

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

Contents

1	Introduction	4
2	Considerations	4
2.1	Acquiring Parts	4
2.2	Mounting the Dish	4
2.2.1	Non-Penetrating Roof Mount	4
2.2.2	Concrete Pier	5
2.2.3	Custom Mount	5
3	Parts List	6
3.1	LNA to Dongle Interface	6
3.2	Feed & LNA	7
3.3	Mount	7
4	Tools List	8
4.1	Assembling the LNA and Feed	8
4.2	Assembling the Dish	8
4.3	Mounting Dish	8
4.4	Building the Mount	9
4.5	Attaching Arms to Dish	9
4.6	Wiring the Rotor	9
5	Installing SRTN Software	10
5.1	MIT Haystack Observatory SRT	10
5.2	Ubuntu	10
5.2.1	Getting the Source Code	10
5.2.2	Adding Project Dependencies	11
5.2.3	Compiling the Source Code	11
5.2.4	Running the Program	11
5.3	CentOS	11
5.3.1	Getting the Source Code	11
5.3.2	Adding Project Dependencies	12
5.3.3	Compiling the Source Code	13
5.3.4	Running the Program	13
5.4	Arch	13
5.4.1	Getting the Source Code	13
5.4.2	Adding Project Dependencies	13
5.4.3	Install gcc-4.4	13
5.4.4	Compiling the Source Code	14
5.4.5	Running the Program	14
5.5	Troubleshooting	15
5.5.1	Blacklisting the Driver	15

6	Using SRTN Software	16
6.1	Configuration Options	16
6.2	Simulated Mode vs Unsimulated Mode	16
6.3	Setting Your Location	16
7	RAS SPID Rotor	17
7.1	Setup	17
7.2	Wiring	17
7.2.1	Tools Needed	17
7.2.2	Instructions	17
7.3	Controller Settings	23
7.3.1	RAS SPID rot2 Settings	23
7.3.2	Controller Reset	23
7.3.3	Controller Operation	24
7.3.4	F - Function Mode	24
7.3.5	S - Setup Mode	24
7.4	Controller Setup	26
7.4.1	Setting Limits	26
7.4.2	Setting to SPID	26
7.4.3	Set to Automatic	27
7.4.4	Controller Wiring	27
8	Building Components	28
8.1	LNA	28
8.2	Building LNA to Dongle Interface	35
8.2.1	Parts Needed	35
8.2.2	Tools Needed	36
8.2.3	Building DC Barrel Jack to F-Type Plug Connector	36
8.2.4	Assembly	37
8.3	Feed	40
8.3.1	Cutting Styrofoam Rod	40
8.3.2	Drilling Aluminum LNA Base Plate	41
8.3.3	Drilling Aluminum Cake Pan	41
8.3.4	Minor concerns	43
8.3.5	Making helical antenna	44
8.3.6	Drilling/cutting PC board	49
8.3.7	Assembly	50
8.4	Roof Mount	52
8.5	Dish	57

1 Introduction

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.

2 Considerations

2.1 Acquiring Parts

Many of the components required for this project are available through major suppliers, such as Digi-Key, Mini-Circuits, and Mouser. Many of the hardware should be available at your local hardware store, or online at a major hardware store's website. A few components are easier to find on Amazon or eBay, such as the cake pan used for the feed, but they may also be available at local retail stores. To get larger parts, such as the rotor, rotor controller, or dish, you can visit the manufacturer's website or find a retailer for what you need online. For our rotor and controller, we used products from Spid Electronik (<http://www.spid.alpha.pl/english/01.php>), and for our dish, RF Hamdesign (<http://www.rfhamdesign.com/>), though other options do exist.

2.2 Mounting the Dish

When choosing a location to mount the dish, it is best to have as few obstructions to the sky as possible. Care should also be taken to place the dish in a location with little RF interference. This means that when scoping out places where you want your dish to be, having a way to measure the RF interference would be helpful. Although you want as clear a view of the sky as possible, it may be useful to have a tree nearby to act as an absorber during calibration.

There are a few options that you can consider when deciding how to mount your dish. The two options suggested by the MIT Haystack documentation are

1. Non-Penetrating Roof Mount
2. Concrete Pier

These are good options to consider, however they are not the only that exist. Our dish is secured on a roof by a custom-built mount. Below are details on each of the three suggested methods so you can make a decision as to which mounting system will work best for you.

2.2.1 Non-Penetrating Roof Mount

Non-Penetrating Roof Mounts are ideal for flat areas, particularly roofs, paved areas, or flat fields, though they are not limited to these locations. Being non-penetrating, this

method is the least permanent, as you are able to move the mount (and thus the dish) as needed. This option appears to have a large footprint, so it would be necessary to make sure you have the area required first. These mounts are available commercially. Below are links to some websites that sell them, as described in the MIT documentation

- <http://www.skyvision.com/store/mi6012006.html>
- <http://www.orbitcommunications.com/cyberstore/cband/mounts.htm>
- <https://www.bairdmounts.com/>

Since the mounts will be dependant on the dish size and may depend on project restrictions, it may be worthwhile to search for one that will best fit your needs.

2.2.2 Concrete Pier

Building a concrete pier mount is sturdy and reliable, if constructed properly, however it is also permanent. If this is the option you choose, take into consideration that soil mechanics vary widely, so the suggestions here (taken from MIT documentation) are just suggestions. You should speak to someone familiar with your local geography/weather to ensure your pier will be stable. Things to take into consideration:

- When digging the hole for the pier to be set in, be sure it goes below the frost line
- Taper the sides of the hole inward as the hole goes down. This will help provide additional stability
- Fill the bottom of the hole with gravel to improve drainage
- Make sure the pipe coming up from the concrete foundation is level as the concrete cures
- Allow enough time for the concrete to cure fully before applying too much weight to your mounting system

2.2.3 Custom Mount

We decided to build a mount to fit our needs, and it may be that this is the easiest option. The pros of this method is that your solution will work for your design and placement of the dish. Our design allowed us to use pre-existing structures, which simplified the mount a bit. If this method is chosen, it is likely everything will have to be custom built and designed, so be sure you have resources in place for this.

3 Parts List

3.1 LNA to Dongle Interface

Part	Qty.	Manufacturer	Cost
SMA-M Crimp Connector Straight	1	Amphenol/Connex	\$5.95
LMR-400	100ft		\$102.99
Type-F Male Crimp Connector	1	Amphenol/Connex	\$4.95
In-Line Amplifier	1	Blonder Tongue	\$8.22
BNC Female Jack to Type-F Male Plug	3		\$10.99
BNC-M to BNC-M Patch Cable	3	Amphenol RF	\$12.55
Power Injector	1	Channel Vision	\$14.71
BNC-F to SMA-M Adapter	1	Vitelec / Emerson Connectivity Solutions	\$12.09
Bandpass Filter	1	Mini-Circuits	
SMA-F to BNC-F Adapter	1	AIM-Cambridge / Emerson Connectivity Solutions	\$4.21
BNC-F to SMB Plug Adapter	1		\$8.49
AC-to-DC Power Supply	1		\$20.18
Breadboard	1	Vector Electronics	\$5.75
DC Jack	1	CUI Inc	\$1.00
F-Type Plug	1	Bomar Interconnect/ Winchester Electronics	\$4.16
Approximate Total:			\$216.24

3.2 Feed & LNA

Part	Qty.	Manufacturer	Cost
Ultra Low Noise Amplifier	1	Mini-Circuits	
Coaxial Bias-Tee	1	Mini-Circuits	
Coaxial Cable 086-6SM+	1	Mini-Circuits	
Coaxial Cable 086-4SM+	1	Mini-Circuits	
SMA-F to SMA-F Adapter	1	Mini-Circuits	
SMA-F to Solder Pin Bulkhead Connector	1	Emerson Network Power Connectivity Johnson	\$8.06
SMA-M to SMA-M Right Angle Adapter	1		
3M Adhesive Copper Foil Tape	1	3M	
22 Gauge Copper Wire	1		
Semi-Rigid Coaxial Cable	1	Micro-Coax	\$4.62
F-Type Male to BNC Female	1		\$0.89
SMA-M to SMA-M Coupler	1		\$7.49
DC Barrel Jack connector	1	Digikey	\$1.00
F-type plug connector	1	Digikey	\$4.16
Cap Screw 1/4-20x3.25"			
Lock Washers 6-32	4		
Machine Screws, 6-32x1/2"	2		
Machine Screws, 6-32X1"	4		
Nuts 6-32	14		
Nuts 1/4-20	1		
Washers 1/4"	4		
Styrofoam Rod 2.5" x 24"	1		
Cake Pan 7"x3"	1		
O-rings 7/32" (inner diameter)	2		
O-rings 1/4" (inner diameter)	2		
Approximate Total:			\$26.22

3.3 Mount

Part	Qty.	Manufacturer	Cost
1/4" (3' x 2') cold rolled steel plate	1		Found scrap metal
2 1/2" (11') long steel pipe	1		Found scrap metal
1/8" 2" (6') angle iron	1		Found scrap metal
Black spray paint	1-3	Any	\$4 - \$12
Approximate Total:			\$4 - \$12

4 Tools List

4.1 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

4.2 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
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

4.3 Mounting Dish

Tool	Description
Socket Wrench + socket set	For bolting the rotor to the mount and the dish to the rotor
Ladder	Somewhere to stand while tightening fasteners
Tap and die	For minor adjustments to mounting plate if customized holes are required

