# Bennington College Small Radio Telescope
## Operations Manual

*Andrew Cencini – Hugh Crowl*
*William Buchanan, Alexander Curth, Erick Daniszewski, Evan Gall,*
*Clemente Gilbert-Espada, Chernoh Jalloh, Brendon Walter*

*with guidance from*

**MIT Haystack Observatory's**
Haystack Small Radio Telescope Project

# Table of Contents

This manual serves as a guide following the work done at Bennington College during the spring of 2014 to build and operate a radiotelescope, following the procedure layed out by MIT Haystack and their Small Radio Telescope (SRT) project. This manual contains information regarding both the hardware and software components of the project which should allow for a simple and successful build and operation of a SRT.

## Parts List

#### LNA to Dongle Interface

| Part | Quantity | Manufacturer | Link | Approx. Cost |
|---|---|---|---|---|
| SMA-M Crimp Connector Straight | 1 | Amphenol/Connex | here | $5.95 |
| LMR-400 | 100ft | | here | $102.99 |
| Type-F Male Crimp Connector | 1 | Amphenol/Connex | here | $4.95 |
| In-Line Amplifier | 1 | Blonder Tongue | here | $8.22 |
| BNC Female Jack to Type-F Male Plug | 3 | | here | $10.99 |
| BNC-M to BNC-M Patch Cable | 3 | Amphenol RF | here | $12.55 |
| Power Injector | 1 | Channel Vision | here | $14.71 |
| BNC-F to SMA-M Adapter | 1 | Vitelec / Emerson Connectivity Solutions | here | $12.09 |
| Bandpass Filter | 1 | Mini-Circuits | here | |
| SMA-F to BNC-F Adapter | 1 | AIM-Cambridge / Emerson Connectivity Solutions | here | $4.21 |
| BNC-F to SMB Plug Adapter | 1 | | here | $8.49 |
| AC-to-DC Power Supply | 1 | | here | $20.18 |

| Part | Quantity | Manufacturer | Link | Approx. Cost |
|---|---|---|---|---|
| Breadboard | 1 | Vector Electronics | here | $5.75 |
| DC Jack | 1 | CUI Inc | here | $1.00 |
| F-Type Plug | 1 | Bomar Interconnect/ Winchester Electronics | here | $4.16 |

### Feed

| Part | Quantity | Manufacturer | Link | Approx. Cost |
|---|---|---|---|---|
| Ultra Low Noise Amplifier | 1 | Mini-Circuits | here | |
| Coaxial Bias-Tee | 1 | Mini-Circuits | here | |
| Coaxial Cable 086-6SM+ | 1 | Mini-Circuits | here | |
| Coaxial Cable 086-4SM+ | 1 | Mini-Circuits | here | |
| SMA-F to SMA-F Adapter | 1 | Mini-Circuits | here | |
| SMA-F to Solder Pin Bulkhead Connector | 1 | Emerson Network Power Connectivity Johnson | here | $8.06 |
| SMA-M to SMA-M Right Angle Adapter | 1 | | | |
| 3M Adhesive Copper Foil Tape | 1 | 3M | here | |
| 22 Gauge Copper Wire | 1 | | | |
| Semi-Rigid Coaxial Cable | 1 | Micro-Coax | here | $4.62 |
| F-Type Male to BNC Female | 1 | | here | $0.89 |
| SMA-M to SMA-M Coupler | 1 | | here | $7.49 |

# Tools List

### Assembling the LNA and Feed

| Tool | Description |
|---|---|
| Soldering iron + solder | soldering components as specified by Haystack documentation |
| Wire cutter/ strippers | cutting 22 gauge copper wire |
| Screwdriver | assembling box of LNA |
| Labeler | labeling LNA |
| Hacksaw | cutting styrofoam |
| Metric measuring tape | measuring styrofoam, and helix position |
| Drill press and 1/4" bit | drilling hole into styrofoam, LNA baseplate, cake pan, pc board |
| X-acto knife | cutting copper tape |
| Straight edge | cutting copper tape |
| Marker | marking position of helix |

## Assembling the Dish

| Tool | Description |
|---|---|
| Socket wrench set | bolting arms onto center |
| Hammer and tap | marking ends of arms and outer band to make drilling holes easier |
| Handheld Riveter/ rivets | riveting the outer band and strips over mesh on the arms. |
| Hacksaw | |
| Power drill + Drill bit | drilling holes to place rivets connecting outerband, mesh, strips, and dish arms |

| Tool | Description |
|---|---|
| Safety Gloves | for your fingers |
| Pliers | Useful for bending mesh when wrapping around edge |
| Tin Snips | cutting mesh |
| Pen/Marker | mark hole locations pre-drilling |

### Building the Mount

### Attaching Arms

### Mounting Dish

## Installing SRTN Software

The original SRT source can be found on the MIT Haystack Observatory website, here: http://www.haystack.mit.edu/edu/undergrad/srt/ . This source appears to work with ArchLinux and CentOS, and reportedly works on REHL (Red Hat Enterprise Linux). The source will not work immediately with Ubuntu, however a modified version of the source can be found here: https://github.com/BenningtonCS/Telescope-2014 , which will work on Ubuntu, as well as CentOS and ArchLinux (It may work with other linux-based operating systems, however, those listed are the only ones we have tested so far).

### *Ubuntu*

These instructions were developed as a guide from a clean download of source files from the MIT Haystack Observatory website. If using the code pulled from the repository, some of the steps (modifying the srtnmake script for example) may not be necessary, but the guide should still hold, overall.

**Note that these instructions were written for Ubuntu 12.10**

```
$ lsb_release -a
No LSB modules are available.
Distributor ID: Ubuntu
Description:    Ubuntu 12.10
Release:    12.10
Codename:   quantal
```

# MIT Haystack Observatory SRT

The MIT Haystack Observatory SRT website includes the manuals, documentation, and program code. The code base we are looking at, SRT Source Code ver 3, can be found here.

## Setting Up the Source Code

To get the source code, either go to the website, linked above, and download SRT Source Code ver 3, or download directly from the link above. Once downloaded, the gzipped tarball should be in your Downloads directory: `~/Downloads/newsrtsource_ver3.tar.gz`.

