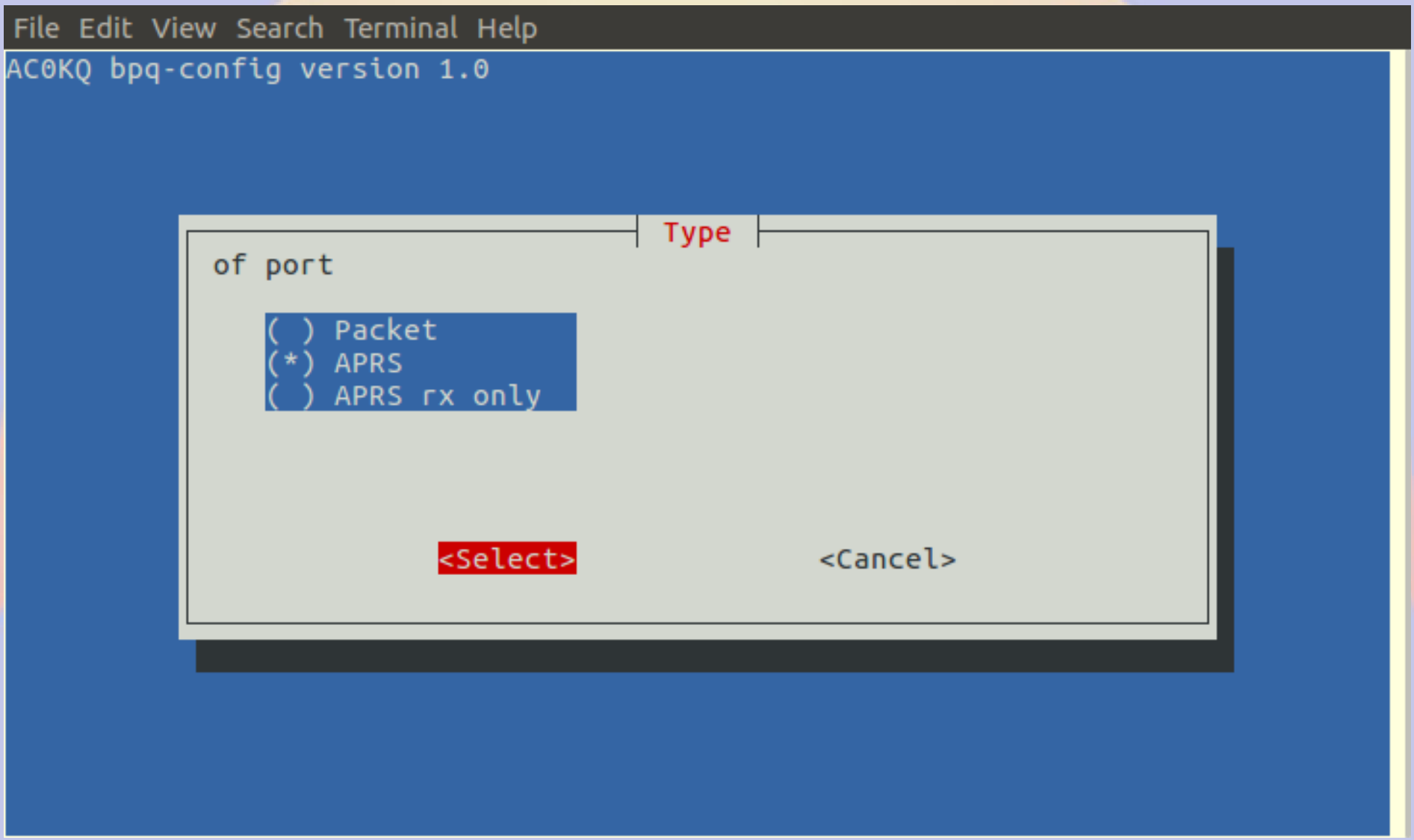
(bpq-config generates password automatically)



iGate Add Port Configuration

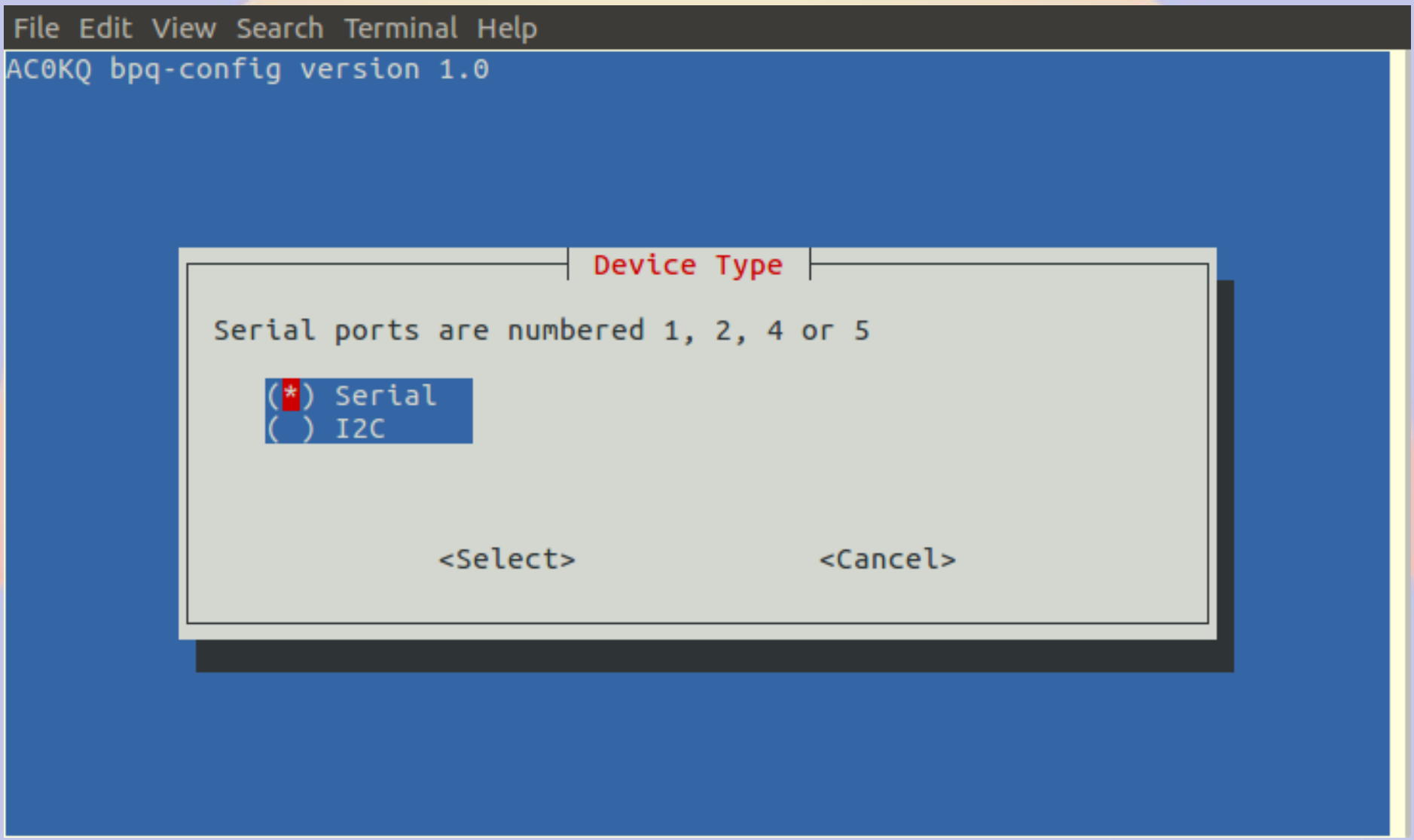


Set Port Type to APRS

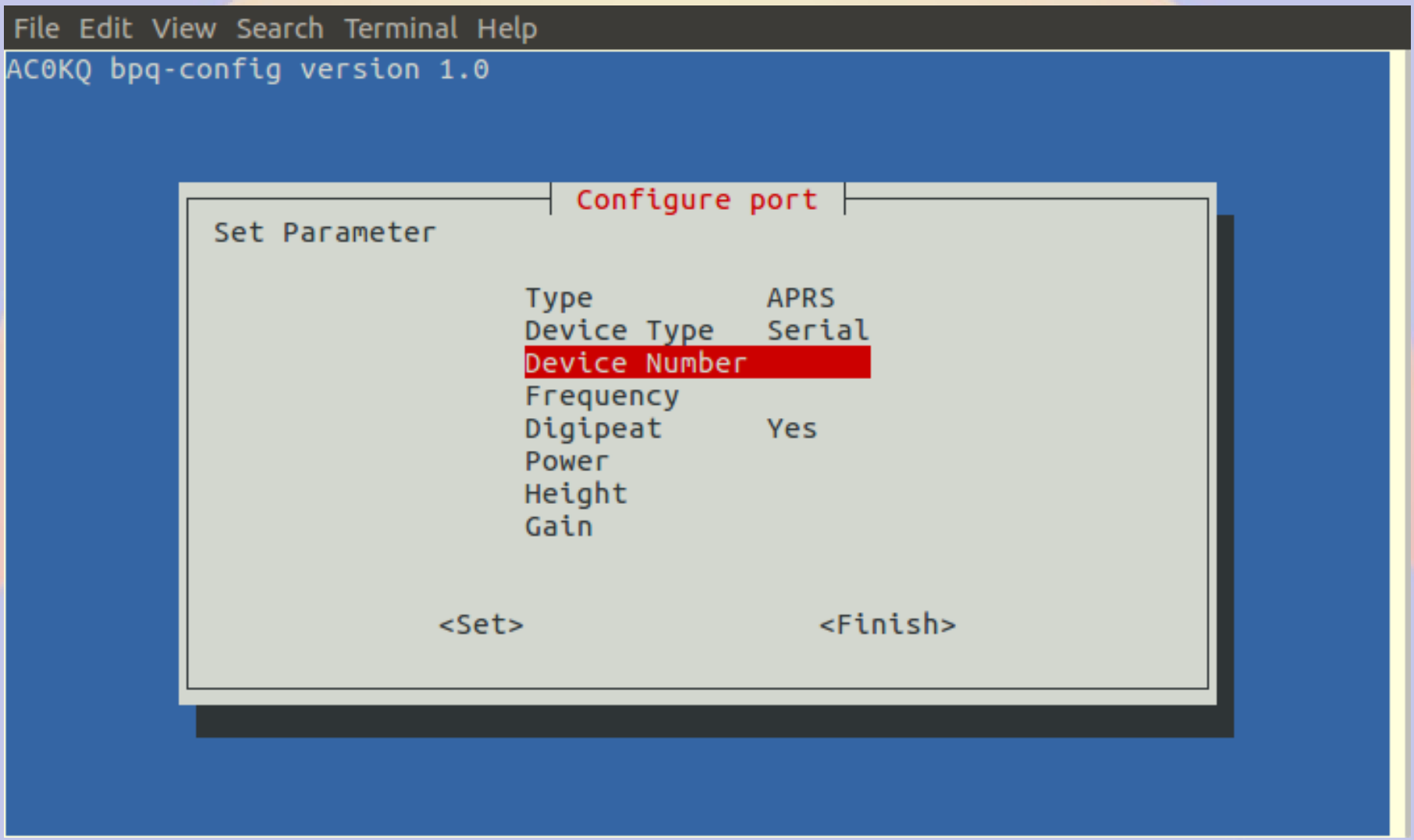


Set Serial Port

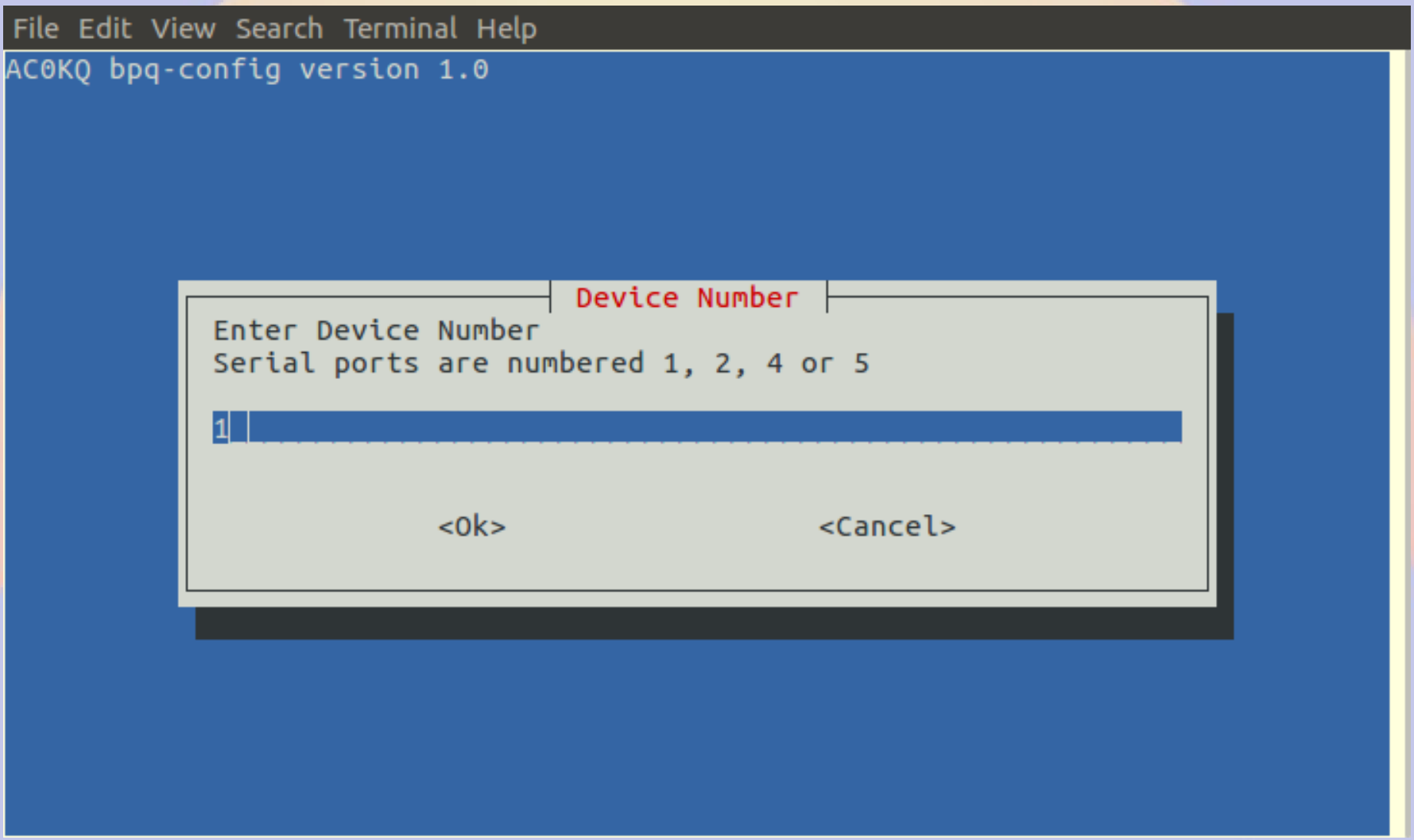
(Note that this is a Beaglebone so 4 ports)



