TERRARAD TECH

Arctic Version Portable L-band Radiometer (PoLRa) v3.x Series:

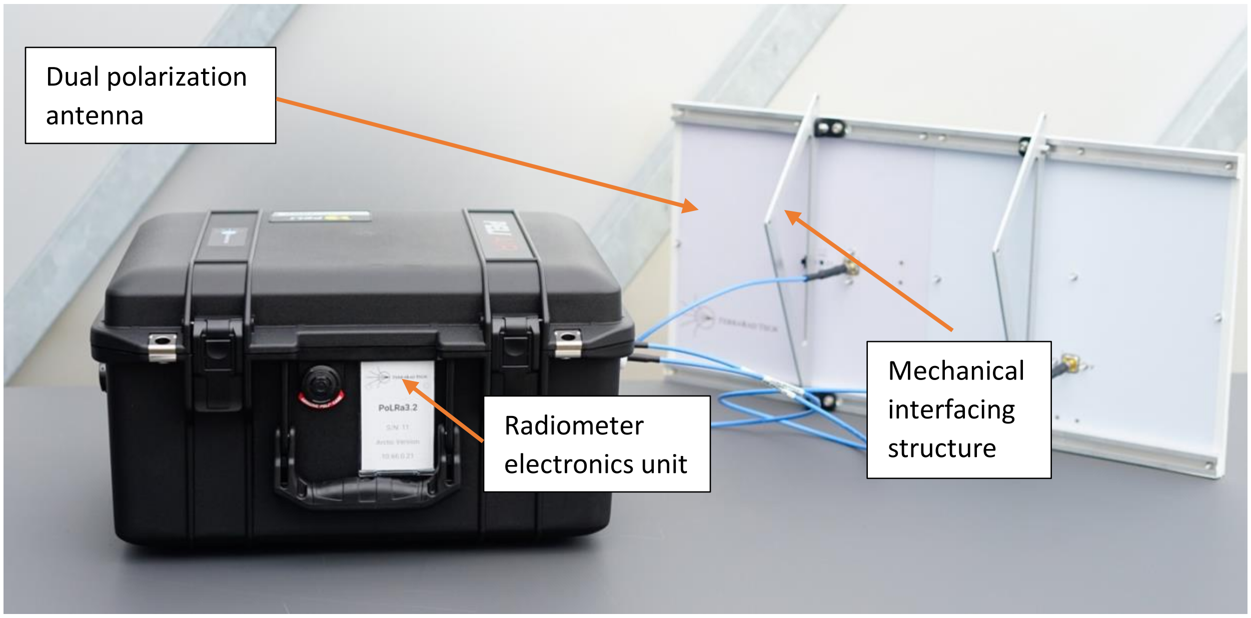
Users’ Manual

Version 1.0 Modified 6 November 2022

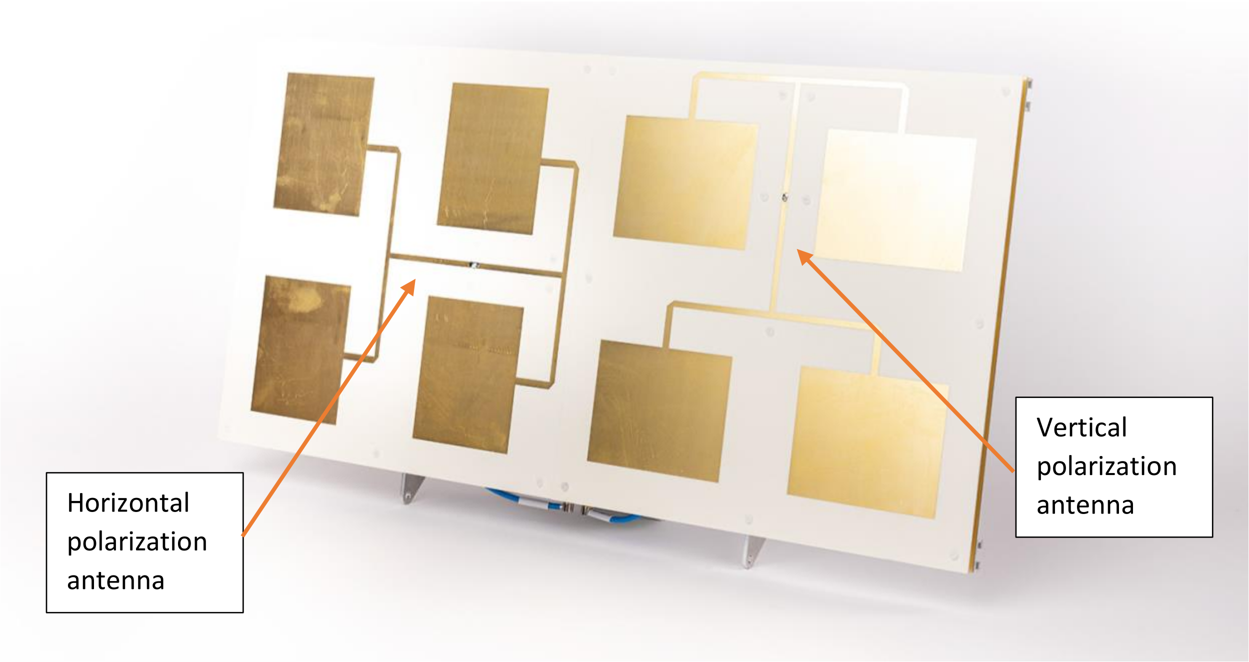
Table of Contents

I. About PoLRa3.x Series

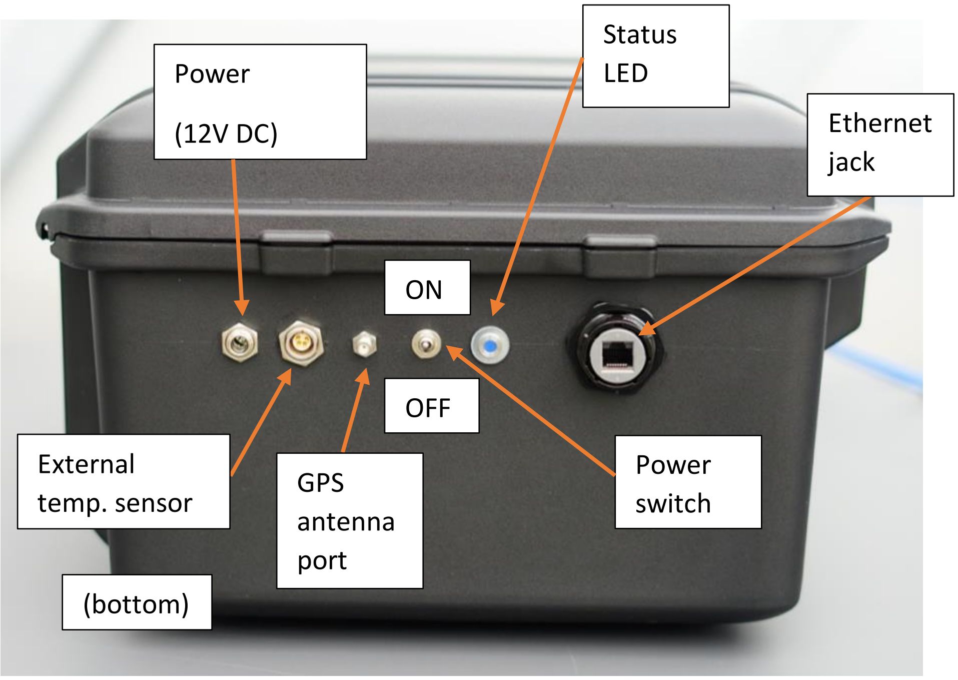
1) The PoLRa unit consists of the radiometer electronics unit, dual polarization antenna, and mechanical interfacing structure. The RF cables

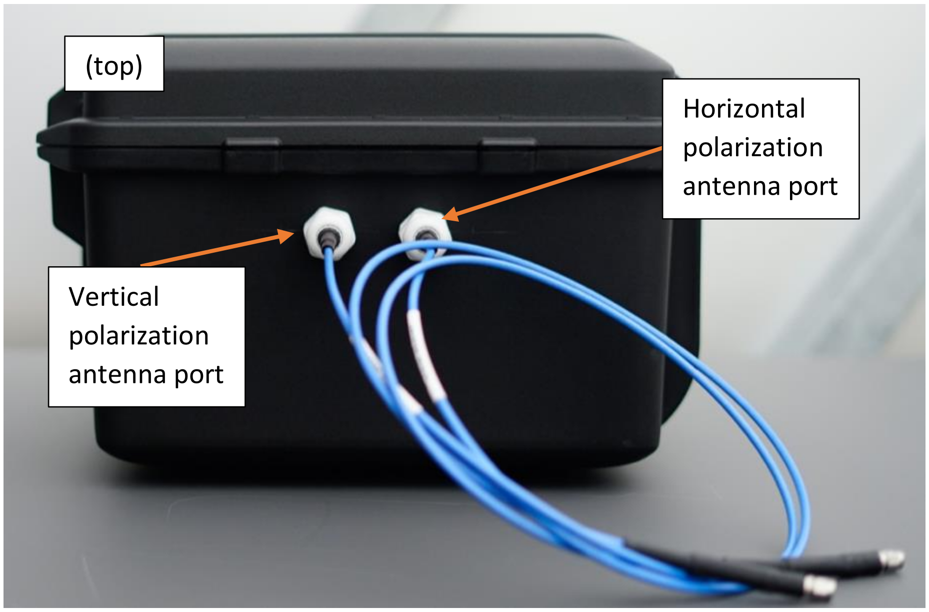

Dual polarization antenna
Mechanical interfacing structure
Radiometer electronics unit
PoLRa3.2

2) The two antenna polarizations are shown below.


Vertical polarization antenna
Horizontal polarization antenna

3) The electronics unit has 6 connection points, 4 on the bottom and 2 on the front. The temperature sensor is a push pull connector. Align the Red Dots and push the sensor in. The power connector will make a slight clicking noise when properly plugged.




(top)
Horizontal polarization antenna port
Vertical polarization antenna port

The Arctic Version comes with two power options, and the source is controlled by a switch inside the Peli enclosure. “Ext.” position will enable the external power option, while “Batt.” Position enables the internal XT30 connector which can be optionally connected to a 3S LiPo battery placed inside the Peli case.

Included cables are: 12V AC/DC Power adapter and cable with weatherproof connector, weatherproof ethernet cable plug, magnetic GPS antenna, Ethernet connector cap (to maintain weather-proofness when ethernet cable not connected) and a spare unsoldered weatherproof 12V connector.

II. How to Perform a Measurement with the PoLRa

1) The Portable L-band radiometer will power on when powered with 12 VDC.
2) Connect via SSH (Follow Section III and IV) and enable desired "crontab" for the preferred measurement type.

When the "@reboot" polra_main crontab is enabled, the system will automatically start a "vehicle-based" measurement on startup.

a) The blue LED will blink (after ~10 seconds from power-on) while the GPS unit waits for lock. A clear view of the sky is needed to obtain GPS lock
b) The instrument will automatically sync its clock to GPS time and the blue LED will go solid. This indicates that data is being collected.
c) You may begin your measurements. The sensor and GPS saves 14 samples per second as long as the blue LED is solid and it is continuously powered.
d) After collecting data, download the datafiles (.dat, and gps.csv) from the PoLRa using your sftp client, as described in section V.

Important: Before starting measurements confirm that the blue LED is solid blue. This indicates that data is being collected.

When the */5 run_g_measurement.sh crontab is enabled the system is set for tower-based interval measurements and will take 30 subsamples every 5 minutes (or 1 minute for */1 setting).

