



Radio Astronomy with Raspberry Operating System

Glen Langston, 2020 June 30

This document describes the Operating System provided by our research group for Raspberry Pi computers configured for Science Aficionado Radio Astronomy. The Operating System (OS) is

courtesy of nearly countless volunteers in the Raspberry Pi Foundation, the Ubuntu Organization, the GnuRadio Collaboration and the folks at SDRplay. The [GnuRadio Collaboration](#) made home radio astronomy possible, thanks to their free software for digital signal processing. The community of volunteers have been very helpful in creating a fun to use package for learning the fundamentals of modern communications. The SDRplay company put together the basis of the OS package including GnuRadio, in order to ease testing of their RSP1A for numerous applications.

The notes here describe only my additions and some hints on setting up networks of radio telescopes, with precise time (hopefully with time tags correct to roughly a microsecond).

Purpose

The goal of this project is to enable everyone to discover that our Milky Way Galaxy surrounds us and is visible day and night, with your own home-built radio telescope. **Figure 1** (below) shows our location in the Milky Way and some calibrated observations, like those you will make. There are several steps involved to get to this point. This document is intended to get you to this level of observations in about a week's work, after you've gathered/constructed all the parts.

After you accomplish your first observations, the universe is yours to explore. There is virtually no limit to what you can discover with your radio telescopes, if you choose to expand the basic telescope with your own additions.

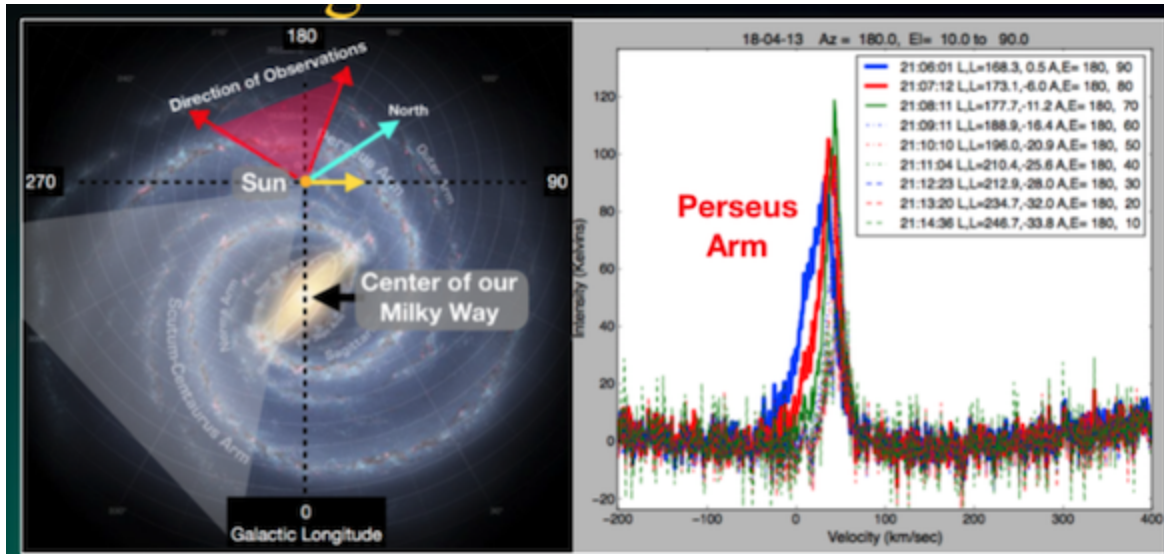


Figure 1: Overview of our place in the Milky Way Galaxy (Left) and 10 Minutes of Observations of the Perseus Arm. The sketch at left shows our Sun (and us) far from the center of the Milky Way. The image was drawn as if we are way above our galaxy. Our galaxy is a disk and the coordinate of the center of our galaxy is at Galactic Longitude = 0. The galactic longitude, latitude coordinates are centered on us. The plot at right shows 9 beautiful minutes of data. With some research, you can figure out that you've discovered the Perseus Arm of our Galaxy. The plot shows calibrated intensity (Kelvins) versus the velocity of the hydrogen measured. The observations were taken with telescope Azimuth=180 degrees, and different Elevations (A,E =). The GRC block calculates the Longitude and Latitude (L,L=) for the time of the observations (21:06 to 21:15 UTC).

