

NOTE: THE FOLLOWING PROCEDURES HAVE BEEN MODIFIED FROM A VERY USEFUL SITE - 'LEARN WITH OMAR' AT [HTTPS://OMAR2CLOUD.GITHUB.IO/RASP/REALVNC/](https://omar2cloud.github.io/rasp/realvnc/) OMAR HAS MANY USEFUL AND EDUCATIONAL TUTORIALS. I HAVE TAILORED 2 OF HIS TUTORIALS TO BE SPECIFIC TO THE PSWS DRF UBUNTU SYSTEMS AND ADDED SOME PROCEDURES OF MY OWN GLEANED FROM OTHER SOURCES ON THE INTERNET.
--- 231102 - N4TVC, CHAS (RANDY) ALLEN ---

Step 1: Setup a -free- Real VNC account

Go to RealVNC to signup (if you do not already have an account) for a free RealVNC Team Account to allow you to save up to 5 systems to be accessed and managed across the internet. The following site will let you sign up and will instruct you on installing the Real VNC Viewer on the system you wish to use (managing computer) to access your PSWS DRF Ubuntu system:

<https://manage.realvnc.com/>

Step 2: Install RealVNC Server on Ubuntu 22.04

Stop execution of the system - On an operational PSWS DRF system, bring the terminal window to the front (click in the terminal window) and do a <Ctrl-c> to halt gnuradio. The terminal window and plot window should close.

Download the RealVNC ARM64 package - Open a new terminal window and enter the following command:

```
wget https://archive.raspberrypi.org/debian/pool/main/r/realvnc-vnc/realvnc-vnc-server_6.7.2.43081_arm64.deb
```

You will get a listing of the download process in your terminal window...

Install the package using the dpkg command - once your prompt returns, enter the command:

```
sudo dpkg -i realvnc-vnc-server_6.7.2.43081_arm64.deb
```

When that install finishes,

Add specific links to files to /usr/lib/aarch64-linux-gnu directory - cd into the /usr/lib directory by entering the following command:

```
cd /usr/lib/aarch64-linux-gnu
```

Now add the following 10 links to this directory (NOTE: Many of these will return no such file/directory... disregard that and try the next one... do all 10, just in case):

```
sudo ln libvcos.so /usr/lib/libvcos.so.0  
sudo ln libvchiq_arm.so /usr/lib/libvchiq_arm.so.0
```

```
sudo ln libbcm_host.so /usr/lib/libbcm_host.so.0
sudo ln libmmal.so /usr/lib/libmmal.so.0
sudo ln libmmal_core.so /usr/lib/libmmal_core.so.0
sudo ln libmmal_components.so /usr/lib/libmmal_components.so.0
sudo ln libmmal_util.so /usr/lib/libmmal_util.so.0
sudo ln libmmal_vc_client.so /usr/lib/libmmal_vc_client.so.0
sudo ln libvcsml.so /usr/lib/libvcsml.so.0
sudo ln libcontainers.so /usr/lib/libcontainers.so.0
```

Finally, lets enable and start the following services - Enter the following commands (NOTE: any error response indicates a typo or format error in the command. Fix any errors before continuing. The first 2 enabled services will return a message that says 'created symlink...' The 2 started services will simply return to the prompt.

```
sudo systemctl enable vncserver-virtuald.service
sudo systemctl enable vncserver-x11-serviced.service
sudo systemctl start vncserver-virtuald.service
sudo systemctl start vncserver-x11-serviced.service
```

Now reboot the system

```
sudo reboot
```

When the system finishes booting up, bring the terminal window to the front (click in the terminal window) and do a <Ctrl-c> to halt gnuradio. The terminal window and plot window should close.

Step 3: Install the lightdm Display Manager

The RealVNC server dialog is exposed to the desktop using the Display Manager.

Open a new Terminal window and follow these steps to install lightdm:

```
sudo apt-get install lightdm
```

This installation will ask if you wish to continue, respond with a 'Y' (just the single letter, not the accents). An information dialog will pop up, select OK (simply hit <enter> as it is already highlighted). Next it will ask how you wish to startup, select 'lightdm' and <enter>. After this install finishes and the prompt returns, reboot the system:

```
sudo reboot
```

Step 4: Setup Autologin

When the system reboots, you will see a 'login' dialog on the desktop. Enter the system password 'HamSCI2023!' unless you have changed it to something else.

After entering the password, you should see the RealVNC dialog box. Don't be concerned if this dialog box disappears - we'll bring it back later.

When the system finishes booting up, bring the terminal window to the front (click in the terminal window) and do a <Ctrl-c> to halt gnuradio. The terminal window and plot window should close.

We will next fix the startup files so autologin works - Making the login dialog unnecessary for unattended restarts (like on a power failure and restoration).

Open a new Terminal window.

Create this file in the lightdm directory by executing the following:

```
sudo nano /etc/lightdm/lightdm.conf
```

Enter the following exactly as presented here into this new file:

```
[SeatDefaults]
autologin-guest=false
autologin-user=ubuntu
autologin-user-timeout=0
autologin-session=lightdm-autologin
user-session=ubuntu
greeter-session=unity-greeter
```

Now do a <Ctrl-o> and then a <return> to write the file out. It will inform you of how many lines it wrote out (8 with a blank line at the bottom or 7 without that blank line - either is OK), then <Ctrl-x> to exit the nano editor back to the system prompt.