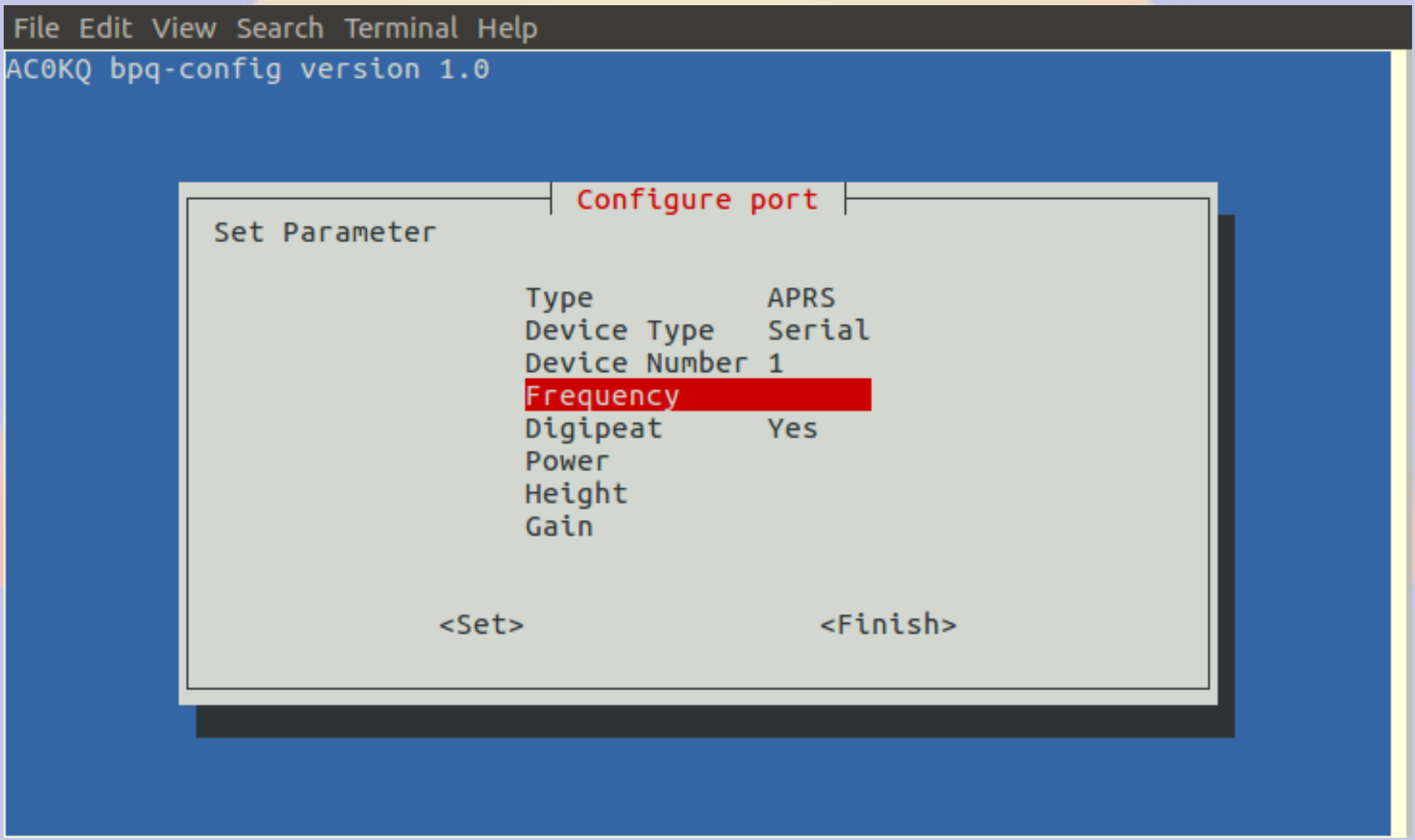
Select Serial Port Number



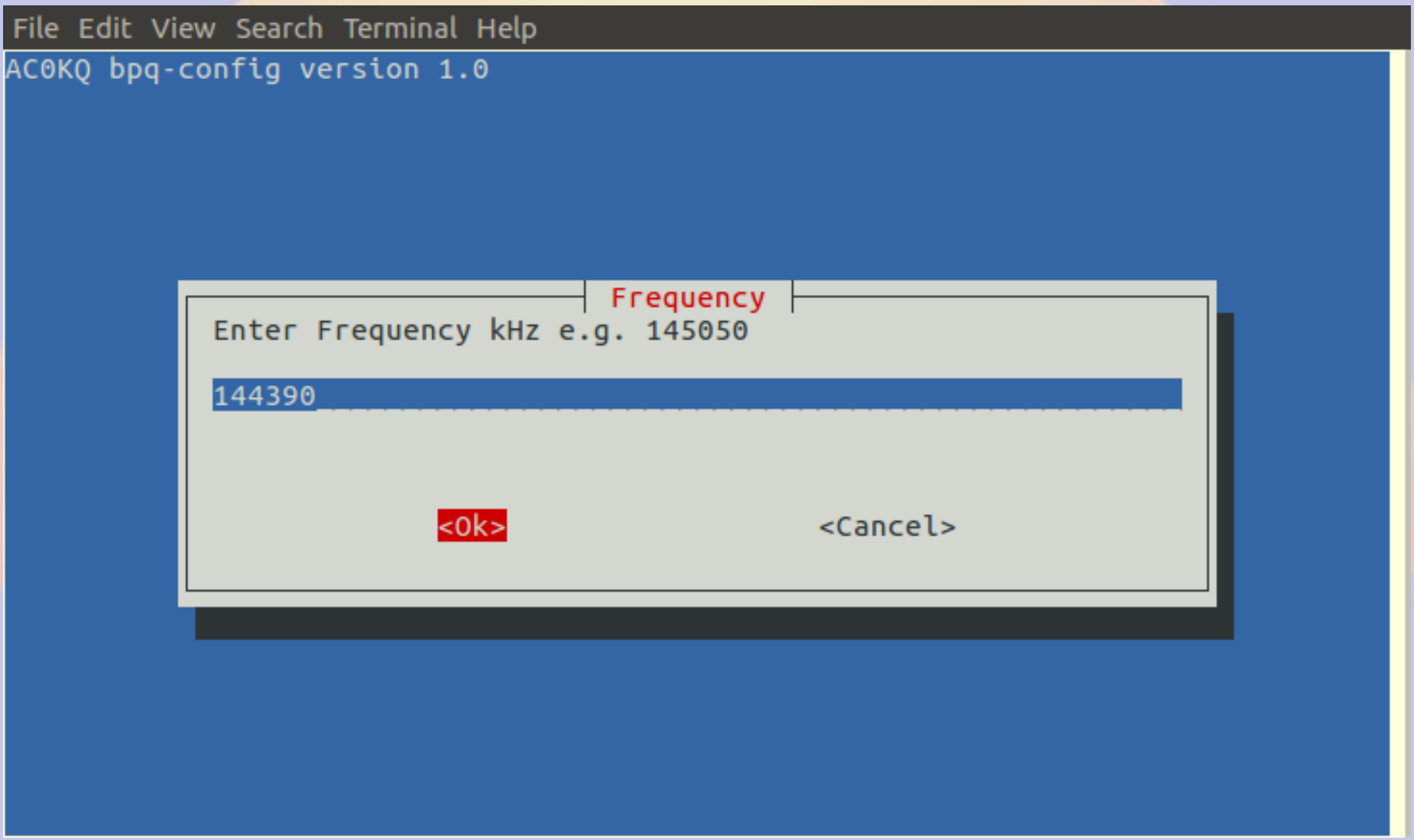
Select Serial Port Number



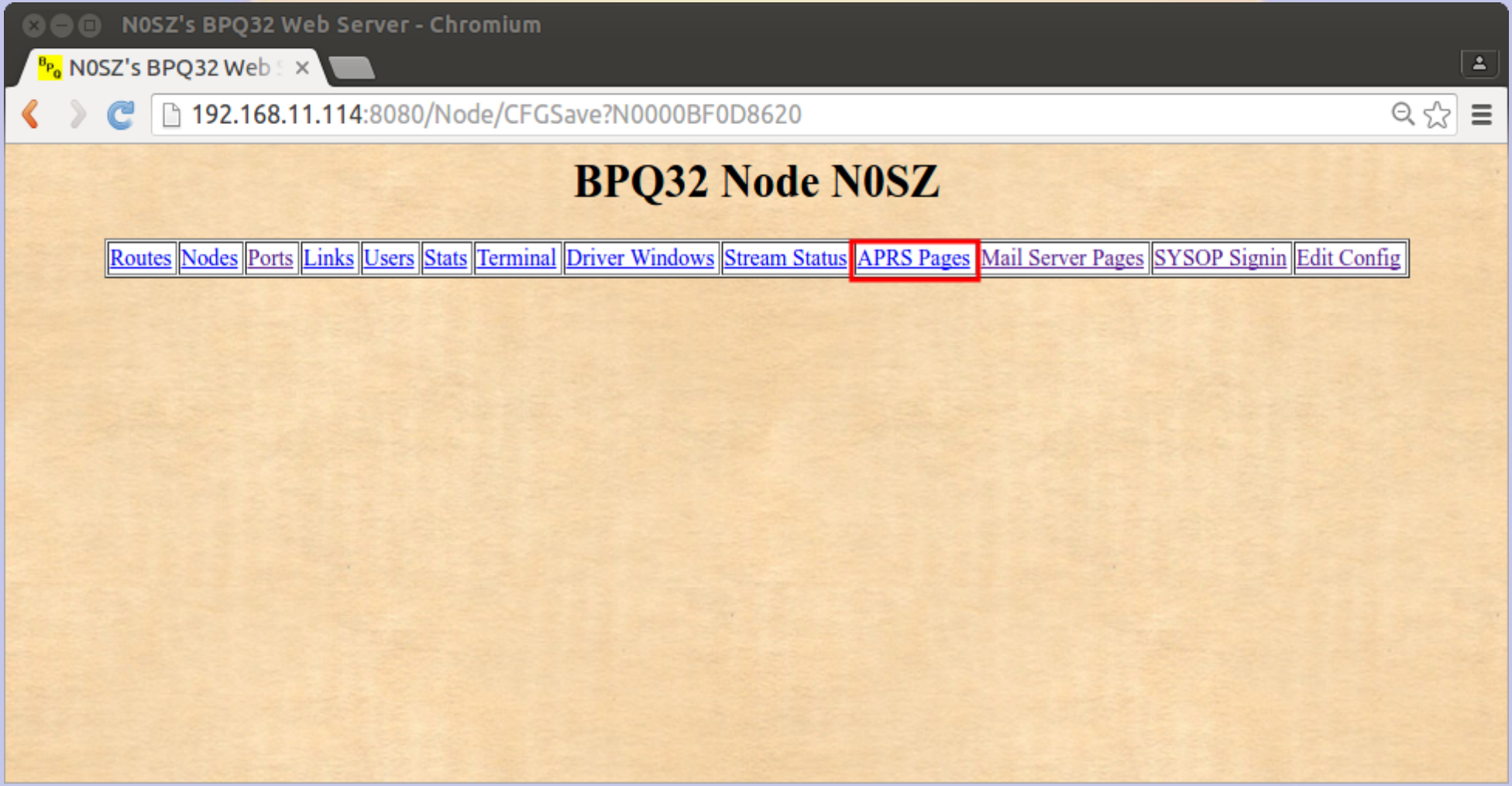
APRS Frequency 1



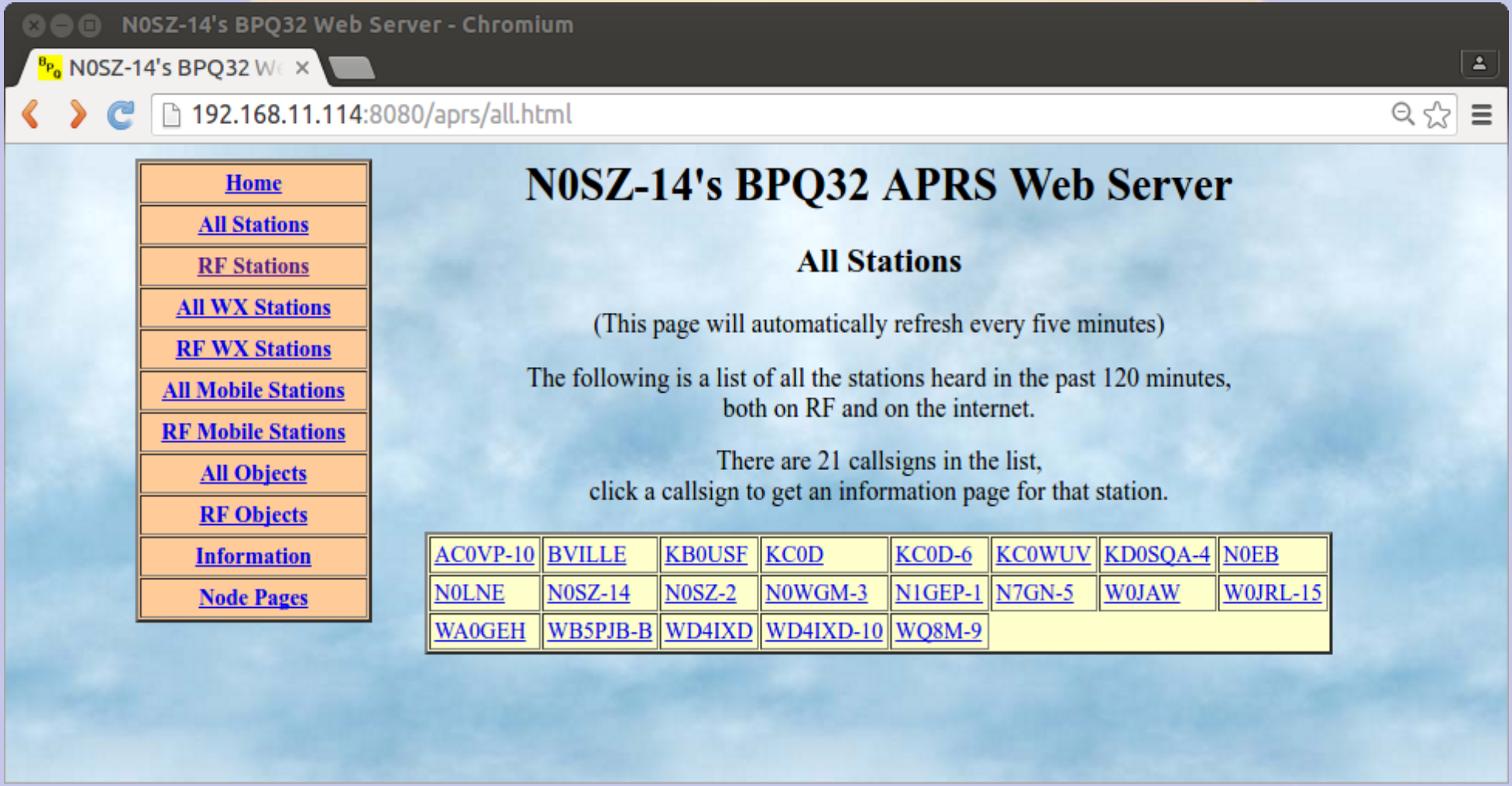
APRS Frequency 2



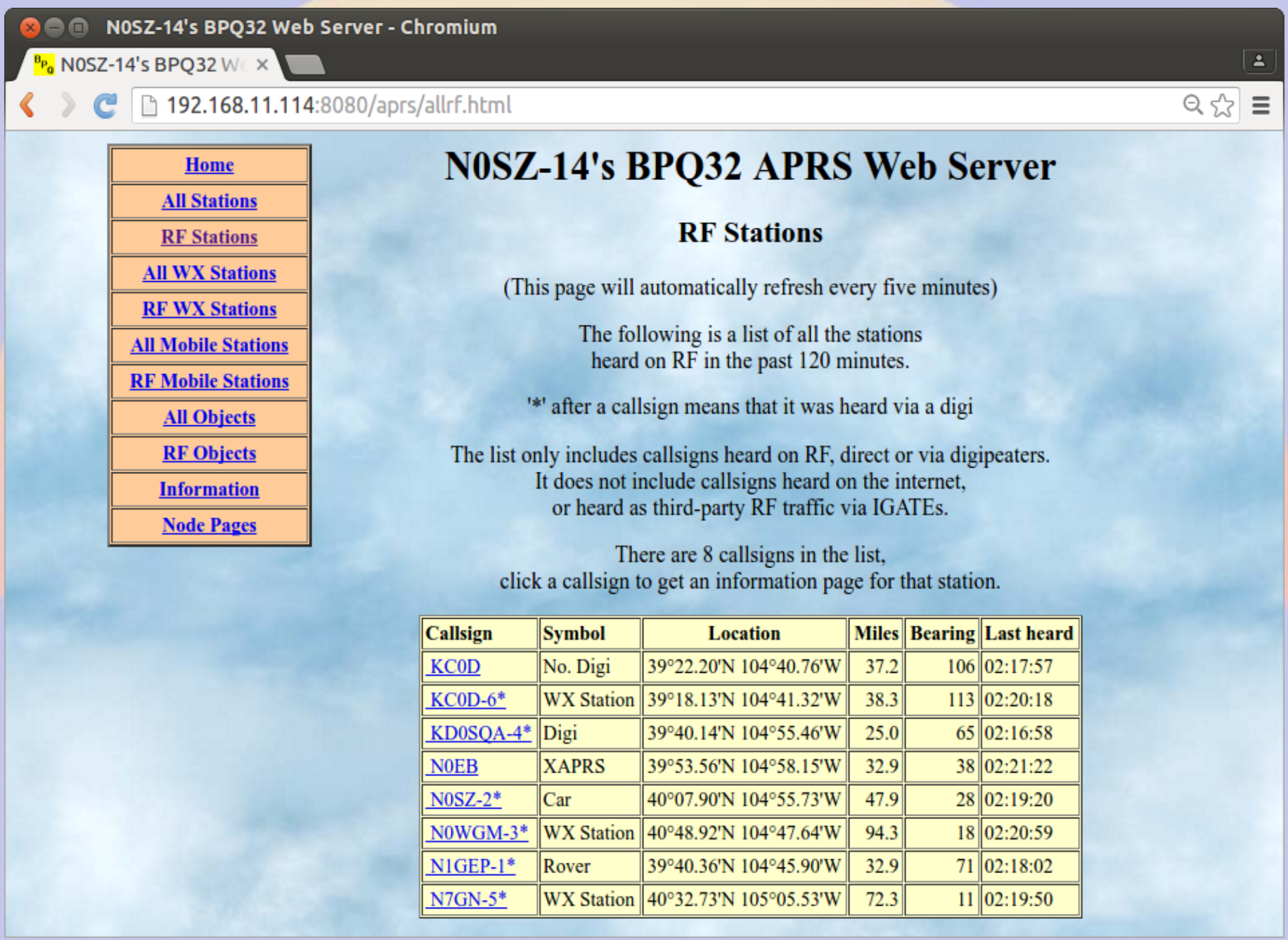
Node Page Update



APRS Main Page



APRS RF Stations



Callsign	Symbol	Location	Miles	Bearing	Last heard
KC0D	No. Digi	39°22.20'N 104°40.76'W	37.2	106	02:17:57
KC0D-6*	WX Station	39°18.13'N 104°41.32'W	38.3	113	02:20:18
KD0SQA-4*	Digi	39°40.14'N 104°55.46'W	25.0	65	02:16:58
N0EB	XAPRS	39°53.56'N 104°58.15'W	32.9	38	02:21:22
N0SZ-2*	Car	40°07.90'N 104°55.73'W	47.9	28	02:19:20
N0WGM-3*	WX Station	40°48.92'N 104°47.64'W	94.3	18	02:20:59
N1GEP-1*	Rover	39°40.36'N 104°45.90'W	32.9	71	02:18:02
N7GN-5*	WX Station	40°32.73'N 105°05.53'W	72.3	11	02:19:50

APRS Station Map

N0SZ-14's BPQ32 Web Server - Chromium

N0SZ-14's BPQ32 W

192.168.11.114:8080/aprs/find.cgi?call=N0SZ-2

- Home
- All Stations
- RF Stations
- All WX Stations
- RF WX Stations
- All Mobile Stations
- RF Mobile Stations
- All Objects
- RF Objects
- Information
- Node Pages

N0SZ-14's BPQ32 APRS Web Server

(This page will automatically refresh every five minutes)