System Components

The radio telescope consists of a simple metal object to funnel the radio signals from the sky into the first very low noise amplifier (LNA). After increasing the very faint signal by a factor of about a million, by the first amplifier, the signal is fed to a Software Defined Radio (SDR) device. These devices range in price from about \$30 to \$200. The more expensive ones have many unique capabilities. The total telescope cost will be between \$280 and \$380, depending on the size of the telescope you build and which SDR you buy.

All the devices plug into computers via USB. A computer is needed to gather the data into a format for display. You can use your own laptop, but we've found that there is more work involved in getting all the software installed than in building the telescope! Therefore we recommend you buy a dedicated Raspberry Pi 4 computer, with power supply, case and SD card for about \$70.¹ ([For the full introductory kit, the cost is a little more](https://www.raspberrypi.org/products/raspberry-pi-4-model-b/)).

¹ <https://www.raspberrypi.org/products/raspberry-pi-4-model-b/>

Figure 2 shows all the telescope hardware you're going to need for your first observations. The picture on the left shows the horn, made out of a 1 foot long, six inch diameter (30x15 cm) stove pipe, with a 8 inch to 6 inch duct reducer (20 to 15 cm). Attached to the side is a small metal box with a single amplifier installed. For scale, the carpenter's square is 18 by 24 inches (46 by 60 cm).



Figure 2: Left is the horn. At right is the side view of the horn and the computer box containing the SDR and the Raspberry Pi 4 computer. The Pi is powered by the blue ethernet cable, which also connects the Pi to the computer network. The Pi is in a weatherproof box.

Software Defined Radio

A Software Defined Radio (SDR) is an electronic device that includes amplifiers and a high speed sampler that measures the incoming signal millions of times a second. The prices of these fantastic devices have dropped dramatically. This document is targeted towards the minimum cost radio telescope and for that system the SDR to use is the \$38 NeSDR,². Figure 3, at right, shows two SDRs attached to one Raspberry Pi computer. Only one SDR is needed. The devices are about 2 cm square and 5 cm long.



Figure 3: Two low cost SDRs attached to a Raspberry Pi computer.

The NeSDR comes with antennas that you can use for testing (and playing). However you'll need to construct the special purpose feed for observations of the Milky Way.

² <https://www.nooelec.com/store/nesdr-smart.html>

Many other SDRs provide more capabilities. The software provided here also works with the Airspy SDRs (mini and R2)³, the Analog Devices PlutoSDR⁴ and the SDRPlay⁵ devices. Each of these devices has different, fantastic features and additional costs.

I've had reasonably good luck with the [Airspy-mini](#)⁶, as it has a flat band pass response and plenty of Bias-tee power for the first LNA.

Currently I'm mostly using the SDRplay RSP1A devices, which have wider bandwidth (9.0 MHz), and very good software support. (Unfortunately the RSP1A does not have sufficient power in their bias tee circuit for my amplifiers, so an additional component, a bias tee is needed).

Computer System Setup

This guide assumes you have one or more Raspberry PI 4 computers for your telescopes and a **host** computer, either Linux Ubuntu on a x86 laptop or a Mac computer. I almost never connect a keyboard and monitor to my Raspberry Pis, unless there is some sort of problem. Rather the **host** computer runs *Virtual Network Connections* to the PIs. After installation you will have a complete computing environment. This includes a python programming system and C++ compiler. There is no end of programs you can create.

The OS includes GnuRadio and gnuradio-companion, an easy to use tool to explore signal processing. GnuRadio can be used without any SDRs. Click this [link to see Sophie and Evan give a quick video introduction to GnuRadio.](#)



Figure 4: Raspberry Pi 4 in hand.