4.4 Building the Mount

Tool	Description
Safety glasses	Eye protection for grinding
Welding Gloves	Hand protection
Welding helmets	Eye protection
Respirators	To mitigate inhalation of steel dust
MIG welder	For welding all the components together
Grinders with various heads	a steel brush head for cleaning surfaces to be welded and a carbide head for fabrication
Hydraulic metal Band saw	For cutting the pipe
Plasma Cutter	For cutting steel sheeting
C Clamps	For holding components while welding
Large roofing square	Useful for making sure things are perpendicular

4.5 Attaching Arms to Dish

Tool	Description
Rivet gun	For attaching cake pan receiver to arms
Band Saw	For making custom L brackets to attach cake pan to dish arms
Drill Press	For making custom L brackets to attach cake pan to dish arms
5mm crescent wrench	For tightening arm to dish fasteners
Flexible tape measure	For spacing arms evenly around a dish

4.6 Wiring the Rotor

Tool	Description
Screwdriver (Phillips)	Remove cover plate of rotor
Utility Knife	Score plate edge to remove excess paint
Adjustable Wrench	Remove/tighten water resistant wire caps
Small Screwdrivers (Phillips & Flathead)	Connect wires to the AZ / EL terminals
Soldering Iron	Tin the ends of the wire
Solder	Tin the ends of the wire
Electrical Tape	Isolate wires to prevent a short
Wire Strippers	Cut and strip the wires to use as the terminal leads

5 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).

5.1 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:

http://www.haystack.mit.edu/edu/undergrad/srt/newsrtsource_ver3.tar.gz

5.2 Ubuntu

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
```

5.2.1 Getting the Source Code

To get the source code, go to <https://github.com/BenningtonCS/Telescope-2014> and download the repository. The source code exists in the srtnver3 folder of the repo.

5.2.2 Adding Project Dependencies

In order to compile and run the program successfully, two libraries must be installed, gtk+-2.0, and libusb-1.0. This can be done using the command below:

```
$ sudo apt-get install libgtk2.0-dev libusb-1.0-0-dev
```

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 above.

5.2.3 Compiling the Source Code

A script exists to compile the source code. To compile, run it:

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

A few warnings may appear, but the code should compile successfully, assuming the needed libraries were installed properly.

5.2.4 Running the Program

To execute the program:

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

5.3 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.

5.3.1 Getting the Source Code

To get the source code, go to <https://github.com/BenningtonCS/Telescope-2014> and download the repository. The source code exists in the srtnver3 folder of the repo.

5.3.2 Adding 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 "xxx ALL=(ALL) ALL", where xxx is the username logged in to the computer, in our case 'radiotelescope'

```
## 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 libusb-devel
```

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

5.3.3 Compiling the Source Code

A script exists to compile the source code. To compile, run it:

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

5.3.4 Running the Program

To execute the program:

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

5.4 Arch

5.4.1 Getting the Source Code

To get the source code, go to <https://github.com/BenningtonCS/Telescope-2014> and download the repository. The source code exists in the srtnver3 folder of the repo.

5.4.2 Adding 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
```

5.4.3 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 gcc 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 (<https://aur.archlinux.org/packages/gcc44/>). Download the tarball (<https://aur.archlinux.org/packages/gc/gcc44/gcc44.tar.gz>). 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.

5.4.4 Compiling the Source Code

First, you will need to edit the compiler script, srtmmake to make the script use gcc-4.4 instead of the default version.

```
$ nano srtmmake
```

Then, change gcc to gcc-4.4, save, and exit out of nano.