III. Setup: Setting a Static IP address on your PC

Parts and Software Required:

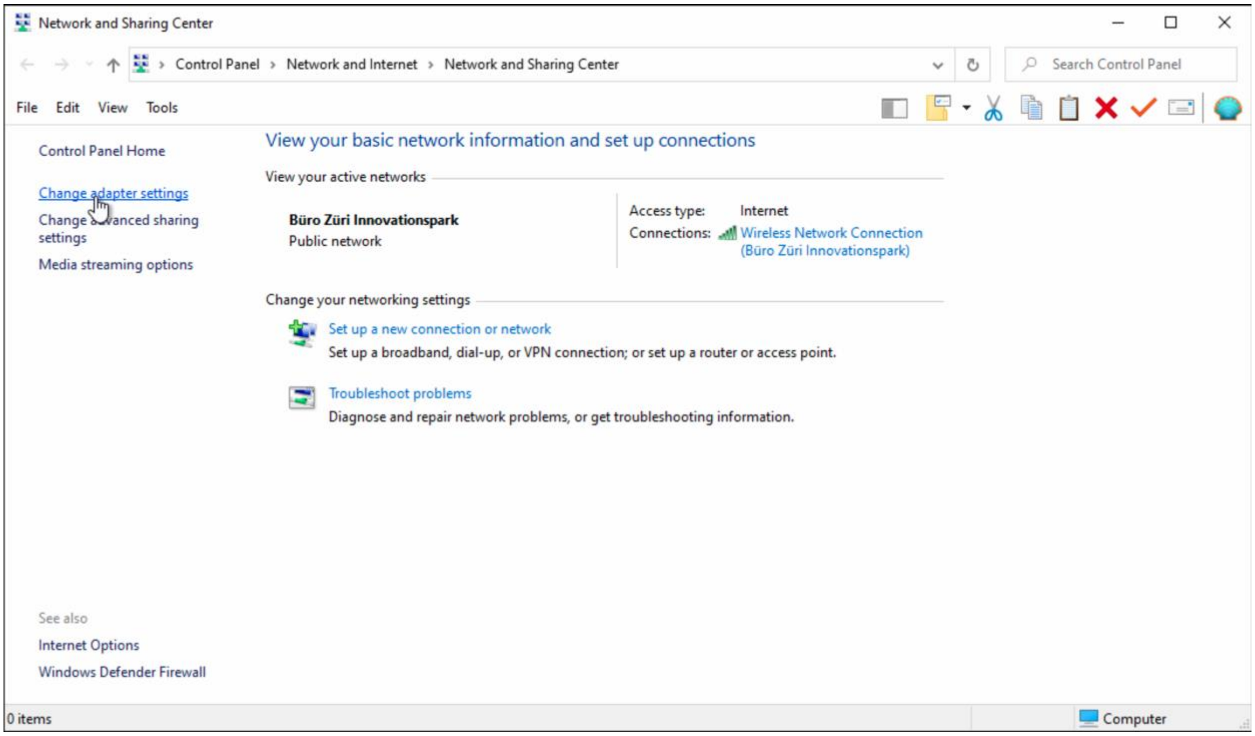
- Computer with Ethernet jack (many modern computers no longer have this and thus a USB to Ethernet dongle is required
- Administrator rights to your computer in order to make changes to the network settings

Windows:

Through the start icon navigate to:

Control Panel\Network and Internet\Network and Sharing Center

Select from the left hand side “Change Adapter Settings”



Network and Sharing Center
Control Panel > Network and Internet > Network and Sharing Center
Search Control Panel
File
Edit
View
Tools
Control Panel Home
Change adapter settings
Change advanced sharing settings
Media streaming options
See also
Internet Options
Windows Defender Firewall
0 items
Computer

View your basic network information and set up connections

View your active networks

Büro Züri Innovationspark
Public network

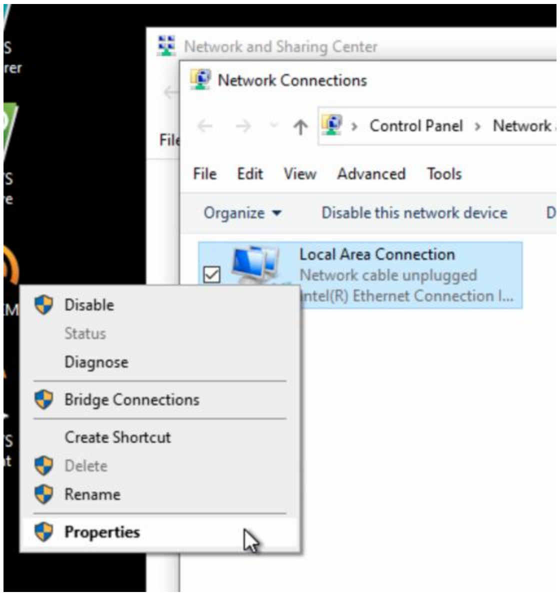
Access type: Internet
Connections: Wireless Network Connection (Büro Züri Innovationspark)

Change your networking settings

Set up a new connection or network
Set up a broadband, dial-up, or VPN connection; or set up a router or access point.

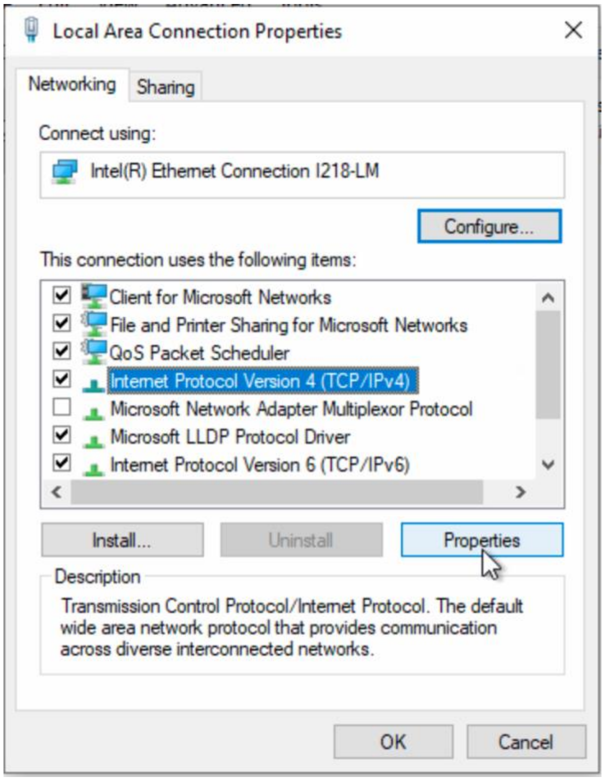
Troubleshoot problems
Diagnose and repair network problems, or get troubleshooting information.

Right click on your ethernet adapter and select Properties:



Local Area Connection
Network cable unplugged
Intel(R) Ethernet Connection I...

Navigate to “Internet Protocol Version 4” and again select “Properties”



Connect using:

Intel(R) Ethernet Connection I218-LM

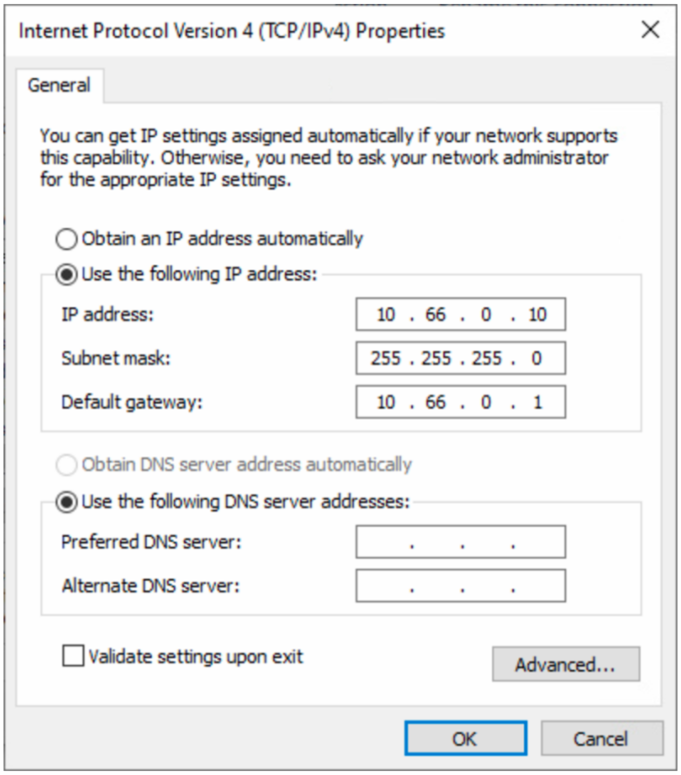
This connection uses the following items:

- ☑ Client for Microsoft Networks
- ☑ File and Printer Sharing for Microsoft Networks
- ☑ QoS Packet Scheduler
- ☑ Internet Protocol Version 4 (TCP/IPv4)
- □ Microsoft Network Adapter Multiplexor Protocol
- ☑ Microsoft LLDP Protocol Driver
- ☑ Internet Protocol Version 6 (TCP/IPv6)

Description

Transmission Control Protocol/Internet Protocol. The default wide area network protocol that provides communication across diverse interconnected networks.

Set the manual static IP address similar to below:



You can get IP settings assigned automatically if your network supports this capability. Otherwise, you need to ask your network administrator for the appropriate IP settings.

○ Obtain an IP address automatically

● Use the following IP address:

IP address: 10 . 66 . 0 . 10

Subnet mask: 255 . 255 . 255 . 0

Default gateway: 10 . 66 . 0 . 1

○ Obtain DNS server address automatically

● Use the following DNS server addresses:

Preferred DNS server: . . .

Alternate DNS server: . . .

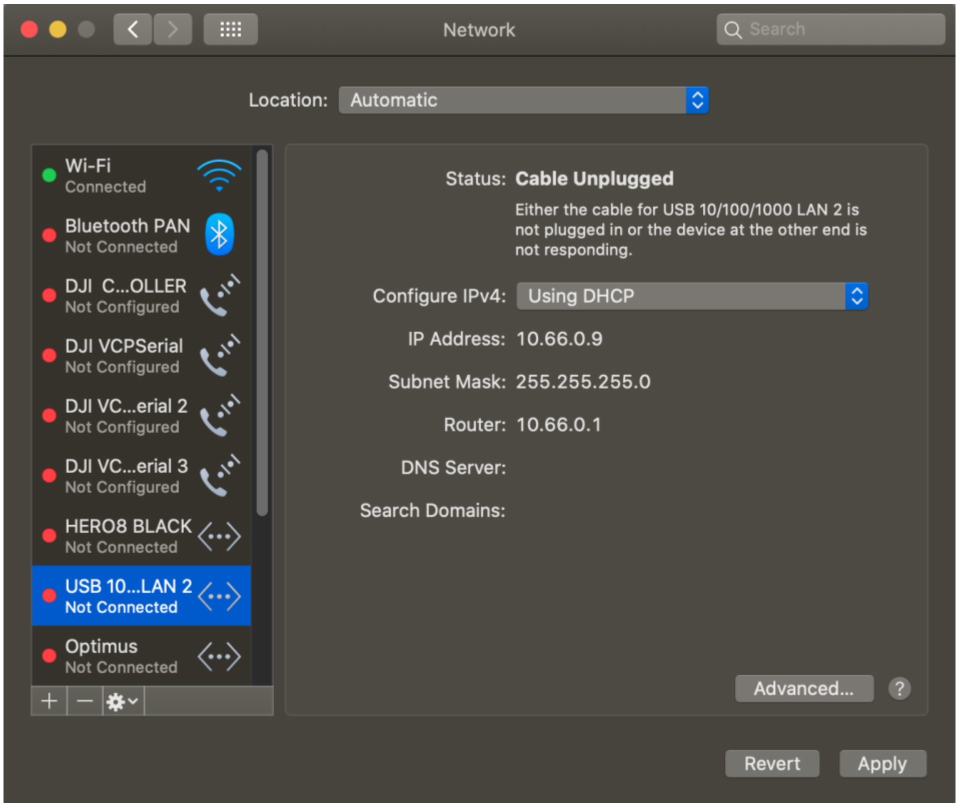
□ Validate settings upon exit

Select “OK”

Apple / OSX:

Open System Preferences and select “Network”

Select your LAN ethernet port or adapter (USB LAN or something similar) and select “Advanced”



Location: Automatic

Status: **Cable Unplugged**

Either the cable for USB 10/100/1000 LAN 2 is not plugged in or the device at the other end is not responding.