Download

The single file, SDRplay-20Jun28.img.xz, containing the OS is available as a link to a very large file:

³ <https://airspy.com/>

⁴<https://www.analog.com/en/design-center/evaluation-hardware-and-software/evaluation-boards-kits/adalm-pluto.html#>

⁵ <https://www.sdrplay.com/>

⁶Supports 3.0 and 6.0 MHz bandwidth. 6.0 MHz is excellent for observations of the Milky Way.

<https://drive.google.com/file/d/12QfEosneqLEIQL4mx86z0q3VjvDUbd2x/view?usp=sharing>

This file is a 2.7 GB compressed version of the full 9.5 GB OS. The OS is fully functional with many standard programs plus configuration for GnuRadio (version 3.7.13.4) and Radio Astronomy software from West Virginia University.

You will need a 16 GB, or larger, SD card to install the OS.

This file must be downloaded to your host computer. Transfer will take a while. This software is the product of many hundreds of volunteers. GnuRadio is a great project built on top of the Linux operating system. This particular version was created by the folks at sdrplay.com, then tuned for Radio Astronomy by our research group.

Transfer to SD Card

I use free download of the program Balena Etcher (<https://www.balena.io/etcher/>) to transfer copies of this OS to SD Cards for my telescopes. I usually use 32 GB cards, but 16 GB cards are sufficient, as the OS is only 9.5 GB. One very nice thing about Etcher is that you don't have to first uncompress the file, just start Etcher writing the compressed file to the SD Card.

After configuring your own Raspberry Pi changes for your experiment, you can easily remove the SD card and make an archive version using the Linux "dd" command. Before writing the SD image, I try to clean up any unused files and observations first.⁷

Gparted

The SD card can be used immediately after the OS is transferred. However the OS was compressed to make downloading quicker. If you want to use the full sized SD card, the card needs to be re-partitioned after writing. There is a Linux program available for free to increase the partition to full size. Using the linux "gparted" command (available via *sudo apt-get install gparted*), I plug the SD card into a USB adapter and increase the partition to the full size of the SD card.

After getting the Radio Astronomy software working (to the best of my ability), I then "gparted" it back down to a small size before saving it for distribution.

Initial Configuration

Because I plan to use this OS in Raspberry PIs for Radio Astronomy, the operating system is configured with WIFI and Bluetooth off. In my configuration, I almost never connect a keyboard and terminal to the Raspberry PI, but rather Virtual Network Connection (VNC) into the PI. For

⁷ The "dd" command: `sudo dd bs=10m count=900 if=/dev/disk3 of=SDRPlay-20Jun28.img`

this to work, you need to know the default IP address of the PI. ***For this version the default IP address is 192.168.1.205.*** You can turn WIFI on after your first login via VNC.

This IP address is chosen because it seems to be the default IP address group for un-managed Power over Ethernet (POE) switches. The switch IP address is 192.168.1.1. The **205** part was chosen because this happened to come from the 5th of the telescopes in my network. I've been using 192.168.1.200 as the IP address of the linux **host** computer that harvests the data from the telescopes and also provides a GPS time reference.

I use the free download version of RealVNC (<https://www.realvnc.com/en/>). This enables multiple connections to several Raspberry PIs.

Each PI automatically starts a passwordless VNC session. You probably want to give your VNC sessions passwords, but to avoid initial problems I removed the password. When you start VNC, the VNC session will complain that the session is not secure, because there is no password.

The PI also starts a protected "ssh" server, allowing direct login, without the graphical interface. To login, the special word is "H1!drogen", without the quotes.

Use your mouse and keyboard to run various tutorials on the Raspberry pi operating system and configuration.⁸ The top line of the PI has a web browser and a terminal. After enjoying the PI features, open a terminal. This is the old fashioned way of running a computer. (I'm old fashioned.) You need only type two commands:

```
cd examples  
python NsfIntegrate24.py
```

The first command changes to the directory (cd) containing the GnuRadio designs for Radio Astronomy. The second command starts a python program. The programs are also available for free from our github site, [West Virginia University Radio Astronomy Instrumentation Lab \(WVURAIL\)](#). These programs have been tested on Linux x86 laptops, Mac computers, Odroid N2 and Raspberry PI (3 and 4) computers.