You can now compile the software.

```
$ ./srtmmake
```

5.4.5 Running the Program

To execute the program:

```
$ ./srtm
```

5.5 Troubleshooting

5.5.1 Blacklisting the 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
```

6 Using SRTN Software

The section above details how to install, compile, and run the software. Below are some tips/guidelines on how to use the software more effectively, including setting it up to work with the controller and the LNA/Feed/dongle.

6.1 Configuration Options

The bulk of the configuration options for the software are found in the *srt.cat* file. The commands in this file are described in the help file, *srt.hlp*, which is accessible either through reading the file directly, or pressing the 'Help' button on the software UI when it is running.

6.2 Simulated Mode vs Unsimulated Mode

In the *srt.cat* file are options for

- SIMULATE ANTENNA
- SIMULATE RECEIVER
- SIMULATE FFT

When active, the command will simulate antenna motion, simulate incoming data from the dongle, and simulating the fast-fourrier transform, respectively.

To see if the software works intially, it may be prudent to simulate the receiver and antenna and run SRTN. This way you will be able to see if the software runs correctly without hardware dependencies. To recieve data from the dongle, make sure the dongle is plugged into the computer, and comment out the SIMULATE RECIEVER command. Similarly, to use the controller via software, make sure the controller is plugged in to the computer and comment out the SIMULATE ANTENNA.

6.3 Setting Your Location

In the *srt.cat* file, under the LOCATION command, update the field with the name of your location, and the latitude and longitude.

7 RAS SPID Rotor

7.1 Setup

Documentation on how to set up the rotor and the controller can be found:

<https://github.com/BenningtonCS/Telescope-2014/wiki/SPID-Antenna-Rotator-Setup—Getting-Started>

Additional information can be found:

http://www.spid.alpha.pl/download/AlfaSpid_Rotator_RAS_EN.doc
http://www.spid.alpha.pl/download/rot2prog/RotorController_sat_A3.pdf
http://www.spid.alpha.pl/download/rot2prog/RotorController_sat_zas_A3.pdf

7.2 Wiring

The document that came with the rotator can be found here for additional information on the rotator and wiring it.

7.2.1 Tools Needed

- Screwdriver (Phillips)
- Utility Knife
- Adjustable Wrench
- Small Screwdrivers (Phillips & Flat-head)
- Soldering Iron
- Solder
- Electrical Tape
- Wire Strippers

7.2.2 Instructions

The rotator we used was an Alpha Spid RAS rotor (<http://www.spid.alpha.pl/english/11.php>). These instructions are a guide to accessing the power/control terminals to the rotator in order to control it with our rotor controller (<http://www.spid.alpha.pl/english/05.php>).



Figure 1: On the rotor, there is a plate with two screws and two waterproofed wire outputs (see pictures below). On the face next to the plate should be a sticker detailing the EL AZ wiring.



Figure 2: First, unscrew/remove the two waterproof outlets using the adjustable wrench and unscrew the two screws on the plate.



Figure 3: Note that the edge of the plate may be painted over, so use a utility knife to edge the plate. The first time you remove the plate, it may be difficult, so, using the back of a screwdriver, tap around the corners of the plate to loosen it.

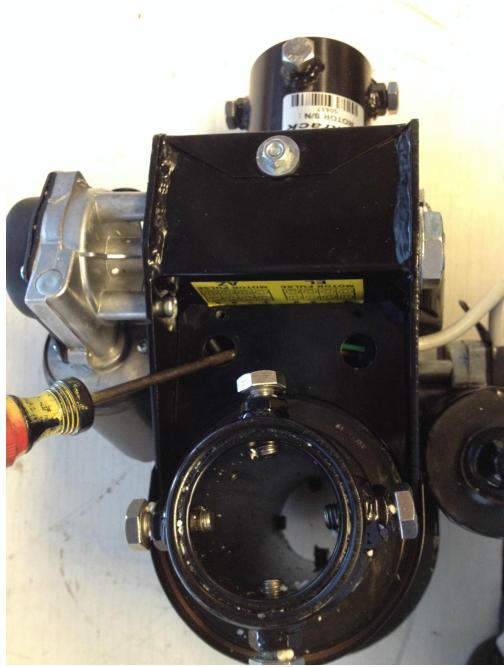


Figure 4: Using a screwdriver (or some other means of leverage), insert one end into a hole left from unscrewing the waterproof outlet, and carefully pry off the plate. This may take some time since the plate has a tight fit initially.