Information about N0SZ-2

Click the callsign to look it up on qrz.com

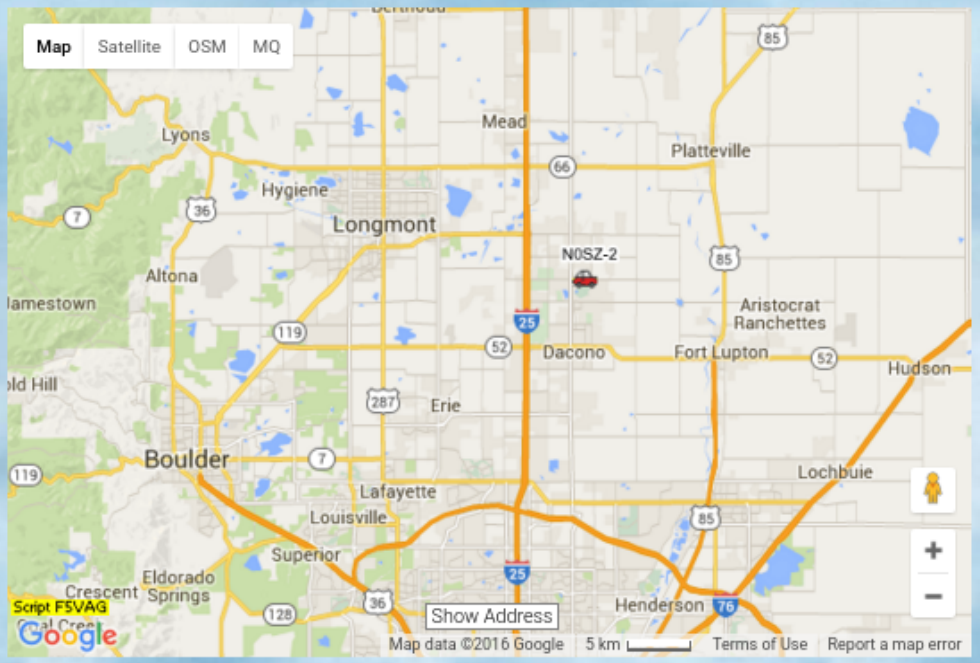
Location: 40°07.90'N 104°55.73'W

The bearing from N0SZ-14 to N0SZ-2 is 028 degrees, the distance is 47.9 Miles

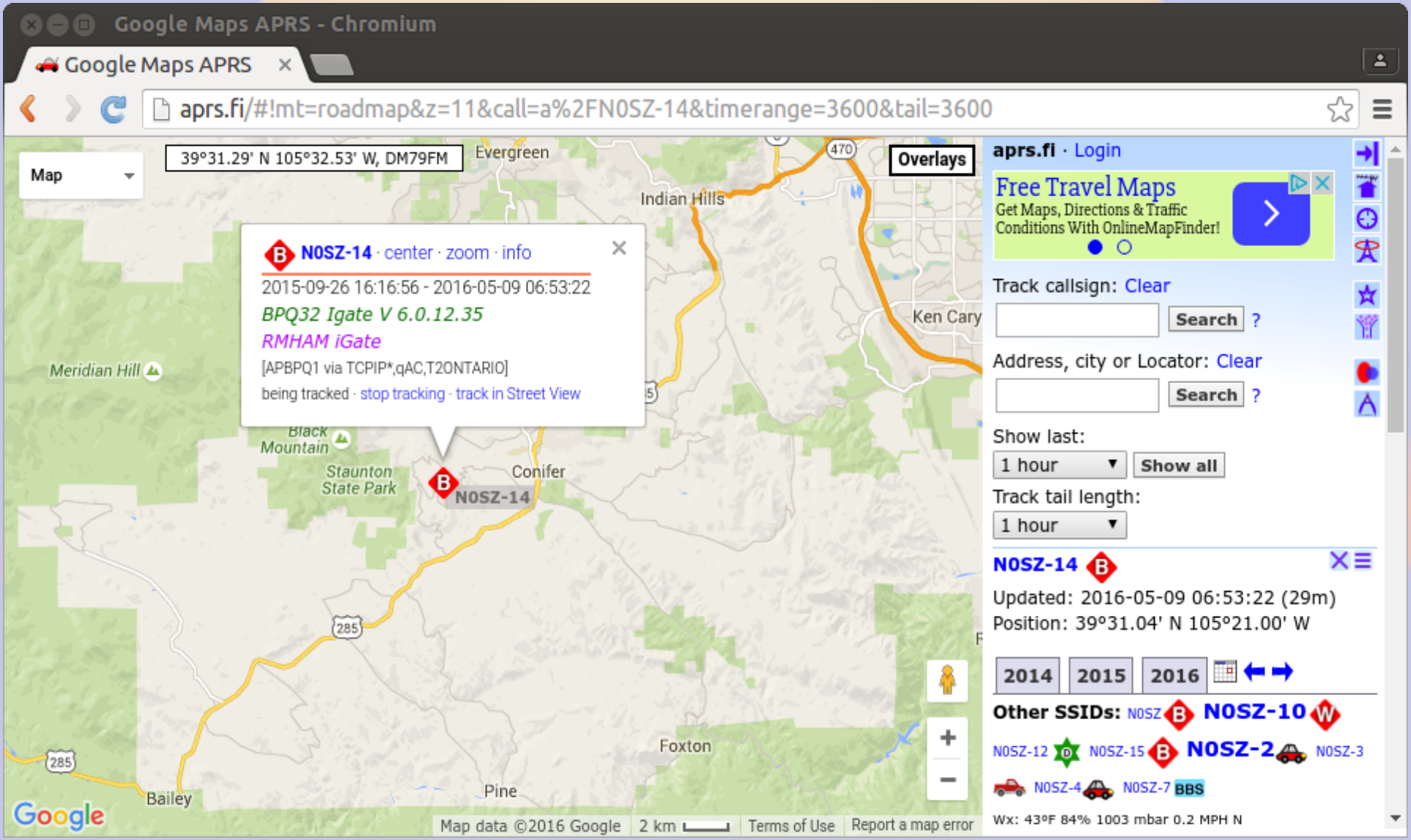
Last posit: TPPWYP,W0UPS-5,WIDE1,KC0D,WIDE2*

Status:

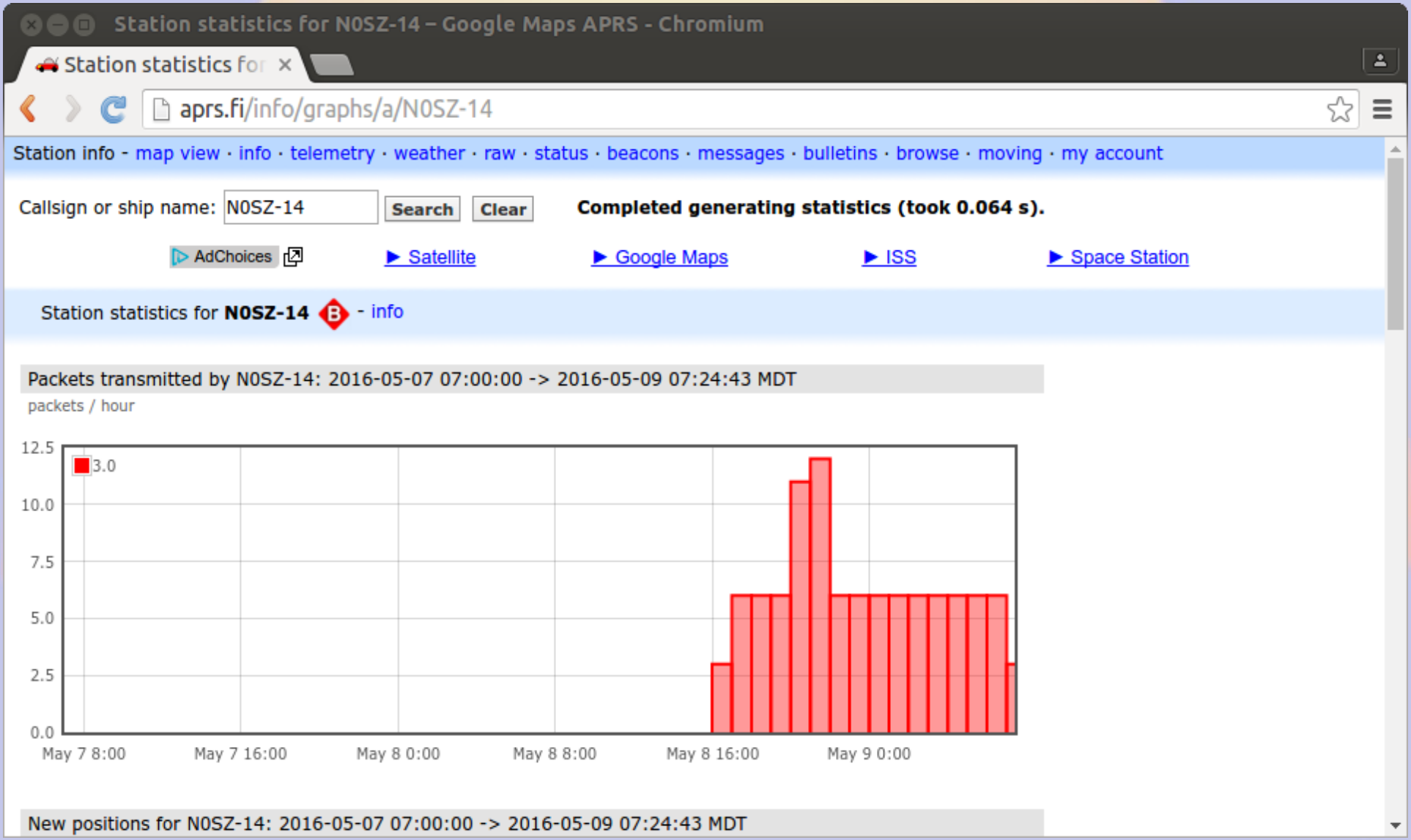
Last heard 00:05:11 ago



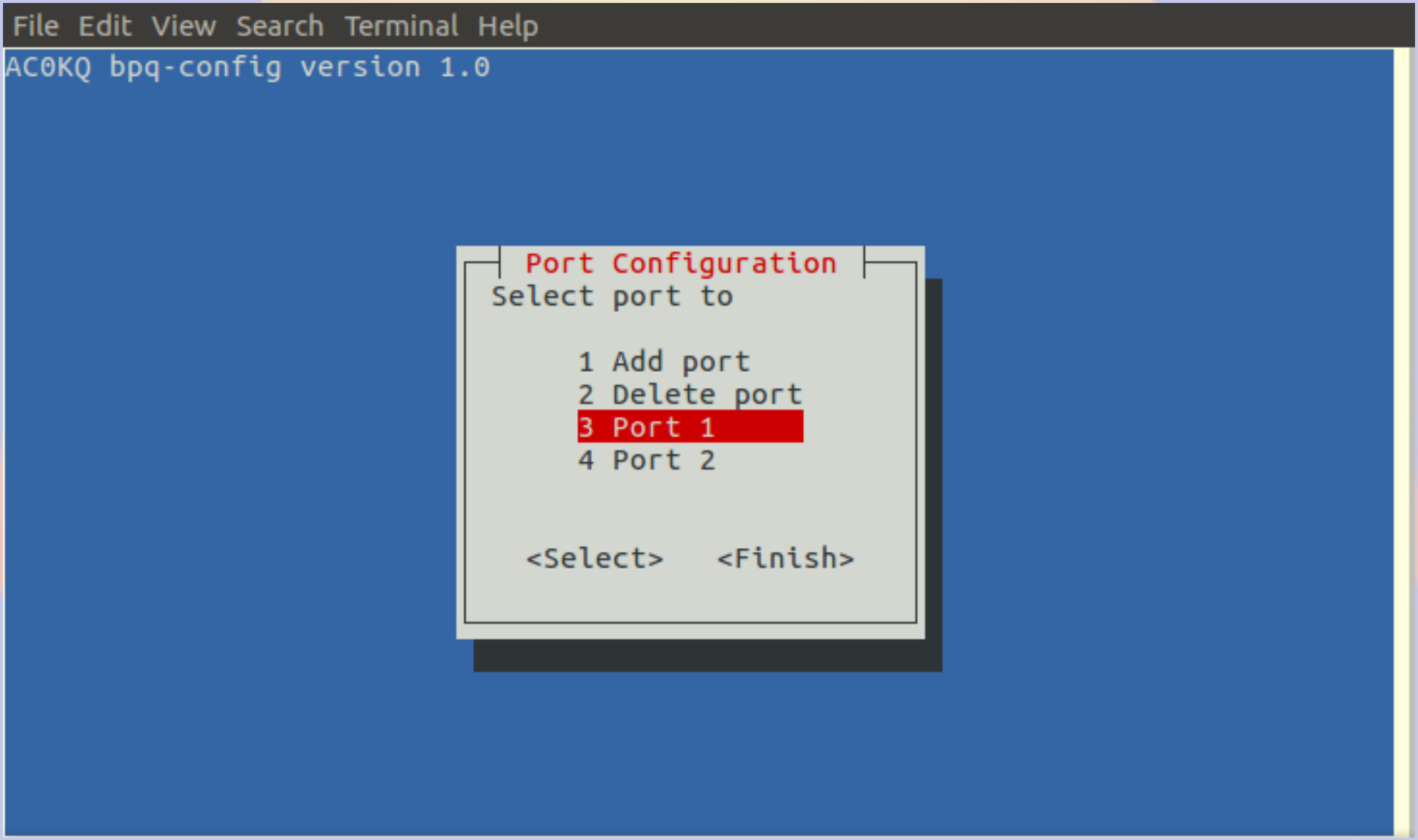
Report on aprs.fi



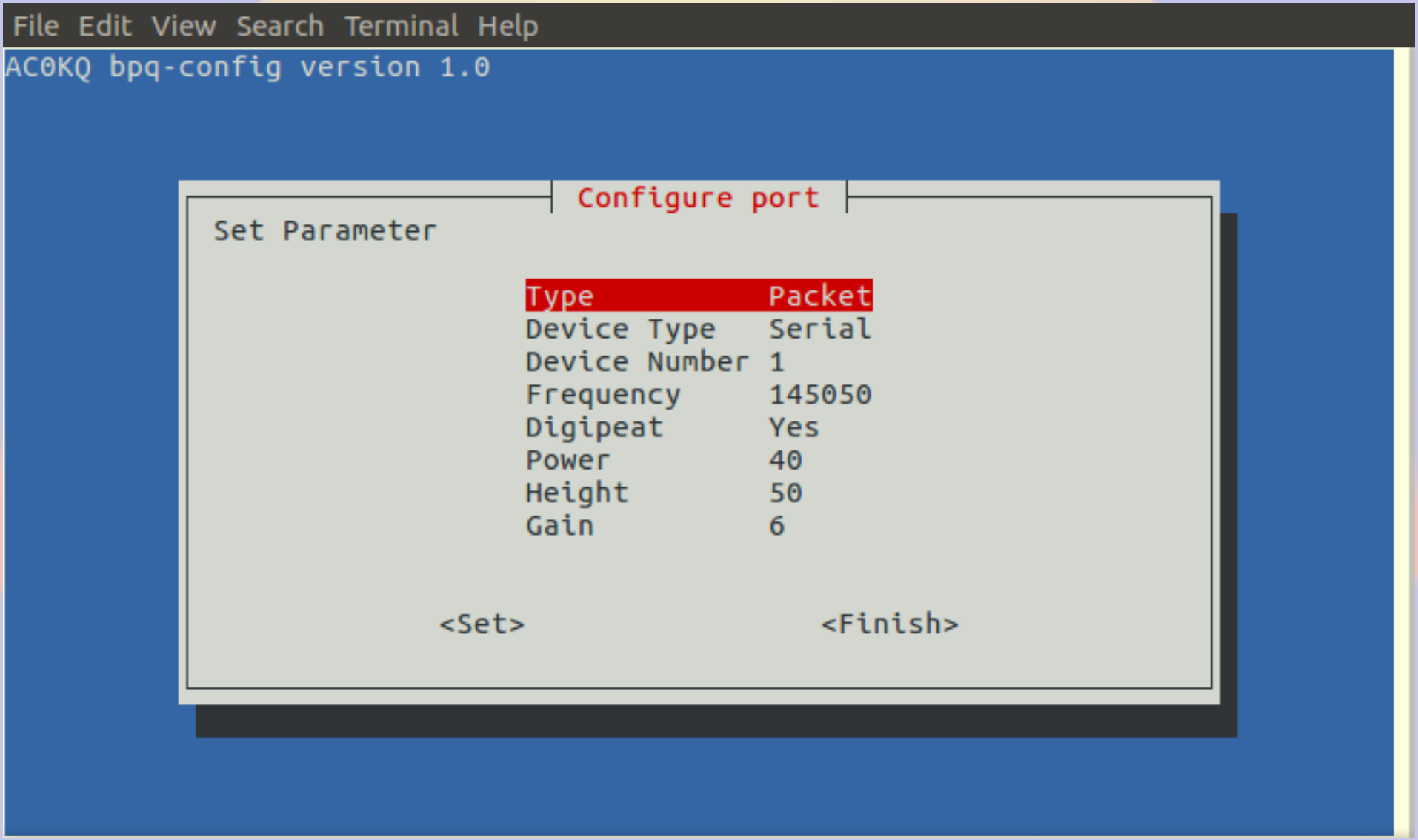
Data graph on aprs.fi



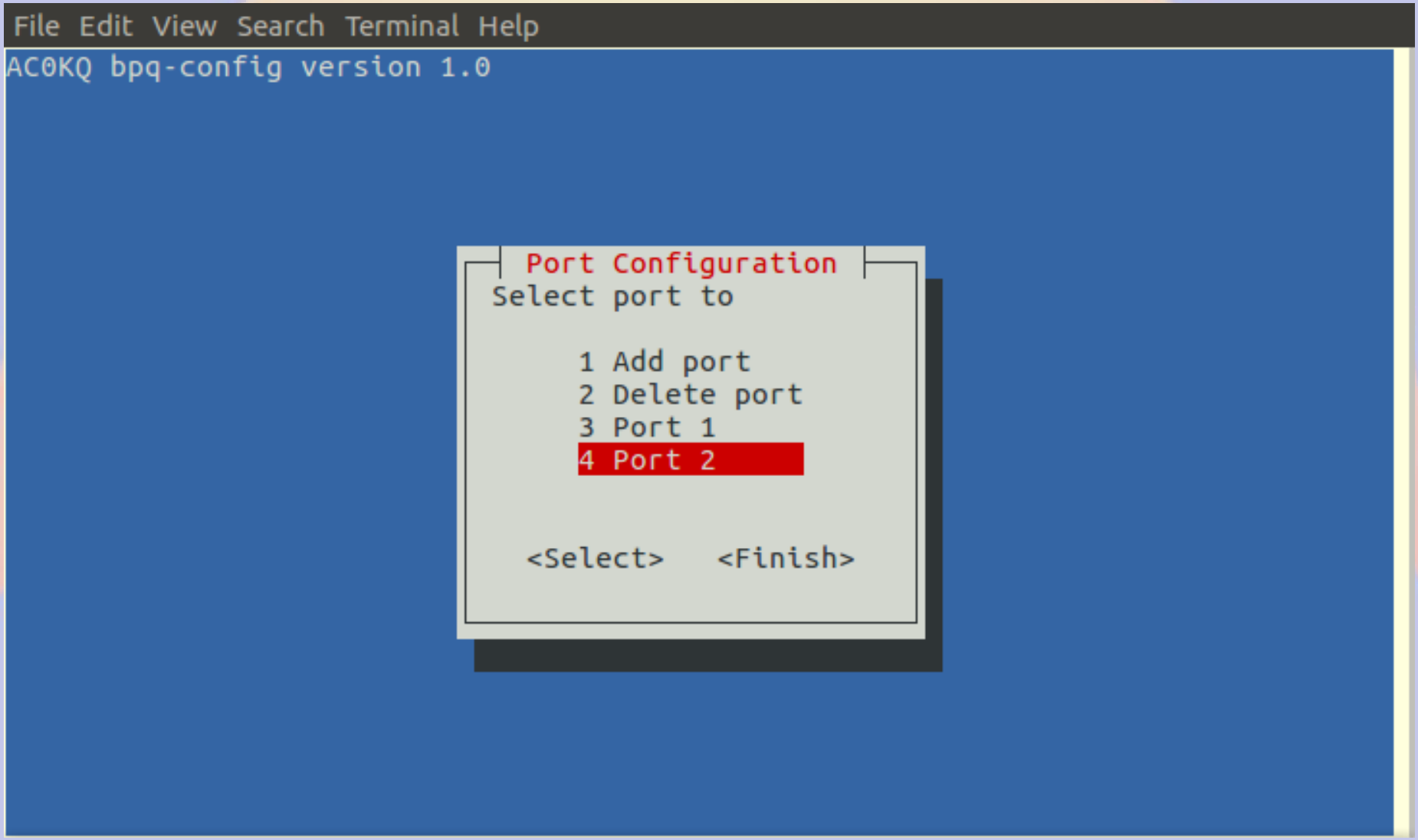
BPQ Port 1


File Edit View Search Terminal Help
AC0KQ bpq-config version 1.0
Port Configuration
Select port to
1 Add port
2 Delete port
3 Port 1
4 Port 2
<Select>
<Finish>

