Documentation for RJOVER

Tyler Kovach

March 23, 2021

Contents

About

This is a short document that should contain all of the necessary information needed to boot and install an OS and prepare a Raspberry Pi 4B for use with the RJOVER project. This doucment will never be complete as likely there will be bugs along the way and new advancements in the field that render this document compltely useless. However, as of the date on the front cover this guide should be up to date. As always, use this document at your own risk.

As the RJOVE project was originally created to be an inclusive learning environment for new students we strive to maintain this ease of use and maximize the learning potential from this new antenna reciever. We the RJOVER team want to strive to continue in the path of predescessors and make this document as inclusive as possible to new users. We will try our best to explain what everything is, but if we miss anything do not be afraid to Google! Sometimes the best way to learn is to do research and try new ideas out. If there are any questions feel free to contact the RJOVE team, or us the RJOVER team at:????? and ?????? respectively.

What is New

As may have come across in the About section, the RJOVER team is not the original creators of this project. This project was originally created by NASA to begin research of Jupiter atmospheric signals sent here to the Earth. Their original devices were created in a different age, where all radio signals were received and analyzed using physical circuits and sound cards. We now live in a truly digital age where information is more easily accessable and have powerful computers lying around nearly everywhere that we can take advantage of to do data analysis.

What is amazing is that engineers have found ways to fully simulate the processes necessary to filter and process signals without the need for physical circuitry anymore. Our project is thus minimizing the cost for you to help with this research and modernize it with more modern devices. The only 2 devices we recomend you to use with this project are an RTL-SDR and a Raspberry Pi 4B.

The RTL-SDR found here Amazon link to RTL-SDR is what we used in our project, what this guide is made for, and what we recomend you use. SDR stands for a software-defined-radio and its acts just like what its name sounds. This is the device that will be replacing the standard RJOVE receiver. We will eventually get to a point of explaining how it works, but for right now it is just a device.

Secondly, a Raspberry Pi is not what it sounds like. In fact, these are small computers that run on ARM Processors. For our project this will act as our hub computer and control the RTL-SDR as well as act as a storage medium for the data the project collects.

Bill of Materials

This section will describe all of the necessary materials needed to setup and properly use the RJOVE project, as well as contain links to each of the necessary elements.

- 1 RTL-SDR Amazon link to RTL-SDR
- 1 Raspberry Pi 4B with atleast 4Gb of RAM Amazon link to RaspPi 4B
- 1 Micro SD card ?????insert size???? Amazon link to SD card
- 1 power adapter Amazon link to Pi Power Supply
- Atleast 1 Micro-HDMI to HDMI, or some other preferred video connection Amazon link to HDMI
- 1 monitor
- 1 keyboard
- 1 computer you are familiar using
- 1 Micro SD card Writer
- An Internet connection

What is an OS

An OS, also known as an Operating System, is a type of program that facilitates your interaction with a computer. Many new users come across one or two of the major OS types. A brief list of a few popular ones include Windows, MacOS, IOS, and Android. What all of these have in common are that they perform a gateway function to allow users to easily control computers without having to know specific machine code instructions. They also form the foundation on which computer programs run from. The first two are similiar since their modern forms are desktop x86 64bit architectures. The second two are made for ARM processors in mobile phones which are more power efficient. Take a look at the Raspberry Pi. Does it look like a modern laptop or PC? No, not really, it is more similiar to a modern smartphone than a desktop. More specifically it uses an aarch64 architecture. This means non of the above operating systems will work "out of the box" and we will have to use alternate forms of operating systems on this.

The OS of our choice is a version of Ubuntu Linux called Lubuntu. Linux is an OS not disimiliar to Windows and MacOS. The largest difference is that it is known for being entirely open source and free. Recently it has been gaining in popularity so it has been less hard to get programs to be coded for Linux users who aren't programmers. Typically this is when a lot of Linux jargon is thrown at the reader and they get all confused on what it all means. We will not take this apporach since a simple google search of how to install Linux, or what is Linux, will answer those questions beyond unreasonable doubt. Instead we will only introduce material only deemed important and leave the rest as an interesting research project for interested readers.

The biggest difference between Linux and Windows for instance is that Windows is a GUI designed OS whereas Linux is a terminal based OS. Recently there have been more versions of Linux that have GUIs, but most users tend to use the terminal for most of the tasks since it jsut makes using your computer more efficient. If we wanted to be more formal, these different versions of Linux I have been mentioning are known as distibutions. There are many, many different distros of Linux available to download and install. I give you permission to leave this tutorial and go google some Linux distributions. Two of the most commmon introductory Linux distros are Ubuntu and Mint. Another more complicated distro is known as Arch. For the purpose of this tutorial we will diregard Linux Mint since as of writing this, they do not support the aarch64 architecture for the Raspberry Pi 4B.