Configure IPv4: Using DHCP

IP Address: 10.66.0.9

Subnet Mask: 255.255.255.0

Router: 10.66.0.1

DNS Server:

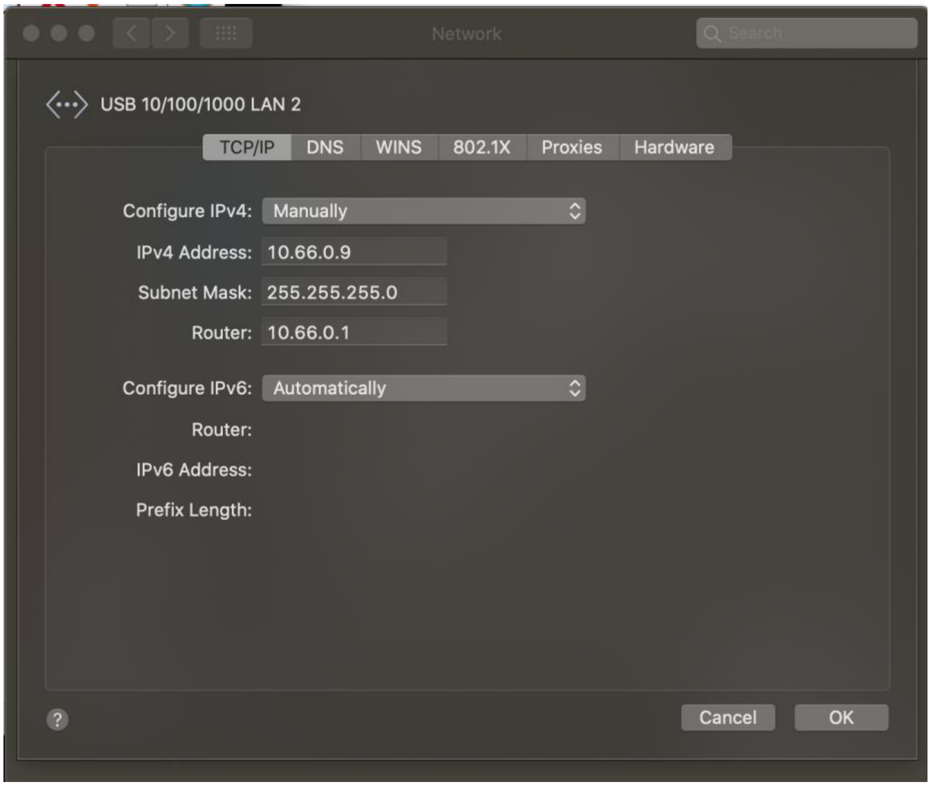
Search Domains:

From “Configure IPv4” dropdown select “Manually”.

Under IPv4 Address enter: 10.66.0.9

Subnet Mask: 255.255.255.0

Router: Doesn’t matter as we connect directly. 10.66.0.1 is fine.



USB 10/100/1000 LAN 2

Configure IPv4: Manually

IPv4 Address: 10.66.0.9

Subnet Mask: 255.255.255.0

Router: 10.66.0.1

Configure IPv6: Automatically

Router:

IPv6 Address:

Prefix Length:

Select “OK”

Select “Apply”

For All Operating Systems:

WiFi should be turned off so that your computer only looks for connections on the Ethernet port

Note: if you do not revert these changes back, for the ethernet adapter you are using, the settings will remain and you only need to perform this step once.

IV. Communicating with the Instrument (ssh)

There are two main methods for connecting and communicating with the PoLRa. The newest firmware version (after July 2022) does not require a computer in the field because it automatically starts the time sync and data collection when the system is powered on. Communicating is only needed for downloading data and changing settings on the internal computer.

Data download is done most-easily with an sftp client such as FileZilla or WinSCP. Changing settings should be done over ssh terminal protocol.

SSH for Changing Settings on the Internal Computer

1) Connect ethernet cable directly from laptop to PoLRa3.x unit. The ground-based version has included waterproof ethernet cable with Amphenol connector. When the ethernet cable is aligned with the port the connector will push and turn right and snap onto the receptacle on the PoLRa unit.

2) The PoLRa3.x does not have a power switch, it will turn on automatically when powered on.

3) Using Terminal on Mac or Linux, or PuTTy on Windows connect to the PoLRa using ssh.

Apple / OSX:

Enter the following command into "terminal":

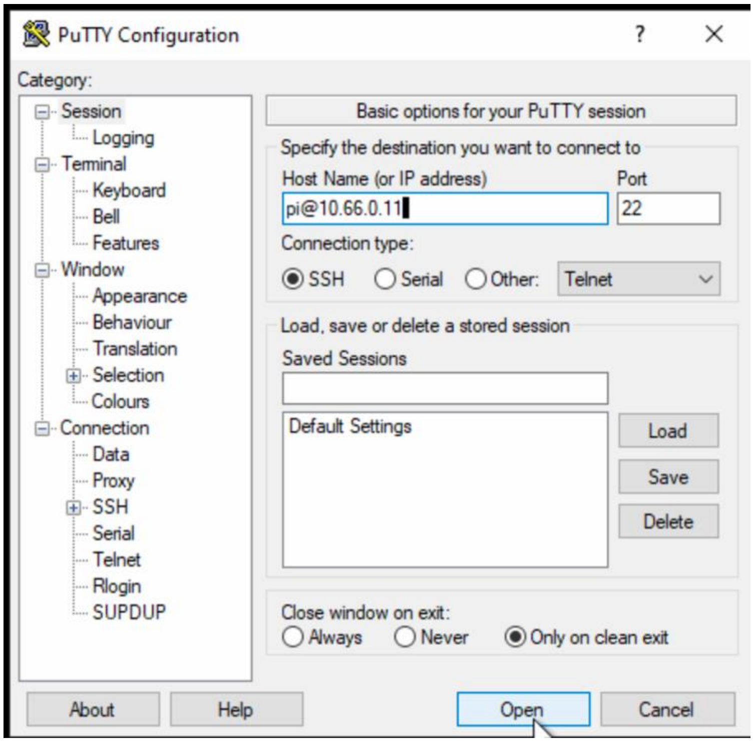
```
ssh pi@10.66.0.xx
```

Where `xx` is found on the sticker on the front of your particular PoLRa unit.

Enter the password: `microwave`

Windows:

In PuTTy window, Type `pi@10.66.0.xx` into the connection Host Name (Where `xx` is found on the sticker on the front of your particular PoLRa unit) and click "Open"



Specify the destination you want to connect to

Host Name (or IP address): pi@10.66.0.11

Port: 22

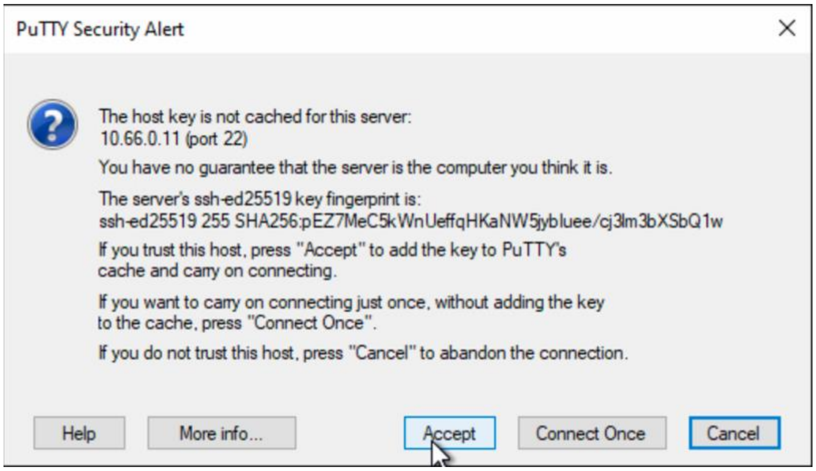
Connection type: ● SSH ○ Serial ○ Other: Telnet

Saved Sessions

Default Settings

Close window on exit: ○ Always ○ Never ● Only on clean exit

Accept security alert



The host key is not cached for this server:
10.66.0.11 (port 22)

You have no guarantee that the server is the computer you think it is.

The server's ssh-ed25519 key fingerprint is:
ssh-ed25519 255 SHA256:pEZ7MeC5kWnUeffqHKaNW5jybluee/cj3lm3bXSbQ1w

If you trust this host, press "Accept" to add the key to PuTTY's cache and carry on connecting.

If you want to carry on connecting just once, without adding the key to the cache, press "Connect Once".

If you do not trust this host, press "Cancel" to abandon the connection.

Enter the password: `microwave`

4) Setting the crontab for desired measurement type.

At the terminal prompt, type the command `crontab -e`

You will get the following screen. Use the down array to move down to the last 4 lines. Uncomment the line(s) for the desired measurement type.

For vehicle-based measurement uncomment the following line:

```
@reboot sleep 10 && /home/pi/software3.1/polra_main >> /home/pi/data/measurementlog.txt 2>&1
```

The PoLRa will now automatically start measurements upon power-up and GPS-lock.

For timed interval measurements (tower or station-based) uncomment the following lines:

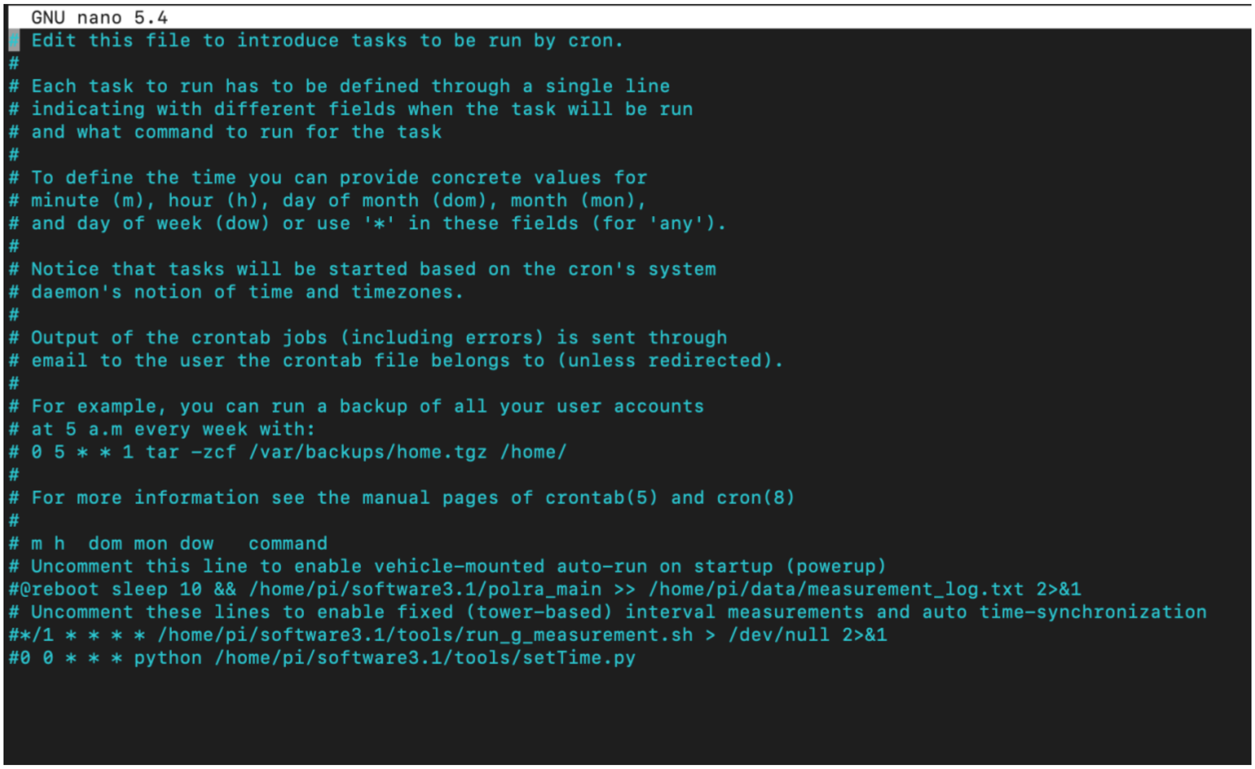
```
*/5 * * * * /home/pi/software3.1/tools/run_g_measurement.sh > /dev/null 2>&1

0 0 * * * python /home/pi/software3.1/tools/setTime.py
```