## Move and Unzip File in New Directory

To create a new directory, move the gzipped tarball into it, and then unzip it, follow the commands below:

```
~$ mkdir radio
~$ mv Downloads/newsrtsource_ver3.tar.gz radio
~$ cd radio
~/radio$ tar -xzvf newsrtsource_ver3.tar.gz
~/radio$ cd srtnver3
```

You are now in the directory housing the SRT codebase. To see what exists in the directory, simply use the command `ls`.

At this point if you try to compile the code, it will fail, as we saw in class.

### Breaking down the above commands

- `ls` will list all the files in the directory you are in
- `cd` allows you to change directories (folder) eg. `cd Downloads` will put you in the downloads directory
- `mv` moves a file from one directory (folder) to another
- `mkdir` creates a new directory (folder)
- `tar -xzvf` a command to un-gzip and un-tar a file (decompress)

For more information on any command, in the terminal window, `man [command]` will display the manual for the given command. Additionally, here's a cheat sheet of commonly used UNIX commands.

## Adding the Library Dependencies

The reason the code would not compile is because SRT uses libraries which the computer you are using does not have. To get them, use the command below:

```
~/radio/srtnver3$ sudo apt-get install libgtk2.0-dev libusb-1.0-0-dev
```

You will be prompted for your password, and may be prompted if you wish to continue after apt-get informs you of the package size to be installed. Input `Y` and press enter to continue downloading. These libraries should automatically be downloaded into the correct directory.
NOTE: If you are unable to download using `apt-get install`, try using the command `apt-get update` first, wait for all updates to be installed, and then retry the install command.

### Breaking down the above commands:

`sudo` is used to give the user super user privilege which allows you to make dire changes to your computer if you so choose. Using the sudo command will require you to input your password, which it will ask for once the command is run.
`apt-get` is a software repository which is used on debian based distributions of linux (although Ubuntu is no longer debian based, it used to be and so still uses `apt-get`). Other software repositories include `zypper` (used by openSUSE), and `pacman` (used by arch). The software repos give you the ability to download software from the command line without having to search the internet to find the program you want to use.
The `install` part of the command is specific to `apt-get`. When using `apt-get`, you have to specify if you want to `install` a program, remove a program, or something else. Next, you specify the name(s) of the program(s) to be installed (or removed).
`libusb-1.0-0-dev` and `libgtk2.0-dev` are the names of the libraries we want to get for the SRT program.

## Modifying the Compile Script

In order for the program to compile with gtk+-2.0, it must know that it has to use it/where to get it. (This was the problem that prevented it from compiling in class). To remedy this, simply

```
~/radio/srtnver3$ sudo nano srtnmake
```

to edit the bash script. Scroll to the bottom of the file to the gcc line, which should look like:

```
gcc -W -Wall -03 $CFLAGS $LIBS main.c vspectra_four.c disp.c plot.c cat.c geom.c time.c outfile.c sport.c
map.c cmdfl.c cal.c srthelp.c velspec.c four.c librtlsdr.c tuner_r820t.c -lm `pkg-config --libs --cflags
libusb-1.0`
```

and change it so the end section is now

```
... `pkg-config --libs --cflags gtk+-2.0 libusb-1.0`
```

Now you have all the libraries and the script should be able to compile successfully.

## Compiling the Source Code

The script to compile the code (the one you just edited) should take care of all the compilation, so it is just a matter of running it

```
~/radio/srtnver3$ ./srtnmake
```

Once you execute this command, a few warnings should appear, with lines ending with `[-Wunused-but-set-variable]`. These errors are fine, and there should be relatively few of them as compared to before (running the compile script without the libraries or modification to the script). The output I got from running the compile script is shown below as a reference for which errors are acceptable.

```
~/radio/srtnver3$ ./srtnmake
main.c: In function ‘main’:
main.c:62:12: warning: variable ‘secstart’ set but not used [-Wunused-but-set-variable]
main.c:60:12: warning: variable ‘ii’ set but not used [-Wunused-but-set-variable]
main.c:59:14: warning: variable ‘color’ set but not used [-Wunused-but-set-variable]
main.c: In function ‘gauss’:
main.c:555:16: warning: variable ‘j’ set but not used [-Wunused-but-set-variable]
vspectra_four.c: In function ‘vspectra’:
vspectra_four.c:33:28: warning: variable ‘min’ set but not used [-Wunused-but-set-variable]
vspectra_four.c:33:12: warning: variable ‘avsig’ set but not used [-Wunused-but-set-variable]
vspectra_four.c:31:43: warning: variable ‘r’ set but not used [-Wunused-but-set-variable]
disp.c: In function ‘clearpaint’:
disp.c:440:18: warning: variable ‘update_rect’ set but not used [-Wunused-but-set-variable]
outfile.c: In function ‘outfile’:
outfile.c:16:16: warning: variable ‘n’ set but not used [-Wunused-but-set-variable]
sport.c: In function ‘rot2’:
sport.c:402:25: warning: variable ‘i’ set but not used [-Wunused-but-set-variable]
sport.c:402:17: warning: variable ‘status’ set but not used [-Wunused-but-set-variable]
sport.c:408:15: warning: ignoring return value of ‘system’, declared with attribute warn_unused_result [-
Wunused-result]
cal.c: In function ‘cal’:
cal.c:18:24: warning: variable ‘ixe’ set but not used [-Wunused-but-set-variable]
srthelp.c: In function ‘display_help’:
srthelp.c:39:14: warning: variable ‘color’ set but not used [-Wunused-but-set-variable]
srthelp.c: In function ‘load_help’:
srthelp.c:254:10: warning: ignoring return value of ‘fgets’, declared with attribute warn_unused_result
[-Wunused-result]
srthelp.c:258:14: warning: ignoring return value of ‘fgets’, declared with attribute warn_unused_result
[-Wunused-result]
velspec.c: In function ‘velspec’:
```

```
velspec.c:19:14: warning: variable ‘color’ set but not used [-Wunused-but-set-variable]
velspec.c: In function ‘vplot’:
velspec.c:162:15: warning: variable ‘jmax’ set but not used [-Wunused-but-set-variable]
librtlsdr.c: In function ‘rtlsdr_open’:
librtlsdr.c:1267:13: warning: variable ‘rt’ set but not used [-Wunused-but-set-variable]
tuner_r820t.c: In function ‘R828_RfGainMode’:
tuner_r820t.c:2859:11: warning: variable ‘LnaGain’ set but not used [-Wunused-but-set-variable]
tuner_r820t.c:2858:11: warning: variable ‘MixerGain’ set but not used [-Wunused-but-set-variable]
~/radio/srtnver3$
```

Now the source code is successfully compiled!