Ubuntu is such a large distribution that it has divided up into a couple of flavours that allow for different functionality while utilizing the same core pacakge manager and default applications. These flavours change up the environment that you are computing in. Some flavours act similiar to Windows in the form of having LXQt and Cinnamon desktop environments. A great example of this is Lubuntu. Other more notable flavours are Kubuntu and Ubuntu Gnome. For a more inclusive list visit this link. Here we will be using Lubuntu for the actual operation of the Raspberry Pi since it is known for being one of the lightest resource consuming flavour of Ubuntu. This is perfect for the Raspberry Pi since it has limited system resources and has to make the most out of the resources it has. We want to minimize the amount of RAM it takes to run the desktop environment so this RAM can be instead put to use actually decoding signals.

Along the way to Lubuntu we will spend a couple of minutes in the Ubuntu Gnome basic environment. It may look a bit strange as it is a completely new form of a desktop created only for Linux. This creates both benefits and disadvantages. On a modern computer it is really reponsive and a neat environment to work in. However, the cost is this environment will on average consume 4GB of RAM all of the time. With a Raspberry Pi with only 4GB of RAM, this would mean nothing can get accomplished on this computer as long as the desktop is beign run. This is why we abandon this environment in favor of Lubuntu in case you were wondering.

People write books on the topic of Linux Distos so I hope this was enough information to atleast describe why we are using the OS we are and leave enough breadcrumbs for interested readers to do some more research and learn more about Linux.

Installation of Boot Media

The first step to begin the install is to navigate your way to this website to begin the install process of the Ubuntu OS onto the SD card.

Plug in the SD card reader and put the SD card in the corresponding slot on the reader. Wait until your computer recognizes that an SD card has been inserted to be used. Typically the computer chimes and a popup appears saying that the SD card is ready to be used. Open up the Raspberry Pi Imager app and click on the "Choose OS" button on the bottom left. A menu should appear asking for which OS you would like to install. We will be installing Ubuntu desktop for this tutorial. In the menu select Other general-purpose OS- Ubuntu- and finally Ubuntu Desktop 64-bit. Right now as of writing this tutorial the desktop we will be using is Ubuntu 20.10 but in the future, this will likely change.

Next, select the SD card from the middle column and select the SD card. The next step will delete all the data on the SD card so realize that this is a not undo-able endeavor. The SD card willl be reformated to be used as the boot drive for the Pi. Follow the rest of the steps as directed by the program after clicking write in the right column.

This should take some time, 30 minutes on Windows. Afterward, plug it into the Raspberry Pi. The port you are looking for is on the opposite side of the Pi from all of the USB plugs. If you are looking down at the PCB of the Pi, insert the SD card upside down, so you should be looking at the golden contacts pads when you slide it into the SD reader.

Setting up the Environment

The next step on our journey will be booting the Pi onto the SD Boot card you made in the previous step. This is important since if you improperly insert commands you may end up breaking the Pi, deleting important data from the SD card, forcing you to reinstall a clean boot image of the OS once more.

Once you have gotten into the desktop hit the Windows/Super key to pull up the menu for Ubuntu. Type in the terminal and let it pop up. Next, we will have to execute a terminal command.

Type: sudo apt install lubuntu-desktop

You should need to enter your password and the terminal should take control. A pink menu should appear 5 minutes in asking for a default system setting. Navigate using arrow keys to highlight sddm3 and hit enter. This will make it so the next time we boot into Ubuntu we will be booting into Lubuntu instead.

After this finishes, we will need to reboot the pi.

Type: sudo shutdown

This will shut down the pi. We will also have to power cycle the pi as well. Unplug the power cable, wait for a second, and then plug it back in. The Pi should boot and you should see a bunch of text with green boxes on the left side. This is perfectly normal and is what you will see when booting into Lubuntu.

You should see a new boot menu asking for your credentials. This is the same password you made earlier so type it in and hit enter. You will be logged in to Lubuntu. You may notice it feels much snappier than Ubuntu felt earlier when you are booting up the terminal.

Lubuntu should carry over the wifi settings from Ubuntu, but this tutorial used an ethernet connection to the Pi. Hit the Super key again and the menu for Lubuntu should appear. Type terminal and select either of the 2 options. The terminal should pop up on the screen.

Type: sudo apt update

This command should update all of the PPA's that Lubuntu installed to get its package manager operational and also Pi-specific support.

Type: sudo apt upgrade

This command should take a while as the Lubuntu package manager, Muon should upgrade all of the various applications that are preinstalled and are ancient versions. After this concludes your Raspberry Pi should be a functional 64-bit computer that can do regular computing tasks.

Next, we will setup GnuRadio.

Type: sudo apt install GnuRadio gnuradio-dev gnuradio-doc

This command will install GnuRadio onto your Pi. After it finishes you should be able to hit the super key, type in GnuRadio and the GnuRadio companion app should become visible to select.

Next, we will make the Pi compatible with the RTL-SDR. There is a great guide here:https://ranous.files.
sdr4linux$_{q}uickstartguidev20.pdf.FollowittosetuptheRTL-SDRtobecomefunctionalonthesystem.$

The next step is to get GnuRadio running on the pi and receiving data from the RTL-SDR.

$_{I}{}^{nsertstuffoncodinginGnuRadio}$

To run your code without a GUI we have to change your python environment.

Type: conda deactivate