After starting the python program, a graphical user interface appears and lots of messages. If this fails the first time, try it again. It seems that the hardware does not wake up on the very first try, but will be fine thereafter. You will see a display looking like the next figure.

⁸ <https://thepihut.com/blogs/raspberry-pi-tutorials/the-raspberry-pi-tutorial-beginners-guide>

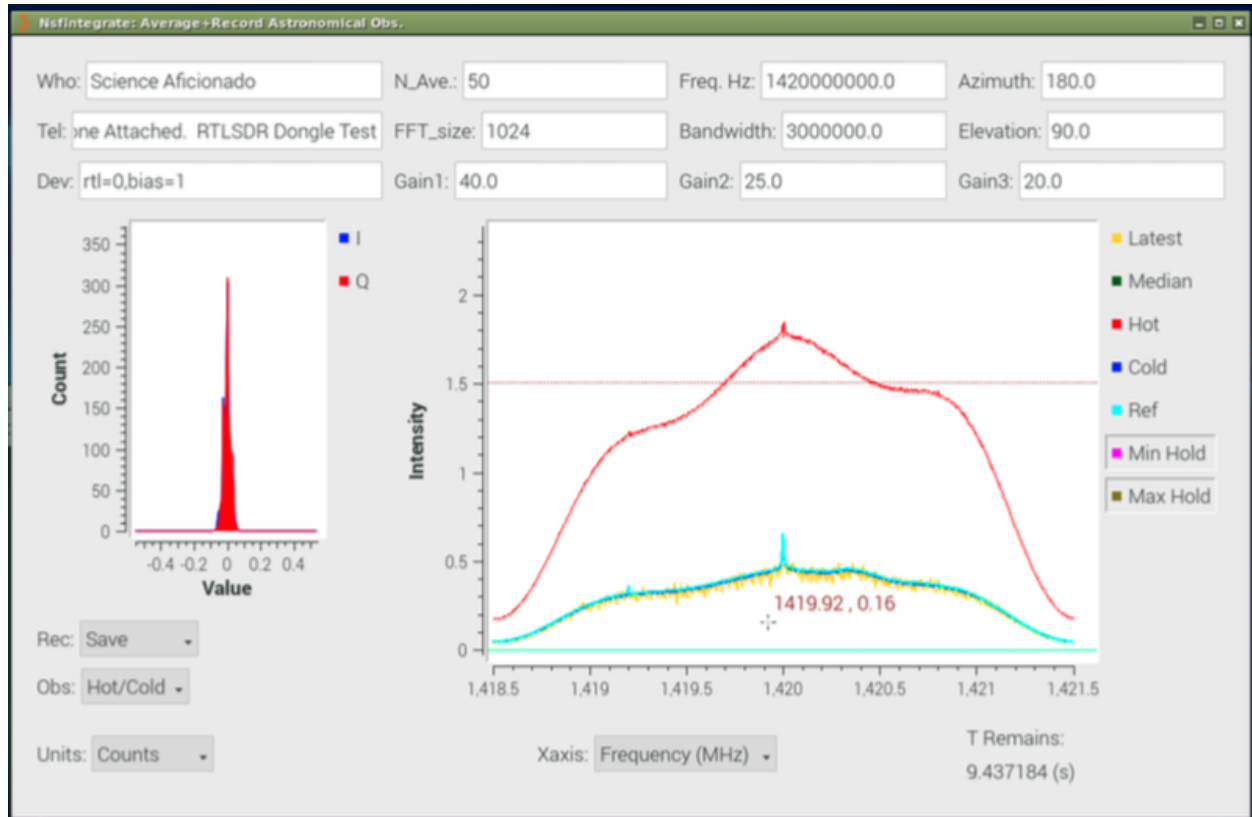


Figure 5: Raw Radio Observations of the Sky.

