NomadicBTS Lab Kit 

INSTALLATION MANUAL

**EMMANUEL ADETIBA, R.ENGR. (COREN), Ph.D**

**VICTOR AKANDE**

**TABLE OF CONTENTS**

PREFACE

[INTRODUCTION](#_op47z6p8opf)

[OpenBTS](#_fn2y2e26wm8t)

[Requirements](#_yphwh57hddl4)

[Installation](#_vg1tvo83jpp4)

[OpenBTS Installation](#_k08206pheube)

[GUI INSTALLATION](#_wyy9cp46esmq)

[REQUIREMENTS](#_falin9z0zl6y)

[USRP DRIVERS](#_j0x0aab4k380)

[TARGET\_TERM](#_gjvsjpv0z0l5)

[ENABLE ADMIN PRIVILEGES](#_zap0uq6odo93)

# 

# 

# 

# 

# 

# 

# 

# 

# 

# 

# 

# 

# 

# 

# 

# 

# 

# 

# 

# 

# INTRODUCTION

The NomadicBTS GUI project has created a wonderful platform that allows the learning and teaching of Network-in-a-box architecture based on the OpenBTS software. Created for both students and researchers, the GUI project makes learning fast and implementation of new ideas easy. This document is aimed at providing intricate details on setting up the NomadicBTS GUI on your computer.

Further documents can be found in our Github.

# OpenBTS

## Requirements

1. Linux Server

This can be a computer or a virtual machine. No recommendation for computer specifications. Although, a 32-bit operating system is the recommended minimum.

1. Software Defined Radio (SDR)

An SDR is a piece of hardware that connects to your computer via a USB cable or ethernet. SDRs from Ettus Research, Fairwaves, Nuand, and Range Networks currently support OpenBTS.

1. Antennas

Select antennas that are tuned for either 850, 900, 1800, or 1900 MHz.

1. Test Phones

Once again pick test phones compatible with the aforementioned bands compatible with the antennas.

1. Test SIMs

Best to purchase blank SIM cards and avoid SIM cards that have a security key programmed into them by a carrier. Blank SIM cards can be programmed yourself or by a reader.

## Installation

### OpenBTS Installation

Follow the itemized procedures. Commands are *italicized*. Outputs are highlighted green. Warnings are highlighted red

1. Setup Git

*$ sudo apt-get install software-properties-common python-software-properties*

*$ sudo add-apt-repository ppa:git-core/ppa*

*$ sudo apt-get update*

*$ sudo apt-get install git*

1. Confirm Git installation

*$ git --version*

git version 1.9.1

1. Generate Git SSH keys

There is a very detailed tutorial on [Github](https://help.github.com/en/github/authenticating-to-github/generating-a-new-ssh-key-and-adding-it-to-the-ssh-agent).

1. Install SSH keys.

Learn to install SSH keys [here](https://help.github.com/en/github/authenticating-to-github/generating-a-new-ssh-key-and-adding-it-to-the-ssh-agent).

1. Clone the Git.

$ git clone https://github.com/RangeNetworks/dev.git

*$ cd dev* (if you get an error, create a new folder elsewhere. Recommended you use a fresh installation of linux and start installation in the ~/ directory)

*$ ./clone.sh*

*$ ./switchto.sh master* (default would be 5.0)

1. Build the code

*$ ./build.sh B200* (Build the code for your specific radio, here we used B200).

You will get a directory “BUILD” containing a folder name by the timestamp of the build. To know if the build was complete, change into the build directory and perform a list directory. The result should be like this:

*$ ls \*.deb*

liba53\_0.1\_amd64.deb range-asterisk-config\_5.0\_all.deb

libcoredumper1\_1.2.1-1\_amd64.deb range-configs\_5.1-master\_all.deb

libcoredumper-dev\_1.2.1-1\_amd64.deb sipauthserve\_5.0\_amd64.deb

openbts\_5.0\_amd64.deb smqueue\_5.0\_amd64.deb

Range-asterisk\_11.7.0.5\_amd64.deb

1. Install dependencies.

Change into the build directory

*$ cd /dev/BUILDS/2020-05-14--13-41-33*

*$ sudo apt-get install software-properties-common python-software-properties*

*$ sudo add-apt-repository ppa:chris-lea/zeromq*

*$ sudo apt-get update*

*$ sudo dpkg -i libcoredumper1\_1.2.1-1\_amd64.deb*

*$ sudo dpkg -i liba53\_0.1\_amd64.deb*

1. Install OpenBTS and Components

*$ sudo dpkg -i range-configs\_5.1-master\_all.deb (use default prompts)*

*If ntp is not installed error occurs, run,*

*$ sudo apt-get -f install*

***ASTERISK***

*$ sudo dpkg -i range-asterisk\*.deb*

(there are 2 asterisk files, see  *6 Build the Code in OpenBTS installation*)

*$ sudo apt-get install -f*

***SIPAUTHSERVE***

*$ sudo dpkg -i sipauthserve\_5.0\_amd64.deb*

*$ sudo apt-get install -f*

***SMQUEUE***

*$ sudo dpkg -i* smqueue\_5.0\_amd64.deb

*$ sudo apt-get install -f*

***OPENBTS***

*$ sudo dpkg -i openbts\_5.0\_amd64.deb*

*$ sudo apt-get install -f*

*OpenBTS is ready.*

# GUI INSTALLATION

## REQUIREMENTS

Listed below are requirements for the GUI to work on your local machine. Take note that steps in the previous section are also requirements for the GUI to fully function.

### USRP DRIVERS

The drivers help your computer to communicate via usb with the USRP. (Note: the USRP needs a USB 3.0 port to function properly).

1. *Install USRP drivers with the command below.*

*$ /usr/lib/uhd/utils/uhd\_images\_downloader.py*

1. *run,*

*$ uhd\_find\_devices*

*to confirm the USRP is discoverable.*

### 

### TARGET\_TERM

1. Install dependencies

*$ sudo apt-get install wmctrl xdotool*

1. Download the script *target\_term* from our git repo and save it in ~/bin (create the directory ~/bin if necessary) .

1. Don’t forget to make script executable

*$ chmod +x target\_term*

1. Refresh

$ *source ~/.profile*

### ENABLE ADMIN PRIVILEGES

*This allows any admin command not to require password input. Do not do this on an internet connected computer with sensitive information.*

1. *Run,*

*$ sudo visudo*

1. *Add the line at the end of the document*

*$USER ALL=(ALL) NOPASSWD: ALL*

### GUI SOURCE CODE

Download GUI source code from git repository.

*$ git clone*

### PYTHON

Python 3.6 or 3.7 recommended

Change default python settings if multiple python installations exist on your machine.

*$sudo update-alternatives --install /usr/bin/python python /usr/bin/python3 10*

## RUNNING GUI

This section will guide you on how to get the source code as well as the django webserver for the web GUI. Follow the guide and report any issues to the Forum.

1. Launch terminal in cloned Nomad repository directory
2. Start virtual environment

*$ source 1/old\_gui/bin/activate*

1. Run the django server.

*$ python 1/manage.py runserver*

1. Launch the GUI in the browser.

Enter <https://localhost:8000> or <https://127.0.0.1:8000> in browser