## Running the Program

If you look in the srtnver3 directory, you will notice an executable file called `srtn`. This is what we just compiled with `srtnmake`. To run:

```
~/radio/srtnver3$ ./srtn
```

Once executed, this should open a window which looks similar to the image below.

## *CentOS*

These instructions were made using CentOS release 6.5 with kernel version:

```
$ uname -r
2.6.32-431.el6.x86_64
```

on a machine with an AMD processor.

# MIT Haystack Observatory SRT

The MIT Haystack Observatory SRT website includes the manuals, documentation, and program code. The code base we are looking at, SRT Source Code ver 3, can be found here.

## Getting the Source Code

From the website, linked above, or the download link provided, download the srtnver3 source files from the MIT

Haystack website. Once the download is complete, the gzipped tarball should be in your Downloads directory, `~/Downloads/newsrtsource_ver3.tar.gz`.

```
$ cd Downloads
$ tar xzvf newsrtsource_ver4.tar.gz
```

If you want to move the source code out of the Downloads directory, and into the Home directory, for example:

```
$ mv srtnver3 ~
$ cd ~
```

## Getting project dependencies

From a clean install of CentOS, you will be missing some of the packages needed to run the software.

### Getting `sudo` permissions

Before you do this though, you need to add your profile to the sudoers file, if you do not already have sudo permissions. To do this:

```
$ su
$ chmod +w /etc/sudoers
$ gedit /etc/sudoers
```

Down by the bottom of the file will be the line:

```
## Allow root to run any commands anywhere
root    ALL=(ALL)    ALL
```

Below this add the line " ALL=(ALL) ALL", where is the username logged in to the computer, in our case 'radio telescope'

```
## Allow root to run any commands anywhere
root    ALL=(ALL)    ALL
radiotelescope    ALL=(ALL)    ALL
```

Save the file and exit gedit, then in the terminal, type `exit` to return to your user status:

```
[root@localhost radiotelescope]# exit
exit
[radiotelescope@localhost ~]$
```

Now your user account should have sudo permission, so we can now install package dependencies.

### Getting dependencies

The packages needed are gtk+2, gcc, and libusb. These can be acquired using the CentOS package manager, `yum`:

```
$ sudo yum install gtk2-devel gcc libusb1-devel
```

If successful, you should now have all the packages you need to run the software.

## Compiling and running

Now, move into the srtnver3 directory

```
$ cd ~/srtnver3
```

and run `srtnmake`:

```
$ ./srtnmake
```

This should complete successfully, but may show some warning messages. Now, to run the software:

```
$ ./srtn
```

## Running with unsimulated dongle

To run with the dongle, receiver simulation must be turned off. To do this, simply

```
$ gedit srt.cat
```

In the edit window, find the line that says

```
SIMULATE RECEIVER
```

and change it to

```
*SIMULATE RECEIVER
```

Save the file and exit gedit. Now, we can run srtn using the dongle with

```
$ sudo ./srtn
```

And it should work!

*NOTE*: Notice how before to run, it was just `./srtn` and this time it is `sudo ./srtn`. If you try `./srtn` after enabling the dongle you will get a message similar to:

```
$ ./srtn
Found 1 device(s)
   0:  ,   SN:  ?????
Using device 0: ezcap USB 2.0 DVB-T/DAB/FM dongle
usb_open error -3
Please fix the device permissions, e.g. by installing the udev rules file rtl-sdr.rules
Failed to open rtlsdr device #0.
```

As we see above, it is possible to run srtn without sudo, but to do so, we must do a little extra work in installing the provided udev rules. **A guide to installing the udev rules will be added later.**

## Running with unsimulated controller/rotator

To run with the controller/rotator, antenna simulation must be turned off. To do this, simply

```
$ gedit srt.cat
```

In the edit window, find the line that says

```
SIMULATE ANTENNA
```

and change it to

```
*SIMULATE ANTENNA
```

Save the file and exit gedit. Now, we can run srtn using the controller/rotator with

```
$ sudo ./srtn
```

To test that the rotator works, try manually changing the az or el in-program using the 'azel' button. (Note: If the the rotator motion is moving in a direction you do not want it to go, you can press the 'S' button on the controller to stop movement.)

If you are unable to run the program successfully, look at the controller setup page to troubleshoot and make sure the controller was set up correctly.

## *ArchLinux*

These intstructions were made to install the srtver3 code from MIT on a clean install of Archbang Linux. However, these instructions should work for any up to date release of Arch.

The MIT Haystack Observatory SRT website includes the manuals, documentation, and program code. The code base we are looking at, SRT Source Code ver 3, can be found here.

## Getting the Source Code

From the website, linked above, or the download link provided, download the srtnver3 source files from the MIT Haystack website. Once the download is complete, the gzipped tarball should be in your Downloads directory, `~/Downloads/newsrtsource_ver3.tar.gz`.

```
$ cd Downloads
$ tar xzvf newsrtsource_ver3.tar.gz
```

## Getting Project Dependencies

From a clean install of Archbang, there are a few necessary things missing, namely `make` and`patch`.

From the command line, run:

```
$ sudo pacman -S make patch
```

## Install gcc-4.4

Unfortunately, the program will not run properly using a current version of gcc, so it is necessary to download and install version 4.4. As gcc-4.4 is not in pacman, gcc-4.4 must be downloaded from the AUR.

You can find gcc-4.4 in the Arch User Repository here

Download the tarball. It should appear in your Downloads folder. Untar it by using:

```
$ cd ~/Downloads
$ tar xzvf gcc44.tar.gz
```

and move into the new folder that was created.

```
$ cd gcc44
```

Use the command `makepkg` by itself to make the package. *NOTE*: It might be necessary to make the package as root. If this is the case, run:

```
$ sudo makepkg --asroot
```

As gcc is a large program, it might take a very long time to download. Once it's downloaded, run:

```
$ sudo pacman -U gcc44-4.4.7-6-x86_64.pkg.tar.xz
```

Once again, as gcc is a large program, this will take a long time to complete.