First Observations

When you've completed the [horn construction](#),⁹ you'll need to first look at the ground to calibration the observations. The calibration consists of two observations. The Hot load is the ground, which has a temperature of about 280 Kelvin. You will need to set the elevation to -90 degrees and set the Record mode to "Average". What about a minute. Set the Observing mode to "Hot/Cold", then change the record mode to "Save" to save the Hot load observation. A **red line** will appear where the average line was. Set the observing mode back to "Survey" and then set the recording mode to "WAIT".

The empty sky is very cold, only about 10 Kelvin (-440 F, -260 C).

Go out and move the horn to point straight up. Next set the elevation to up (elevation +90 degrees). Set the Record mode to "Average" again. Wait another minute. Then set the observing mode to "Hot/Cold" and set the Record mode to "Save". A **blue line** will appear where the average plot was. You've completed calibration!

⁹ More Horn Construction documents are needed. The WVURAIL group has also put together a good construction guide for a different shaped horn. For a classroom construction guide see <https://wvurail.org/cra/>

The blue line will show a very weak bump in the plot, which is our Milky Way Galaxy. This can be much more clearly seen in the calibrated plot. Switch the Units to “Kelvins” and the program computes the calibrated spectrum. The broad peak at 1,420.4 MHz is the signal from the spiral arms of our Galaxy. Your plot should look similar to the next figure.