Additionally, to make sure that the time is synchronized when interval measurements begin, add the following line to the crontab:

```
@reboot sleep 60 && python /home/pi/software3.1/tools/setTime.py >> /home/pi/data/gps_log.txt 2>&1
```

The PoLRa will now take 30 samples every 5 minutes (the screenshot shows settings for every minute: */1).



```
  GNU nano 5.4
# Edit this file to introduce tasks to be run by cron.
#
# Each task to run has to be defined through a single line
# indicating with different fields when the task will be run
# and what command to run for the task
#
# To define the time you can provide concrete values for
# minute (m), hour (h), day of month (dom), month (mon),
# and day of week (dow) or use '*' in these fields (for 'any').
#
# Notice that tasks will be started based on the cron's system
# daemon's notion of time and timezones.
#
# Output of the crontab jobs (including errors) is sent through
# email to the user the crontab file belongs to (unless redirected).
#
# For example, you can run a backup of all your user accounts
# at 5 a.m every week with:
# 0 5 * * 1 tar -zcf /var/backups/home.tgz /home/
#
# For more information see the manual pages of crontab(5) and cron(8)
#
# m h  dom mon dow   command
# Uncomment this line to enable vehicle-mounted auto-run on startup (powerup)
#@reboot sleep 10 && /home/pi/software3.1/polra_main >> /home/pi/data/measurement_log.txt 2>&1
# Uncomment these lines to enable fixed (tower-based) interval measurements and auto time-synchronization
#*/1 * * * * /home/pi/software3.1/tools/run_g_measurement.sh > /dev/null 2>&1
#0 0 * * * python /home/pi/software3.1/tools/setTime.py
```

The time will also synchronize every day at midnight to fix any small drift of the internal system clock.

The data will be saved with one file per hour in the following data structure:

./data/year/month/POLRA3-YYYYMMDD/POLRA3GYYYYMMDD_HH.dat

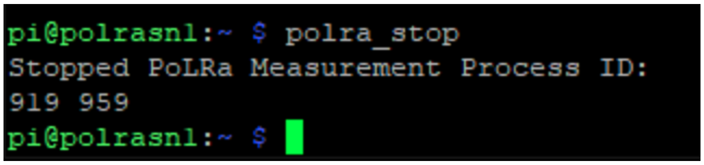
Where HH is hour on a 24 hour clock.

5) First time Setup: This only needs to be performed when first setting up the PoLRa.

- Type command: `sudo raspi-config`
- Navigate with the down array to `'4 Internationalisation Options'` and press <Enter>.
- Highlight `'2 Change timezone'` and hit <Enter>
- Follow the instructions to select your country and change your local timezone.
- When done use Tab to move to `'Finish'` and press <Enter>
- You may be prompted to restart when finished. This will close your ssh session and you will need to login again in about 30 seconds when the reboot is complete.

6) If you want to stop the automatically run vehicle-based data-collection process (e.g. if you want to connect inside to the unit to download data etc.). Run the command `polra_stop` to cleanly stop the data collection processes. This will stop the blinking blue LED but the unit will remain

powered. When power is removed and turned back on the measurement procedure will start again.



```
pi@polrasn1:~ $ polra_stop
Stopped PoLRa Measurement Process ID:
919 959
pi@polrasn1:~ $
```

Note: process IDs will vary.

V. Using FileZilla (or WinSCP etc.) to Browse Files on PoLRa and download data

1) Open FileZilla (or preferred sftp client).

Connect to PoLRa with FileZilla using `sftp://10.66.0.xx`

Username: `pi`

Password: `microwave`

Port: `22`

Where `xx` is found on the sticker on the front of your particular PoLRa unit

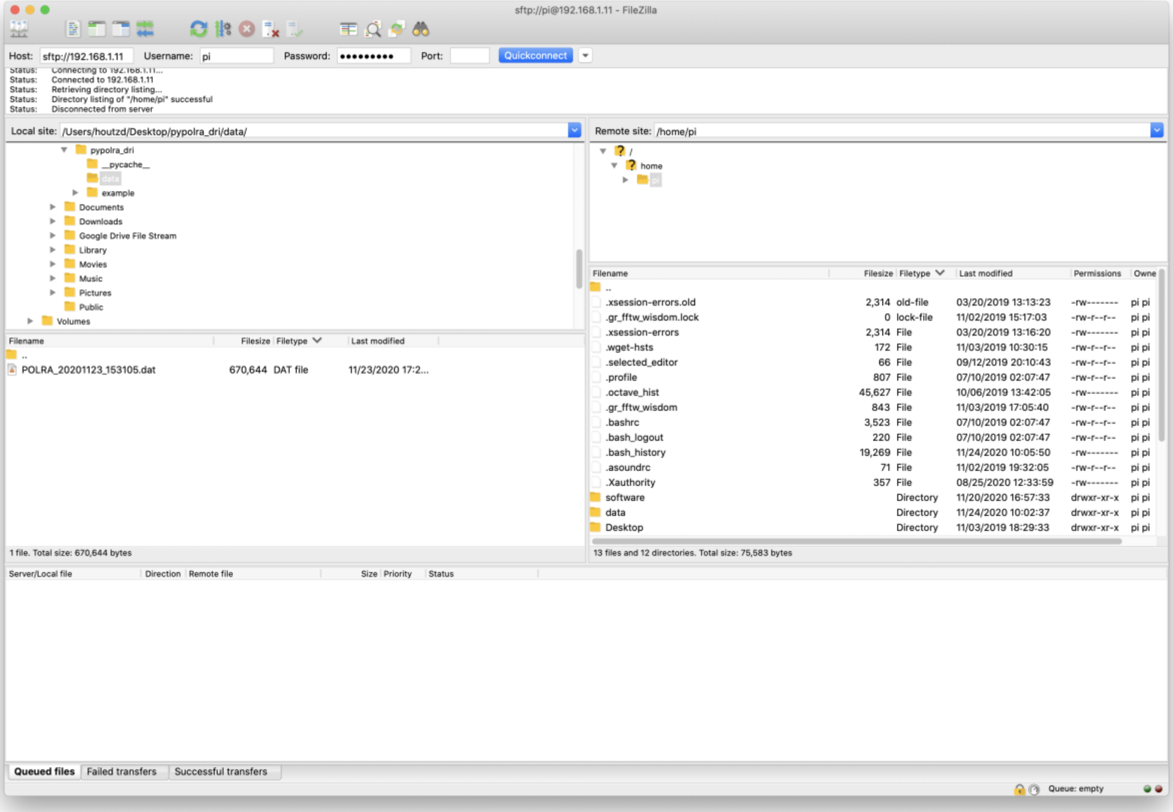
The data files will be saved in the directory `/home/pi/data`. FileZilla automatically opens the directory `/home/pi` also known as the “home directory”.

The data is saved with the naming structure: `‘POLRA3_yyyymmdd_hhmmss.dat’`

And the corresponding GPS file is “ `‘POLRA3_yyyymmdd_hhmmss_p31gps.csv’`

The data files are readable in any text editor. We recommend downloading and using Notepad++ (Windows) or Atom (Mac) as it preserves fixed width columns for quick checks or data files.

2) Your local computer files will show on the left and the PoLRa on the right. Drag the data files (`‘POLRA3_yyyymmdd_hhmmss.dat’ and` “`‘POLRA3_yyyymmdd_hhmmss_p31gps.csv’)` from the right to the left to copy to your local computer.



sftp://pi@192.168.1.11 - FileZilla

Host: sftp://192.168.1.11 Username: pi Password: ••••••••• Port:

```
Status:     Connecting to 192.168.1.11...
Status:     Connected to 192.168.1.11
Status:     Retrieving directory listing...
Status:     Directory listing of "/home/pi" successful
Status:     Disconnected from server
```

Local site: /Users/houtzd/Desktop/pypolra_dri/data/

- pypolra_dri
  - __pycache__
  - data
  - example
- Documents
- Downloads
- Google Drive File Stream
- Library
- Movies
- Music
- Pictures
- Public
- Volumes

Filename	Filesize	Filetype	Last modified
..			
POLRA_20201123_153105.dat	670,644	DAT file	11/23/2020 17:2...

1 file. Total size: 670,644 bytes

Remote site: /home/pi

- /
  - home
    - pi

Filename	Filesize	Filetype	Last modified	Permissions	Owne
..					
.xsession-errors.old	2,314	old-file	03/20/2019 13:13:23	-rw-------	pi pi
.gr_fftw_wisdom.lock	0	lock-file	11/02/2019 15:17:03	-rw-r--r--	pi pi
.xsession-errors	2,314	File	03/20/2019 13:16:20	-rw-------	pi pi
.wget-hsts	172	File	11/03/2019 10:30:15	-rw-r--r--	pi pi
.selected_editor	66	File	09/12/2019 20:10:43	-rw-r--r--	pi pi
.profile	807	File	07/10/2019 02:07:47	-rw-r--r--	pi pi
.octave_hist	45,627	File	10/06/2019 13:42:05	-rw-------	pi pi
.gr_fftw_wisdom	843	File	11/03/2019 17:05:40	-rw-r--r--	pi pi
.bashrc	3,523	File	07/10/2019 02:07:47	-rw-r--r--	pi pi
.bash_logout	220	File	07/10/2019 02:07:47	-rw-r--r--	pi pi
.bash_history	19,269	File	11/24/2020 10:05:50	-rw-------	pi pi
.asoundrc	71	File	11/02/2019 19:32:05	-rw-r--r--	pi pi
.Xauthority	357	File	08/25/2020 12:33:59	-rw-------	pi pi
software		Directory	11/20/2020 16:57:33	drwxr-xr-x	pi pi
data		Directory	11/24/2020 10:02:37	drwxr-xr-x	pi pi
Desktop		Directory	11/03/2019 18:29:33	drwxr-xr-x	pi pi

13 files and 12 directories. Total size: 75,583 bytes

Server/Local file	Direction	Remote file	Size	Priority	Status

Once you have gathered the datafile from your measurement you are ready to process this data.

The PoLRa has roughly 100 GB of internal storage (hundreds of hours) so you do not need to worry about cleaning the internal memory.

VI. Vehicle-based Processing: Installation of python3 and dependencies

Parts and Software Required:

- Computer- Mac OSX/Linux/Windows
- Python 3 (version 3.5-3.8+) installation with all dependencies installed

Step 1: Installing Python3:

Processing PoLRa data requires python3 and a number of other python packages.

Recommended Installation: Anaconda

We recommend installing python3 and the required dependencies via anaconda. This will automatically package many of the required dependencies including numpy, scipy, pandas, etc, and will make it easier to install some of the dependencies, such as gdal.

Installation is also possible using “brew” on mac, and other methods on Windows, but these are not directly supported.

Anaconda on Apple / OSX:

Step 1. Download and install anaconda from here: https://www.anaconda.com/products/individual

Step 2. When the installation completes,

open Terminal (from Applications or Spotlight) and type,

```
python3
```

this should bring up the python command line prompt

```
>>
```

we can now close python. Type,

```
exit()
```

If you do not get the above prompt you may need to activate anaconda with the command `conda activate` and/or restart your computer. Anaconda installation instructions can also help you with the latest up-to-date instructions on this.