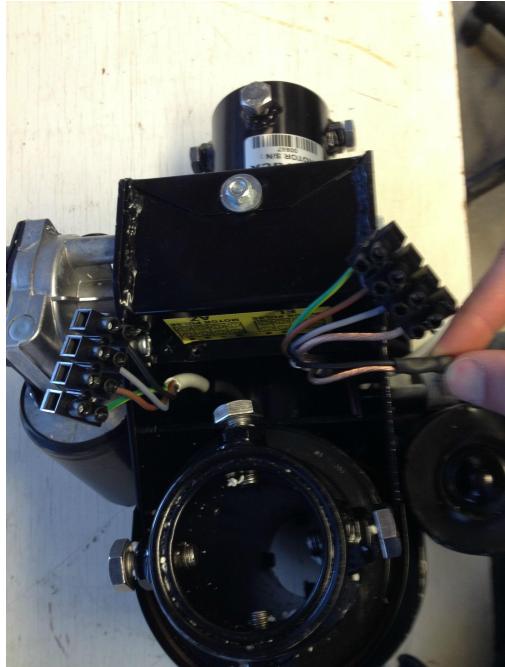


Figure 5: Underneath the plate are two wiring terminals, one for AZ and one for EL. (See the sticker on the rotor to determine which terminal corresponds to AZ and which corresponds to EL).



Figure 6: For testing purposes, a set of test cables were made, approx. 75 cm long. (Later, these wires were replaced with more permanent ones to fit our mounting strategy). Our test cables consisted of four cables, each cable containing two wires, so there were two cables per terminal, or four wires per terminal. The ends of the wires were stripped and tinned.

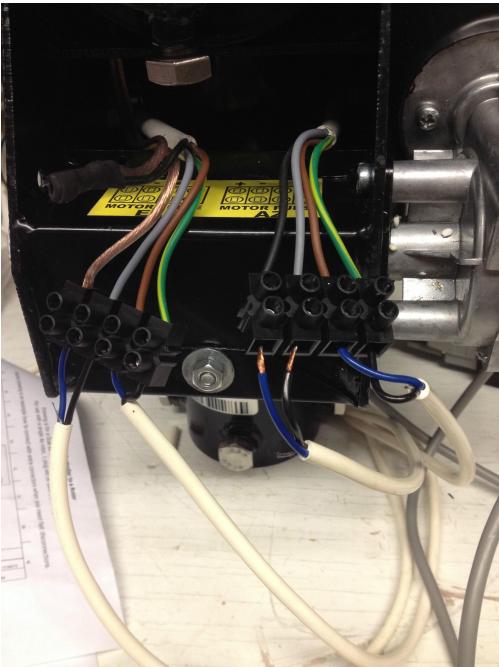


Figure 7: The ends of the wires were secured in their respective rotator terminal.



Figure 8: The free end was then soldered to a head which connects to the controller. When doing this, note the order of the wires. On the head, the pins are labeled 1-4, which will correspond sequentially to the wires coming from the terminal. Electrical tape was wrapped around the wire near the connection to ensure each pin and wire were isolated from one another. Stripping a small amount of wire on this end may diminish the need to wrap the connection in electrical tape.



Figure 9: The head was re-assembled and connected to the controller.



Figure 10: Once everything was secure and connected, the power supply and controller were turned on, with the mouse attached to the controller.

Adjusting the controller to be in 'A' (automatic mode, see the controller settings section below for a full list of controller options), we were able to control the rotor with the mouse. Note that in order for the controller to be given commands via the computer (serial), the controller must be in 'A' mode.

7.3 Controller Settings

Full documentation can be found:

(http://www.spid.alpha.pl/download/AlfaSpid_Rotator_RAS_EN.doc).

Much of this documentation was pulled directly from the source linked above.

7.3.1 RAS SPID rot2 Settings

In our setup, we are using the RAS SPID rot2 Rotator. We want to have our settings such that PS (program simulation) is configured to SP. If the mode is not SP, the controller will not work. With the rotor we are using (1 degree per pulse accuracy), we also want our rotor transmission, P, to be set to 1.0. To have the computer software or mouse control the rotor, the mode will need to be set to A (automatic). For more detailed information, see the "Controller Setup" section, below.

7.3.2 Controller Reset

Since there are no mechanical limits in the rotor, it may be installed with the antenna pointing in any direction. There is no reason to locate "TRUE NORTH" until you are ready to calibrate the control box. Use the controller to position the antenna to physically point north, then reset the controller as follows:

- Turn the unit OFF.
- While holding the F button down, turn control unit back on.
- The display should then read 0.0 0.0

This feature can be used if, for any reason, the direction of the antenna becomes incorrect. This may be caused by antenna to mast slippage or incorrect initial alignment.

Important Note:

The SPID rotator is now set at the counter-clockwise end of its normal rotation range. Normal rotation range is in a clockwise direction for 360 degrees.

From the reset position, you can rotate counter-clockwise an additional 180 degrees in over-travel, as well 360 degrees clockwise, plus an additional 180 degrees into clockwise over-travel.