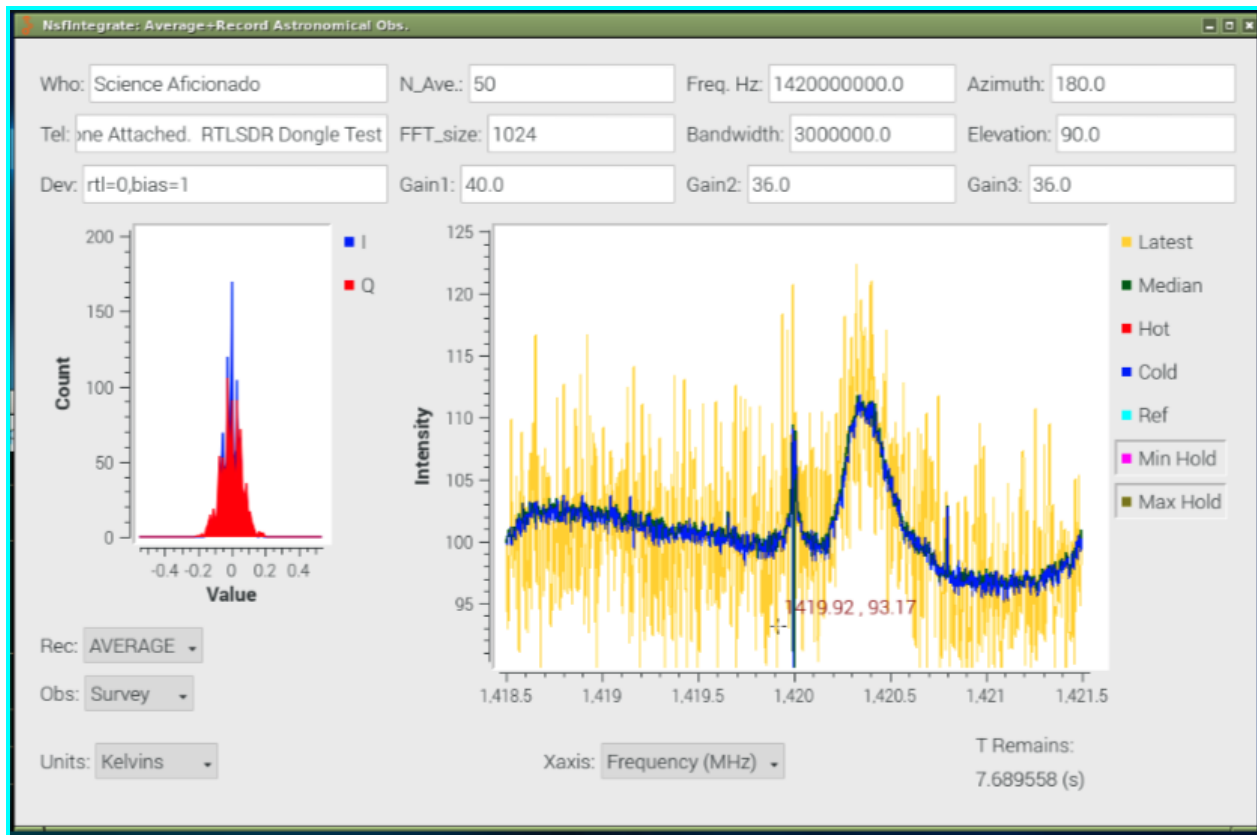


Figure 6: Calibrated Radio Observations of the Sky. The Milky Way galaxy spiral arms are seen as the broad peak between 1420.2 and 1420.6 MHz.

NTP Server

Because the time is critical to Astronomical Observations, it is very important to set the time correctly. Setting the computer clocks after power outages is tedious, so everyone should get a fairly low cost (< \$35) GPS mouse to attach via USB to some computer in their network. Then set a fixed IP address for that computer. (Not required to use the software.)

The IP address of a computer is fixed by editing the file “dhcpd.conf” in the /etc directory. You will also want to edit the “/etc/hosts” file to give your telescopes some names.

The PI also starts a GPS demon, the Precision Time Protocol demon and the Chrony version of the Network Time Protocol (NTP). You will have to change the IP addresses to set up the time for your network.