Go to the section labeled "For Both Mac and Windows"

Anaconda on Windows:

Step 1. Download and install anaconda from here:
https://www.anaconda.com/products/individual#windows

Step 2. Open the newly installed "Anaconda Navigator" program to initiate the environment. Once this loads, you can close the program.

Step 3. Open program "Anaconda Prompt (Anaconda3)" from the start menu.

Step 4. Type "python" <Enter> to check that python is correctly installed. You should see the standard python prompt of ">>>"

we can now close python. Type,

```
exit()
```

For Both Mac and Windows:

Optional: you can create a separate environment for pypolra if you regularly use anaconda and want to keep track of the dependencies. In this case, you would use the command

"conda create --name pypolra", and "conda activate pypolra"

Step 4. Start installing the dependencies required for data processing using the convenient conda command and repositories.

One by one, type the following commands into the terminal to install the dependencies:

```
conda install numpy

conda install -c conda-forge pyproj

conda install -c anaconda xarray

conda install -c conda-forge shapely

conda install -c conda-forge Descartes

conda install -c oggm oggm

conda install -c conda-forge motionless

pip3 install salem

conda install -c conda-forge gdal

pip3 install netCDF4
```

Depending on versioning and dependencies of other packages you may get messages such as “`# All requested packages already installed`”. This is not an error and you can proceed to the next package installation.

alternatively, the salem install can be replaced with “`conda install -c oggm salem`“ which may or may not work depending on versions of other packages. The above should work regardless.

Some of the packages will ask you to type “`yes`” or “`y`” and hit <enter> to confirm the usage of system resources.

Dependencies can occasionally be difficult with version updates and changes to the package managers. If you have difficulties with an import, or with installing any package, please first consult google or stack overflow and second contact TerraRad for support.

Your system should now be ready to run the PoLRa processing scripts.

VII. Vehicle-based Processing: Running polra3_proc.py and viewing the outputs

For Windows:

Step 1. Open Anaconda Prompt

Step 2. From the terminal, change the current directory to pypolra3.1 processing software

```
cd C:\\path\to\directory\pypolra3.x
```

And hit <Enter>. You can check that this worked by typing `dir` to list the files in the current directory.

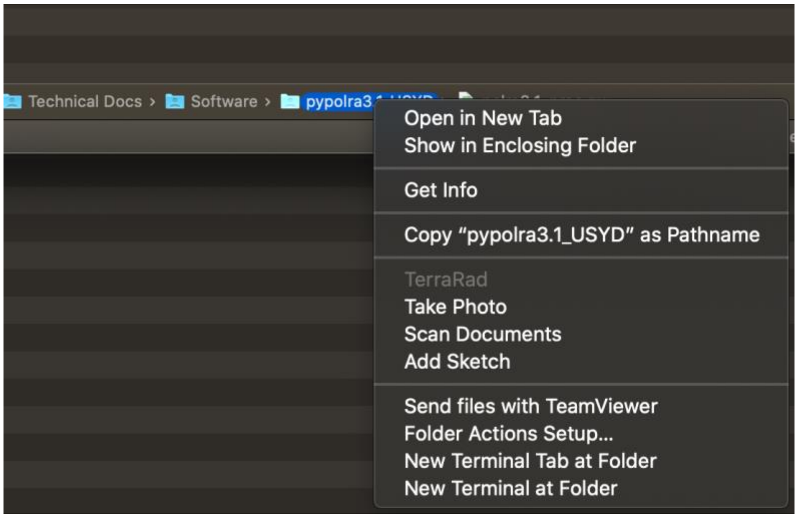
For Mac:

Step 1. Open Terminal

Step 2. Navigate to directory

```
cd /Users/username/path/to/pypolra3.x
```

Hit <Enter>. Alternatively, you can make sure that you are showing the path bar. From 'Finder' go to View>Show Path Bar. You can now right click on the path bar and click "New Terminal Tab at Folder".



For Windows and Mac:

Step 3. Edit `polra3.1_proc.py` (using Notepad++, Atom, or your favorite editor) to include the path to files, radiometer data file, and flightlog you wish to process. Currently the file is setup to process the example dataset included with the software package.

```
fpath="./data/example/"

flogfile = "2022-03-12_12-00-03_v2.csv"

radfile = "POLRA_20220312_115847.dat"
```

The explanation of variables in `polra3.1_proc.py`:

`fpath` is the directory containing the data, and `config.py` file. We recommend to create a new directory for each flight and **make sure there is a `config.py` file in this directory!** You can copy this over from the previous flights but you should modify and update soil temperature, and the retrieval type you wish to perform. In a future release we will investigate using infrared skin temperature as a soil temperature estimate, but this has not yet been validated. The PoLRa3.2 and beyond enable this through hardware.

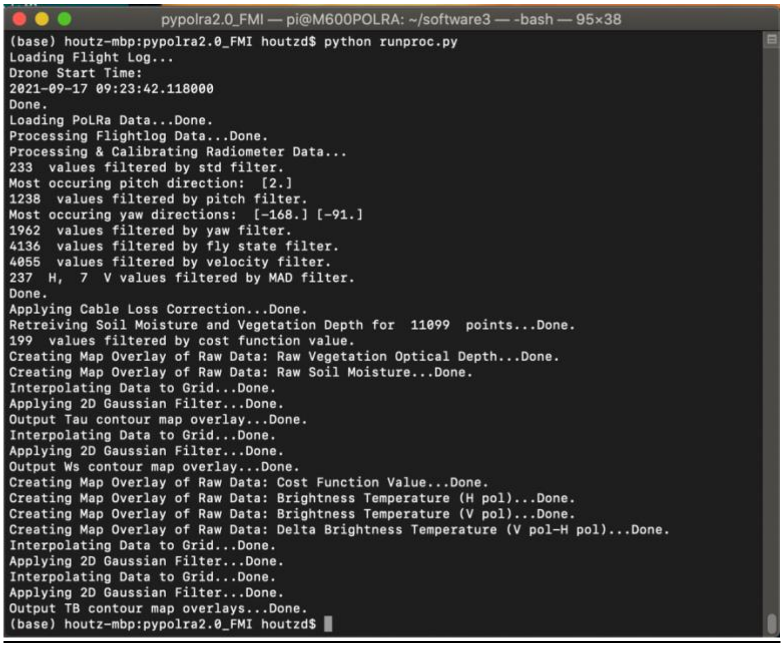
`flogfile` stands for flight-log file (inherited from the drone-based system) but refers to the GPS file containing the telemetry data.

`radfile` is the radiometer data (.dat) file

Step 3. Run the processing by typing,

`python polra3.1_proc.py` On Windows Anaconda Prompt or Mac Terminal while in the directory containing this file.

The terminal will give you many progress messages as you process the PoLRa data file.



```
(base) houtz-mbp:pypolra2.0_FMI houtzd$ python runproc.py
Loading Flight Log...
Drone Start Time:
2021-09-17 09:23:42.118000
Done.
Loading PoLRa Data...Done.
Processing Flightlog Data...Done.
Processing & Calibrating Radiometer Data...
233  values filtered by std filter.
Most occuring pitch direction:  [2.]
1238  values filtered by pitch filter.
Most occuring yaw directions:  [-168.] [-91.]
1962  values filtered by yaw filter.
4136  values filtered by fly state filter.
4055  values filtered by velocity filter.
237  H,  7  V values filtered by MAD filter.
Done.
Applying Cable Loss Correction...Done.
Retreiving Soil Moisture and Vegetation Depth for  11099  points...Done.
199  values filtered by cost function value.
Creating Map Overlay of Raw Data: Raw Vegetation Optical Depth...Done.
Creating Map Overlay of Raw Data: Raw Soil Moisture...Done.
Interpolating Data to Grid...Done.
Applying 2D Gaussian Filter...Done.
Output Tau contour map overlay...Done.
Interpolating Data to Grid...Done.
Applying 2D Gaussian Filter...Done.
Output Ws contour map overlay...Done.
Creating Map Overlay of Raw Data: Cost Function Value...Done.
Creating Map Overlay of Raw Data: Brightness Temperature (H pol)...Done.
Creating Map Overlay of Raw Data: Brightness Temperature (V pol)...Done.
Creating Map Overlay of Raw Data: Delta Brightness Temperature (V pol-H pol)...Done.
Interpolating Data to Grid...Done.
Applying 2D Gaussian Filter...Done.
Interpolating Data to Grid...Done.
Applying 2D Gaussian Filter...Done.
Output TB contour map overlays...Done.
(base) houtz-mbp:pypolra2.0_FMI houtzd$
```

The script has now created contour plots (saved as .png images) and geoTiff outputs in the

`./pypolra3.1_xxx/fpath/` directory

Based on the specific output options set in `./pypolra3.1_xxx/fpath/config.py` you will get different number of plots output as pngs/geotiff files in the fpath directory.

The comments in the provided example config file give some indication of the meaning of the different parameters. Please request a training session with TerraRad support for a demonstration.

The plots that will be output from the provided example data are from a small test drive in a citrus orchard. You will note the micro-drip irrigators are visible at each tree location, the “Raw Vegetation Optical Depth” (L-VOD or Tau) also increases at the tree locations due to the hanging canopy.

This data was taken at the University of California, Riverside in May 2022.

The below plot shows the raw soil moisture and Raw Vegetation Optical Depth output you should see from the example files:

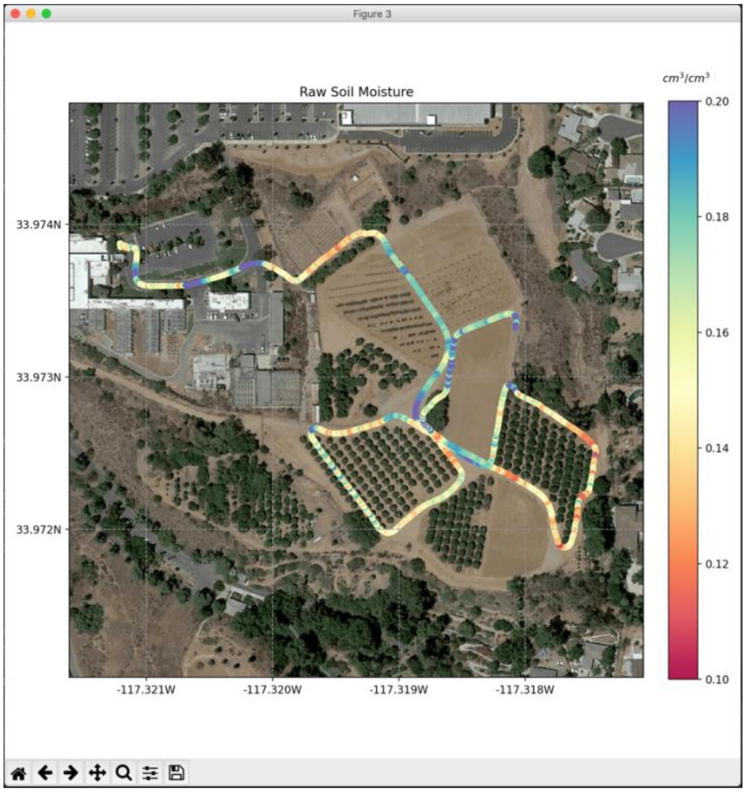


Figure 3
Raw Soil Moisture
cm^3/cm^3
0.20
0.18
0.16
0.14
0.12
0.10
33.974N
33.973N
33.972N
-117.321W
-117.320W
-117.319W
-117.318W