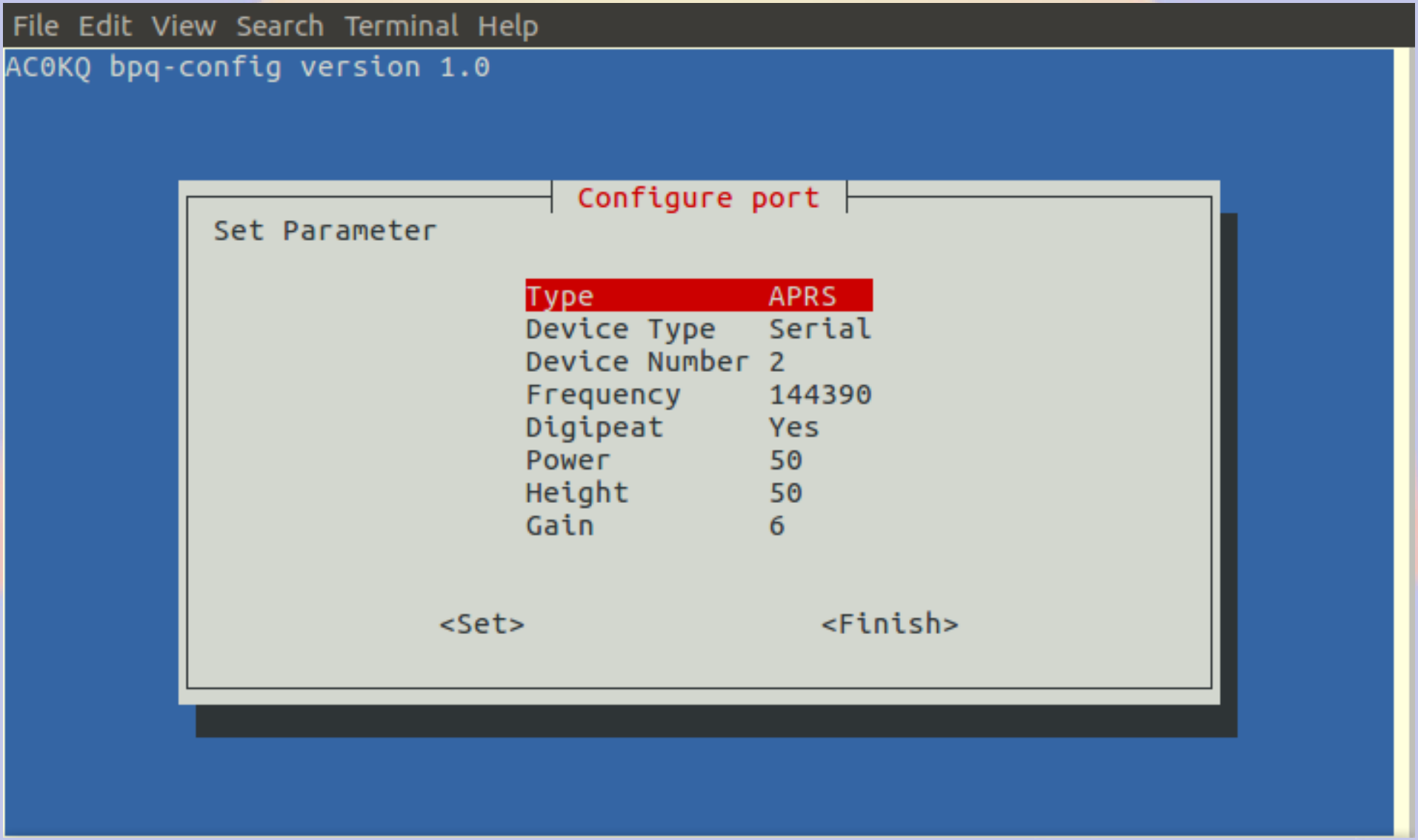
Port 1 is for BBS/RMS



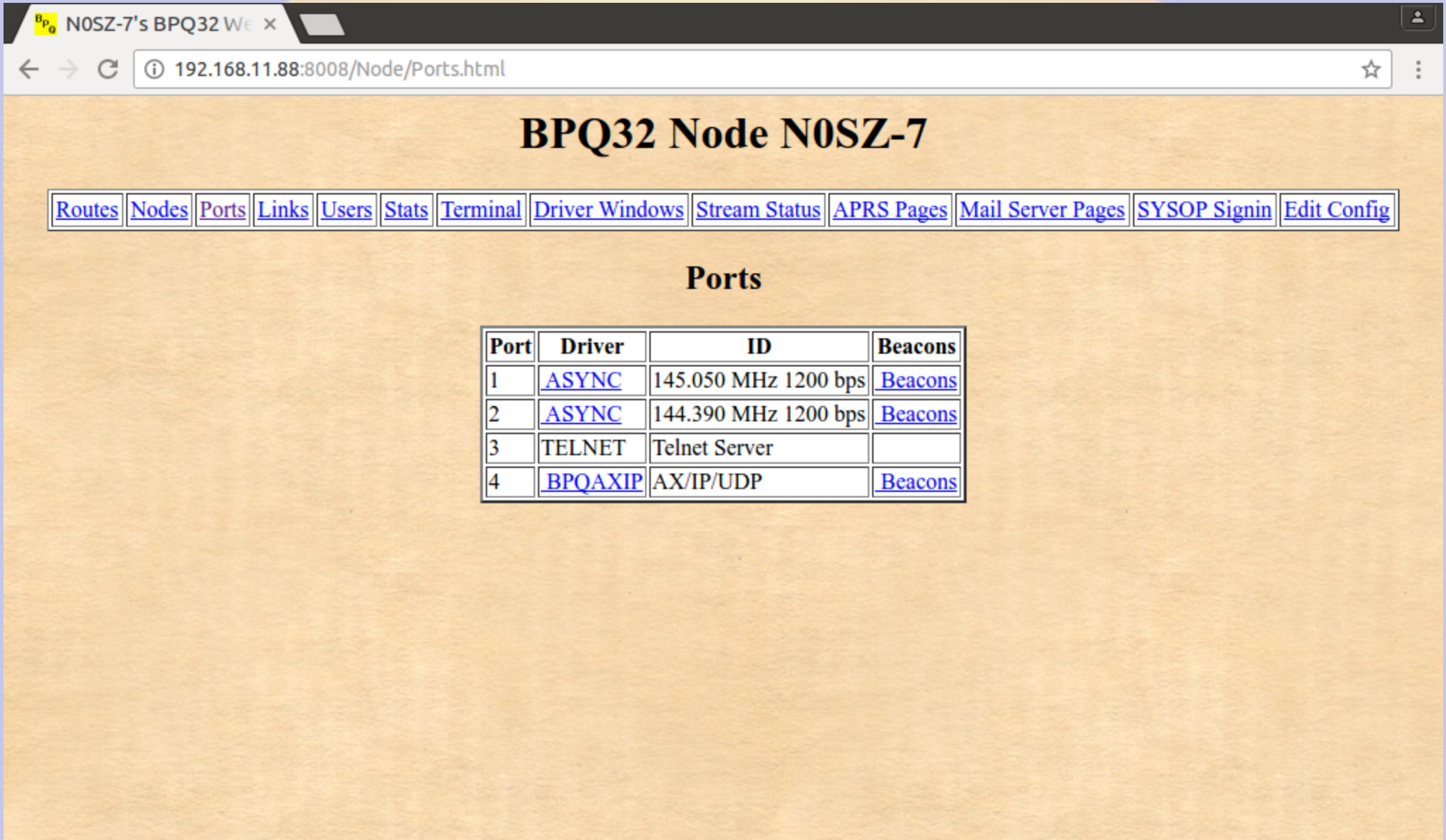
BPQ Port 2


File Edit View Search Terminal Help
AC0KQ bpq-config version 1.0
Port Configuration
Select port to
1 Add port
2 Delete port
3 Port 1
4 Port 2
<Select>
<Finish>

Port 2 is for APRS



BPQ Ports Page



About bpq-config

- ***bpq-config*** **is designed to get you started**
  - **It covers most installations, but not all**
- **It keeps its on configuration file .bpqconfig**
  - **Easier to parse**
  - **Hand edits are lost when using bpq-config**
  - **Version 1.1 *may* parse bpq32.cfg instead**
- **This is new software**
  - **Bug reports and improvements are welcome**
  - **Patches are even more welcome**

Part 3
AllStarLink Repeater

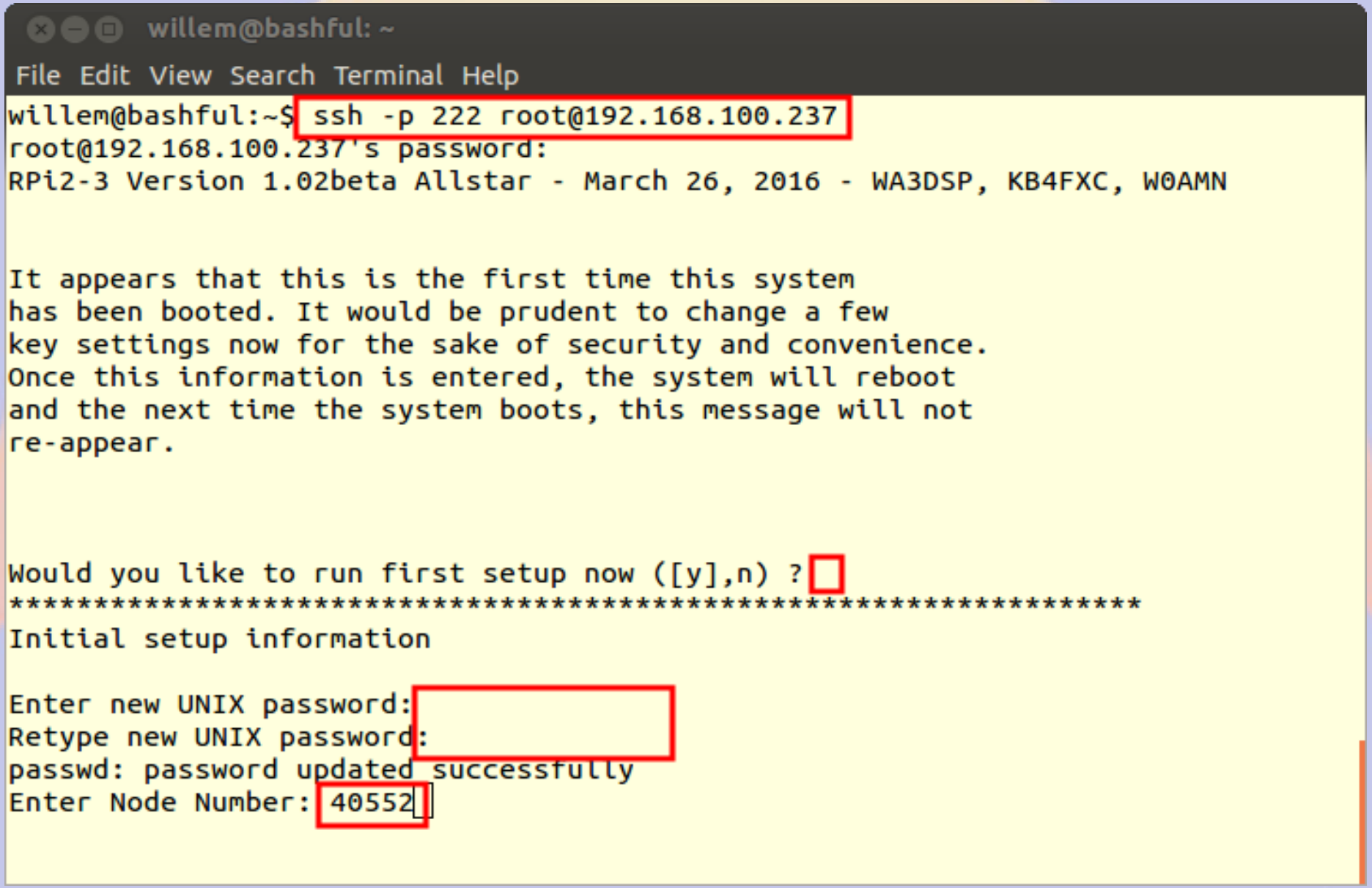
What is AllStarLink?

- Asterisk VOIP software for radio
- Interfaces with radio via URIx
  - CM119 USB audio chip
  - DB25 connector
- Can roll your own with equivalent fob

Installing AllStarLink

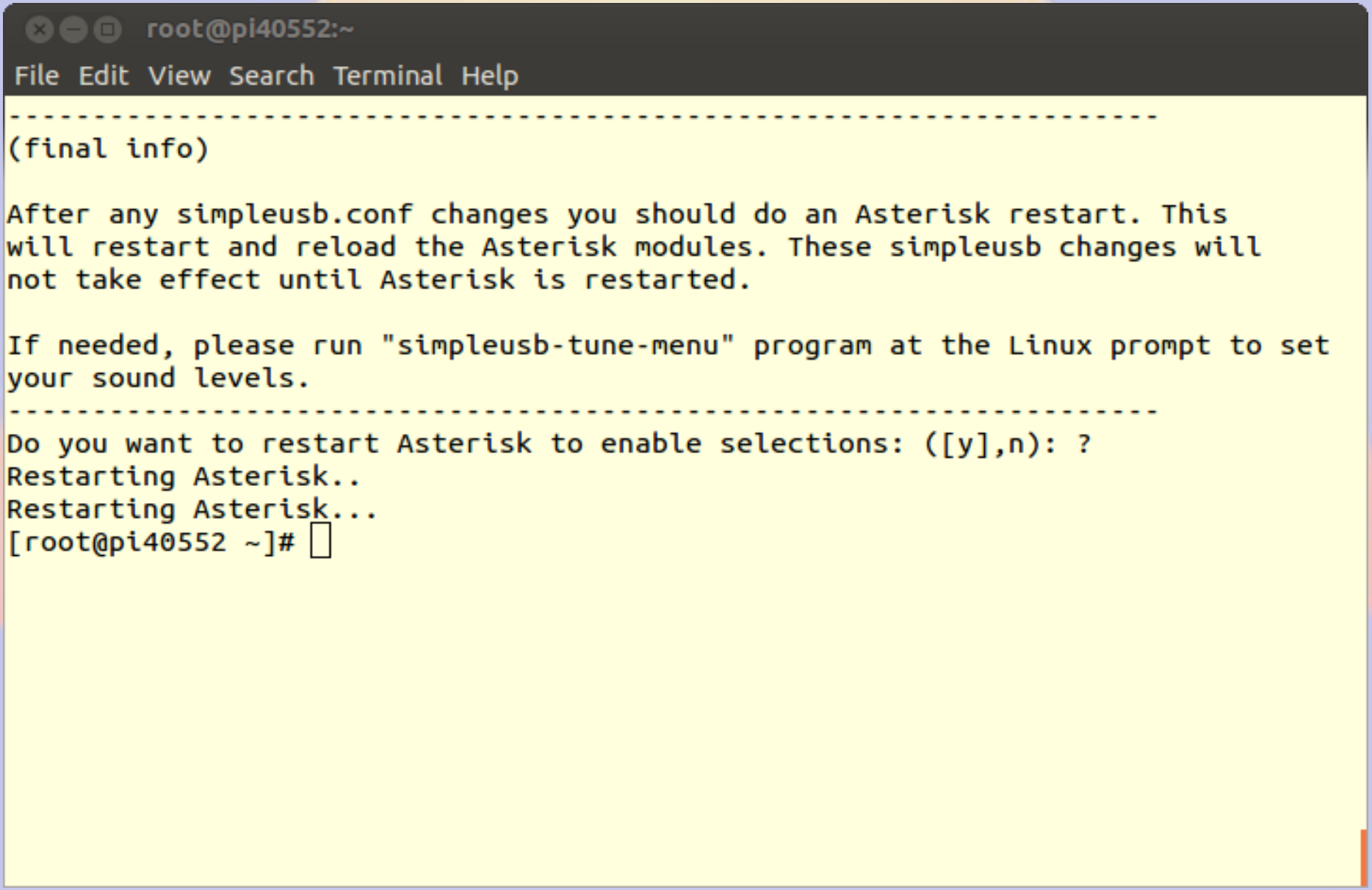
- **Download from** *www.hamviop.com*
  - Burn image to SD card
- Program your radio/repeater
  - Set radio to encode/decode CTCSS
  - On Motorola set accessories to output COS & PL on pin 8
- Tune a receiver to to the radio frequency
- Power up rPi
  - Allison will announce the IP address

Initial Login (password is root)