Counter-clockwise over-travel is indicated by a steady dot above the over-travel icon [<->]. [<->] Rotation past 359 degrees into the clockwise over-travel is indicated by a blinking dot above the over-travel icon. [<->]

Technical Note:

You will need to leave sufficient coax length to accommodate the additional 180 degrees of over-travel on each end of normal rotation. Failure to do so can cause damage to your coax and/or antennas.

7.3.3 Controller Operation

There are multiple modes of operation for the SPID controller. The two main modes, F and S are described below, along with all of their sub-modes.

7.3.4 F - Function Mode

The F button steps through the function menus. The leftmost character on the display indicates the function mode you are currently in.

(no letter) = Normal Operations Mode

In Normal Operations Mode, the up, down, left, right buttons cause rotation as long as the buttons are pressed. Pressing S while in normal operations mode will take you to setup mode.

H = Half-Auto Mode

In Half Auto Mode, the up, down, left, right buttons can be used to pre-select the desired beam heading. The heading displayed on the controller will rapidly change in the direction of desired rotation. Once the desired beam heading is shown on the display, release the key. Approximately $\frac{1}{2}$ of a second after no key presses have been detected, the display will revert back to the actual beam heading, and rotation towards the desired heading will take place. Pressing any key while in transit to the desired heading will cancel the action.

A = Auto Mode

In Auto Mode, the controller will respond to commands from control software running on an attached computer. The up, down, left, right buttons can still be used, but pressing of any of them will cause canceling the data from software.

7.3.5 S - Setup Mode

The S button steps through the setup menu.

P = Rotor Transmission

This value defines the accuracy of rotator operation. 1.0 means operating with up to 1 degree per pulse from rotator accuracy.

PS = Program Simulation

Program Simulation allows the user to set the serial communication protocol used by the rotator. When set to emulate another brand of rotator, the Spid will respond to commands, and send responses back to the computer as if it were the rotator brand selected. If your favorite software supports a rotator, chances are, the Spid will be able to interface to your software. There are 2 modes available:

- PS SP = SPID
- PS 4A = Yaesu (GS232 protocol)

(RS232: 600N1, 8 bits)

(data rate bound 600, 1 STOP bit, no even parity bit)

Operating mode can be changed using left, right.

PH = Programmable High Limit

The Programmable High Limit is a user adjustable clockwise travel limit value. By reducing this value, the maximum clockwise rotation travel can be restricted. Use the buttons:

- *left* and *right* to adjust the azimuth value,
- *up* and *down* to adjust the elevation value.

PL = Programmable Low Limit

The Programmable Low Limit is a user adjustable counter-clockwise travel limit value. By increasing this value, the minimum counter-clockwise rotation travel can be restricted. Use the buttons:

- *left* and *right* to adjust the azimuth value,
- *up* and *down* to adjust the elevation value.

PP = Heading Adjust

This setting can be used to make minor heading adjustments without causing the rotator to turn. If you notice that the heading displayed on the controller to a known signal source is out by a few degrees, you can change the heading displayed on the LED readout to match the known heading, rather than having to turn back to North and reset the controller. These settings are made by up, down, left, right buttons.