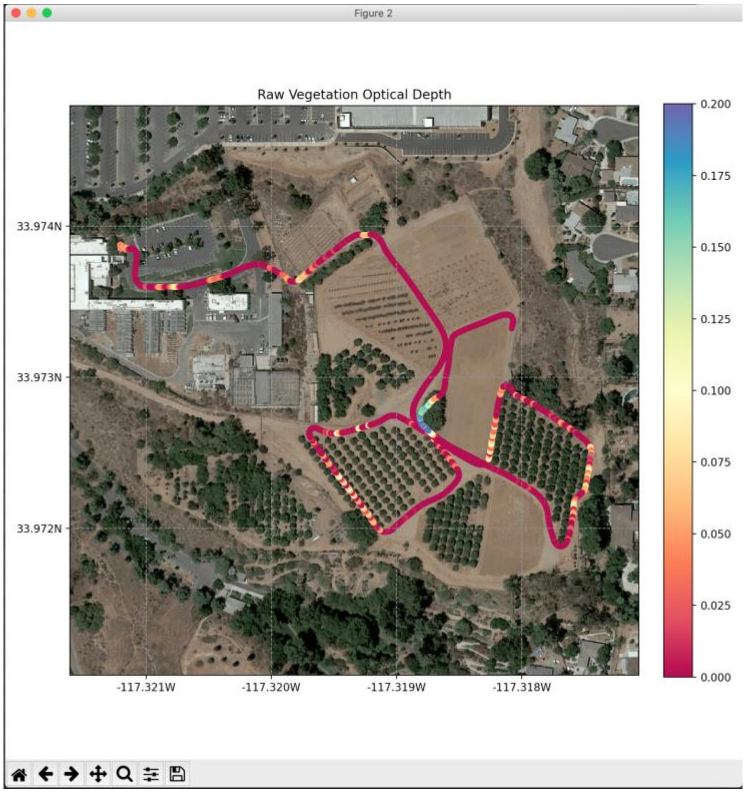
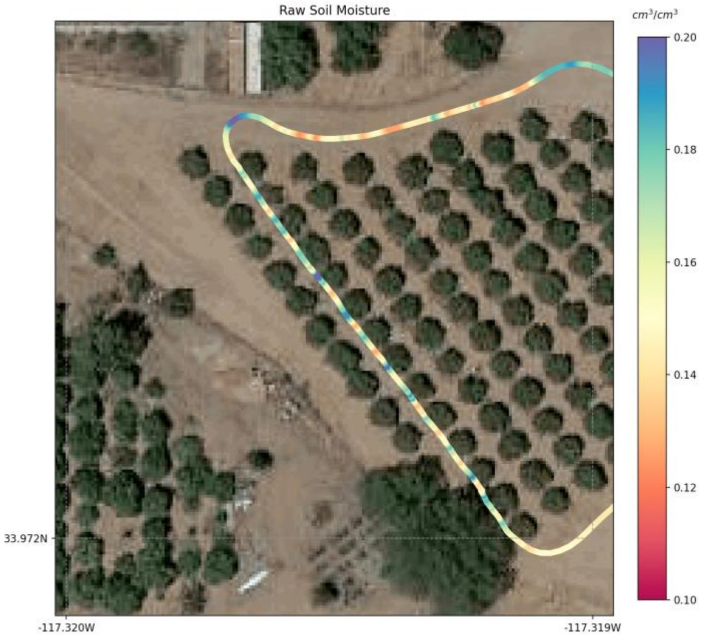


Figure 2
Raw Vegetation Optical Depth
0.200
0.175
0.150
0.125
0.100
0.075
0.050
0.025
0.000
33.974N
33.973N
33.972N
-117.321W
-117.320W
-117.319W
-117.318W

And when you Zoom in on the orchard rows:



You can prevent the script from showing you these plots every time by changing the line 6 in the config.py

```
showfigbool = ‘False’:
```

Additional handy tips for using the terminal:

`~` refers to your home directory or `C://Windows/Users/your_username/`

Or

`/Users/username/` on Mac

`.` refers to the current directory, so when we run `./polra3_flight_proc.py` this is saying run the script that exists in the current directory. This is equivalent to saying:

```
“C://Windows/Users/your_username/pypolra3.1_xxx/polra3_flight_proc.py”
```

Or

```
“~/pypolra3.1_xxx/polra3.1_flight_proc.py”
```

`..` refers to the directory one level up. If we type `cd ..` from pypolra3.1_xxx this takes us back up one level to the home directory or directory containing the main software directory.

If you ever want to cancel a command in the terminal that is taking too long or whatever other reason, pressing <CTRL> (control) and <C> will terminate the process.

What to do when you get an error message during processing:

1) The most common error is timezone errors that can occur from improper match of timezones on the PoLRa software and GPS. The GPS will always receive UTC time and therefore the software needs to make a correction.

   The simplist way to identify this is the messages on the terminal. If the "GPS file start time" and "PoLRa start time" are not the same (within ~3 seconds), you probably have this issue.

   The most "hacky" fix to get your data processed is to add a paramters "toffset = 3600*(hours offset) to the config.py file. This will add a time offset manually to the PoLRa data. We will also want to make sure that the correct timezone is set on the instrument for future runs.

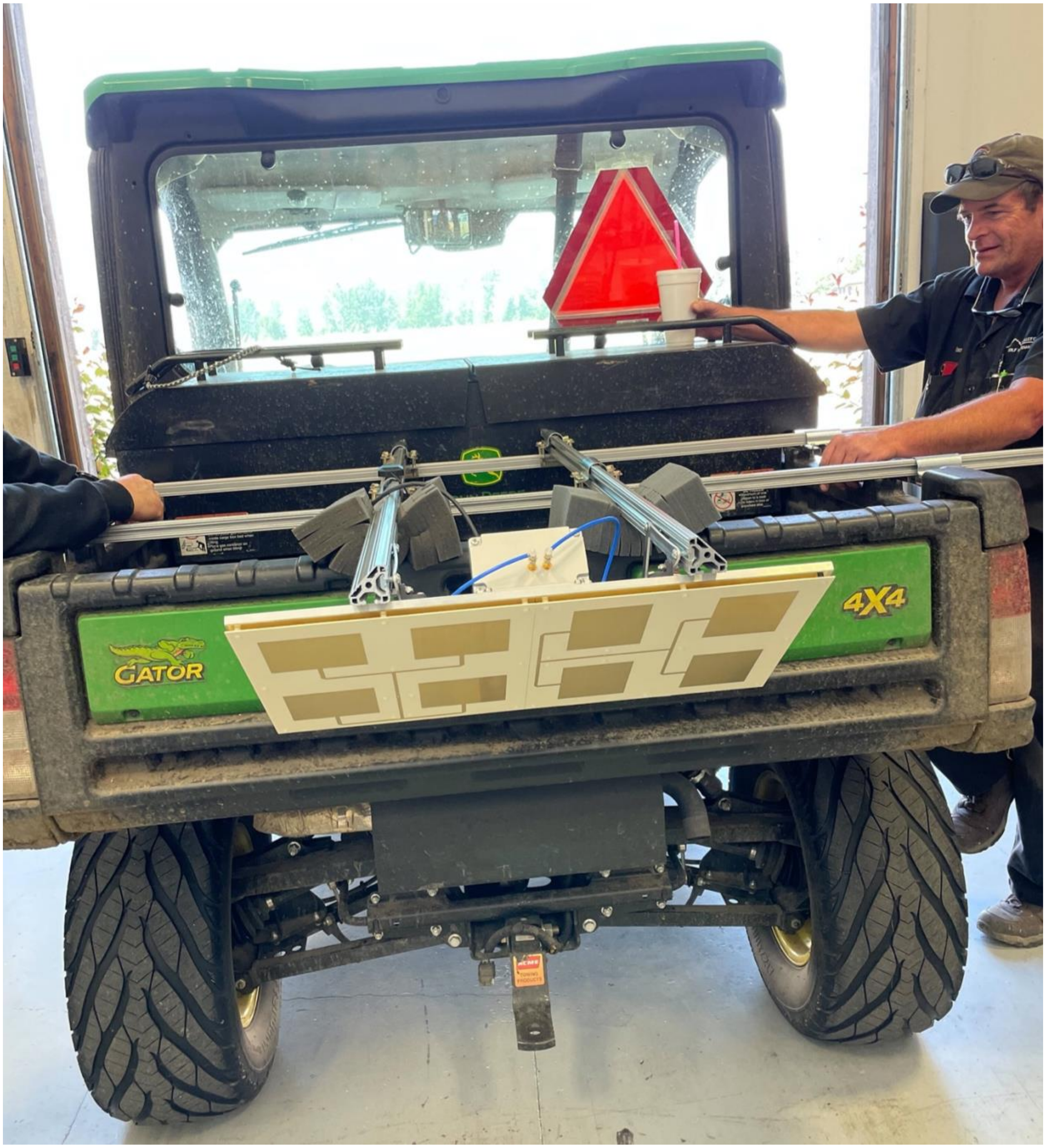
2) At the very start you may see error due to "import" commands. This means that you have missed on of the dependencies or that there is a version issue with compatibility. If you cannot get the necessary module installed please contact TerraRad for support.

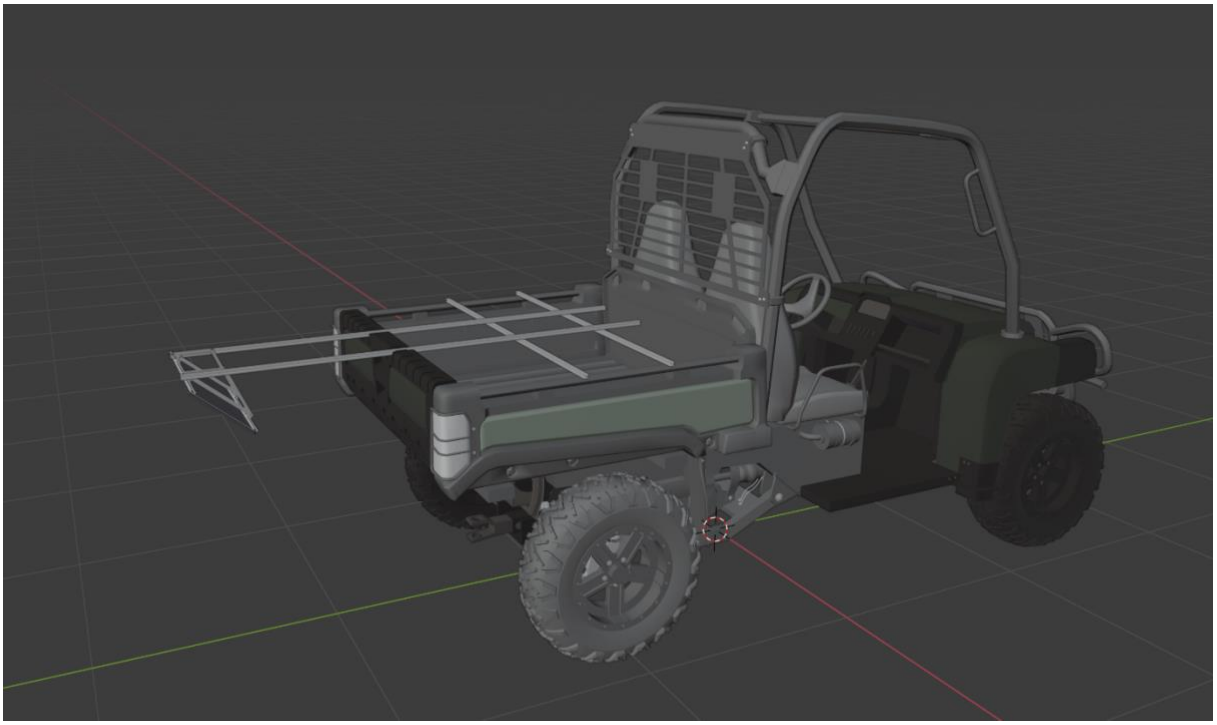
3) Other issues can be handled directly by TerraRad support.

