

VMware Fusion

Installing Parrot OS in the VM

- Download Parrot Security <https://www.parrotsec.org/download-security.php>
 - Add a new machine
 - Drag Security Parrot ISO into appropriate box
 - Choose operating system as follows: Linux and then Debian 7.x 64 bit
 - Choose boot firmware as follows: UEFI
 - Click finish
 - Name it Parrot Security and save it
-
- Once it loads in the vm, and you see parrot security scroll down and click on install
 - Then go through the process, pick appropriate language and region
 - Create a username and password
 - Choose time zone
 - Partitioning method: Guided – use entire disk
 - Select disk to partition: SCSI3 (0,0,0) (sda) – 21.5gb VMware, vmware virtuals
 - Selected for partitioning: Partitioning scheme – All files in one partition (recommended)
 - Do you want to return to the partitioning menu? Select **NO**
 - Write changes to disks? **YES**
 - Install the GRUB boot loader to the master boot record? **YES**
 - Device for boot loader install? **/dev/sda**
 - Install complete, press enter
 - Once completed, another pop up will come up. Pick appropriate languages
 - Check for updates? **Yes, and install**

Install Instrument Cluster Simulator (ICSim)

- Open terminal (MATE terminal)
- `$ sudo apt-get install libSDL2-dev`
- `$ sudo apt-get install libSDL2-image-dev`
- `$ sudo apt-get install libSDL2-ttf-dev`
- `$ sudo git clone https://github.com/zombieCraig/ICSim.git`

Open ICSim (The following commands set up a virtual can interface)

- `$ sudo modprobe can`
- `$ sudo modprobe vcan`
- `$ sudo ip link add dev vcan0 type vcan`
- `$ sudo ip link set up vcan0`

(The following commands start the ICSim and the controls)

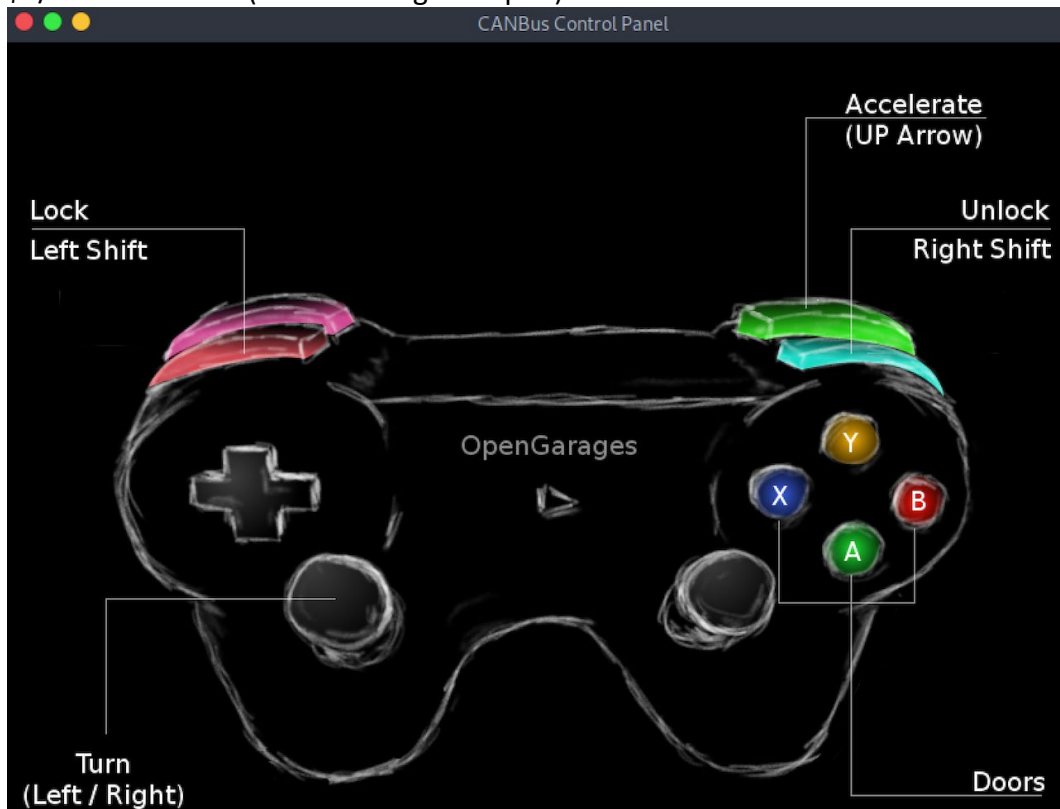
- Type in: `$ ls`
- Then: `$ cd ICSim`

- `$./icsim vcan0` (it will open the following:)