Figure 11: The easiest way to be sure there are no limits set on the controller is by doing a controller reset. To do this, turn off the controller, hold down the 'F' button, and while still holding down the button turn the controller on again.

7.4 Controller Setup

The antenna controller will likely need a bit of setup in order to work properly with the software. To properly set up the controller, three things must be done:

1. Make sure the controller has no limits set
2. Set the controller to SPID mode
3. Set the controller to Automatic mode so it is able to communicate with a computer via serial

7.4.1 Setting Limits

To change the remaining settings, we will use the buttons on the controller, designated by the red arrows. (*F*, *S*, *left*, *right*, *up*, *down*)

If there are limits set, it is possible that when the software is run, a buffer overflow may occur, and that the GUI logic breaks, leading the program to be ineffective. If no radio data is being displayed on the UI and large values/*Nan* appear on the readout, having limits set is a potential cause.

7.4.2 Setting to SPID

In the controller's program simulation setting, two options are available, SPID or Yaesu. Since the controller we used is SPID, the controller should be set to SPID. To set, press the '*S*' button until the readout says PS. Then, if the readout does not say PS SP, press the < or > arrows to change it to SP.

If the controller is not in SPID mode, nothing will happen and the program may hang, as it is unable to communicate with the controller.



Figure 12: Controller set to SP mode (SPID mode)



Figure 13: Controller readout in Automatic (A) mode

7.4.3 Set to Automatic

In order for the computer to be able to communicate with the controller, it must be in automatic mode. To do this, you would press the 'F' button until an A shows up on the readout.

7.4.4 Controller Wiring

While the controller does not have many/complicated connections, and the documentation is fairly good for what connects where, below is a picture detailing our connection setup, for reference.

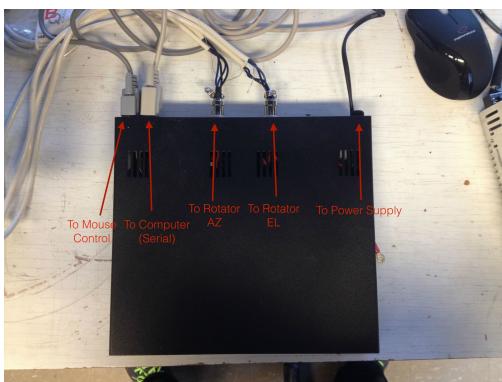


Figure 14: Controller wiring guide



Figure 15: Attached the coupler to the "In" side of Amp 1. Then attached the bandpass filter to the "Out" side of Amp 1.

8 Building Components

8.1 LNA

8.2 Building LNA to Dongle Interface

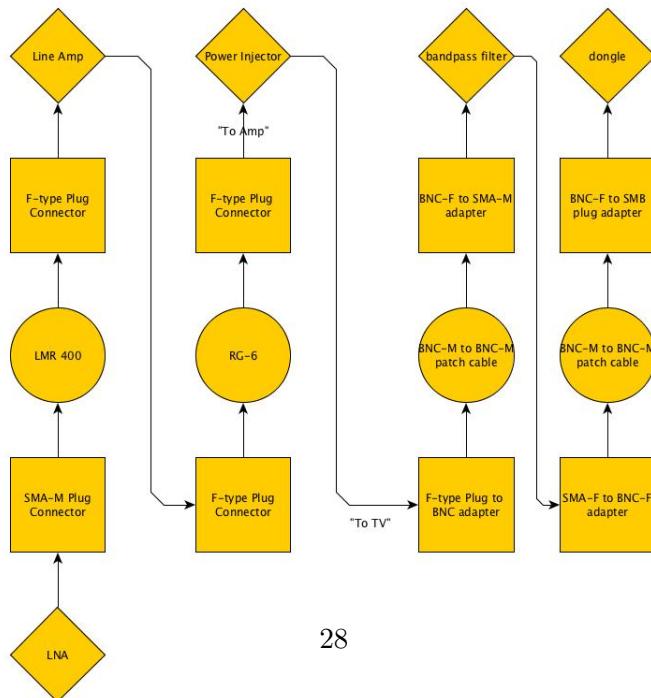




Figure 16: Connected one coaxial cable to bandpass.



Figure 17: Connected opposite end of coaxial cable to "In" side of Amp 2.

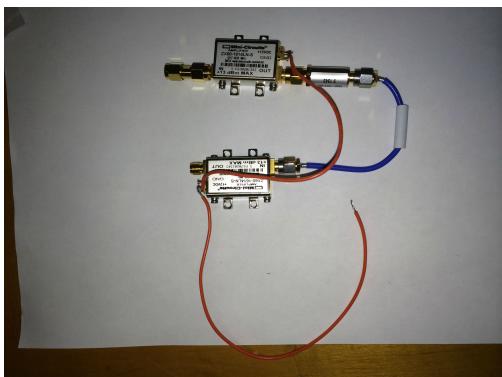


Figure 18: Cut two lengths of 15-20 centimeter long 22 gauge copper wire. Soldered one wire to both Amps. Soldered the other wire to Amp 2.



Figure 19: Soldered the other end of the second wire to the 12 volt pin on the bias tee. Connected "RF" side of the bias tee and "out" side of Amp 2 with coaxial cable.



Figure 20: Placed this group into the case, with coupler and "RF DC" side of the bias tee protruding from the holes.



Figure 21: Placed on rubber lining, and case cover, wrote LNA on top, and screwed into place.

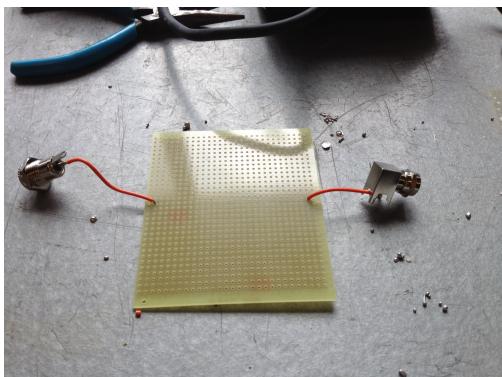


Figure 22: Pics of assembly of dc jack connector to f type coax adapter. Solder middle pin of dc plug connector to perf board. Solder middle pin of f type plug connector to same conductive strip of perf board.