VIII. Mounting on Vehicle

1) We propose the following integration on the John Deere Gator:

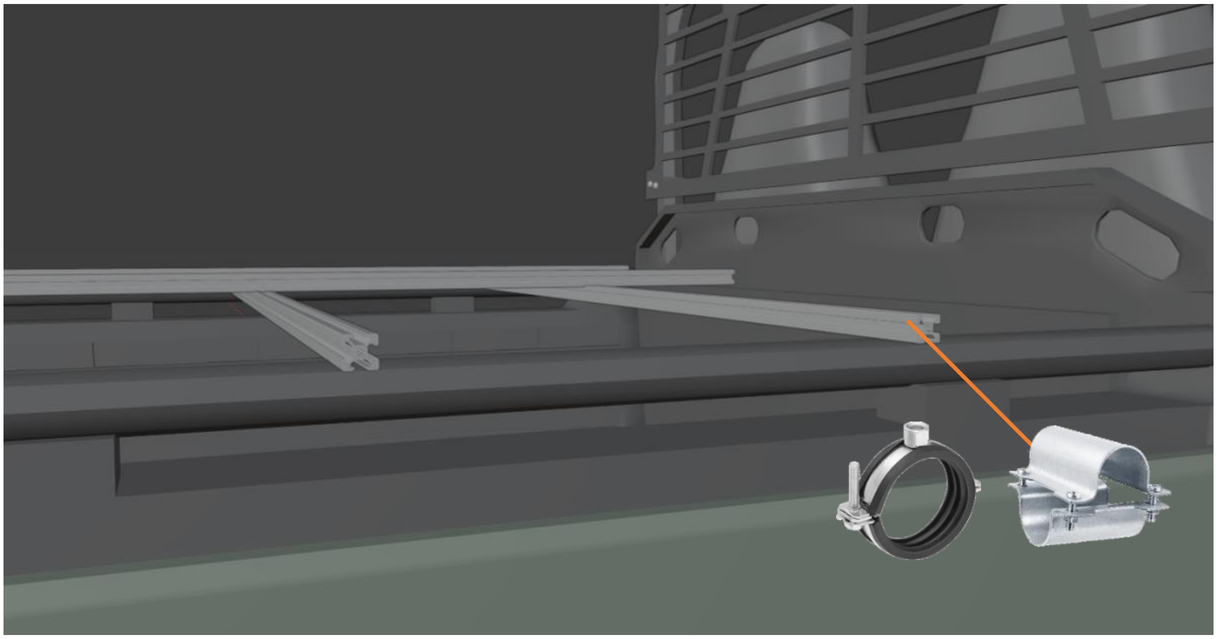
Mounting can be performed in a very similar manner for side-looking configuration. In the side-looking configuration the cross-bars will directly mount to the mount interface triangles of the PoLRa.

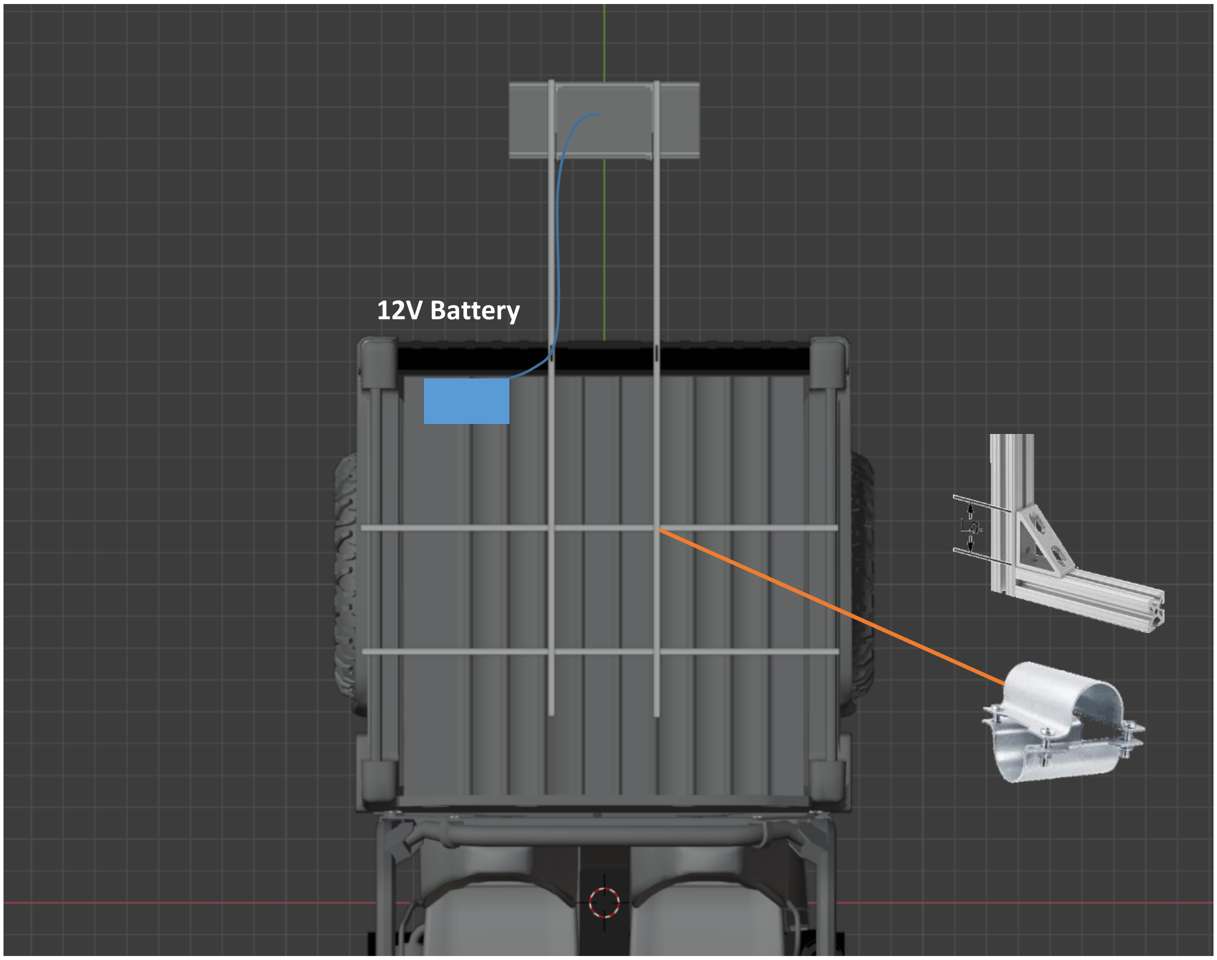


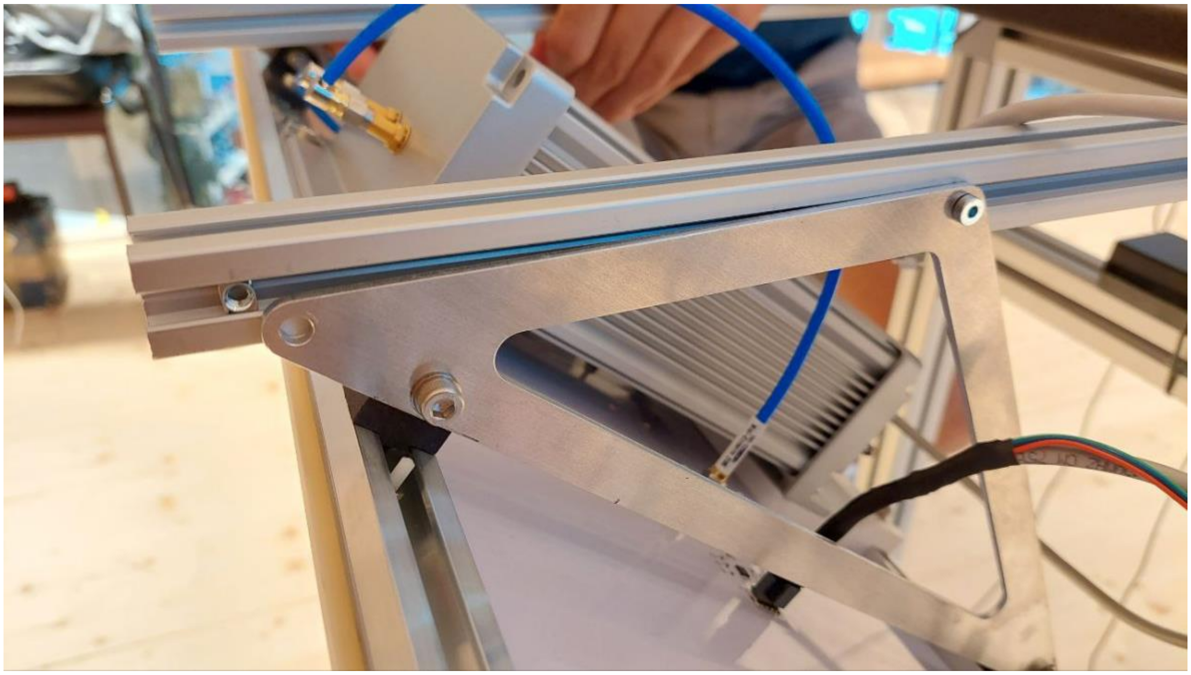


The rods are 1” or 1.5” (20mm x 20mm or 30mm x 30mm equivalent metric) **extruded aluminum profiles** that are mounted to the gators round handrails as shown in **Error! Reference source not found.**. For that **pipe clamps** could be used. These profiles are extremely convenient and adjustable. The T-slot nuts will allow the PoLRa to direct mount with M5 screws.

The aluminum profiles are crossed with two additional profiles and again made together with pipe clamps, for example. 90 degree profile mounts also make for a solid connection. The 12V LiFePo or PB battery is placed in the trunk or passenger cab and included cable powers the PoLRa. The antenna is mounted with **slide-in T-nuts** to the aluminum profiles as shown in **Error! Reference source not found.**.




12V Battery



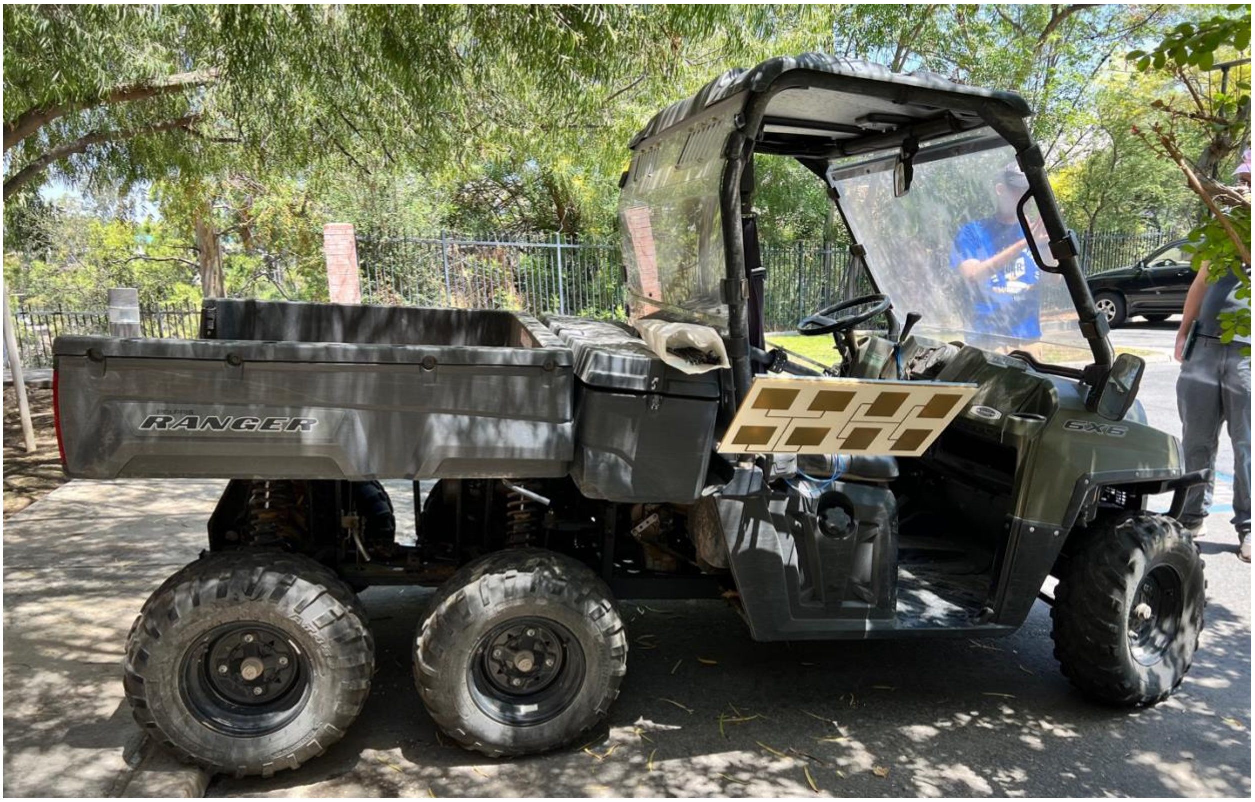
The following proposed part-list includes parts easily sourcable in Northern America. Similar options should be available internationally. Please consult TerraRad support if you have difficulties with aspects of components or mounting procedure.