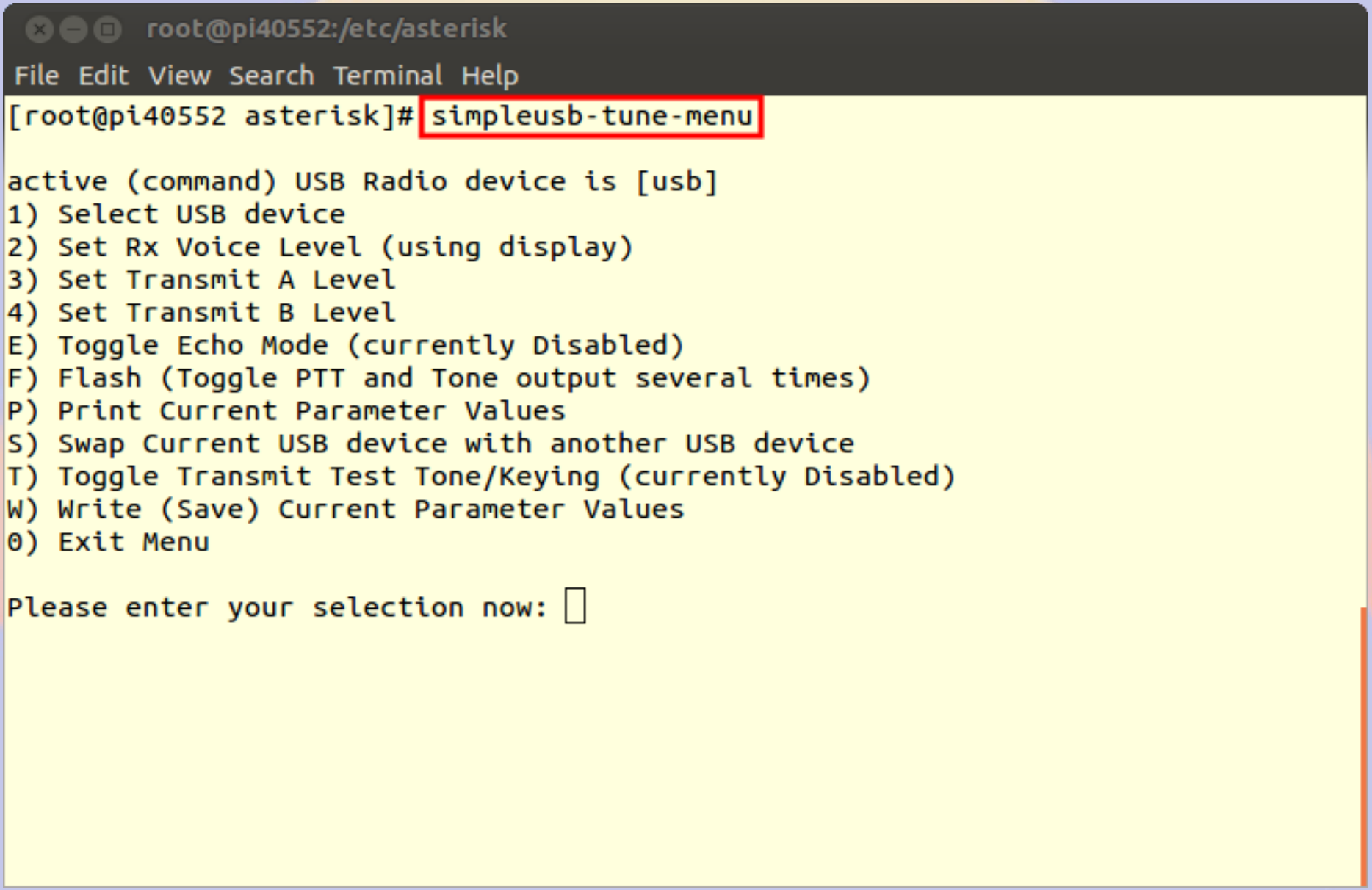


Skipped a dozen slides of options

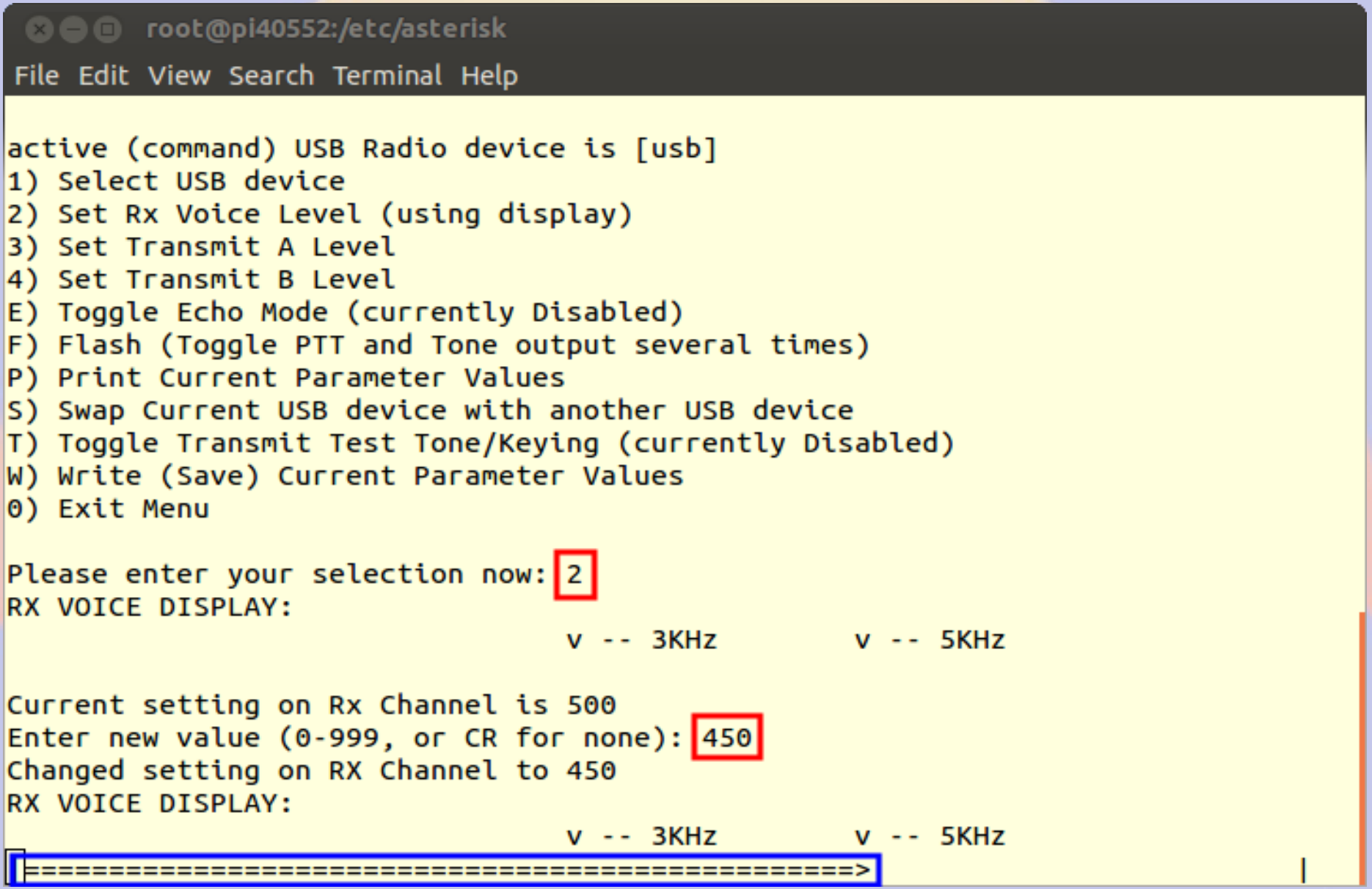
Done


root@pi40552:~
File Edit View Search Terminal Help
(final info)
After any simpleusb.conf changes you should do an Asterisk restart. This
will restart and reload the Asterisk modules. These simpleusb changes will
not take effect until Asterisk is restarted.
If needed, please run "simpleusb-tune-menu" program at the Linux prompt to set
your sound levels.
Do you want to restart Asterisk to enable selections: ([y],n): ?
Restarting Asterisk..
Restarting Asterisk...
[root@pi40552 ~]#

Set Levels


root@pi40552:/etc/asterisk
File Edit View Search Terminal Help
[root@pi40552 asterisk]# simpleusb-tune-menu
active (command) USB Radio device is [usb]
1) Select USB device
2) Set Rx Voice Level (using display)
3) Set Transmit A Level
4) Set Transmit B Level
E) Toggle Echo Mode (currently Disabled)
F) Flash (Toggle PTT and Tone output several times)
P) Print Current Parameter Values
S) Swap Current USB device with another USB device
T) Toggle Transmit Test Tone/Keying (currently Disabled)
W) Write (Save) Current Parameter Values
0) Exit Menu
Please enter your selection now:

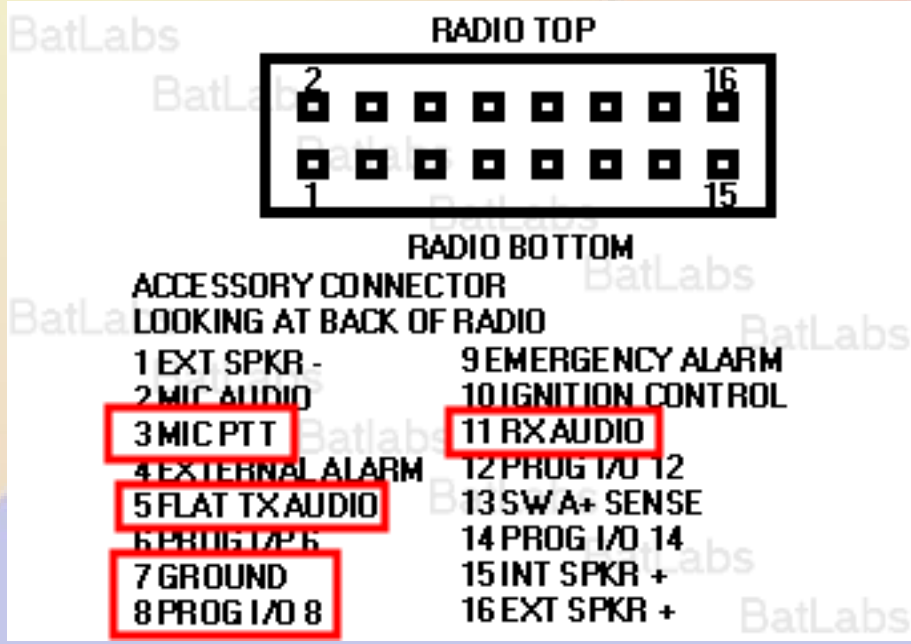
Set Receive Levels



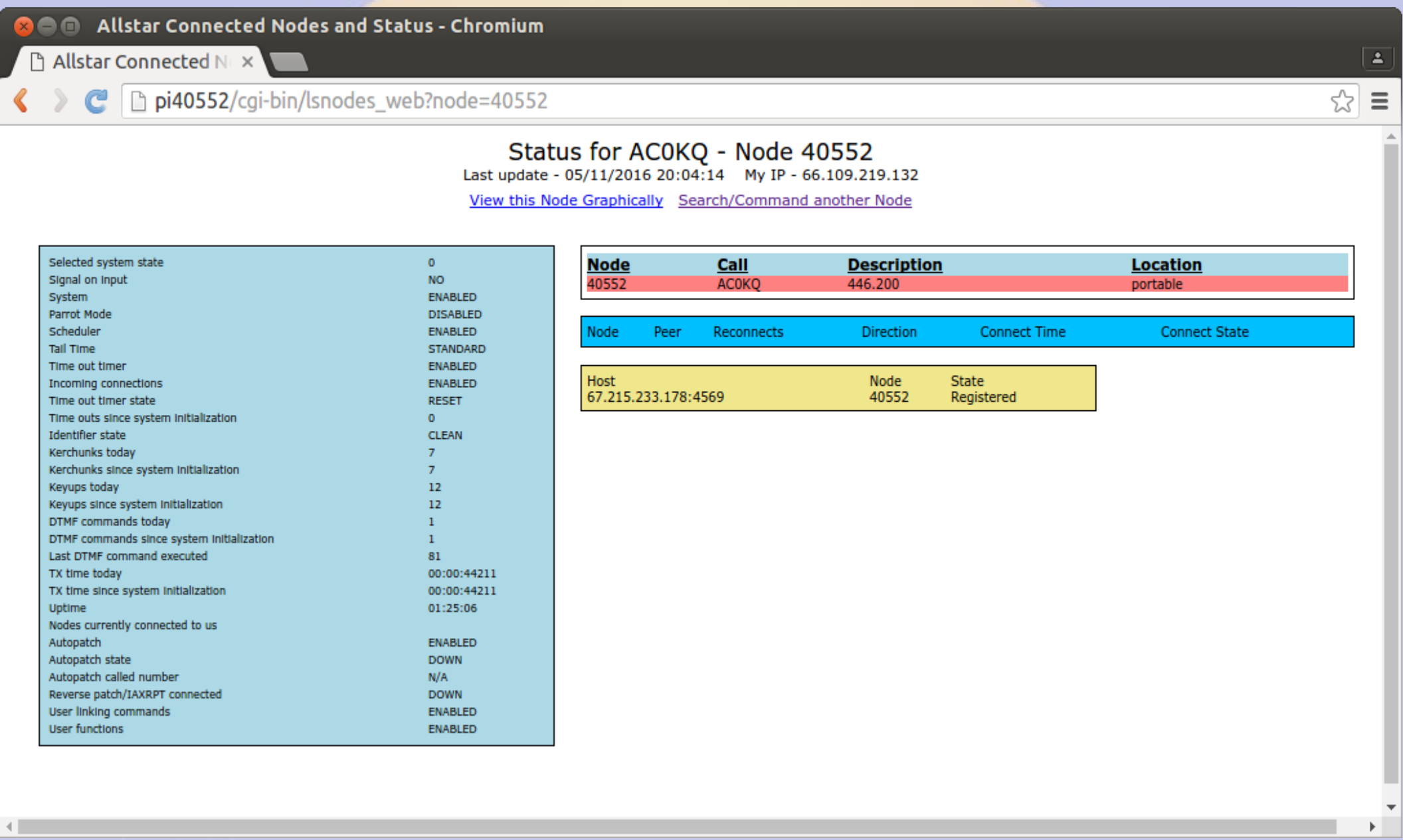
/etc/asterisk/simpleusb.conf

- **Configuration for Motorola SM50**

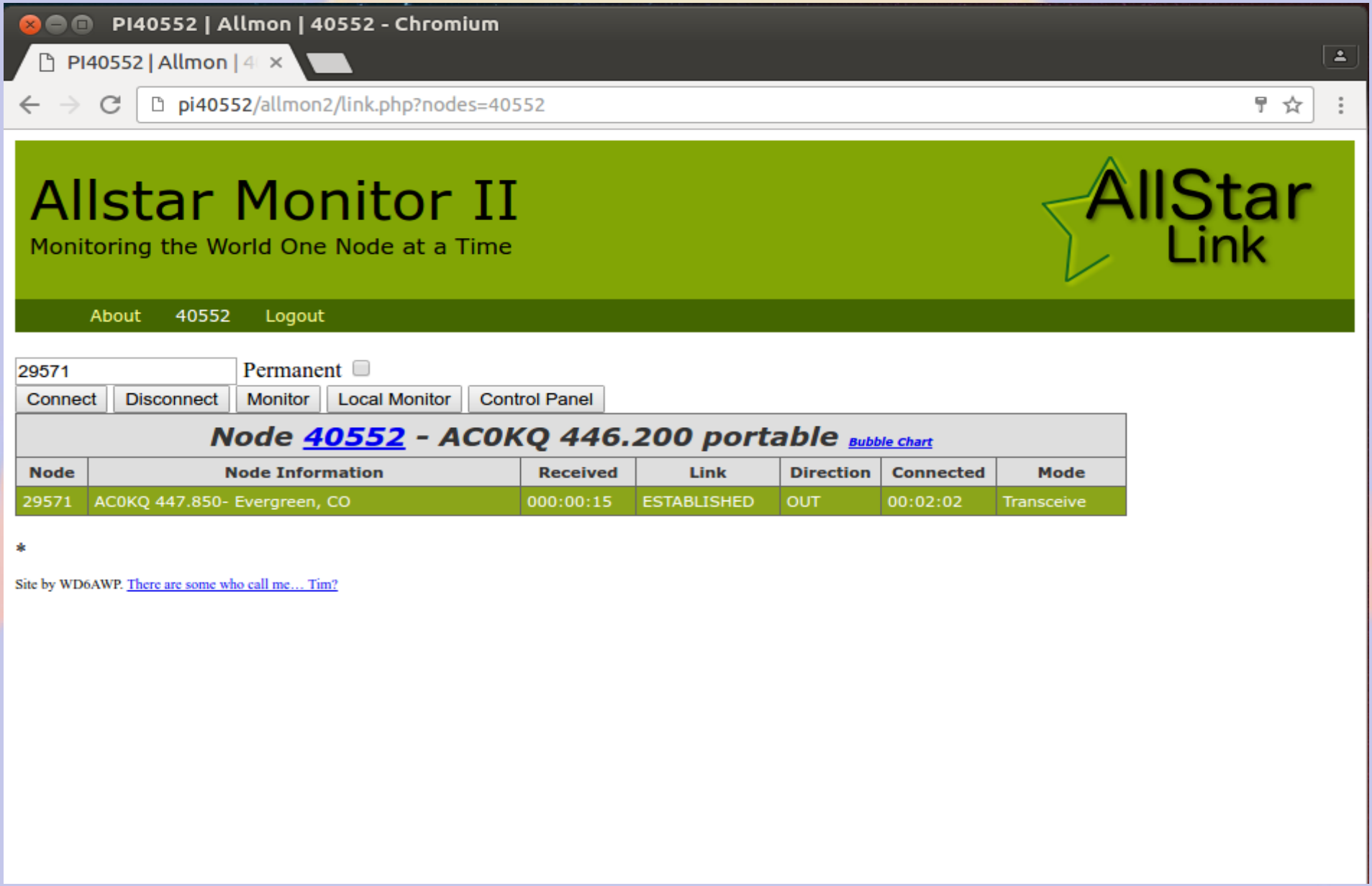
```
[usb]
eeprom=0
hdwtype=0
rxboost=1
carrierfrom=usb
ctcssfrom=usb
txmixa=voice
txmixb=no
invertptt=0
duplex=0
plfilter=yes
deemphasis=no
preemphasis=yes
rxaudiodelay=0
```



lsnode

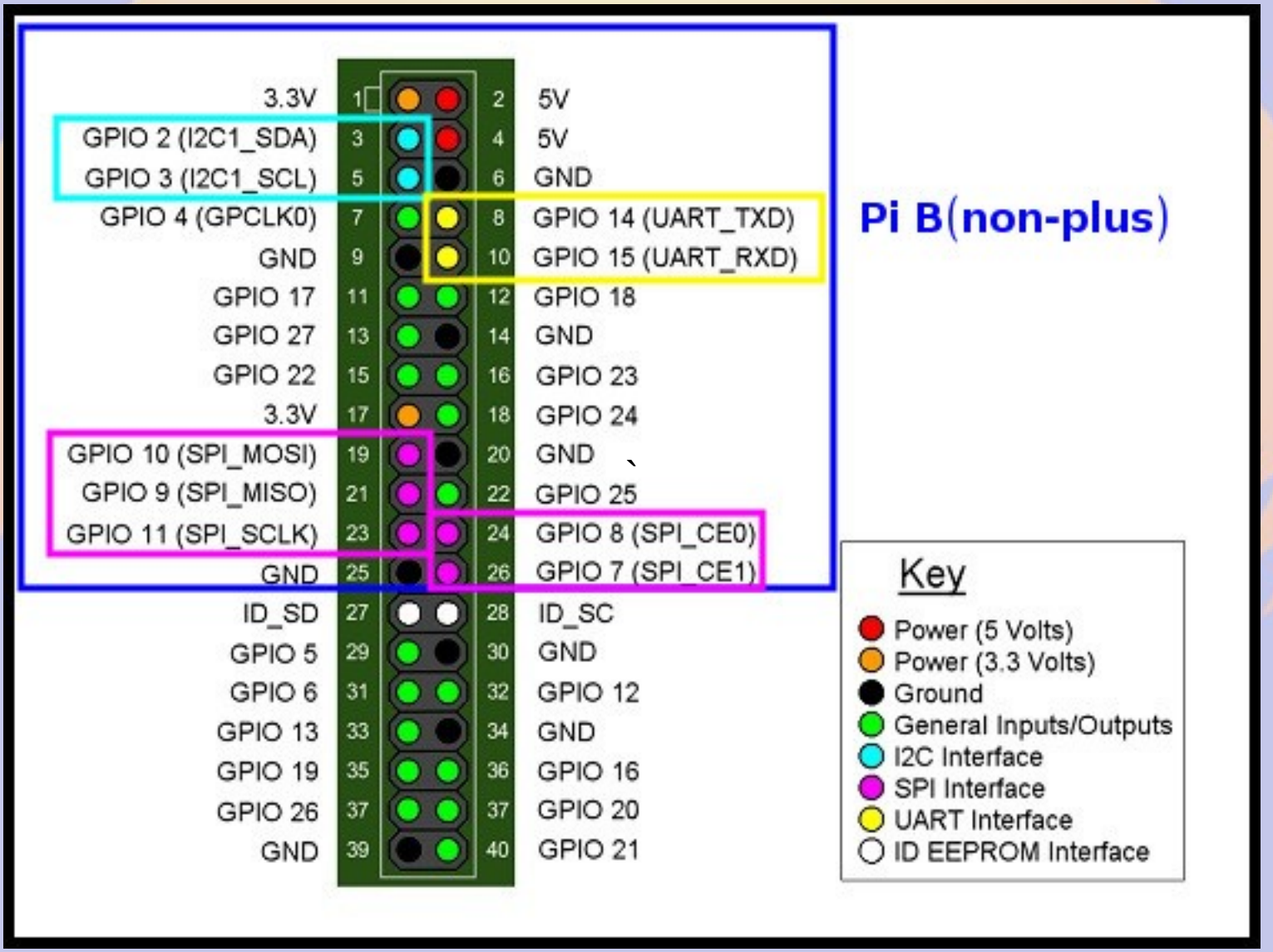

Allstar Connected Nodes and Status - Chromium
Allstar Connected N
pi40552/cgi-bin/lsnodes_web?node=40552
Status for AC0KQ - Node 40552
Last update - 05/11/2016 20:04:14 My IP - 66.109.219.132
View this Node Graphically
Search/Command another Node
Selected system state 0
Signal on input NO
System ENABLED
Parrot Mode DISABLED
Scheduler ENABLED
Tail Time STANDARD
Time out timer ENABLED
Incoming connections ENABLED
Time out timer state RESET
Time outs since system initialization 0
Identifier state CLEAN
Kerchunks today 7
Kerchunks since system initialization 7
Keyups today 12
Keyups since system initialization 12
DTMF commands today 1
DTMF commands since system initialization 1
Last DTMF command executed 81
TX time today 00:00:44211
TX time since system initialization 00:00:44211
Uptime 01:25:06
Nodes currently connected to us
Autopatch ENABLED
Autopatch state DOWN
Autopatch called number N/A
Reverse patch/IAXRPT connected DOWN
User linking commands ENABLED
User functions ENABLED
Node Call Description Location
40552 AC0KQ 446.200 portable
Node Peer Reconnects Direction Connect Time Connect State
Host Node State
67.215.233.178:4569 40552 Registered