## Compile and Run Source

Move into the srtnver3 folder that you downloaded and untared from MIT. Assuming it's in your Downloads folder, use:

```
$ cd Downloads/srtnver3
```

## Edit srtnmake

To make the makefile use gcc-4.4 instead of the default version, you will have to edit the make file.

```
$ nano srtnmake
```

Change `gcc` to `gcc-4.4`

Save and exit out of nano.

## Compile srtnmake

Run:

```
$ ./srtnmake
```

If everything is done right, the program should compile with few errors.

## Running srtn

To run the program:

```
$ ./srtn
```

## Running with unsimulated dongle

**NOTE**: It might be necessary to blacklist the driver in order to run properly. Please look at 'Blacklist Driver' under the troubleshooting section.

To run with the dongle, receiver simulation must be turned off. To do this, simply

```
$ nano srt.cat
```

In the edit window, find the line that says

```
SIMULATE RECEIVER
```

and change it to

```
*SIMULATE RECEIVER
```

Save the file and exit nano. Now, we can run srtn using the dongle with

```
$ sudo ./srtn
```

And it should work!

*NOTE*: Notice how before to run, it was just `./srtn` and this time it is `sudo ./srtn`. If you try `./srtn` after enabling the dongle you will get a message similar to:

```
$ ./srtn
Found 1 device(s)
   0:  ,   SN:  ?????
Using device 0: ezcap USB 2.0 DVB-T/DAB/FM dongle
usb_open error -3
Please fix the device permissions, e.g. by installing the udev rules file rtl-sdr.rules
Failed to open rtlsdr device #0.
```

As we see above, it is possible to run srtn without sudo, but to do so, we must do a little extra work in installing the provided udev rules. **A guide to installing the udev rules will be added later.**

## Running with unsimulated controller/rotator

To run with the controller/rotator, antenna simulation must be turned off. To do this, simply

```
$ nano srt.cat
```

In the edit window, find the line that says

```
SIMULATE ANTENNA
```

and change it to

```
*SIMULATE ANTENNA
```

Save the file and exit nano. Now, we can run srtn using the controller/rotator with

```
$ sudo ./srtn
```

To test that the rotator works, try manually changing the az or el in-program using the 'azel' button. (Note: If the the rotator motion is moving in a direction you do not want it to go, you can press the 'S' button on the controller to stop movement.)

If you are unable to run the program successfully, look at the controller setup page to troubleshoot and make sure the controller was set up correctly.

## Troubleshooting

## Blacklist driver

It may be necessary to blacklist the driver so it will not already be in use when we try to call on it:

```
$ sudo -i
# echo "blacklist dvb_usb_rtl28xxu" > /etc/modprobe.d/librtlsdr-blacklist.conf
```

This will require a reboot before the effects can be seen.

```
$ sudo reboot
```

# RAS SPID Rotator

### *Setup*

### *Wiring*

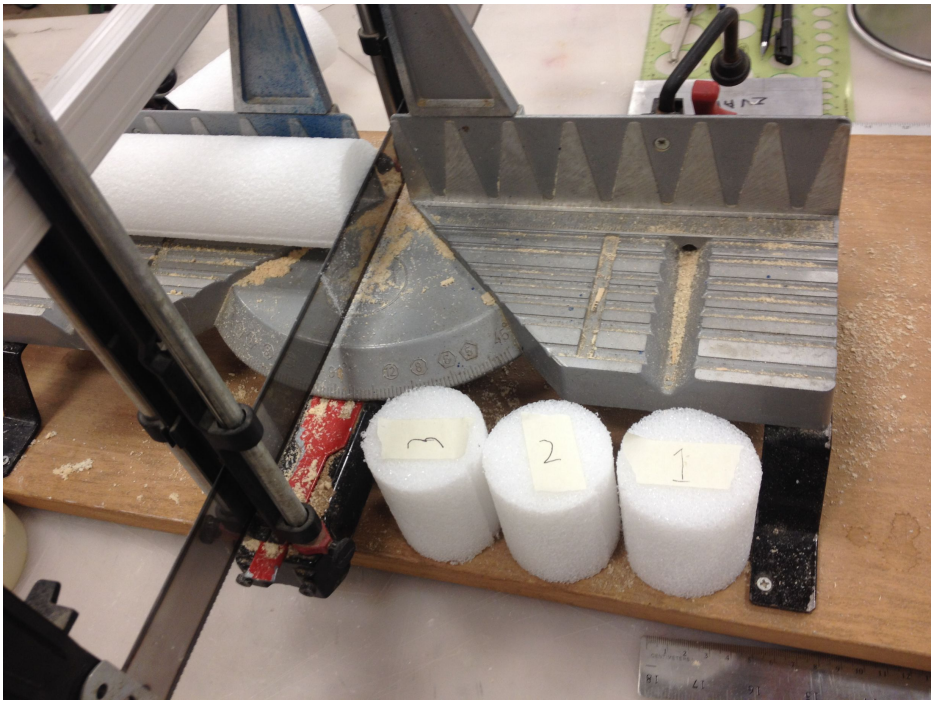
### *Controller Settings*

## Constructing

### *Feed*

## Cutting and Drilling the 2.5" styrofoam rod.

We used a hack saw in a jig to get perfectly straight cuts in the Styrofoam rod.




The Styrofoam was measured from multiple sides to determine the center. The rod we have ranges from 2 5/16" to 2 1/2" in diameter.

We used a drill press with a 1/4" bit. The Styrofoam was held by a vice on the drilling table. (The Styrofoam is a bit messy so we put down a rag under the vice)