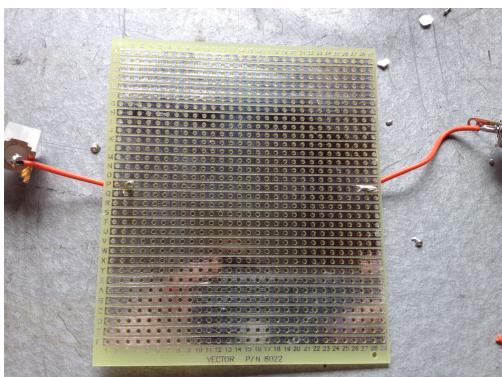


Figure 23: Under side.

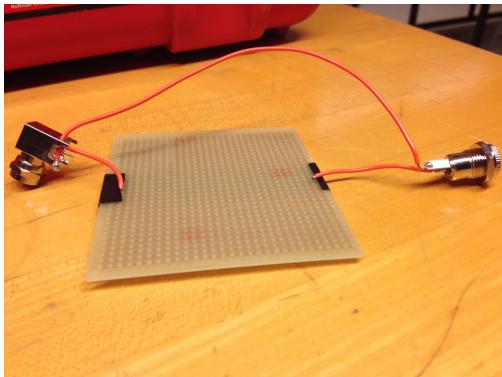


Figure 24: Solder ground pin on dc jack connector directly to one of the ground pins on the f type plug connector.

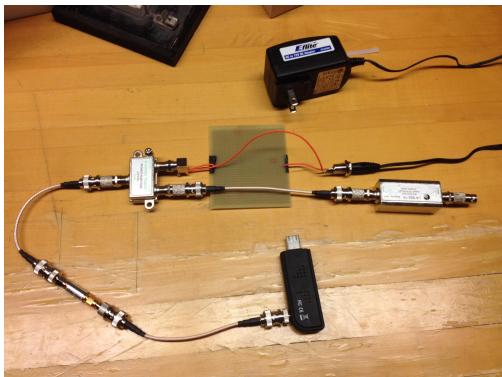


Figure 25: Back End b/t Dongle and LNA Put Together

8.2.1 Parts Needed

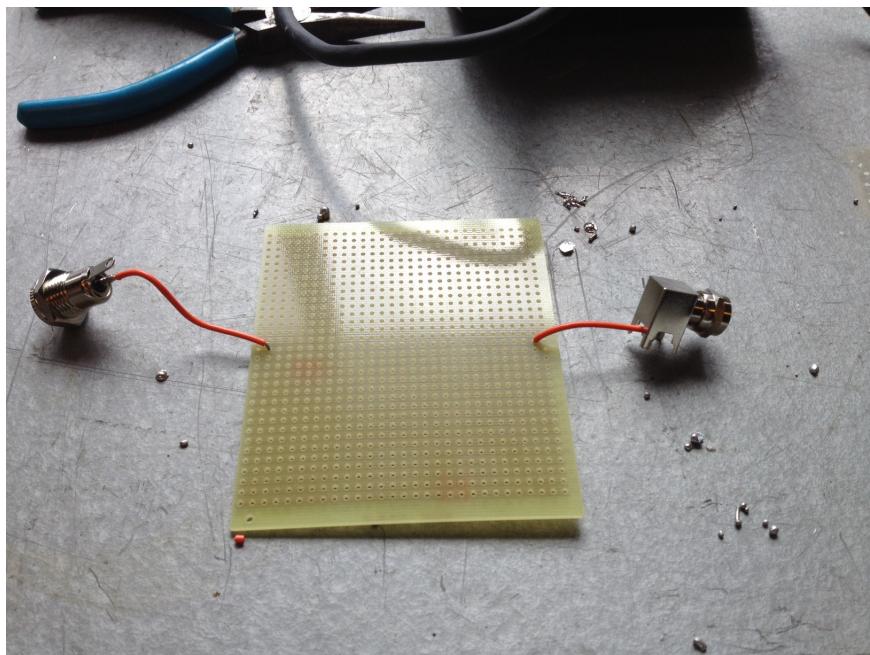
- 4 BNC jack (female) to BNC jack patch cables
- Line Amp
- Power Injector
- (DIY) DC plug (male) to F-type Plug adapter
- 1 SMA-M to BNC-M adapter
- 1 SMA-F to BNC-M adapter
- 5 F-type plug to BNC-M adapters
- Perf board/breadboard
- 22-gauge copper wire
- DC-Barrel jack connector
- Right-angle F-type plug connector

8.2.2 Tools Needed

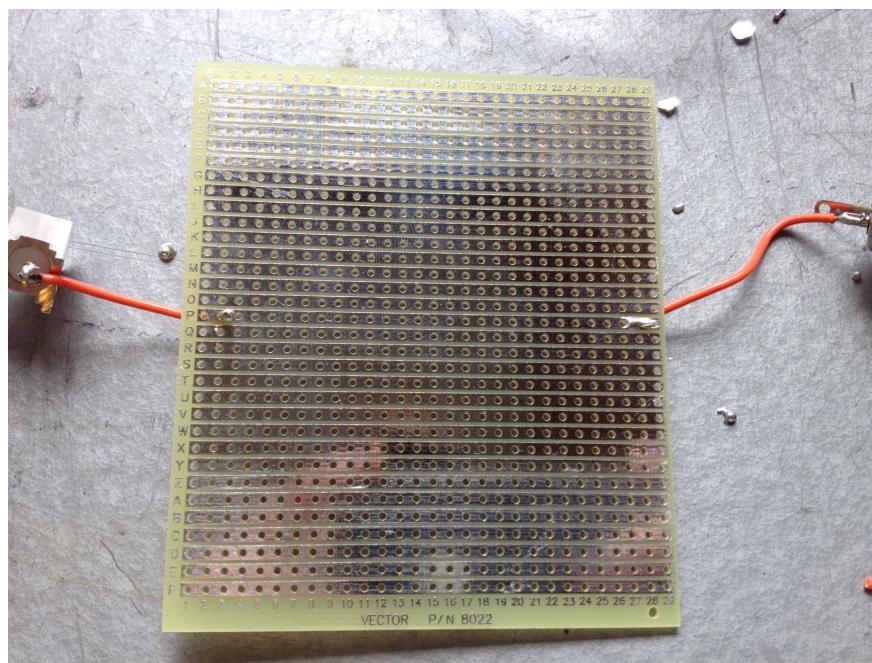
- Soldering Iron + Solder
- Wire cutters/strippers
- Electrical Tape

8.2.3 Building DC Barrel Jack to F-Type Plug Connector

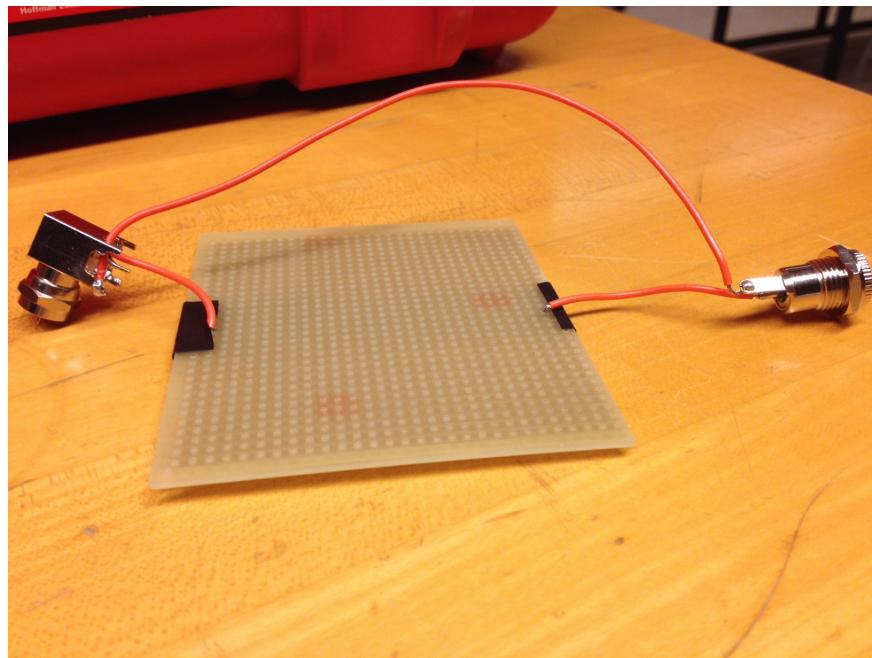
- Assemble the perf board/breadboard, dc barrel jack connector, F-type plug connector, and copper wire.
- Solder middle pin of dc plug connector to perf board.
- Solder middle pin of f type plug connector to same conductive strip of perf board.



Under side:



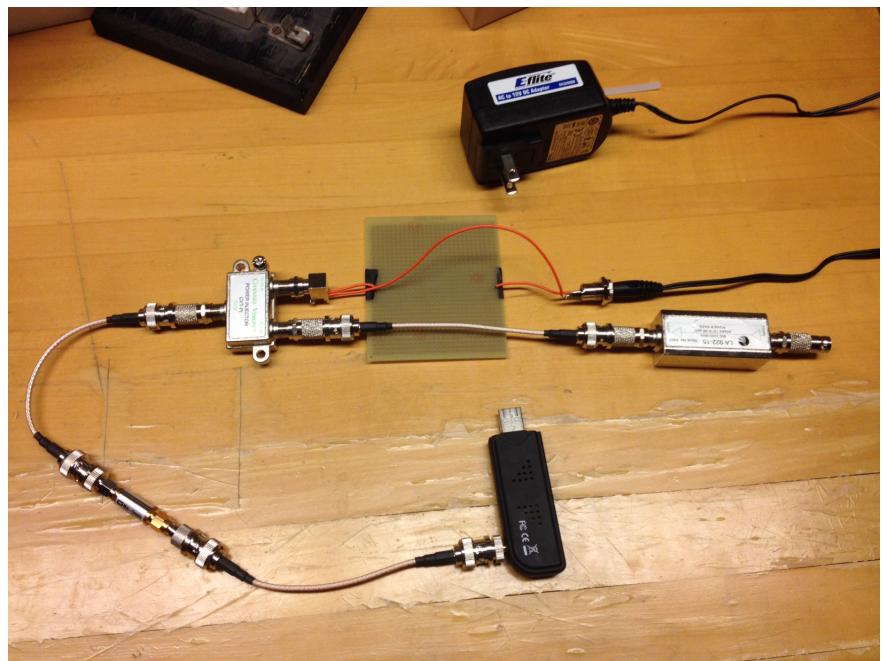
Solder ground pin on dc jack connector directly to one of the ground pins on the f type plug connector.



8.2.4 Assembly

- Attach one F type plug to BNC-M adapter to one end of one of the patch cables.

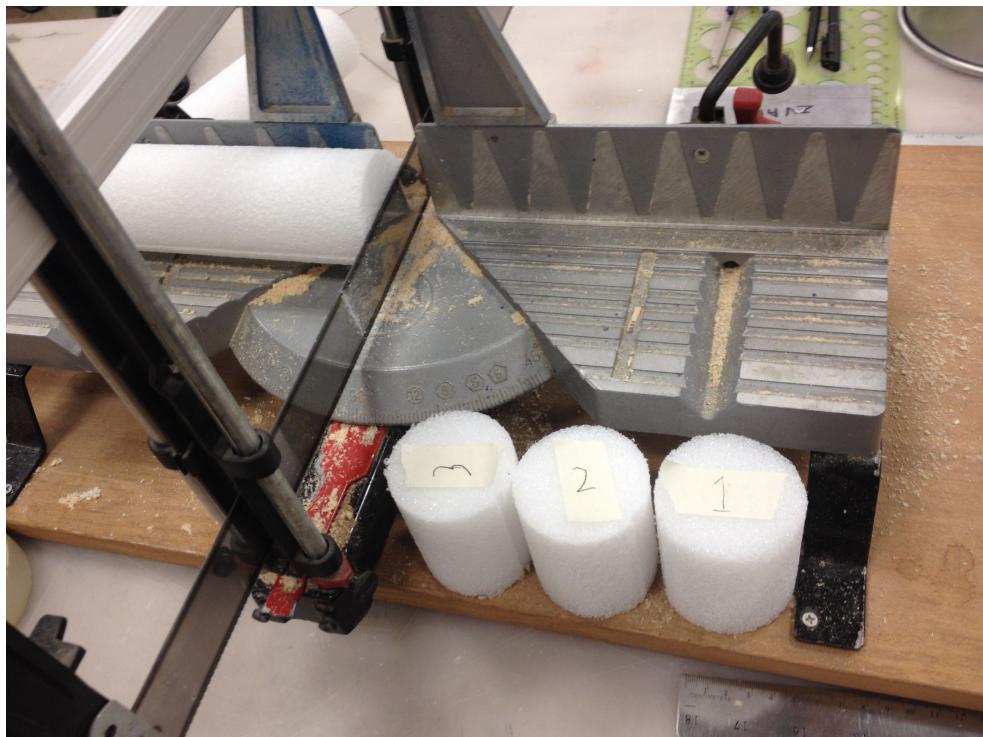
- To the other end of the patch cable, attach another F-type plug to BNC-M adapter.
- To the F-type side of the adapter, attach the "in" side of the line amp.
- To the "out" side of the line amp, attach another F-type plug to BNC-M adapter.
- Attach another patch cable to the BNC side of the adapter.
- To the other end of the patch cable, attach another F-type plug to BNC-M adapter.
- Plug the F-type side of the adapter into the "to amp" input on the Power Injector.
- Screw an F-type plug to BNC-M adapter onto the "out" of the power injector.
- Attach another patch cable to the adapter.
- Onto the free side of the patch cable, screw on your SMA-M to BNC-M adapter.
- Attach the "in" of the bandpass filter to the free (SMA-M) side of the adapter.
- To the "out" of the BPF, screw in the SMA-F side of the SMA-F to BNC-M adapter.
- Attach another patch cable to the previous adapter.
- Attach the BNC-M to SMB-M adapter to the patch cable.
- Plug the SMB-M side of your adapter into the dongle.
- Into the "12V DC IN" portion of the power injector, screw the F-type side of the dc plug to F type adapter
- Plug your 12V DC wall wart power supply into the DC Barrel Jack side of your DIY adapter.
- Plug dongle into computer.



8.3 Feed

8.3.1 Cutting Styrofoam Rod

We used a hack saw in a jig to get perfectly straight cuts in the 2.5" 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.

8.3.2 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.



8.3.3 Drilling Aluminum Cake Pan

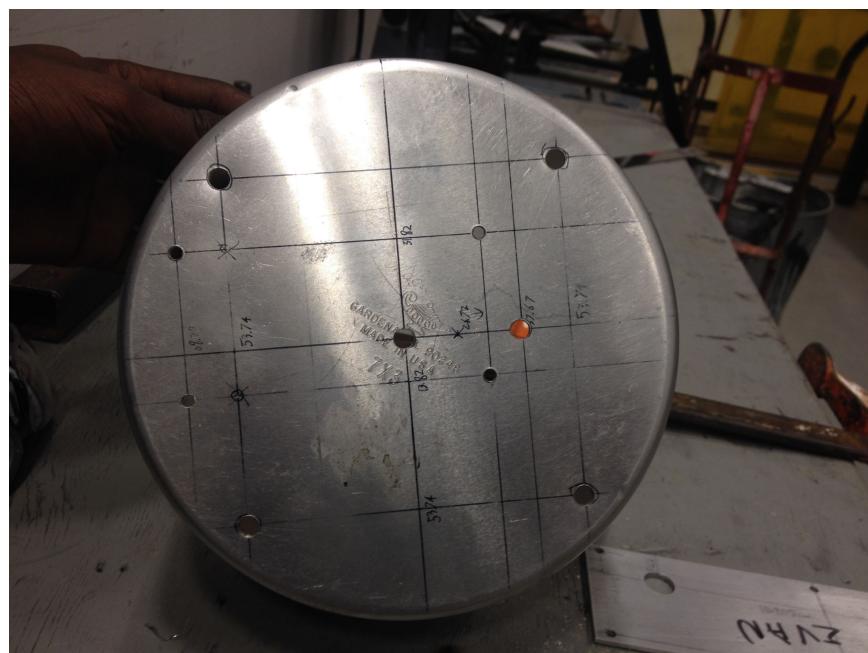
We made the measurements of the cake pan working out from the center which we found by transecting the circle and then making perpendicular rays from the center of each line cutting across the arc. This proved to be adequately precise.



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



8.3.4 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.



8.3.5 Making helical antenna

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



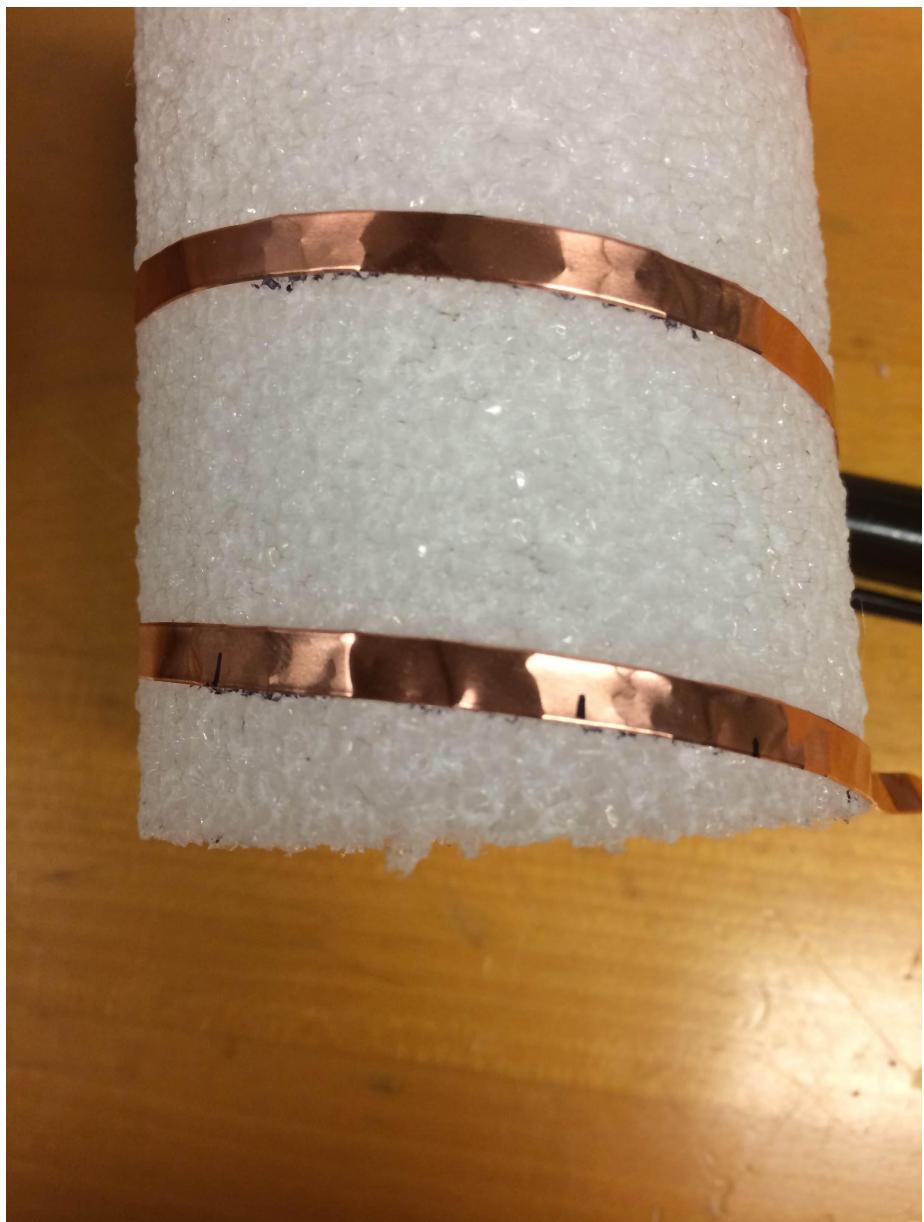