Incoming Audio



Part 4
Control and Monitoring

Raspberry Pi Header



Pins are multiplexed

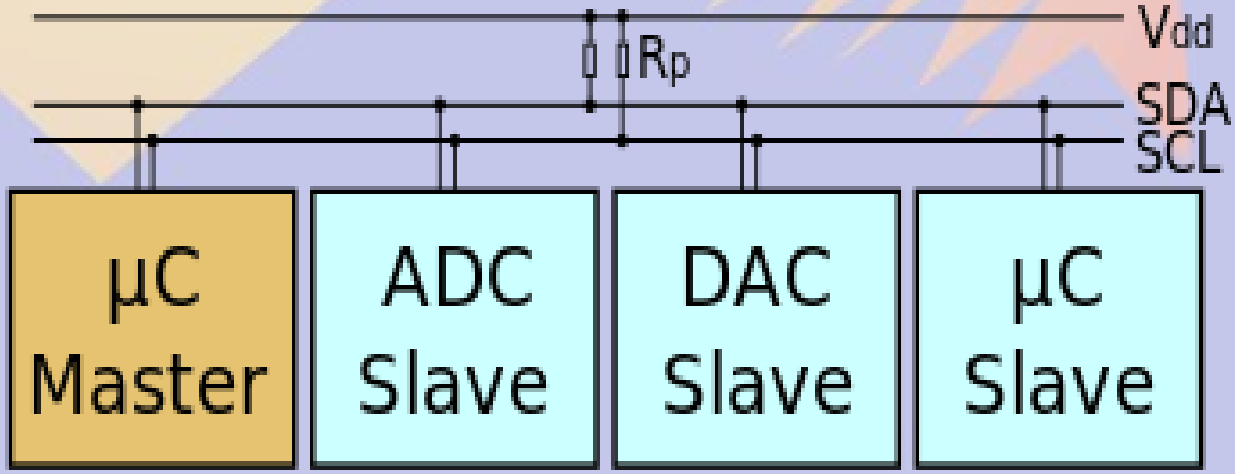
- Pins configured for different uses
- GPIO 14&15 <=> UART TxD/RxD
- GPIO 2&3 <=> I²C SDA&SCL
- GPIO 7&8&9&10&11 <=>
  SPI MOSI&MISO&SCL&CE0&CE1
- GPIO 18&19 <=> PWM 0&1
- 16-26 GPIO pins

Raspberry Pi Serial

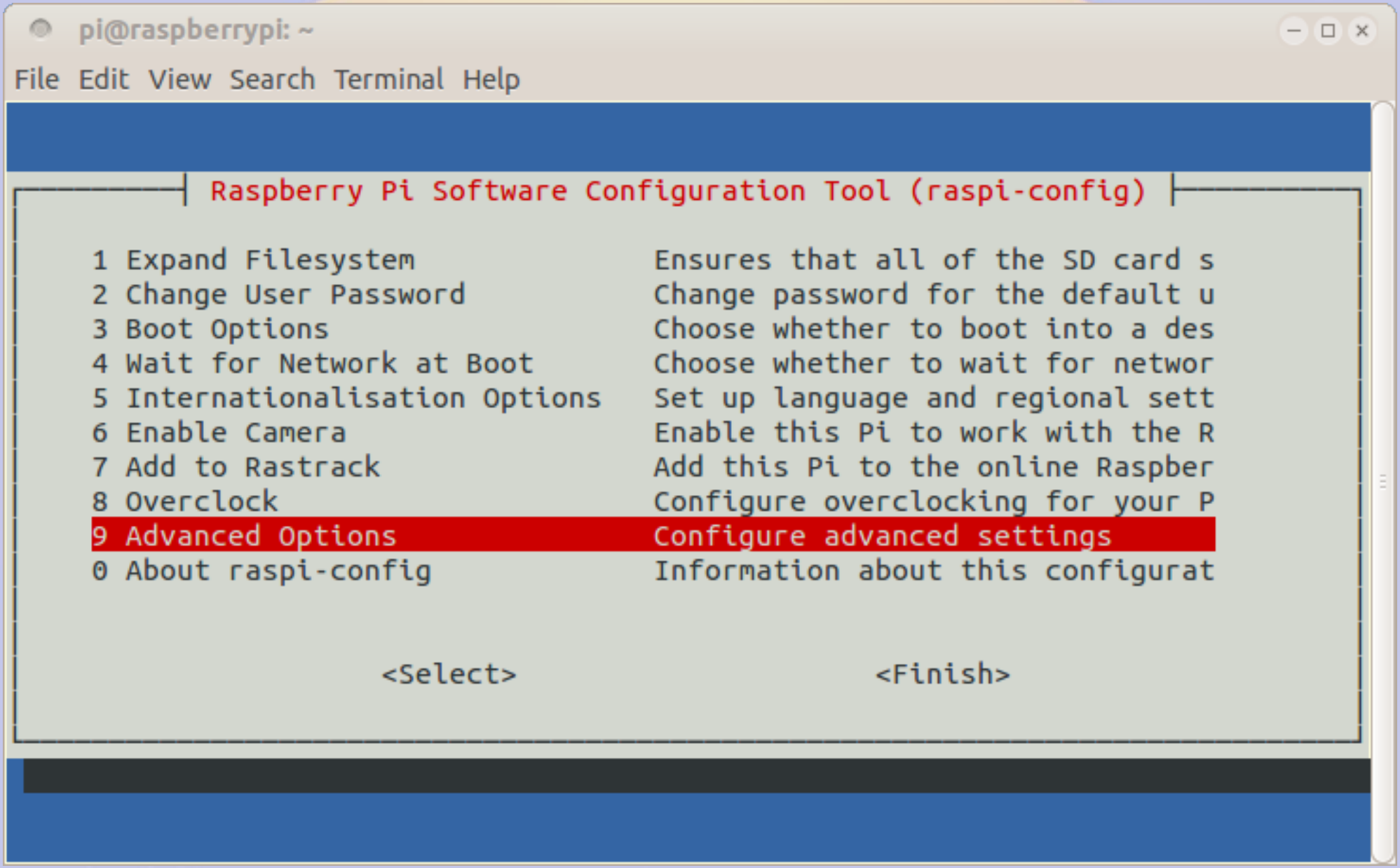
- Single serial port
  - /dev/ttyAMA0
- Speeds up to 115200 bps
- TTL level signals
- By default connected to getty

Raspberry Pi I^2C

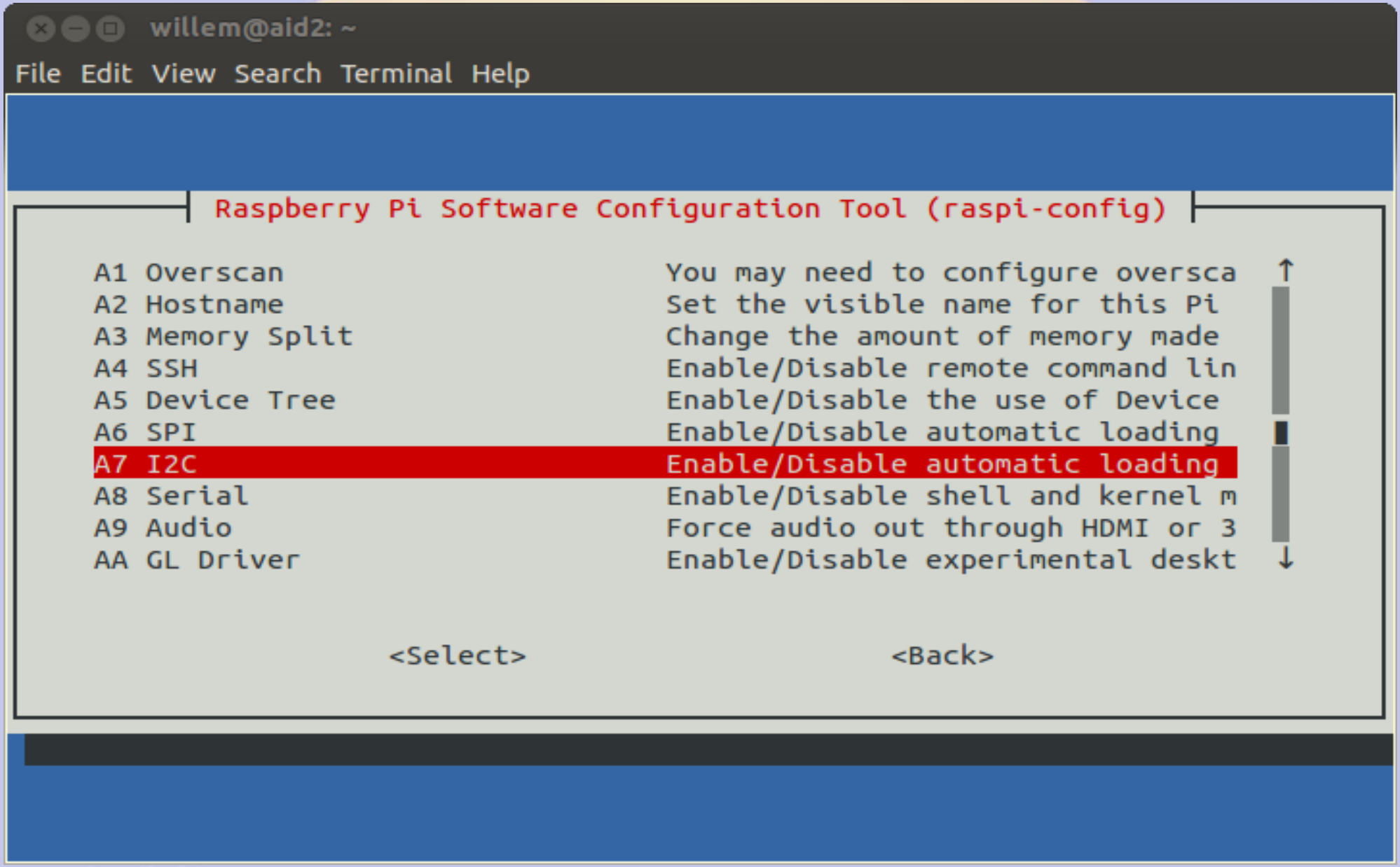
- **Inter-Integrated Circuit**
  - **Serial bus (a.k.a SMBus)**
- **Default speed 400,000 bps**
- **rPi has single external I^2C bus**
  - **127 devices**
- **Control lines**
  - **SDA (data)**
  - **SCL (clock)**



Enable I²C with raspi-config 1



Enable I²C with raspi-config 2

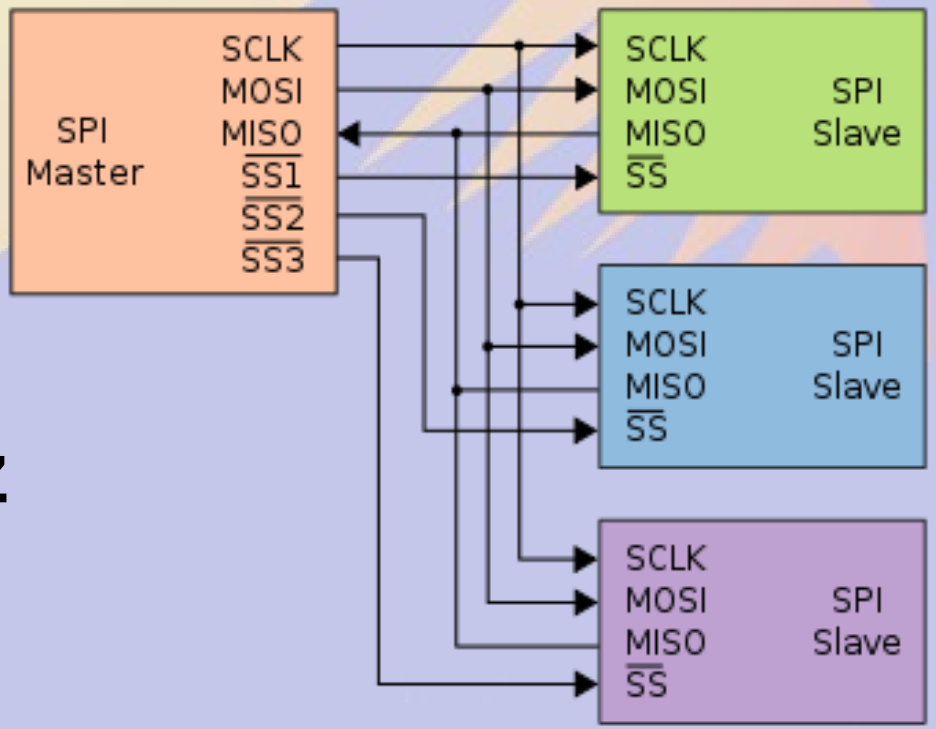


I^2C devices

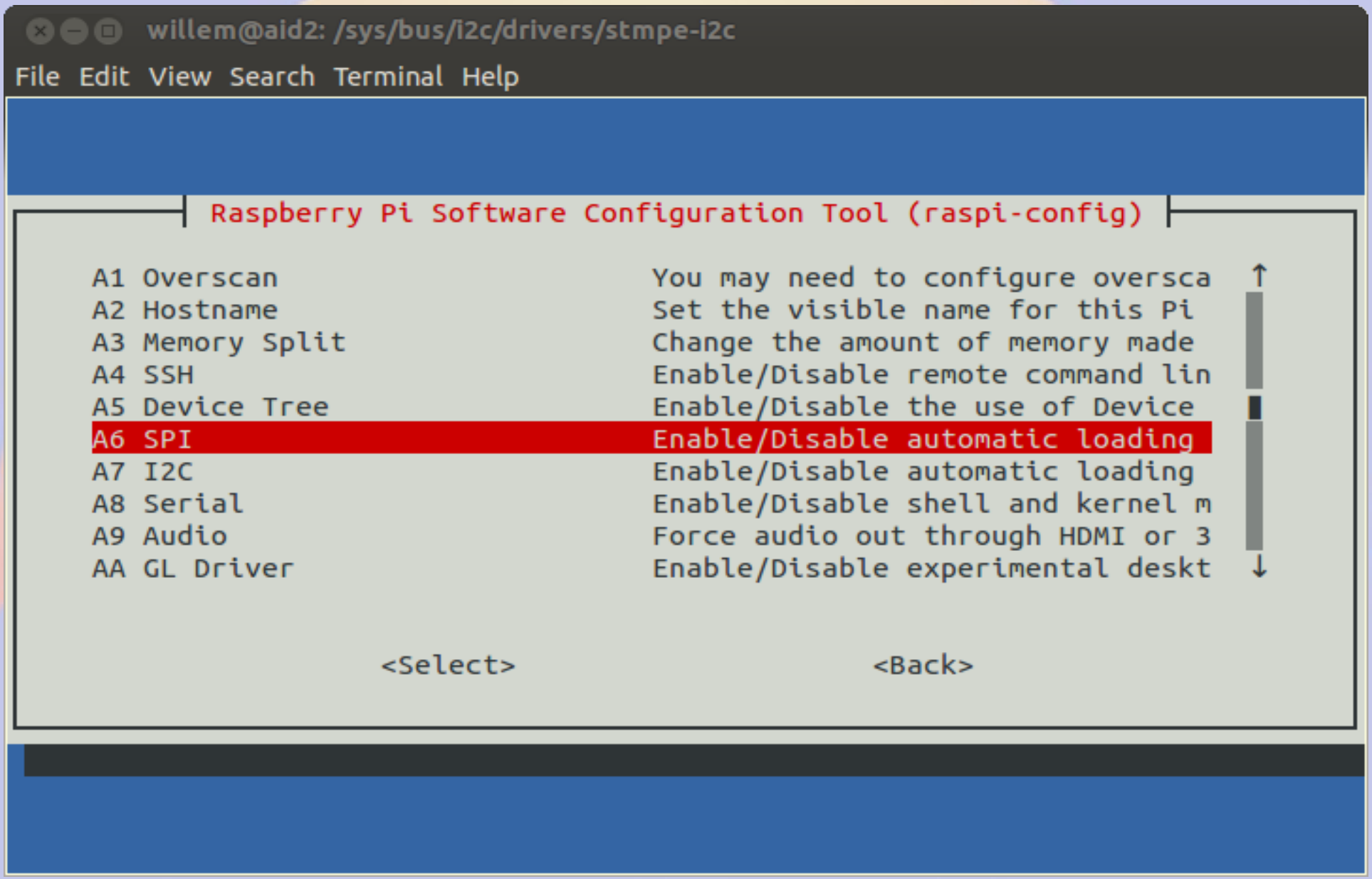
- TNC-Pi
- INA219 current sensor
- Temperature/pressure/RH sensors
- LCD displays
- Accelerometers
- Digitial I/O pins
- Analog<>Digital I/P pins