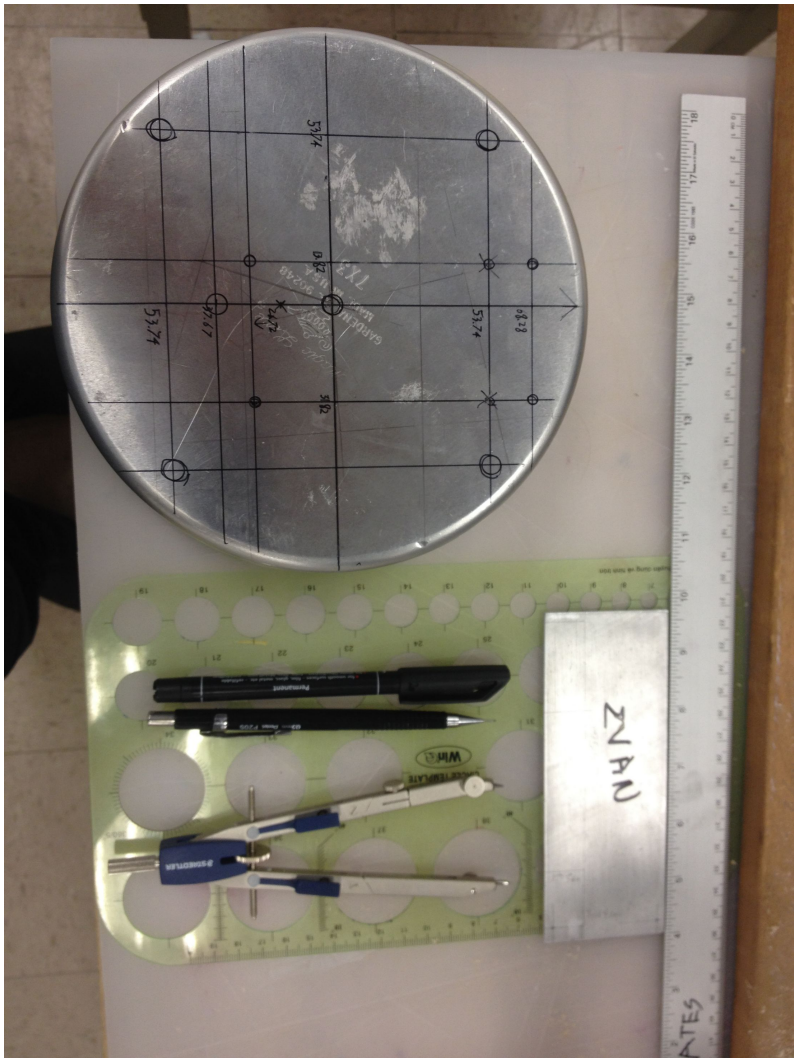

Note: we made three rods to provide room for error in the application and soldering of the copper tape later.

## Drilling Aluminum LNA Base Plate

We put the plate in a vice which we secured on the drill press table with some heavy steel blocks.

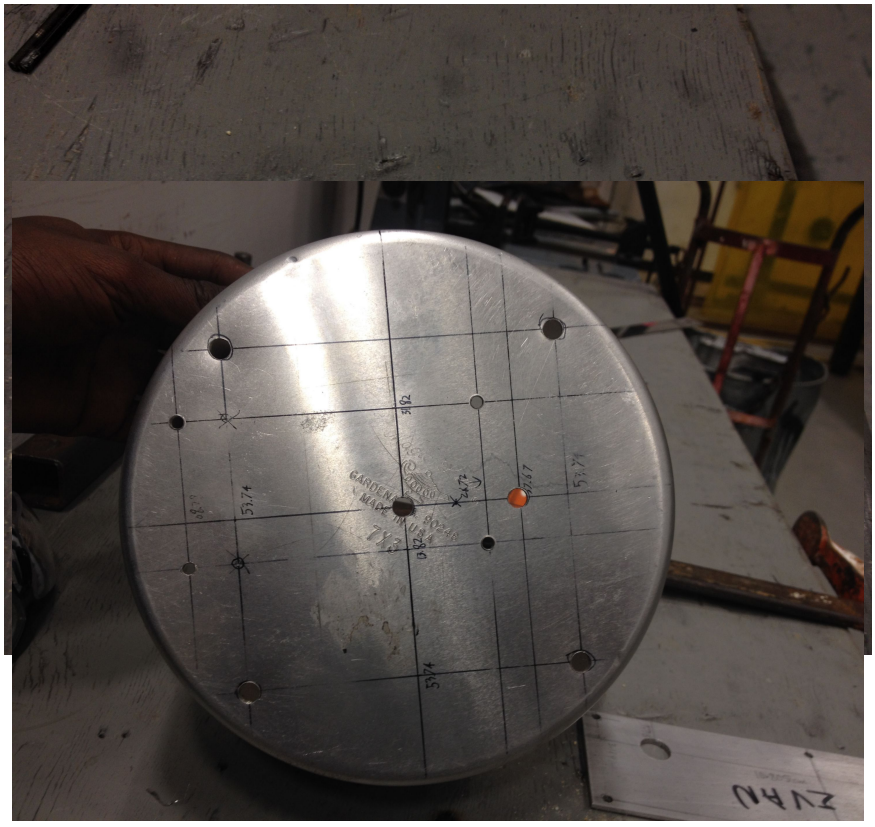

## Drilling Aluminum Cake Pan

We made the measurements of the cake pan working out from the center which we found by transecting the circle and then making perpendicular rays from the center of each line cutting across the arc. This proved to be adequately precise.



The cake pan was more of a challenge to drill than the base plate as it was tricky to secure to the drill press table so we could make accurate holes. We ended up putting a 4" piece of angle iron on the pan and then clamping that down to the edges of the table.