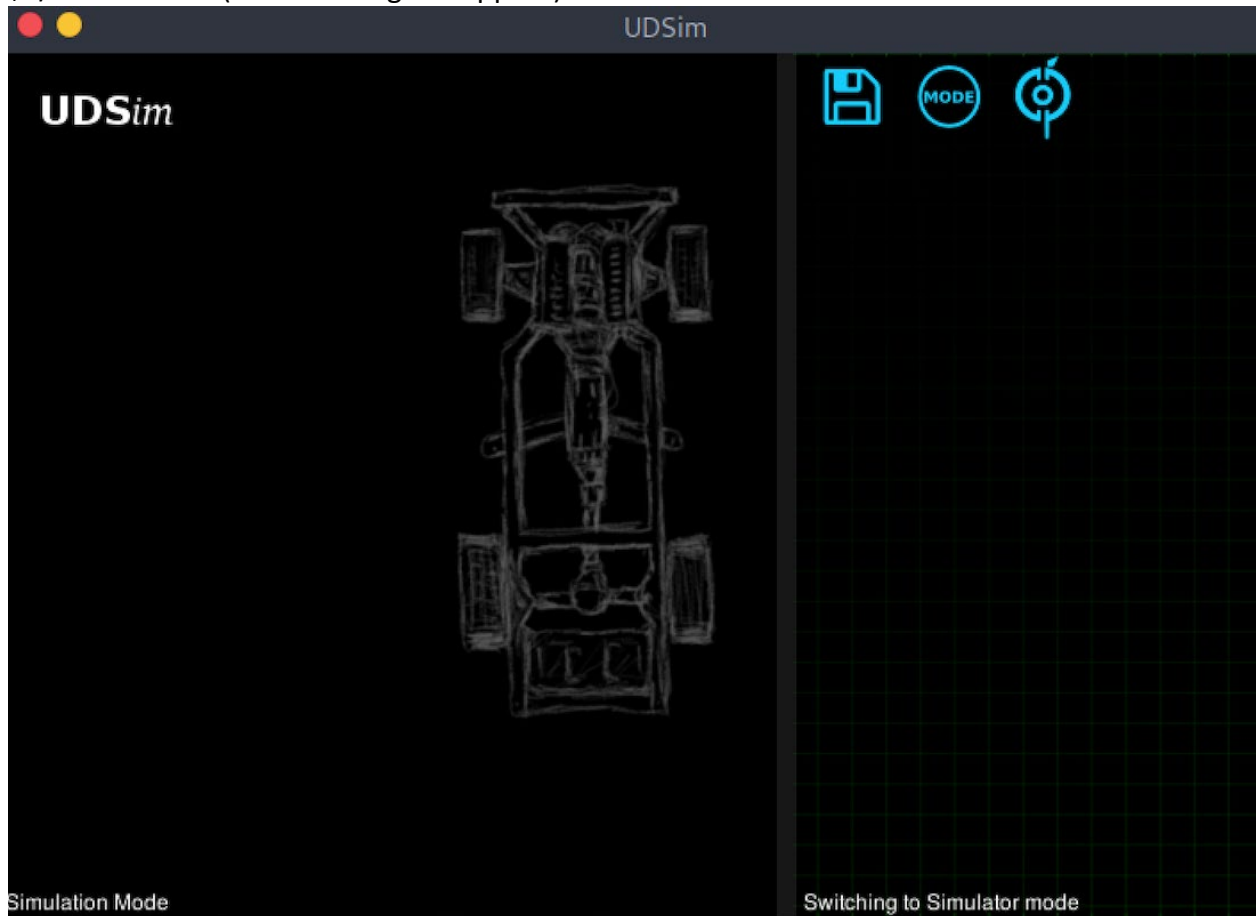
(open another MATE terminal)

- Type in: `$ ls`
- Then: `$ cd ICSim`
- `$./controls vcan0` (the following will open)



Open Unified Diagnostic Services Simulator (UDSim)

- Top left corner of the vm, you will see **Applications**
 - Click on it and scroll down to **Parrot**, then scroll down to **Automotive**, you will see **udsim**
- Click on it and another terminal (**SocketCAN tools**) will appear
- `$ cd src`
- `$ sudo make`
 - Enter password for parrot
- `$./udsim vcan0` (the following will appear)



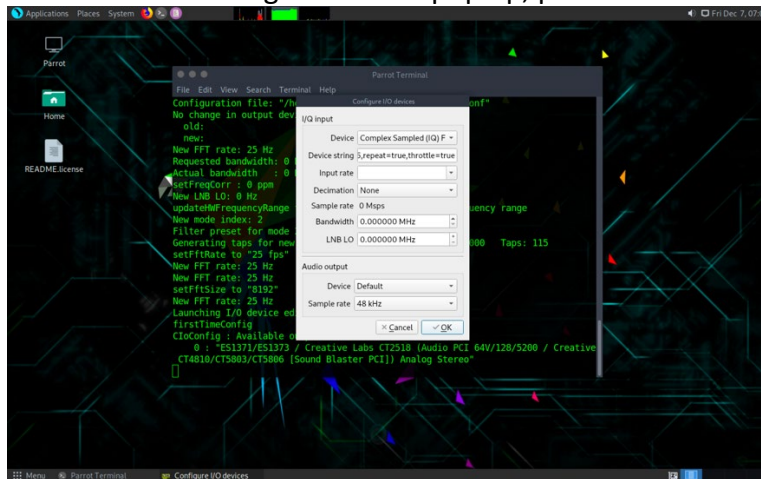
(open another MATE terminal)

