# ezRA - Easy Radio Astronomy – Installation - Linux

- Oct-25-2022

ezRA - Easy Radio Astronomy
https://github.com/tedcline/ezRA

## Linux Operating System Installation

On Windows10, from
https://ubuntu.com/download/desktop
I downloaded a 3.6 GB
Ubuntu 22.04.1 LTS (Long Term Support)
ubuntu-22.04.1-desktop-amd64.iso
file. With burncdcc.exe from
https://burncdcc.en.softonic.com
I wrote that bootable .iso file to a 4.7 GB DVD+R disk.

Booting from that DVD, I installed the
Ubuntu 22.04.1 LTS
Linux operating system on an old PC.
I chose
Normal installation,
Download updates while installing Ubuntu,
Install third-party software
and
Erase disk and install.

When finally prompted, I restarted the PC.

I now start with a new Ubuntu 22.04.1 LTS installed.

# ezRA Installation

I copied these installation instructions onto the old PC, to allow copy and paste of command lines.
Or use the ezRA file you are about to download.

Download the ezRA - Easy Radio Astronomy files.
Open a web browser (like Firefox or Chrome) to
    https://github.com/tedcline/ezRA
Left-click on the top right   Green button   and choose "Download ZIP".
A file arrives in the Downloads subdirectory of the home directory, as
    ~/Downloads/ezRA-master.zip

Open a Terminal.  Maybe with ctrl-alt-T.
A terminal window pops up.
Maybe maximize that terminal window.

I suggest copying the single command lines from below, by individually highlighting them and tapping
    ctrl-C
and then using
    shift-ctrl-V
to paste them into the terminal window.
One at a time, enter these commands,

```
    cd  ~/Downloads
    sudo  apt-get  install  unzip
    unzip   ezRA-master.zip
```

Which creates an    ezRA-master   subdirectory in the Downloads directory.

Move or copy the ezRA-master’s    ezRA   subdirectory to your home directory,
and make a new     demo1    directory in your home directory.
One at a time, enter these commands,

```
    mv  ezRA-master/ezRA  ~
    mkdir  ~/demo1
```

Enter the command,

```
    python3  --version
```

I see Python version “3.10.4”.
Good, Python3 is already installed.

Python3 has many commands, but more are available from modules (libraries) that can be downloaded.
The ezRA programs can tell us which of those additional modules are needed.

The ezRA programs require at least Python 3.6, so to download the additional modules we will need to get    pip3    , the Python3 version of the "Pip Installs Packages" or "Pip Installs Python" or "Preferred Installer Program" software.

One at a time, enter these commands,

```
sudo  apt-get  update
sudo  apt-get  install  python3-pip
pip3  --version
```

I see version “22.0.2”.
Good, pip3 is installed.
Enter the shorter command,

```
pip  --version
```

I see the same version “22.0.2”.
Enter the command,

```
python3  --version
```

I now see an upgraded Python version “3.10.6”.

What additional Python3 modules are needed ?
Enter the command,

```
python3  ../ezRA/ezCon.py
```

I see an error message,

```
ModuleNotFoundError: No module named ‘seaborn’
```

Enter the command,

```
pip3  install  seaborn
```

“Seaborn” is large and brings several needed modules with it.

Similar commands for ezCon, ezSky, and ezGal would say 2 additional modules are needed.
One at a time, enter the commands,

```
pip3  install  astropy
pip3  install  scipy
```

Now the command,

```
python3  ../ezRA/ezCon.py
```

should run without error, but because no data filenames were provided, ezCon prints out its help page.
Same for ezPlot, ezSky, and ezGal .
Good.

If you do not need the ezCol program to create .txt data files, you are done.

# ezCol Needs More

The ezCol program collects radio signals into ezRA .txt data files.  You may already have radio data, and not need to install the ezCol program.

ezCol needs an additional module and a lower level library to control the SDR radio, and perhaps more software to control a USB relay.  This requires many installation commands.

Similar to above, enter this command in the terminal window,

```
pip3  install  pyrtlsdr
```

If you care, this higher level module comes from

https://github.com/pyrtlsdr/pyrtlsdr

That module calls this lower level library,

https://github.com/librtlsdr/librtlsdr

We follow a subset of this web page’s installation instructions.
Left-click on the top right   Green button   and choose "Download ZIP".
A file arrives in Downloads subdirectory of the home directory, as

```
~/Downloads/librtlsdr-master.zip
```

One at a time, enter these commands,

```
cd  ~/Downloads
unzip  librtlsdr-master.zip
```

Which creates a    librtlsdr-master   subdirectory in the Downloads directory.

Collect the tools and make the software.
One at a time, enter these commands,

```
sudo  apt-get  install  build-essential  cmake
sudo  apt-get  install  libusb-dev  libusb-1.0-0-dev

cd  librtlsdr-master
mkdir  build  &&  cd  build
cmake  ../  -DINSTALL_UDEV_RULES=ON
make

sudo  make  install
sudo  ldconfig
```

ezCol will write files, so collect them in that new   demo1   directory.
Trying ezCol, with no receiver plugged in, one at a time, enter these commands,

```
cd  ~/demo1
python3  ../ezRA/ezCol.py
```

we should see an error message ending with "Could not open SDR (device index = 0)"
With no receiver plugged in, that is understandable.

Plug one USB SDR receiver into the PC.

The USB SDR receiver I used was either the
    Nooelec NESDR SMArt v4 SDR
    https://www.nooelec.com/store/sdr/sdr-receivers/nesdr-smart-sdr.html
or the version with the bias output,
    Nooelec NESDR SMArTee v2 SDR,
    https://www.nooelec.com/store/sdr/sdr-receivers/nesdr-smartee-sdr.html

Trying ezCol again, with one receiver plugged in, enter this command,

```
    python3  ../ezRA/ezCol.py
```

and again, we should see an error message ending with "Could not open SDR (device index = 0)".
But the paragraph above that error message says
"Kernel driver is active, or device is claimed by second instance of librtlsdr."