## Finished drilling

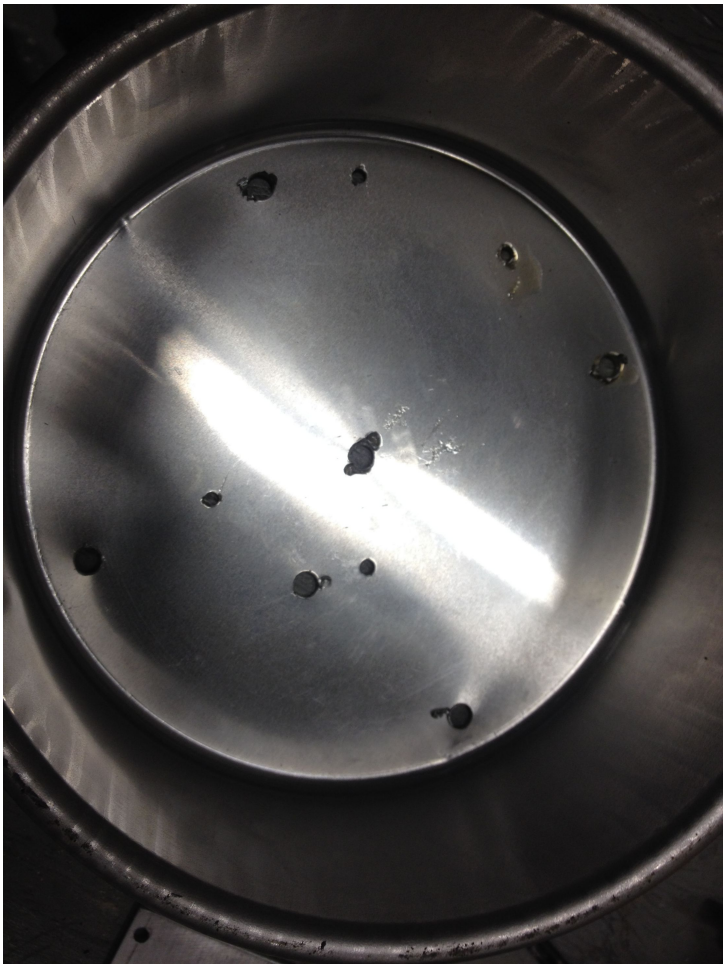


## Minor concerns

The drilling instructions are precise to a 100th of a millimeter. We did the drilling with a drill press, not a CNC machine, so I would say our precision was closer to a 10th of a millimeter.

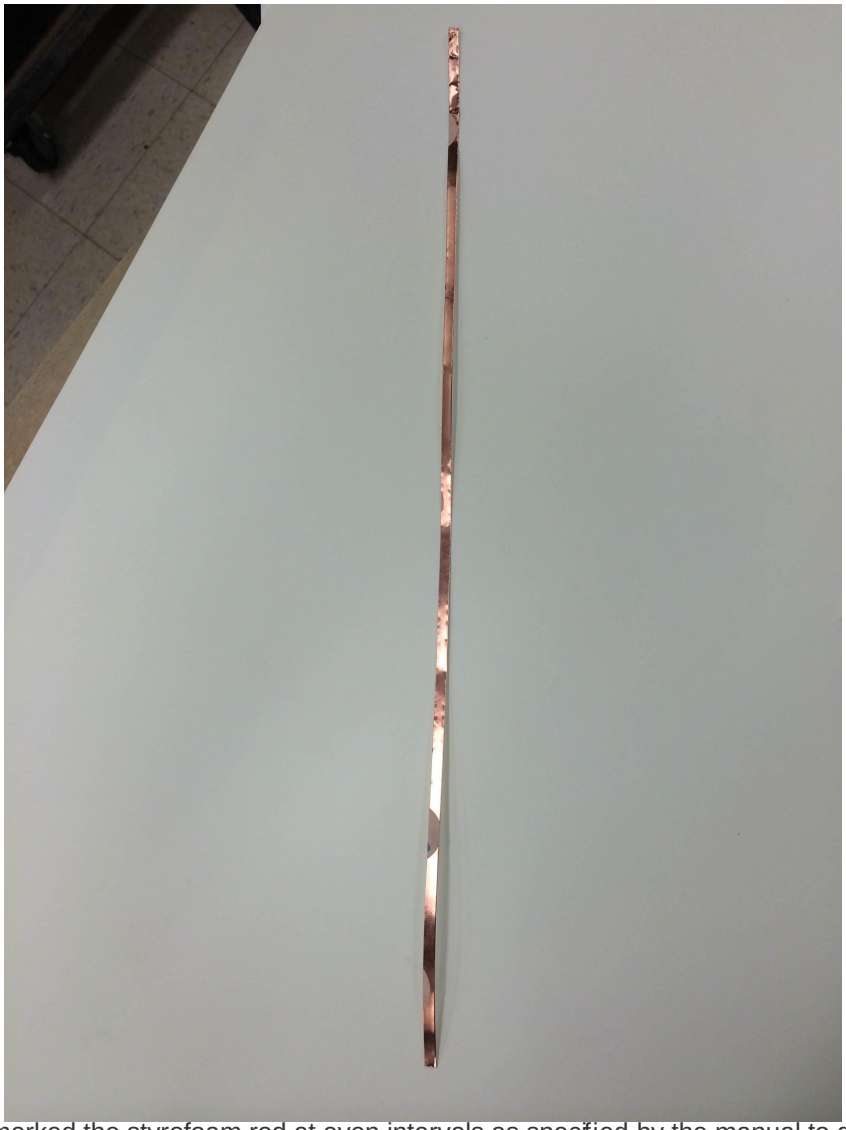
Also, the cake pan is really not a precisely made object, it is certainly not as precise as the schematics given by MIT. I think that indicates that our slight reduction in precision will be fine.

The cake pan is made of what seems to be a softer aluminum than the LNA base plate. This made the metal not cut cleanly on the inside of the pan (the bit went through the outside first). The holes are the correct dimensions but there is some extra material around the holes. I have tried to clean it up with an x-acto knife and been relatively successful. I think the metal is so soft that these small bits of extra material will be squished when we insert the hardware, making their effect negligible.