1. Battery:
   https://www.amazon.com/gp/product/B09WXGNBYR/ref=ox_sc_act_title_2?smid=A2TJ1F8MWBEBG9&psc=1
2. Charger:
   https://www.amazon.com/gp/product/B09CGYLD8Z/ref=ox_sc_act_title_1?smid=A2XHS8UE99G688&psc=1
3. Aluminium profiles:
   https://www.mcmaster.com/profiles/structural-framing/t-slotted-framing-rails/
4. T-Slot Nuts:
   https://www.mcmaster.com/nuts/structural-framing/t-slotted-framing-fasteners/

5. 2x 5 Pipe Clamps (for 1” !):
   https://www.amazon.com/gp/product/B09YRMBRD8/ref=ox_sc_act_title_5?smid=A2KAGKE1Y9Z8AI&psc=1
6. Brackets:
   https://www.mcmaster.com/profiles/structural-framing/t-slotted-framing-structural-brackets/
7. Pipe Clamps:
   https://www.mcmaster.com/hose-clamps/vibration-damping-threaded-rod-mount-clamping-hangers-8/

2) Mounting PoLRa on Polaris RANGER 6x6

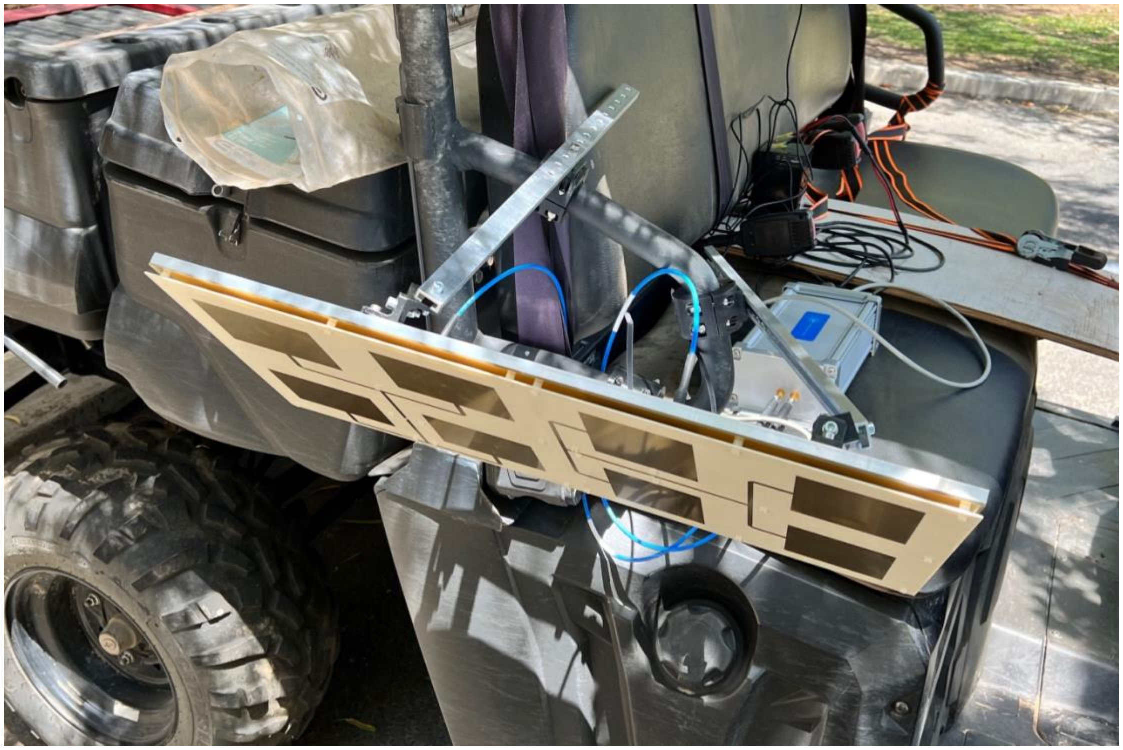
The Polaris does not provide many straight and straightforward access points for mounting, so we take advantage of the metal door on each side of the passenger cabin. The following picture shows the side-looking configuration installed on the RANGER 6x6.



Side-looking mounting on the Polaris can be done most-easily with a combination of commercial aluminum profile hardware and 3D printed parts.

For this mount, we remove the antenna mount triangles and find the desired angle by adjusting the aluminum profile rails. The electronics box is also preferably kept in the passenger cab with alternative SMA cable option. Consult TerraRad if you have an antenna mounted unit and would prefer to transfer to the cabin mounted unit.

The following picture shows a close-up of the 40 degree sensor position with 3D-printed handrail brackets and adjustable aluminum rails.



3) Other vehicles (e.g. snowmobile, trucks, trailors) can also be mounted with the PoLRa. We have provided the above mounting examples to provide mechanical inspiration. For snowmobile-mounted use we would recommend mounting with side-looking antenna (or forward looking) to avoid measuring the rough tracks left by the vehicle.

IX. Interval Measurement Processing (fixed / tower -based)

The crontab for interval measurements creates a new file for each hour. This is to avoid excessively large file sizes and to minimize data loss risk in the event of a misplaced or corrupted file.

The interval measurement processing software is similar to the vehicle-based processing, but generally simpler due to the lack of spatial data. The main script is contained inside:

```
./pypolra3.2_Sherbrooke/ground_based/
```

And is called `main_interval_proc.py`

The first step in processing the interval data is appending the hourly files into a single csv file. Second the "ground_calibrator" function performs much of the same processing steps as pypolra31 for the vehicle-based routine. The plotter function takes the mean of the 30 subsamples taken every 1 minutes, this is the setting in "measperiod" (or whatever */x setting was in the crontab).

Open "`main_interval_proc.py`" and modify the following variables to your dataset:

```
# path to directory containing hourly interval data
datadir = './data/SN8/interval_sample'

# filepath to output appended csv file
rawappended = './data/SN8/sample_appended.csv'

# filepath to config file for correct PoLRa S/N
configfile = "./data/SN8/config.py"

# Output filepath for L1 datafile (L1)
L1filepath = './data/SN8/sample_L1.csv'

# Measurement period (minutes)
measperiod = 1 # minute
```

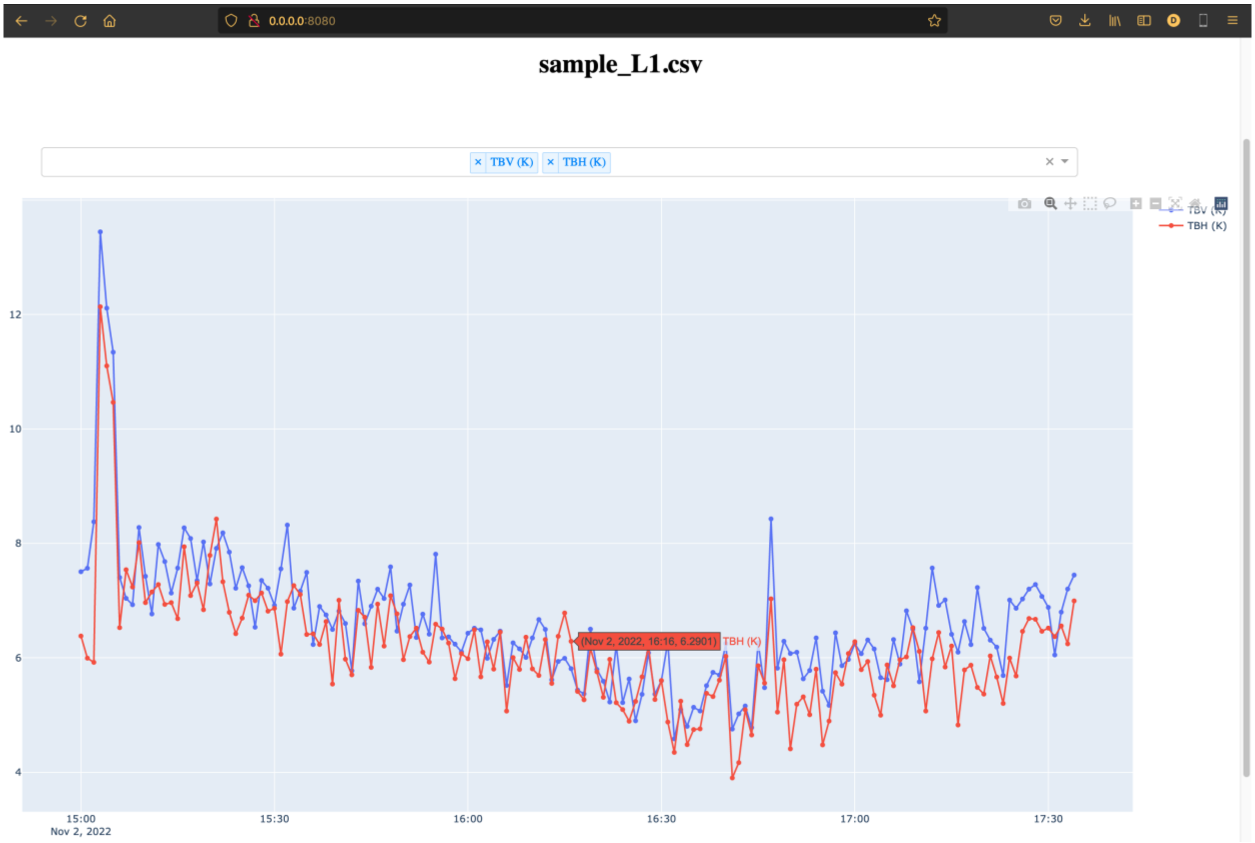
Run the script with "`python main_interval_proc.py`" command. Once the processing routine completes you will see a message:

```
* Running on http://10.66.0.205:8080/ (Press CTRL+C to quit)
```

Open a browser and enter in the URL bar: 0.0.0.0:8080

You will have a plot of brightness temperature time series. You can select other variables from the dropdown menu. This web-based plot is simply displaying the contents of the "L1" file that has been generated by the "ground_calibrator" function. You will see something like the below screenshot.

Lines 44-54 in “interval_L1plotter.py” provides an example of how to load the L1 file using python Pandas, an easy way to do further manipulation and processing of the data.



X. Maintenance

The PoLRa needs to be occasionally maintained to keep perfect performance levels and long lifetime.

The recommended maintenance interval is 3 months.

The recommended tools for maintenance include:

- Isopropyl alcohol in spray or liquid form
- Compressed air can or similar
- 4mm allen key
- 8 mm open-ended wrench

1) Clean the antenna face (gold and bare white substrate) with isopropyl and towel.

2) Clean dust and dirt from in between the two antenna layers using the compressed air.

3) Use 4mm allen key and wrench to make sure all screws connecting mounting hardware are tight.

XI. Firmware Updates

Firmware on the PoLRa v3.x may occasionally need to be updated / replaced. Firmware replacements will be made using FileZilla similarly to downloading data (see section III).

1) Download and decompress .zip file containing firmware update software

2) Drag new “software3.x” directory onto PoLRa3.x unit with FileZilla, replacing the /home/pi/software3.x directory.

3) You may also be provided with additional commands that will need to be run on PoLRa terminal through ssh.