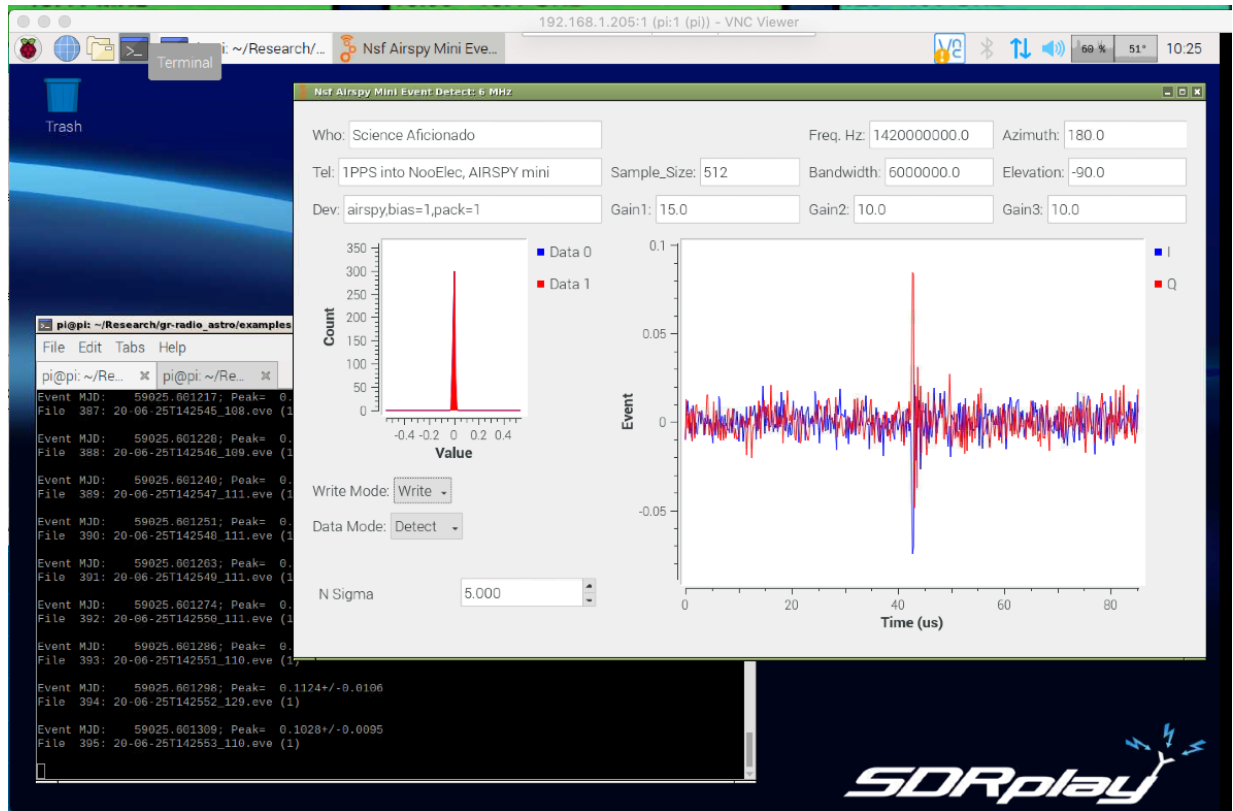


Figure 7: Image of the OS with radio astronomy software running a test capture of a GPS 1 Pulse Per Second signal Event. The goal of this software release is to enable an easy introduction to Radio Astronomy and start detecting cosmic ray events.

FYI, I use “chrony” as the time control software as it is advertised to provide smoother time transitions. If I understand NTP correctly, it changes only the time, but does not correct the reference oscillator that feeds the clock. Chrony changes the time only once, then only tweaks the rate of the reference oscillator, to keep the time tracking accurately. Note that I have not yet been able to confirm a significant difference between time accuracy between using the “ntpd” demon and the “chrony” demon.

Event Detection

It is our intention that everyone can become a *Science Aficionado*, by joining a world-wide network of radio telescopes that monitor the sky for cosmic ray flashes. A major part of our software work has been perfecting detection of very short duration (millionths of a second) bright radio flashes when cosmic rays hit the Earth’s atmosphere. These cosmic rays have energies much, much higher than the highest energy particles that can be created in the world’s foremost physics laboratories.

Each *Aficionado* can make a unique contribution, because the flashes extend only a hundred meters on a side, so unless a radio telescope is at that location, the event is missed.

The Raspberry Pi 4 designs have been tested with Event capture. The test was performed by feeding a 1 Pulse-Per-Second (1PPS) output from a GPS receiver into a NooElec Neutral Hydrogen amplifier¹⁰. The early, barebones, version of this amplifier has a control port which is intended to be used for calibration. When the control signal is high, the input is grounding, causing a power change. The 1PPS is a few millions of a second long (a few microseconds). These pulses show up as flashing radio signals in the data stream. These flashes occur with great timing precision, enabling tests of the timing accuracy of the event captures software.

The event detection python programs have names beginning with "NsfDetect??.py". The last two characters are the sampler clock rate. For example the SDRplay design, sampling at 9MHz has "NsfDetect90.py". When running on a Raspberry Pi 4 the design has been tested, and is found to capture all 600 1 PPS pulses in 10 minutes.

The Raspberry Pi 3 computers are slower and only designs working at less than 3 MHz less are able to accurately capture events without losing many samples. In a 10 minute test, with a Raspberry Pi 3, running with an Airspy-mini sampling at 3.0 MHz sampling, is able to gather about 75% (484 out of 600) 1 PPS samples. With the Raspberry Pi 3 and an RTL-SDR Dongle, sampling at 2.4 MHz, in a 10 minute sampling of 1 PPS events, all 600 events were detected.

The event detection system is still under [development](#) and collaborators are welcomed.

Summary

You have the opportunity to discover the Milky Way for yourself, with the tools provided. These tools are the door to a marvelous, unseen universe. Join the collaboration and make your own contributions!

10

https://www.amazon.com/Nooelec-SAWbird-H1-Barebones-Applications/dp/B07XJLKQDN/ref=sr_1_8?dc_hlid=1&keywords=nooelec+amplifier&qid=1593445896&sr=8-8

LightWork Memos

The more detailed tests of the system are in a set of memos. You're invited to contribute to these memos:

<http://wvurail.org/lightwork>

Live Video Demo

We gave a live demo of the current horn design in an online webinar during the American Astronomical Society Meeting in June 2020. The demo shows the telescope in operation, easily finding our Milky Way Galaxy spiral arms in a few seconds.

<https://www.gb.nrao.edu/~glangsto/aas2020/Langston2020AasTelescopeWebinar.mp4>

Thanks to Katherine LaFleur for pointing the telescope at the Milky Way!