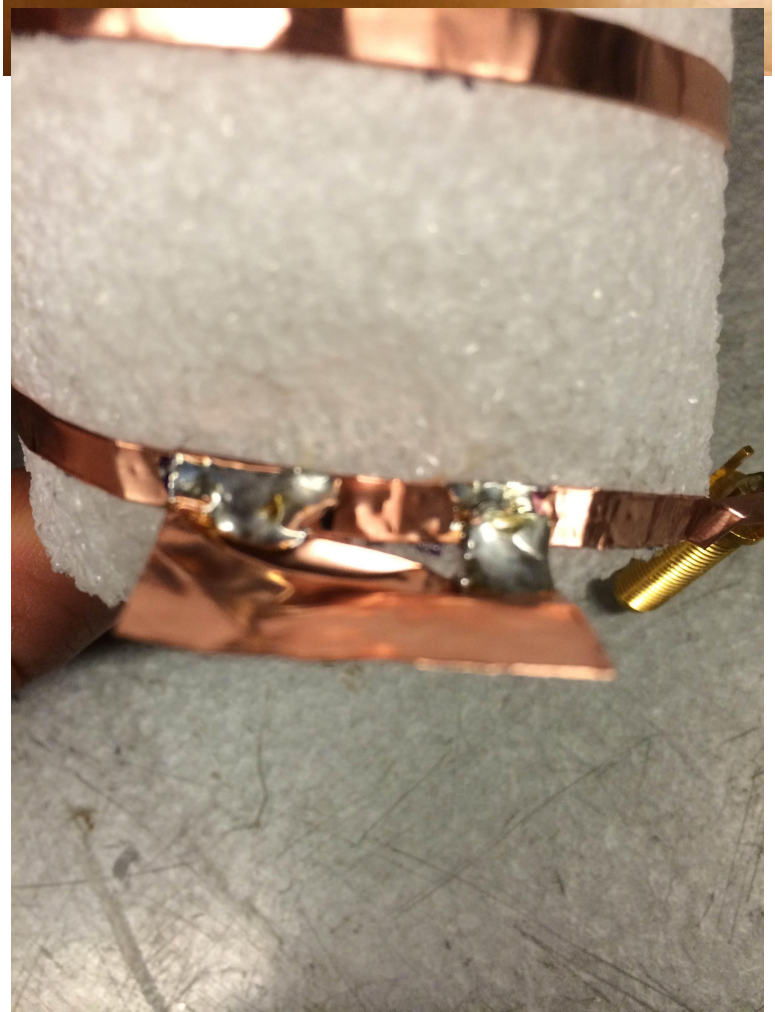
# Making helical antenna

We cut out strips of copper tape with an X-acto knife and a straight edge.

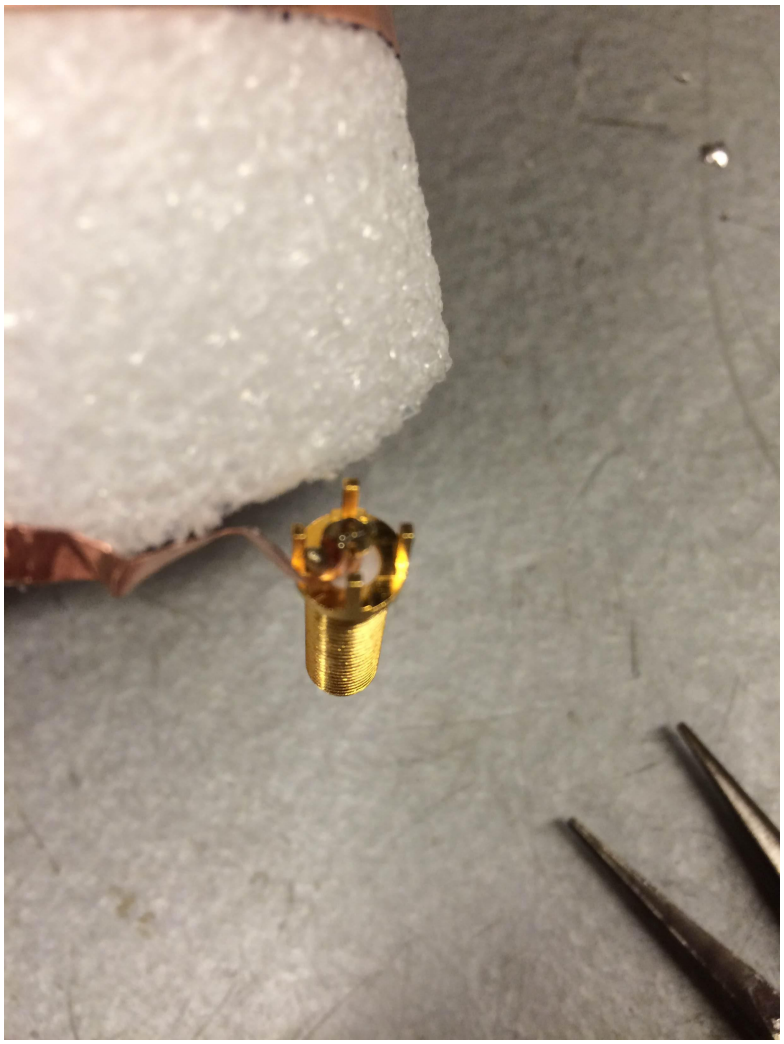


Then we measured marked the styrofoam rod at even intervals as specified by the manual to get a consistent helix with the tape.

Once the tape was applied we soldered on 26.25mmx13mm piece of copper tape for impedance matching section. Note: It is important to include an extra 3-4mm on the long side of the rectangle so that it can be attached to the helix. In the process we melted some of the rod. This should be ok as long as the tape stays in a helix.

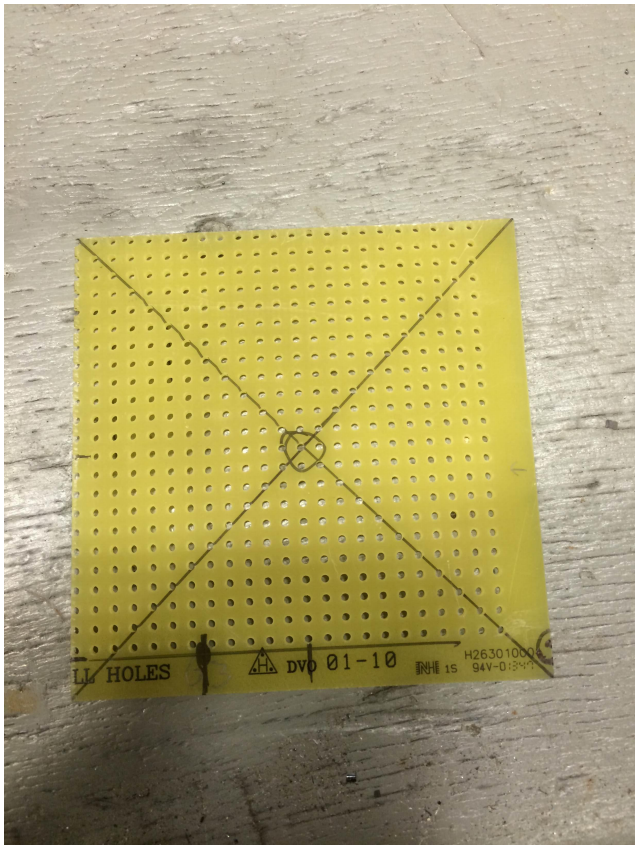


The we soldered the end of the helix to the SMA connector. MIT cut off some of the ground pins to make that easier. We left them on and it worked out the same.