SPI bus

- Serial Peripheral Interface
- Signals (supports 2 slaves)
  - MasterOutSlaveIn
  - MasterInSlaveOut
  - Clock
  - CE0 (SS1)
  - CE1 (SS2)
- Speeds up to 250 MHz



raspi-config enable SPI



SPI Devices

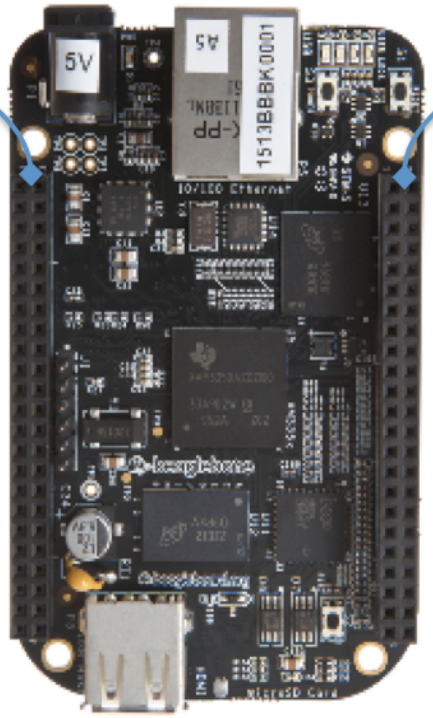
- Faster than I^2C, but uses more pins
- Same devices as I^2C, but adds
  - GPS
  - Ethernet/WiFi/Bluetooth/RFID
  - Memory
- Full duplex

Beagle Bone Black

Cape Expansion Headers

P9

DGND	1	2	DGND
VDD_3V3	3	4	VDD_3V3
VDD_5V	5	6	VDD_5V
SYS_5V	7	8	SYS_5V
PWR_BUT	9	10	SYS_RESETn
UART4_RXD	11	12	GPIO_60
UART4_TXD	13	14	EHRPWM1A
GPIO_48	15	16	EHRPWM1B
SPI0_CS0	17	18	SPI0_D1
I2C2_SCL	19	20	I2C2_SDA
SPI0_D0	21	22	SPI0_SCLK
GPIO_49	23	24	UART1_TXD
GPIO_117	25	26	UART1_RXD
GPIO_115	27	28	SPI1_CS0
SPI1_D0	29	30	GPIO_112
SPI1_SCLK	31	32	VDD_ADC
AIN4	33	34	GNDA_ADC
AIN6	35	36	AIN5
AIN2	37	38	AIN3
AIN0	39	40	AIN1
GPIO_20	41	42	ECAPPWM0
DGND	43	44	DGND
DGND	45	46	DGND



LEGEND
Power/Ground/Reset
Available Digital
Available PWM
Shared I2C Bus
Reconfigurable Digital
Analog Inputs (1.8V)

P8

DGND	1	2	DGND
MMC1_DAT6	3	4	MMC1_DAT7
MMC1_DAT2	5	6	MMC1_DAT3
GPIO_66	7	8	GPIO_67
GPIO_69	9	10	GPIO_68
GPIO_45	11	12	GPIO_44
EHRPWM2B	13	14	GPIO_26
GPIO_47	15	16	GPIO_46
GPIO_27	17	18	GPIO_65
EHRPWM2A	19	20	MMC1_CMD
MMC1_CLK	21	22	MMC1_DAT5
MMC1_DAT4	23	24	MMC1_DAT1
MMC1_DAT0	25	26	GPIO_61
LCD_VSYNC	27	28	LCD_PCLK
LCD_HSYNC	29	30	LCD_AC_BIAS
LCD_DATA14	31	32	LCD_DATA15
LCD_DATA13	33	34	LCD_DATA11
LCD_DATA12	35	36	LCD_DATA10
LCD_DATA8	37	38	LCD_DATA9
LCD_DATA6	39	40	LCD_DATA7
LCD_DATA4	41	42	LCD_DATA5
LCD_DATA2	43	44	LCD_DATA3
LCD_DATA0	45	46	LCD_DATA1

Pins are multiplexed

- **Default configuration**
  - **Power&Reset Buttons**
  - **4 serial ports**
  - **8 analog inputs (1.8V max)**
  - **1 external I^2C bus (127 devices)**
  - **19-128 GPIO pins**
  - **Switched 5V/3.3V DC**

Limitations

- **Pins connect directly to CPU**
  - **Long wires are CPU antennas!**
- **rPi & BBB GPIO Pins are 3.3 V**
  - **Max current 16 mA in or out**
  - **Max combined output current 50 mA**
- **BB Analog In Pins are 1.8V**

Device Tree

- **Unix: Everything is a File**
- **/sys maps to hardware**
  - **In kernel virtual file system**
- **Get status by reading**
- **Set status by writing**

Reading analog pins on BBB

- **Enable analog pins in device tree**

  echo cape-bone-iio>/sys/devices/bone_capemgr.*/slots

- **Read value of pin AIN0 in mV**

  cat /sys/devices/ocp.*/helper.*/AIN0

  `580`

- **Voltage on pin AIN0 is 0.580V**

Show pin voltages in Python 1

```
#!/usr/bin/python

for i in range(0,8):
   #  Snarf file
   fd = open("/sys/devices/ocp.3/helper.16/AIN%d" % i)
   text = fd.read()
   fd.close()
   #  Decode voltage
   V = float(text)/1000
   #  Print voltage
   print "AIN%d = %5.3fV" % (i,V)
```

Show pin voltages in Python 2

./aread

```
AIN0 = 1.740V
AIN1 = 1.481V
AIN2 = 1.645V
AIN3 = 0.867V
AIN4 = 0.589V
AIN5 = 0.709V
AIN6 = 0.852V
AIN7 = 1.678V
```

Limitations

- **Maximum voltage is 1.8V**
- Use a voltage divider to increase
  - Use 1% or better resistors
  - Max 1 kohm for lower leg
- No analog in on rPi
  - use MCP3008 or similar and SPI

Assigning pins to GPIO

- **/sys/class/gpio/export**
  - **Maps pin to GPIO**
  - **echo 18 > /sys/class/gpio/export**
- **/sys/class/gpio/unexport**
  - **Removes pin from GPIO map**
  - **echo 18 > /sys/class/gpio/unexport**
- Root access required

Manipulating GPIO

- **When mapped to GPIO, a new directory is created for that pin**
  - **/sys/class/gpio/gpioXX**
- Files in this directory controls pin
  - direction = in or out
  - value = 0 or 1

Checking pin value

- In or out?
  - **cat /sys/class/gpio/gpio18/direction**
- High or low?
  - **cat /sys/class/gpio/gpio18/value**

Changing the GPIO direction

- Set pin for input
  - echo in > /sys/class/gpio/gpio18/direction
- Set pin for output
  - echo out > /sys/class/gpio/gpio18/direction

Changing the GPIO value

- Set pin voltage high
  - echo 1 > /sys/class/gpio/gpio18/value
- Set pin for output
  - echo 0 > /sys/class/gpio/gpio18/value

python access to pins

- **Import the GPIO package**

  import Rpi.GPIO as GPIO

- Name the pins by their GPIO#

  GPIO.setmode(GPIO.BCM)

- Name pins by their board number

  GPIO.setmode(GPIO.BOARD)

python set pins for in/out

- **Set pin 18 for output**

  GPIO.setup(18,GPIO.OUT)

- Set pins 18,23,24&25 for output

  GPIO.setup([18,23,24,25],GPIO.OUT)

- Set ping 18 for input

  GPIO.setup(18,GPIO.IN)

python set/get pin value

- **Set pin 18 high**

  GPIO.output(18,1)

- Set pin 18 low

  GPIO.output(18,0)

- Read ping 18 value

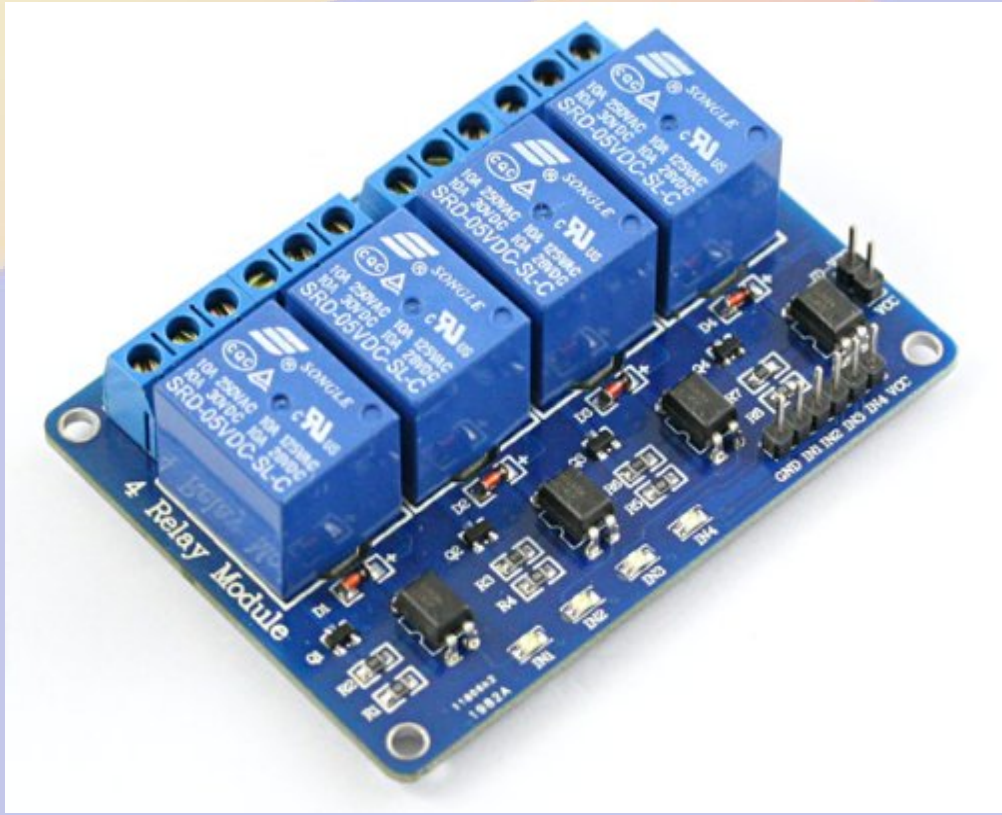
  p18 = GPIO.input(18)

Input pin status

- Set ping 23 to input with pull up
- GPIO.setup(24,GPIO.IN,pull_up_down=GPIO.PUD_UP)
  - ground to activate
- Set pin 24 to input with pull down
- GPIO.setup(24,GPIO.IN,pull_up_down=GPIO.PUD_DOWN)
  - Pull up to 3.3V
- A 1k series resistor is typically a good idea

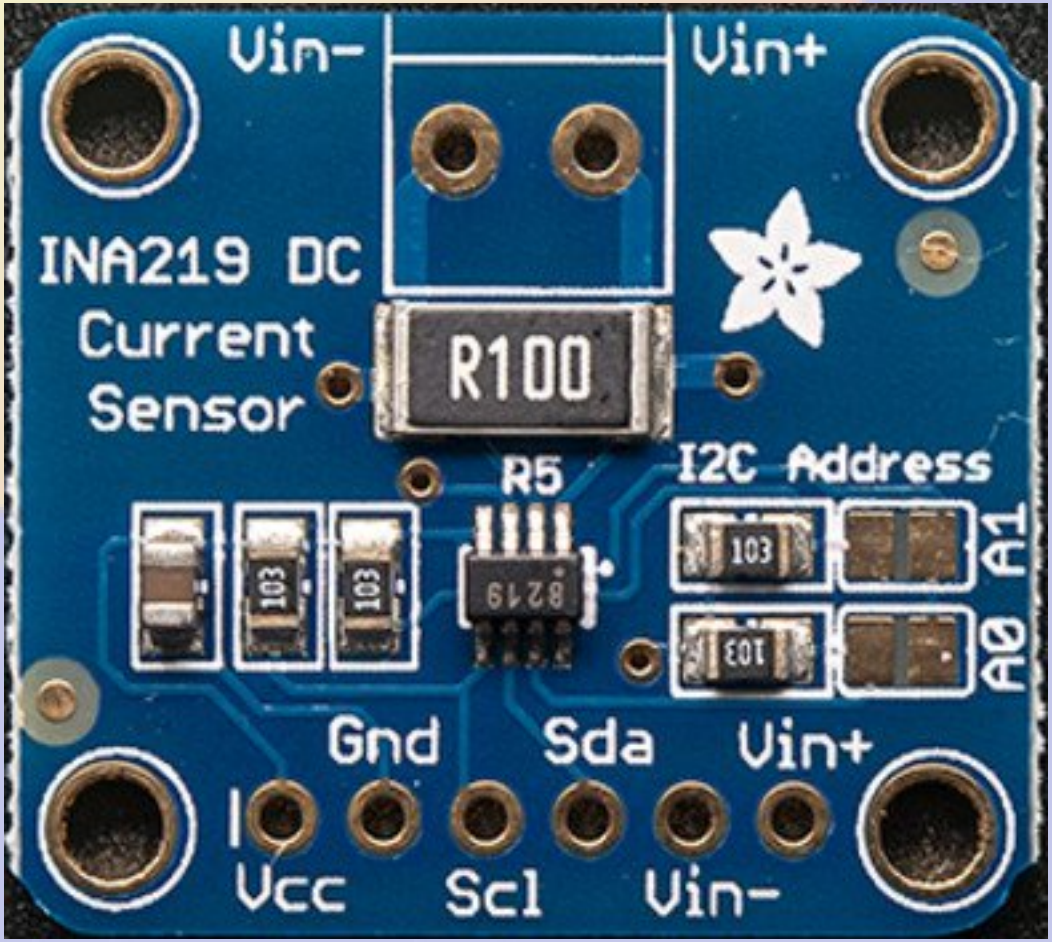
Important Limitations

- **GPIO pins are 3.3 V**
- **Current limited to 16mA**
- **Opto-isolate relays**



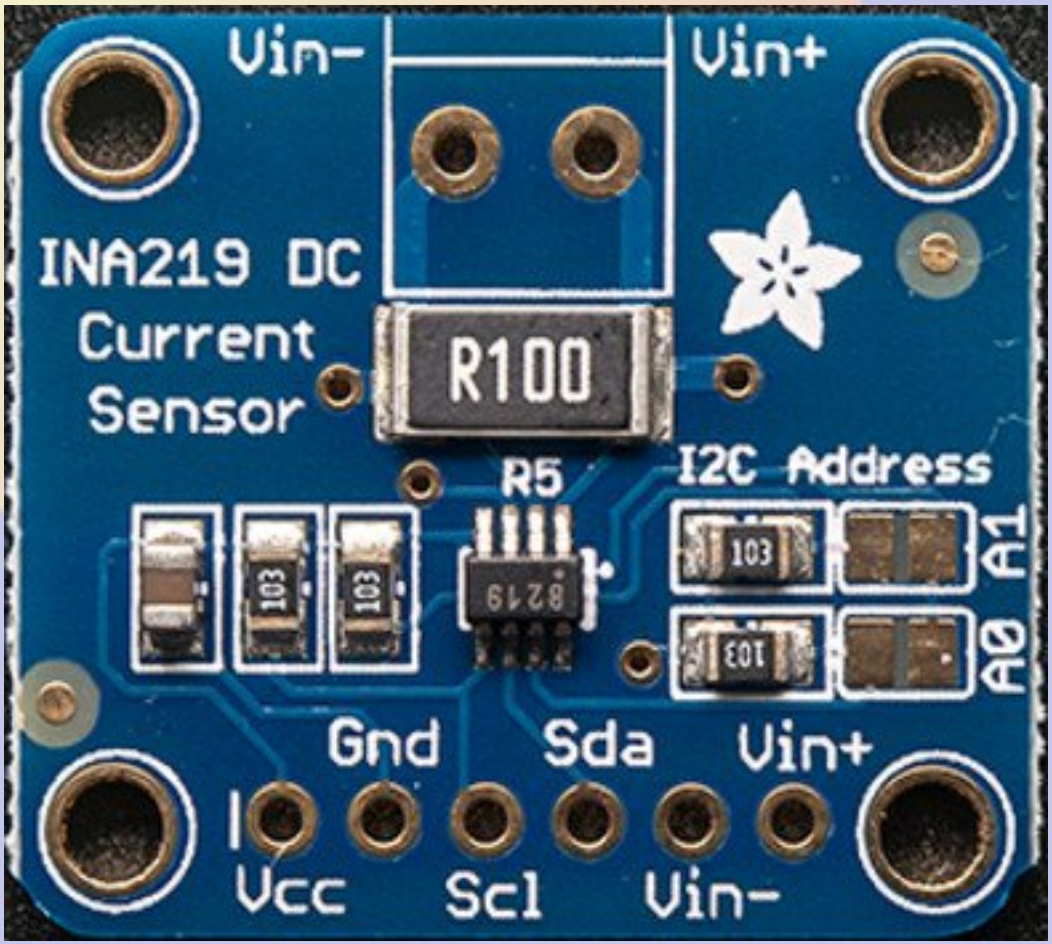
I²C Example: Voltage&Current

- TI INA219 I²C high side monitor
- Max 26V
- Current Sense 40-320mV shunt
- Chip $2.50
- Adafruit $10



Adafruit Breakout

- **I²C address 0x40 0x41 0x42 0x43**
  - **solder jumpers**
- **0.1 ohm shunt reads to 3.2A**

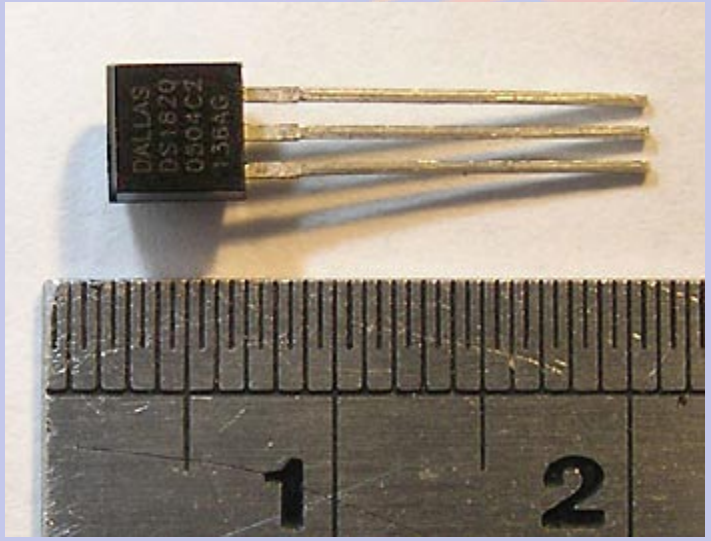


Python Usage

```
import Subfact_INA219 as INA219
ina = INA219()
V = ina.getBusVoltage_V()
mA =  ina.getCurrent_mA()
```