We need to “blacklist kernel modules” by adding lines to a Linux system file.
Enter the edit command,

```
    sudo  nano  /etc/modprobe.d/blacklist.conf
```

At the bottom of that file,
add a blank line, and then
add these 7 lines:

```
    #ozone blacklisting for the rtl-sdr
    blacklist dvb_core
    blacklist dvb_usb_rtl28xxu
    blacklist rtl2832
    blacklist rc_core
    blacklist mei
    blacklist mei_me
```

I copy the 7 lines from above and then use
    shift-ctrl-V
to paste them into the “nano” editor.
Then save to the same filename with
    ctrl-O
        and then tap the keyboard   Enter   key to agree to the displayed filename in the lower left.
Then exit the “nano” editor with
    ctrl-X

Check your work.
To print the whole short file to the screen, enter the command,

```
    cat  /etc/modprobe.d/blacklist.conf
```

Does the end of the file appear as you intended ?

Those blacklist.conf file changes require a PC reboot.
    Restart your PC.

Open a Terminal.  Maybe with ctrl-alt-T.
A terminal window pops up.
Maybe maximize that terminal window.

Trying ezCol again, with one receiver plugged in, one at a time, enter these commands,

```
    cd  ~/demo1
    python3  ../ezRA/ezCol.py
```

This time I see

```
    "Found Rafael Micro R820T/2 tuner"
```

and more ezCol text.
Eventually a large "org.matplotlib.Matplotlib3" graphics window pops up.
Success !

When ready, stop the ezCol program by tapping

```
    ctrl-C
```

on the keyboard, a couple of times, into the running terminal window.

I see one or more new .txt data files in the data directory, with this command,

```
    ls  -ltrh  ~/demo1/data/
```

For testing, try inserting an open metal paper clip into only the center contact of the receiver input coax connector, and record the USA FM broadcast band centered on 100 MHz, with the command,

```
    python3  ../ezRA/ezCol.py  -ezColCenterFreqAnt  100
```

I see the spectra of local FM radio stations signals slowly bounce up and down.
But with this data, I am not quite sure what to look for in the ezRA analysis plots.

# ezCol With A USB Relay Needs Even More

```
#############################################################################
#############################################################################
#############################################################################
#############################################################################
#############################################################################
#############################################################################
#############################################################################
#############################################################################
#############################################################################
#############################################################################
#############################################################################
```

blah, blah, blah ....

Windows:

```
# by operating system, initialize (reset) feedRef relay system, if any
if os.name == 'nt':     # Windows
    if ezColUsbRelay:
        # https://github.com/pavel-a/usb-relay-hid
        # https://github.com/pavel-a/usb-relay-hid/releases/tag/usb-relay-lib_v2.1
        # C:\Users\c\Documents\EZRA01\usb-relay-hid_bin-20150330a\bin-Win64> hidusb-relay-
cmd.exe on 1
        os.system('hidusb-relay-cmd.exe off 1')
        os.system('hidusb-relay-cmd.exe off 2')
        sleep(0.5) # Sleep for 0.5 seconds
```

Linux:

```
else:                   # (posix) Linux assumed
    if ezColUsbRelay:
        # https://github.com/darrylb123/usbrelay
        # sudo apt-get update
        # sudo apt-get install usbrelay
        os.system('sudo usbrelay BITFT_1=0 BITFT_2=0')
        #####################
        #    Serial: BITFT, Relay: 1 State: ff --- Not Found  <=================== BITFT Not Found !
        #####################
        #    > lsusb -v -d 16c0:05df
        # - output looks just like on https://github.com/darrylb123/usbrelay
        #####################
        #    > sudo usbrelay
        #    Device Found
        #      type: 16c0 05df
        #      path: /dev/hidraw2
        #      serial_number:
        #      Manufacturer: www.dcttech.com
        #      Product:      USBRelay1
        #      Release:      100
        #      Interface:    0
        #      Number of Relays = 1
        #    HW348_1=0  <================================= OK, use HW348_1 not BITFT_1 ==========
        # https://www.npmjs.com/package/node-red-contrib-usb-hid-relay/v/0.2.3
        # says also available are
        #    HW-348
        #    HW-343
        #    HW-341
        #    Models with USB-Relay-1, USB-Relay-2 or USB-Relay-4 printed on the PCB
        ####os.system('sudo usbrelay HW348_1=0')        # works !

        # also may be helpful ?:
        #   Human Interface Device (HID)
        #   http://vusb.wikidot.com/project:driver-less-usb-relays-hid-interface
        #       https://github.com/pavel-a/usb-relay-hid
        #       http://vusb.wikidot.com/hosted-projects
        #           http://vusb.wikidot.com/examples
        #               https://www.workinprogress.ca/v-usb-tutorial-software-only-usb-for-mega-tiny/
        #   https://www.giga.co.za/ocart/index.php?route=product/product&product_id=229
        #       - part is out of stock, but has pictures and links to
        #           https://github.com/pavel-a/usb-relay-hid
        #           http://www.giga.co.za/Kit_Drivers/USB_Relay2.zip
        #           https://github.com/darrylb123/usbrelay
        #           and says
        #               Here is an example how to control the relay in command line.
        #               CommandApp_USBRelay.exe [device id] [close / open] [relay nr]
        #                   CommandApp_USBRelay.exe J34EL close 01
        #                   CommandApp_USBRelay.exe J34EL open 01
```