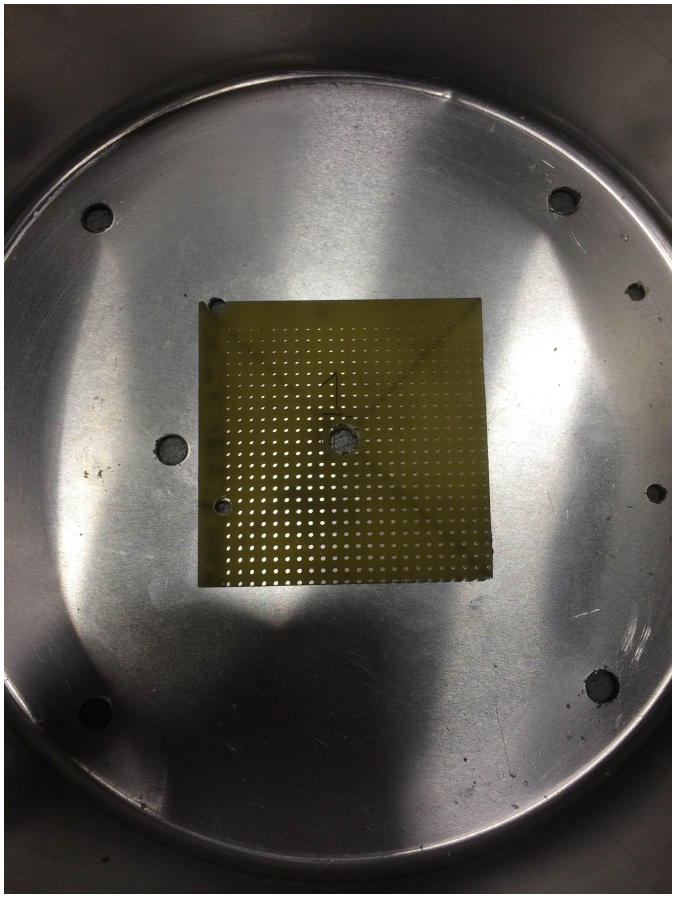


## Drilling/cutting PC board

We cut the PC board into 63mm squares (We had enough PC board to make a couple back ups).

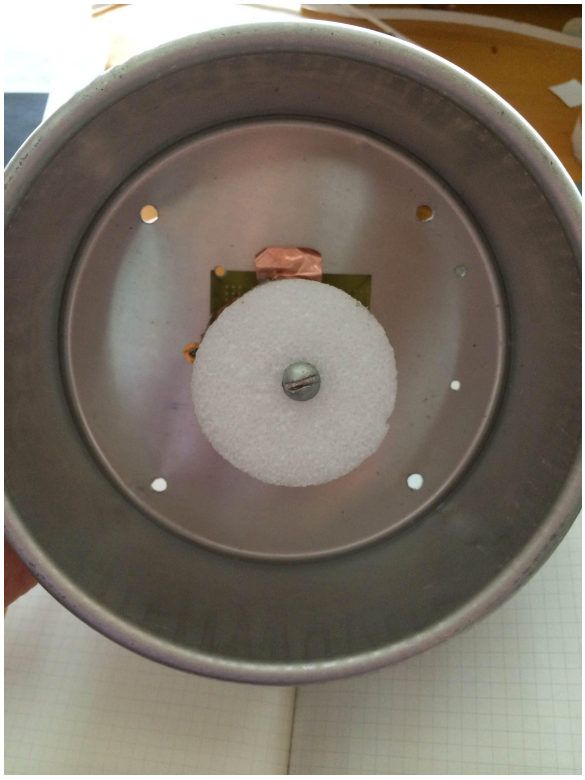


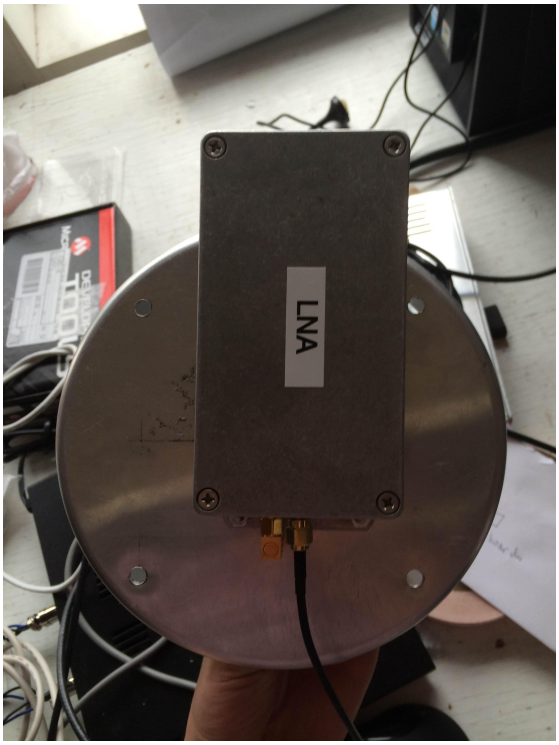

Then we drilled holes in it as specified by the manual using a drill press. It was relatively easy to secure the board in a vice for drilling.

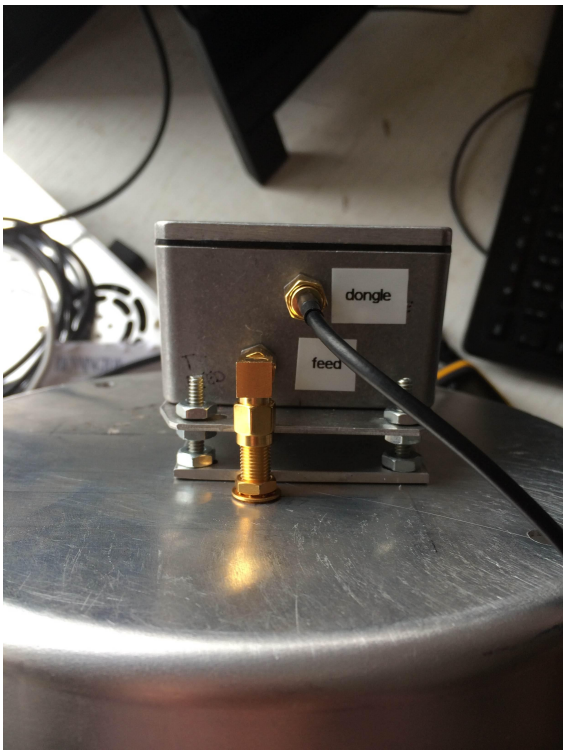


## Assembly

All the components were then bolted together.




LNA


dongle
feed

***LNA***

***Dish***

***Roof Mount***