Reading 1wire Temperatures

- 1wire uses a single data bus
- Each device has unique address
- DS18S20 is a TO-92 temperature sensor with 0.5C resolution for $2.50
- Can use parasite power (but not on rPi)
  Use 4k7 pullup



Getting 1wire output

- **ls /sys/bus/w1/devices**

```
10-000802fba50d
10-000802fbe2f6
10-000802fbf0f9
w1_bus_master1
```

- **10 means it is a DS18S20 temp, the test is a unique serial number**

Getting the Data

cat /sys/bus/w1/devices/w1_bus_master1/w1_master_slaves

```
10-000802fbe2f6
10-000802fbf0f9
10-000802fba50d
```

cat /sys/bus/w1/devices/10-000802fbe2f6/w1_slave

```
2c 00 4b 46 ff ff 0e 10 17 : crc=17 YES
2c 00 4b 46 ff ff 0e 10 17 t=21875
```

Temperature of first sensor is 21.875 ^{o}C

Reading Temps in Python 1

```
#  Snarf the slave list  file
fd=open("/sys/bus/w1/devices/w1_bus_master1/w1_master_slaves")
text = fd.read()
fd.close()
#  Split text on line breaks
slaves = filter(None,text.split("\n"))
#  Sort so that order is predictable
slaves.sort()
```

Reading Temps in Python 2

```
#  Blank dictionary
temps = {}
#  Loop over devices
for slave in slaves:
      if slave=="": continue
      #  Snarf device file
      fd = open("/sys/bus/w1/devices/"+slave+"/w1_slave")
      text = fd.read()
      fd.close()
      #  Split lines
      lines = text.split("\n")
      words = lines[1].split(" ")
      #  Get temperature
      C = float(words[9][2:])/1000
      F = 9*C/5+32
     #  Add result to dictionary
      temps[slave] = "%.1fF" % F
```

Observations

- **Temperature conversion occurs when you cat the file**
  - **About 700mS per device**
- **Temperature reads are best done using a separate thread**
- **rPi 1wire support in *raspi-config***

Part 5
Software Defined Receiver

ADSB SDR Receiver

- **Receiver based on RTL2832 USB**
- **About $20 on Amazon**
- **Also used in many ham related SDR projects**

Software Build

Build and install rtl-sdr module and software

```
git clone git://git.osmocom.org/rtl-sdr.git
cd rtl-sdr
mkdir build
cd build
cmake ../ -DINSTALL_UDEV_RULES=ON
make
cd ..
```

/usr/local/bin/rtl_tcp is a TCP server for remote monitoring

dump1090 Build

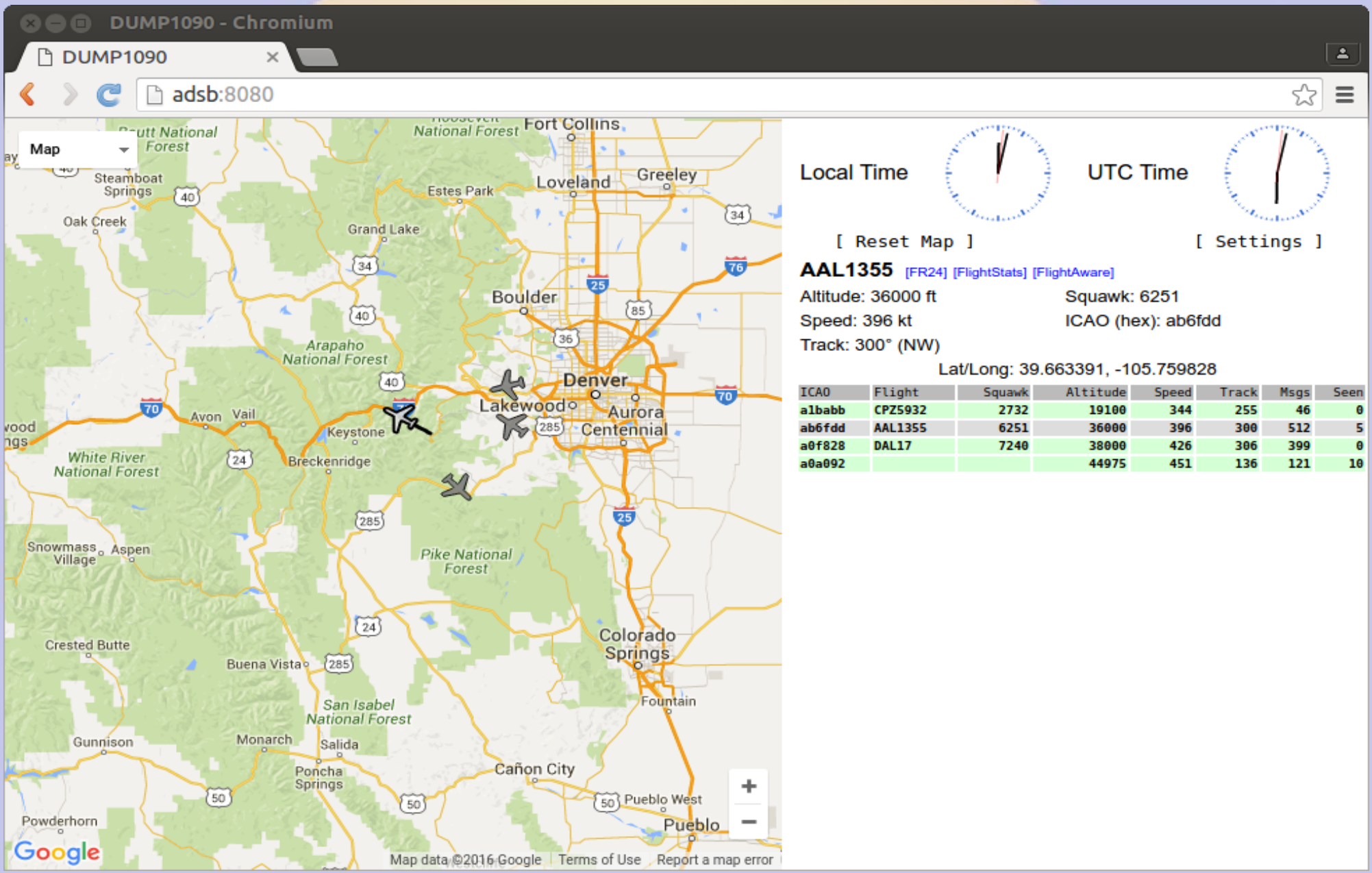
Build and install dump1090 and related software

```
git clone git://github.com/MalcolmRobb/dump1090.git
cd dump1090
make
cd ..
```

Running web interface

- **./dump1090 --net --lon -105 --lat 39**
  - **--net enables web interface port 8080**
  - **--lon and –lat sets location**
- **Run at boot from rc.local**

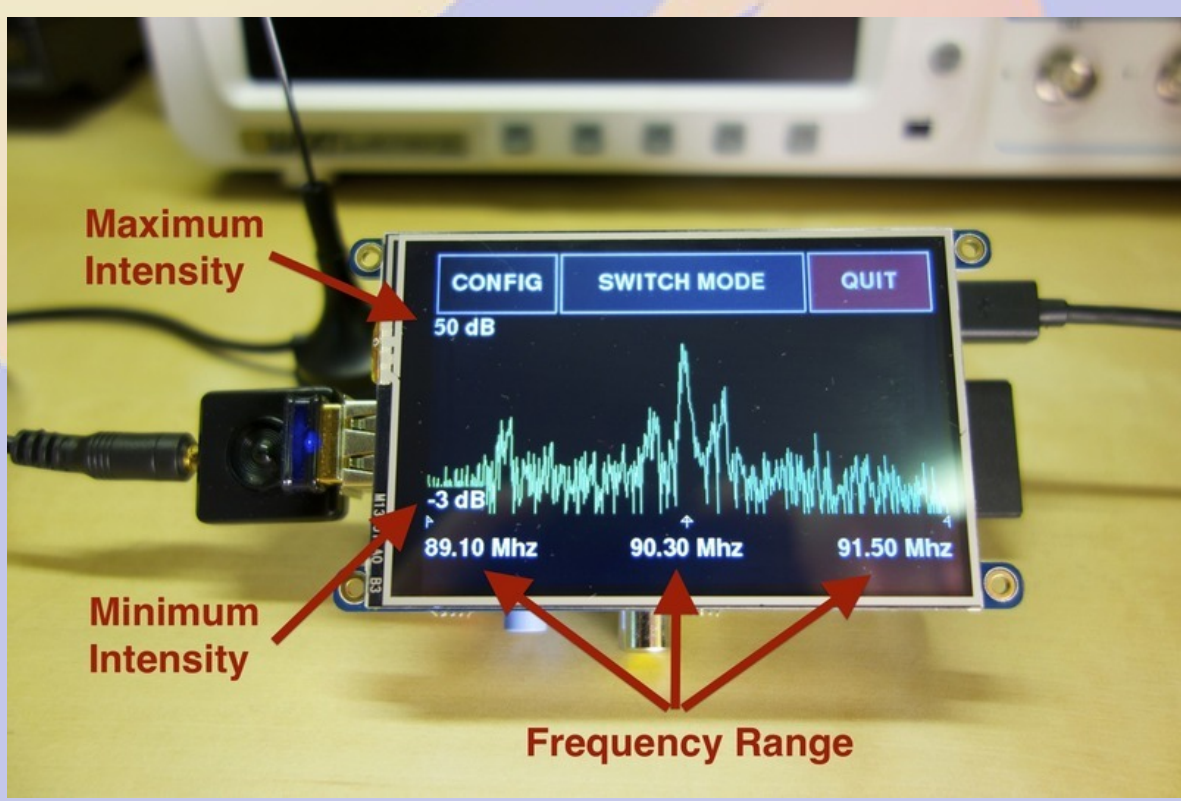
Running



ICAO	Flight	Squawk	Altitude	Speed	Track	Msgs	Seen
a1babb	CPZ5932	2732	19100	344	255	46	0
ab6fdd	AAL1355	6251	36000	396	300	512	5
a0f828	DAL17	7240	38000	426	306	399	0
a0a092			44975	451	136	121	10

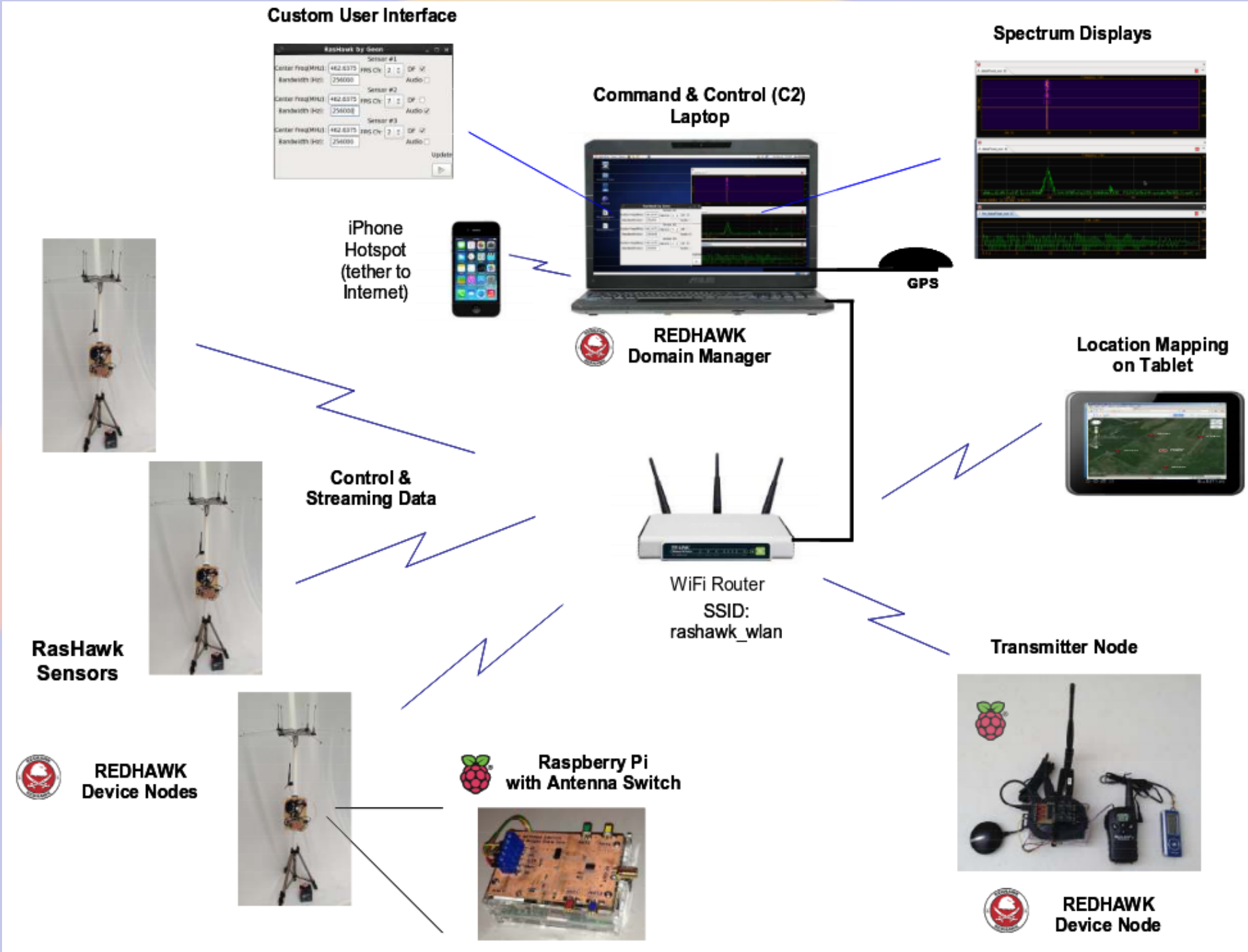
Other SDR Projects

- The rPi 3 is a 1.2GHz 64 bit quad core machine with 1GB memory
  - Processing power to do cool stuff
- Adafruit Freq Show
- GNU radio



Eric Schneider
RMHAM U
April 15, 2017

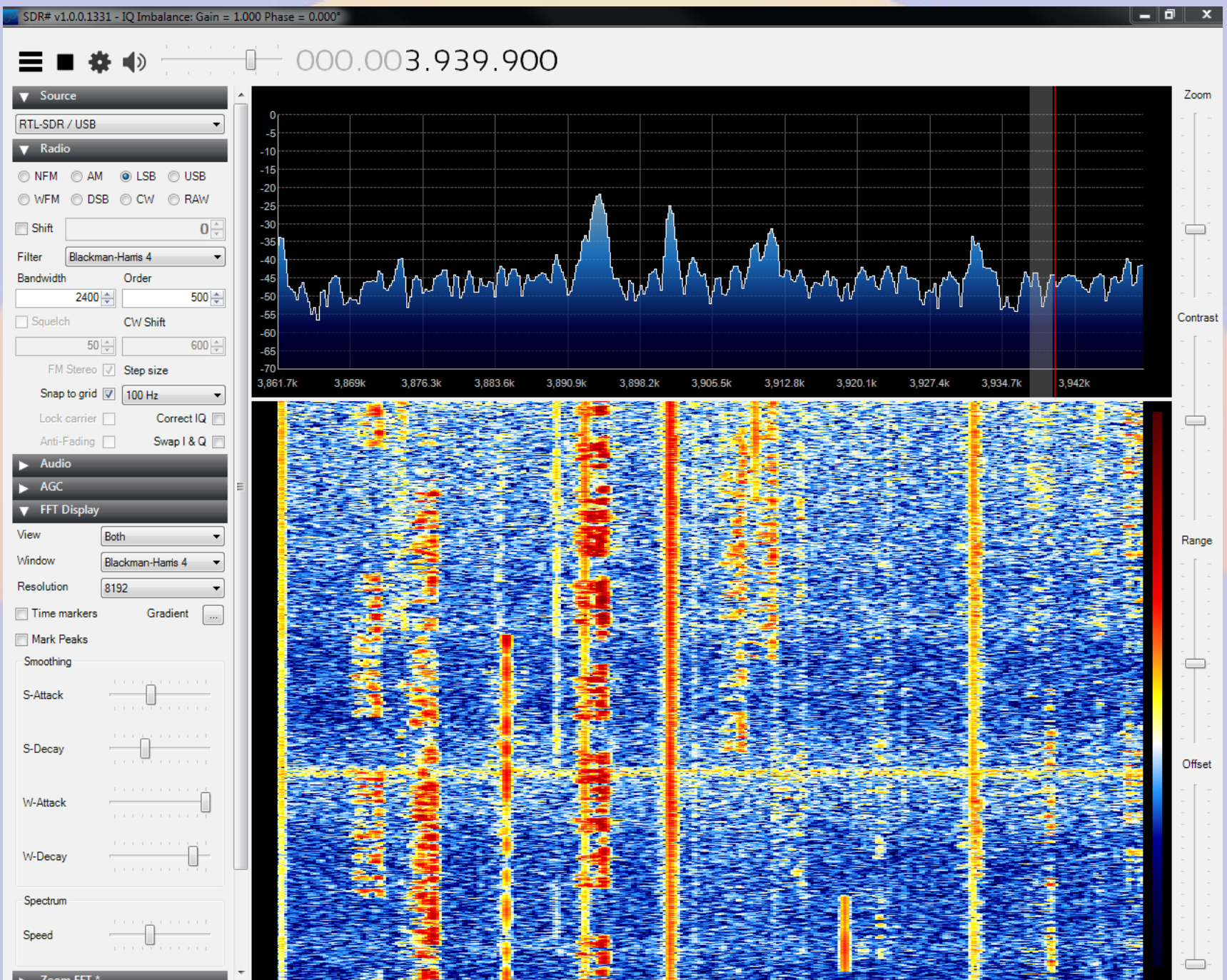
RasHAWK


Custom User Interface
Spectrum Displays
Command & Control (C2) Laptop
iPhone Hotspot (tether to Internet)
GPS
REDHAWK Domain Manager
Location Mapping on Tablet
Control & Streaming Data
WiFi Router
SSID: rashawk_wlan
RasHawk Sensors
Transmitter Node
REDHAWK Device Nodes
Raspberry Pi with Antenna Switch
REDHAWK Device Node

SDR TCP server

- **Start *rtl_tcp* as root**
  - **rtl_tcp -a <ipaddress>**
  - **Default port is 1234 (set with -p)**
- **Connect to it with an SDR program such as SDR# or GNU Radio on a device with enough power to process the data**

SDR# Screenshot



Part 6
Other Projects

rPi / TNC-Pi / screen / xastir

SmokePi (SmokePing rPi)