If you want to check your entries, execute the following command:

```
cat /etc/lightdm/lightdm.conf
```

This will list the file contents and you can compare to the above.

Next, Add the user 'ubuntu' to the 'nopasswdlogin' group - By entering the following:

```
sudo gpasswd -a ubuntu nopasswdlogin
```

You should get a message back saying user 'ubuntu' has been added.

Reboot the system and note that the login dialog no longer appears on the desktop when the system restarts.

```
sudo reboot
```

Step 5: Add this PSWS DRF system to your Team

Wait until the system has come up completely and then bring the terminal window to the front (click in the terminal window) and do a <Ctrl-c> to halt gnuradio. The terminal window and plot window should close.

You should now see a small icon on the top/right side of the desktop window. It is a blue square and has 'VNC' in it. Click on this icon.

Log in to your Real VNC account - The VNC Server dialog window should appear. The third item on the left side of this dialog is 'Sign In'. Click this link and login to your RealVNC account.

The resultant dialog will list, at the bottom, a place to name this computer in your Team. Enter a meaningful name here so you can find it easily. Spaces are permitted. Upon entering the name and continuing, it will ask for the password of the ubuntu computer. Enter 'HamSCI2023!' or whatever you use if you changed the password on your system.

Your system will now be listed in your Team.

You can now launch RealVNC Viewer on your 'managing computer' and in the dialog field at the top, enter the name of your ubuntu PSWS DRF computer. The viewer will search and return your PSWS DRF system.

Double click on the found system and at your option, you can save your system user name and password to automatically sign into this computer from VNC Viewer each time. The user name is 'ubuntu' and the password, again, is 'HamSCI2023!'.

You should now see the desktop of your PSWS DRF system and you can now control your system from your managing computer.

Step 6: Boot 'Headless' or to a Locally Connected HDMI Monitor.

You can now choose to boot to the locally connected Keyboard/Mouse/Monitor or boot headless (meaning with no Keyboard/Mouse/Monitor connected to your Raspberry Pi). Either way, you will still be able to connect across the Internet to your PSWS DRF system with RealVNC.

If you wish to continue booting to locally connected Keyboard/Mouse/Monitor, and also have connectivity via RealVNC Viewer, you are finished.

Should you wish to disconnect your Keyboard/Mouse/Monitor and run your PSWS DRF system 'headless', we will need to tell the ubuntu operating system that we have no monitor connected, otherwise it will hang on boot with no monitor connected. To do this, complete the following

steps either from your locally connected Keyboard/Mouse/Monitor OR from your RealVNC Viewer on your managing computer:

This should already be closed, but just in case...Bring the terminal window to the front (click in the terminal window) and do a <Ctrl-c> to halt gnuradio. The terminal window and plot window should close.

This will allow you to start with no Keyboard/Mouse or HDMI Monitor attached and simply use the RealVNC Viewer to access the system. To do so, edit the following file and note that one line in red gets commented out with a '#' and 3 lines in red get added...

```
sudo nano /boot/firmware/config.txt
```

```
[all]
```

```
kernel=vmlinuz  
cmdline=cmdline.txt  
initramfs initrd.img followkernel
```

```
[pi4]
```

```
max_framebuffers=2  
arm_boost=1
```

```
[all]
```

```
# Enable the audio output, I2C and SPI interfaces on the GPIO header. As these  
# parameters related to the base device-tree they must appear *before* any  
# other dtoverlay= specification  
dtparam=audio=on  
dtparam=i2c_arm=on  
dtparam=spi=on
```

```
# Comment out the following line if the edges of the desktop appear outside  
# the edges of your display  
disable_overscan=1
```

```
# If you have issues with audio, you may try uncommenting the following line  
# which forces the HDMI output into HDMI mode instead of DVI (which doesn't  
# support audio output)  
#hdmi_drive=2
```

```
[cm4]
```

```
# Enable the USB2 outputs on the IO board (assuming your CM4 is plugged into  
# such a board)  
dtoverlay=dwc2,dr_mode=host
```

```
[all]
```

```
# Enable the KMS ("full" KMS) graphics overlay, leaving GPU memory as the  
# default (the kernel is in control of graphics memory with full KMS)  
#dtoverlay=vc4-kms-v3d  
hdmi_force_hotplug=1  
framebuffer_width=1920  
framebuffer_height=1080
```

```
# Autoload overlays for any recognized cameras or displays that are attached  
# to the CSI/DSI ports. Please note this is for libcamera support, *not* for
```

```
# the legacy camera stack
camera_auto_detect=1
display_auto_detect=1
```

```
# Config settings specific to arm64
arm_64bit=1
dtoverlay=dwc2
```

<Ctrl-o> and <Return> to write the file back out and then <Ctrl-x> to exit the nano editor.

Now restart the system (without a monitor/keyboard/ or mouse attached).

```
sudo reboot
```

Wait a minute, or so and then connect via RealVNC and verify the system has started and runs normally.

NOTE: You can play with the following fields in the boot config file to get the screen resolution you want in the RealVNC window:

```
framebuffer_width=1920
framebuffer_height=1080
```

You are finished. Enjoy remote connectivity to your PSWS DRF system from anywhere and save the Keyboard/Mouse/Monitor for your next project.