- `$ sudo git clone https://github.com/dschanoeh/socketcand.git`

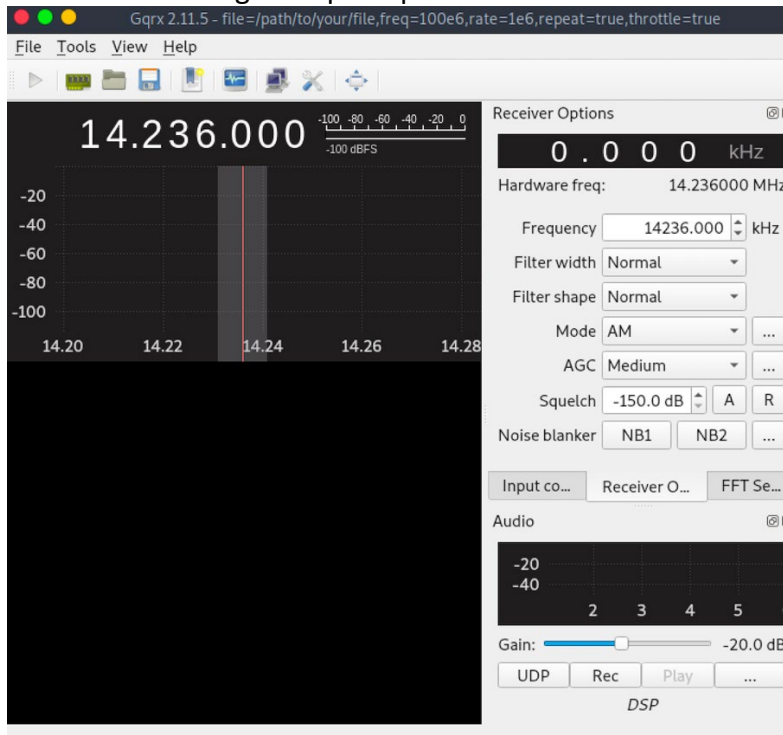
Installing PyBOMBS

- `$ pip install --upgrade git+https://github.com/gnuradio/pybombs.git`
- `$ pybombs auto-config`
- `$ pybombs recipes add-defaults`
- `$ export PATH=$PATH:$HOME/.local/bin`
- `$ mkdir ~/pybombs/`
- `$ pybombs prefix init ~/pybombs/bladeRF -a bladeRF -R gnuradio-default`

- ***If an error with bison occurs use the following**
 - \$ sudo apt-get update
 - \$ sudo apt-get install help2man
- ***If an error installing package uhd occurs**
 - \$ pybombs install uhd gnuradio
 - also install \$ sudo apt-get install python-apt (to make the downloads faster)
- \$ pybombs -p bladeRF install bladeRF gr-iqbal gr-osmosdr gqrx
- \$ pybombs -p bladeRF run bladeRF-cli -- -i
- \$ pybombs -p bladeRF run gqrx
 - The following screen will pop up, press OK



Then the following will open up



Setting up the Arduino CANbus Shield v2

- Open Firefox in Parrot Security
- Download <https://www.arduino.cc/en/Main/Software>
 - Linux 64 bits
- \$ ls
- \$ cd Downloads
- Delete compressed file
 - \$ sudo rm -r arduino-1.8.8-linux64.tar.xz
- Move extracted arduino file into opt directory
 - \$ sudo mv arduino-1.8.8 /opt
- Change into /opt directory
 - \$ cd /opt/arduino-1.8.8
- Make script executable
 - \$ sudo chmod +x install.sh
- Execute with
 - \$./install.sh

Setting up BladeRF A9

- Clone repository by
 - \$ git clone https://github.com/Nuand/bladeRF.git
- \$ mkdir -p build
- \$ cd build
- \$ cmake {options} ../
- \$ make
- \$ sudo make install
- \$ sudo ldconfig
- \$ sudo tee /etc/ld.so.conf.d/local.conf <<EOF
- \$ /usr/local/lib
- \$ /usr/local/lib64
- \$ EOF
- \$ sudo ldconfig
- Go to https://www.nuand.com/fgpa_images/ and download the compatible hosted image to your bladeRF
- Go to https://nuand.com/fx3_images/ and download the latest firmware image and install to your bladeRF
- \$ mkdir build
- \$ cd build
- \$ cmake -DFX3_INSTALL_PATH=/opt/cypress/fx3_sdk -DCMAKE_TOOLCHAIN_FILE=./cmake/fx3-toolchain.cmake ../
- \$ make

- Follow the instructions in the host directory to build and install libbladeRF and the bladeRF-cli utility.
- Attach the bladeRF board to your fastest USB port.
- You should now be able to see your device in the list output via `bladeRF-cli -p`
- You can view additional information about the device via `bladeRF-cli -e info -e version`.
- If any warnings indicate that a firmware update is needed, run:`bladeRF-cli -f <firmware_file>`.
 - If you ever find the device booting into the FX3 bootloader (e.g., if you unplug the device in the middle of a firmware upgrade), see the recovery command in bladeRF-cli for additional details.
- See the overview of the bladeRF-cli for more information about loading the FPGA and using the command line interface tool