

SDRuno (ExtIO) on Linux using Wine 4.0 on Ubuntu 16.04

A Guide Put Together by The Community

The following procedure outlines the approach used to get the ExtIO version of SDRuno running under Wine within a Linux environment. This was pulled together as a result of the efforts of various individuals on the SDRplay forum and we have found that it works on Ubuntu 16.04. Please note the following:

- This is for x86 platforms ONLY and will not work on ARM systems
- If you have little or no experience with Linux, this is not recommended. Linux is in our view, not for the inexperienced or faint-hearted users.
- This is NOT an 'official' SDRplay flow. We will not be providing ANY support for users who try this and have problems. We recommend that if you are having problems that you reach out via our community forum and hopefully the people that figured this out will be in a better position to advise you.
- This flow also works for HDSDR. It may also work for SDR# or any other Windows applications that use the RTL_TCP interface or the ExtIO interface.
- This will NOT work with the main version of SDRuno (non ExtIO version) at present, but we hope that we will be able to provide hooks in the future so that it will.
- This does not diminish our commitment to develop a fully-fledged native Linux version of SDRuno as part of our cross-platform development program. This is intended to open more options for dedicated Linux users.
- This has only been tested on Ubuntu 16.04 by us and we have no idea what other versions of Linux it will work with.

Observations

- As the acronym suggests, Wine is not an emulator. It requires both 32bit and 64bit binaries to support running 32bit and 64bit applications. SDRuno is a 32bit application, so whether you are using a 32bit or 64bit OS, the 32bit build of Wine is necessary.
- Having tried installing from both pre-built packages and source code, unless you are doing this on a 32bit OS, I would highly recommend installing from pre-built packages. The source code build method on a 64bit OS requires you to have a way of building 32bit binaries. It is suggested that a virtual 32bit environment is constructed to do this, which is yet another step for you to have issues with.
- When I finally got this running, SDRuno seems to hang if the RSP TCP server is not running locally. Most likely this is an issue with the ExtIO plugin auto starting.
- This guide is for Ubuntu 16.04, this can be done on other Linux distributions but will most likely have differences. Check <https://wiki.winehq.org> for details
- This guide is also for a 64bit OS, this can be done on 32bit OS, but again, the procedure will be different. Check the website link above for details. 32bit support is required for SDRuno
- There are different levels of build for Wine - Stable, Development and Staging - this guide is only going to focus on the Stable path. See above website for differences relating to the other paths

High Level Tasks

1. Install Wine including dependencies and fonts
2. Install SDRuno
3. Put the RSP TCP ExtIO plugin in My Documents
4. Install API 2.13
5. Start RSP TCP Server
6. Run SDRuno (ExtIO)

Task Walkthrough

1. **Get to a known position**
`sudo apt update`
`sudo apt upgrade`
2. **Add package support for 32bit (i386)**
`sudo dpkg --add-architecture i386`
3. **download and apply WineHQ repository key**
`wget -nc https://dl.winehq.org/wine-builds/winehq.key`
`sudo apt-key add winehq.key`
4. **add the repository**
`sudo apt-add-repository 'deb https://dl.winehq.org/wine-builds/ubuntu/ xenial main'`
5. **update the packages**
`sudo apt update`
6. **install wine packages and dependencies**
`sudo apt install --install-recommends winehq-stable`

At this point, Wine "should" install. However, in our experience of installing this on 2 Ubuntu 16.04 systems, one worked like this and the other failed. The one that failed had complaints that different packages were either uninstalled or broken and the installation could not continue. Here's how we fixed that...

```
sudo apt install aptitude
sudo aptitude install winehq-stable
```

This then reported what dependencies couldn't be installed. By running this on each of the listed dependencies, eventually they were all loaded...

e.g.

```
sudo aptitude install wine-stable-i386
sudo aptitude install wine-stable
sudo aptitude install winehq-stable
```

At this point the last thing to load was the font package (otherwise SDRuno could start with all the text missing). If you have had a failed installation previously of Wine, the system could already think that the font package was installed and so the only way this could be corrected was to force a reinstallation...

```
sudo apt install --reinstall ttf-mscorefonts-installer
```

This requires some input during the installation, just follow the prompts.

SDRuno Installation

Download SDRuno V1.22 from the SDRplay Downloads page as per normal and then in a terminal window, navigate to where the installer exe is, then...

```
wine ./SDRplay_SDRuno_Installer_1.22.exe
```

This will start the installation process. The first time you run wine, there will most likely be a series of windows that appear telling you that either the configuration is being setup or that Mono or Gecko don't exist and there will be a prompt asking you if you want the system to automatically download them, which you want to accept.

Note: All of the files related to Wine are stored in ~/.wine, where ~ represents your user account top level, normally /home/{username}

At this point the SDRuno icon will have been added to the Desktop. Copy this to SDRuno.ExtIO and then edit the ~/.Desktop/SDRuno.ExtIO.desktop file in your favourite text editor. Change the contents to look like...

```
[Desktop Entry]
Name=SDRuno.EXTIO
Exec=env WINEPREFIX=~/.wine" wine C:\\\\Program\\ Files\\
\\(x86\\)\\\\SDRplay\\\\SDRuno\\\\SDRuno.EXTIO.exe
Type=Application
StartupNotify=true
Path=~/.wine/dosdevices/c:/Program Files (x86)/SDRplay/SDRuno
Icon=0FC5_SDRuno.0
StartupWMClass=sdruno.EXTIO.exe
```

Save it, this is SDRuno installation complete.

ExtIO Plugin Installation

Download the ExtIO RSP TCP plugin from the SDRplay Downloads page again as normal for Windows. This will be a zip file. Extract the contents of the zip file and copy the DLL file to your Documents folder (by default ~/Documents is equivalent to the Wine My Documents folder)

```
cp ExtIO_RSP_TCP.dll ~/Documents
```

Install API 2.13

Before building the RSP TCP Server, the appropriate RSP API for your OS/processor needs to be downloaded and installed. Navigate to SDRplay Downloads page and download the Linux x86 API. Then in a terminal window change to where that is downloaded and perform the following to install it...

```
chmod 755 ./SDRplay_RSP_API-Linux-2.13.1.run
./SDRplay_RSP_API-Linux-2.13.1.run
```

Follow the prompts and the API will install.

Start RSP TCP Server

The RSP TCP server can be downloaded and built using these commands...

```
git clone https://github.com/SDRplay/RSPTCPServer.git
cd RSPTCPServer
mkdir build
cd build
cmake ..
make
sudo make install
```

At this point, connect the RSP and launch the RSP TCP server...

```
rsp_tcp -E -a 0.0.0.0
```

Start SDRuno

Now that the RSP TCP server is running, double click on the SDRuno (ExtIO) icon on the desktop. The first time this is run it could take some time to start and you may see the messages regarding Mono and Gecko not being installed, these will be installed at this point if that's the case.

If everything has gone right, the SDRuno Main Window will appear. The SP1, SP2 and RX control panels can be opened by pressing the corresponding buttons to get you started.

Note: This is the ExtIO version of SDRuno and it will lack a lot of the functionality of the main SDRuno application that we are actively developing.

I would view this ExtIO on Linux method as an interim flow until we have SDRuno developed for cross platform.

Acknowledgement

Thanks to StandingWave on the SDRplay Community Forum for making us aware of this flow. You can find the full discussion about this on the community forum here:

<https://www.sdrplay.com/community/viewtopic.php?f=11&t=4187>