Then we measured marked the styrofoam rod at even intervals as specified by the manual to get a consistent helix with the tape.

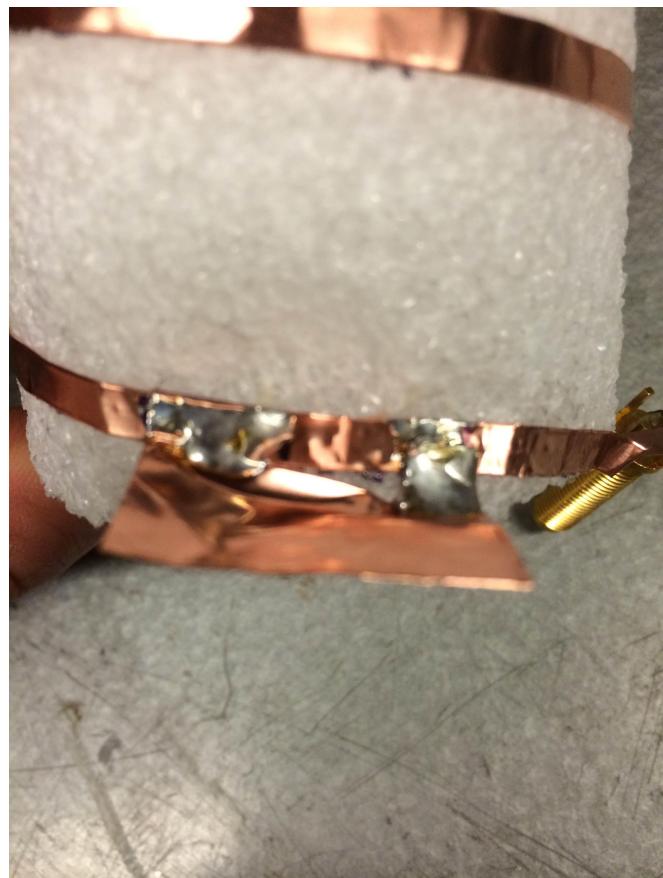




Once the tape was applied we soldered on 26.25mmx13mm piece of copper tape for impedance matching section. Note: It is important to include an extra 3-4mm on the long side of the rectangle so that it can be attached to the helix.

In the process we melted some of the rod. This should be ok as long as the tape stays in a helix.



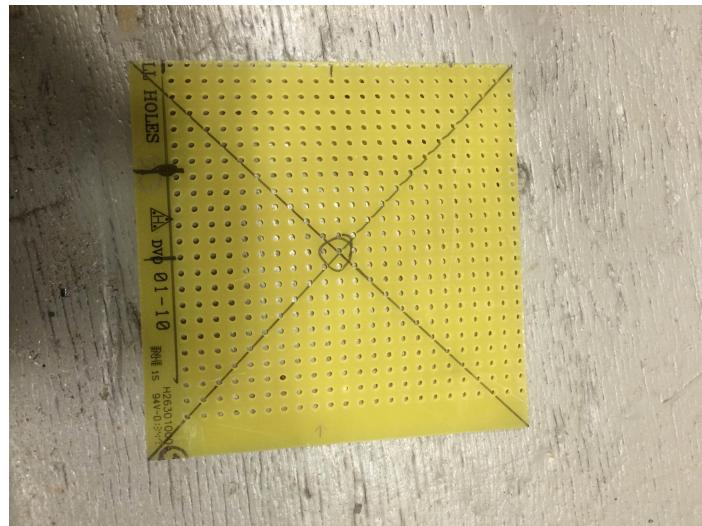


The we soldered the end of the helix to the SMA connector. MIT cut off some of the ground pins to make that easier. We left them on and it worked out the same.

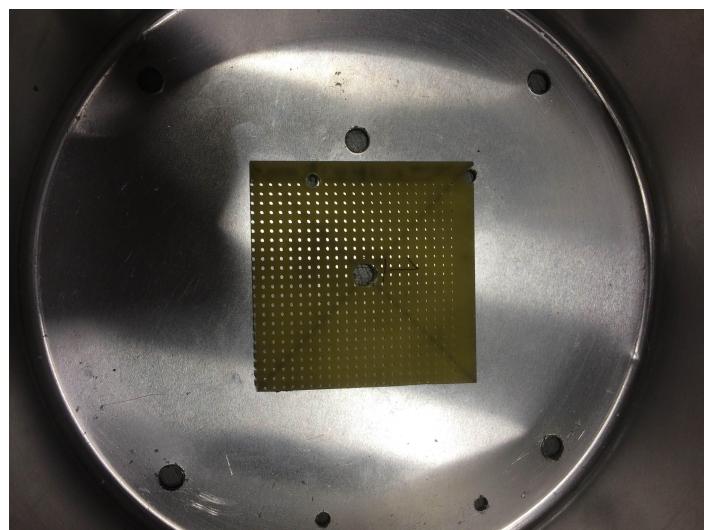


8.3.6 Drilling/cutting PC board

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

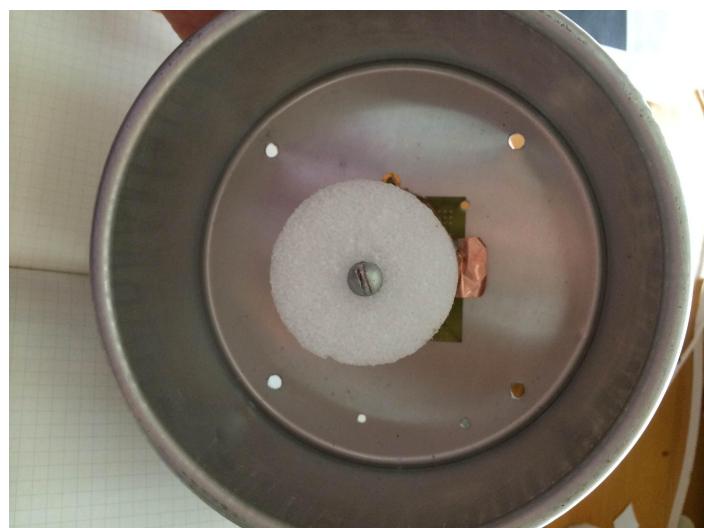


Then we drilled holes in it as specified by the manual using a drill press. It was relatively easy to secure the board in a vice for drilling.



8.3.7 Assembly

All the components were then bolted together.







8.4 Roof Mount

This section outlines the method we used for constructing our custom-designed dish mount. It was designed to use few materials, mostly scrap metal, and attaching the whole mount to pre-existing structures (in our case, a clerestory) on the roof of a building. Other options for roof mounts exist and not all methods will follow from what we have done below, but it may still be worthwhile to read through to get an idea of the process and the considerations made.

We used scrap metal to make our dish mount, so we started by cleaning the metal and cutting it to size. We plasma cut a 1.5' x 3' rectangle of 5 gauge steel (about 3/16" thick, it later turned out this is too thin) and cleaned up the edges with a grinder.



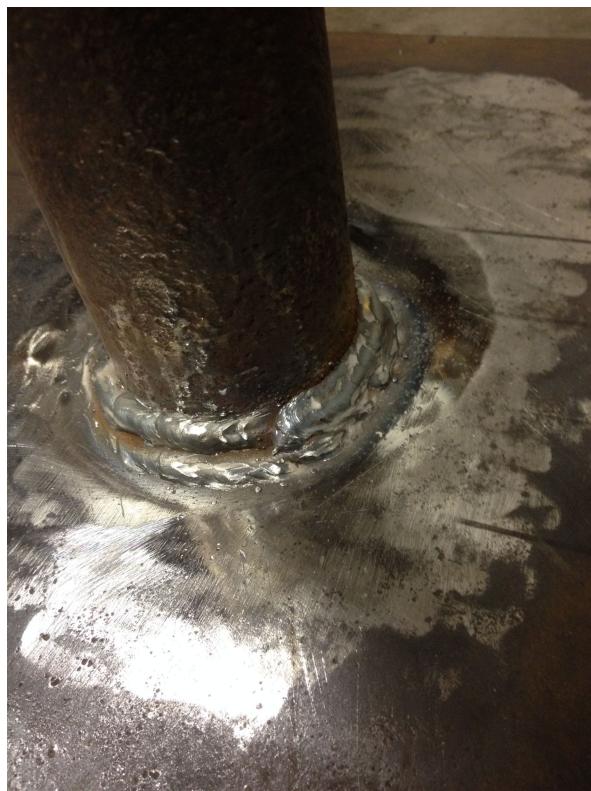
⁴⁸Figure 26: The clerestory our mount would be attached to.



We cut an 11.5' length of 2.5" (exterior diameter) pipe and cleaned it using a steel brush head on a grinder.



We then MIG welded the pipe to the plate. We reinforced the weld with a couple extra beads because most of the force in the system will be directed to this point.



The real trick was getting the pipe perpendicular to the plate.



We ended up clamping the plate on one side of the table and then using the corner of the table as a perpendicular straight edge in two dimensions.

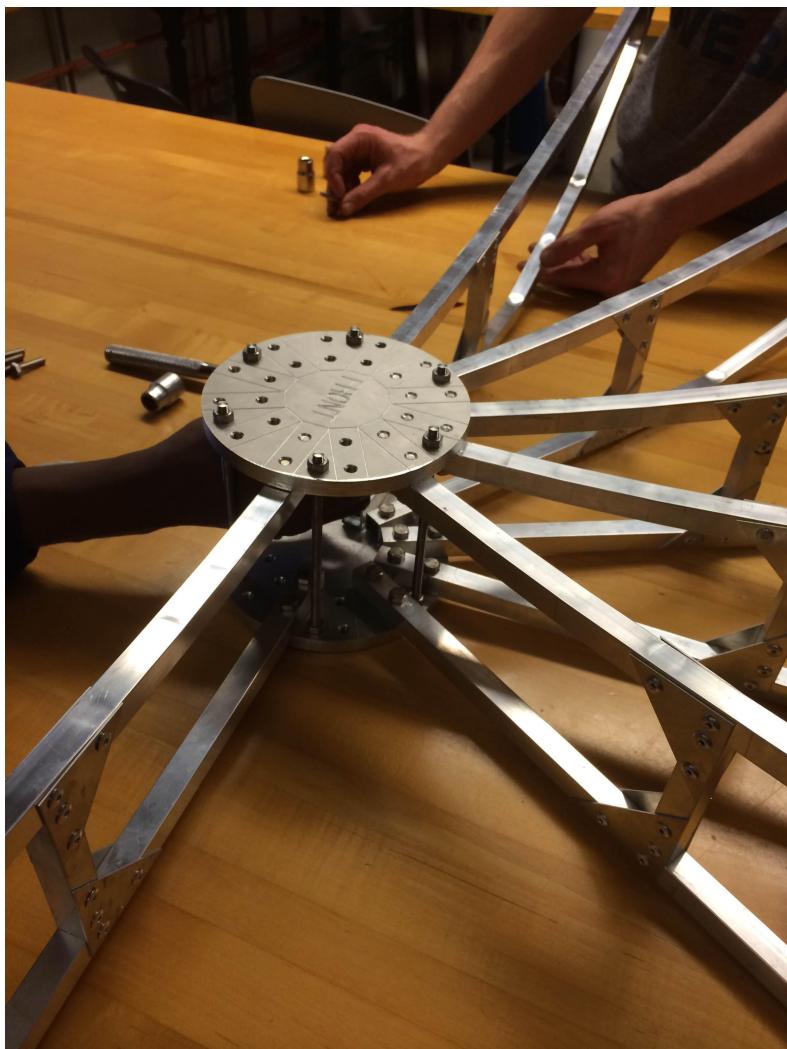
After some tests we realized the weight of the pipe was warping the plate so we successfully reinforced it with some angle iron.

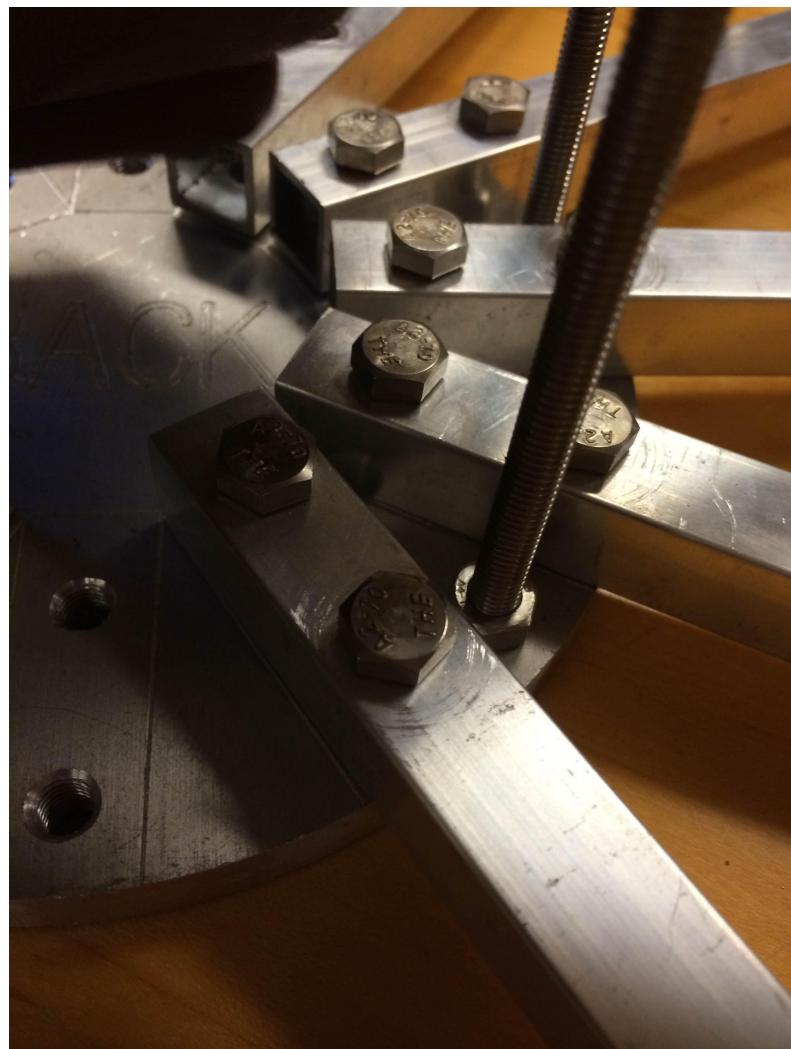


8.5 Dish

We bolted together the central plates, trying to make sure they are parallel to each other.

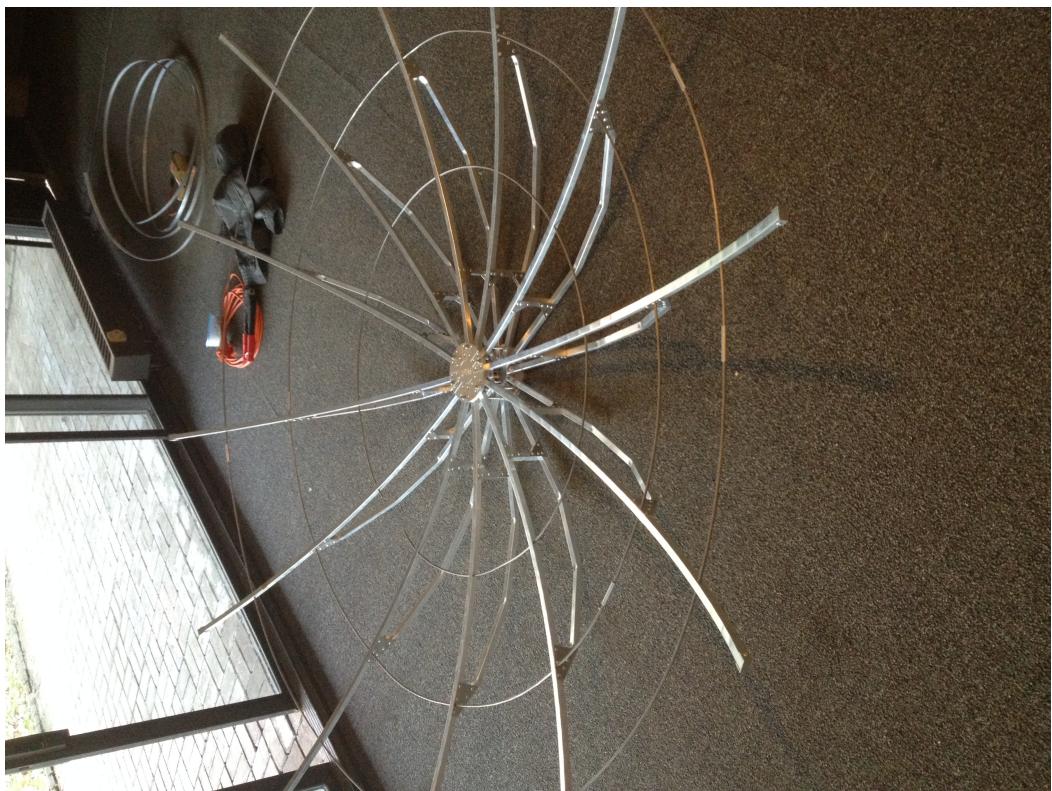
Then we bolted on half of the dish arms which confirmed the accuracy of our central plate construction. This is as much of the dish as will fit out the door to the roof we will be mounting the dish on. The rest of the assembly happened on site.







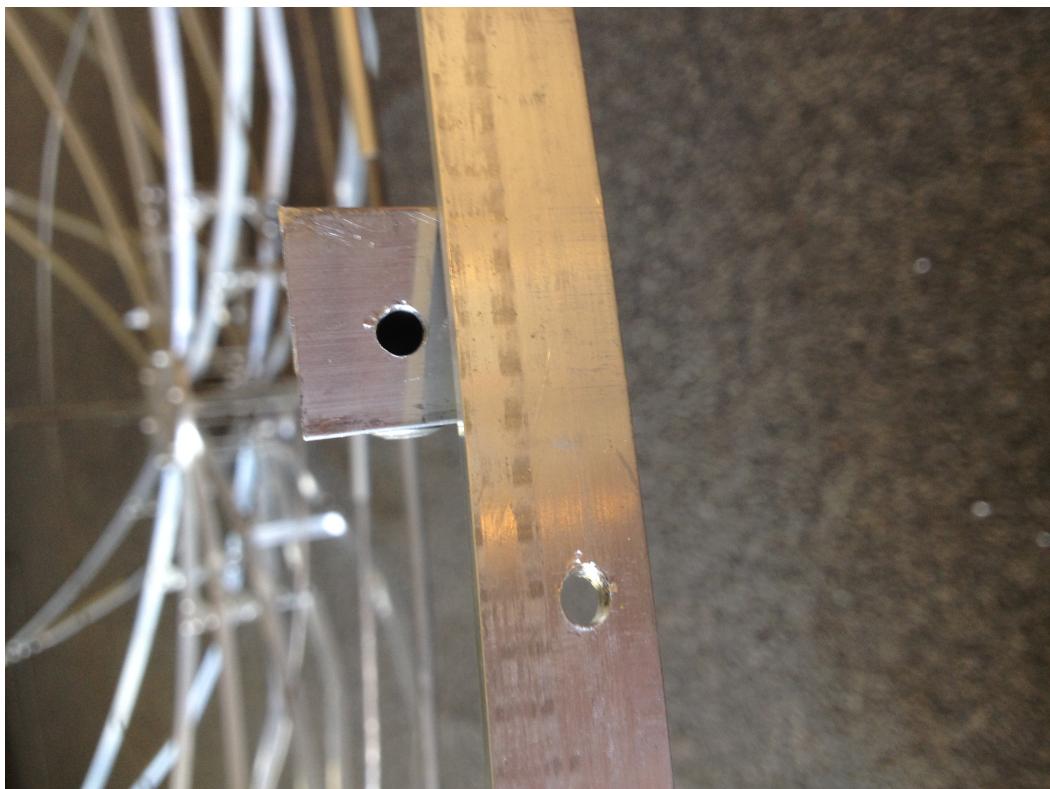
Once the ribs are put together and the support rings are in place, we are ready to put on the outer band.



To do this, a hammer and tap were used to make dents in the end of the rib and the band

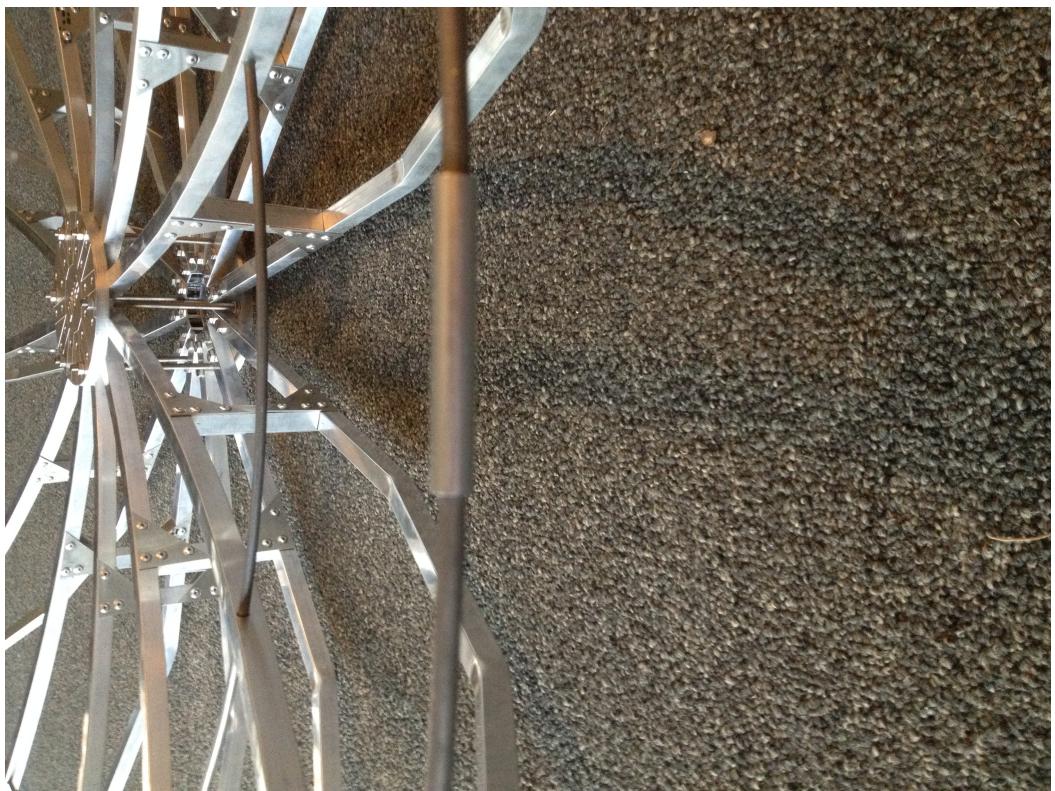


Then, using a power drill, holes were cut in both so they could be lined up and riveted.



This process of tapping, drilling, and riveting was repeated for each of the 12 ribs of the dish. Once complete, we were ready to start putting the mesh on.

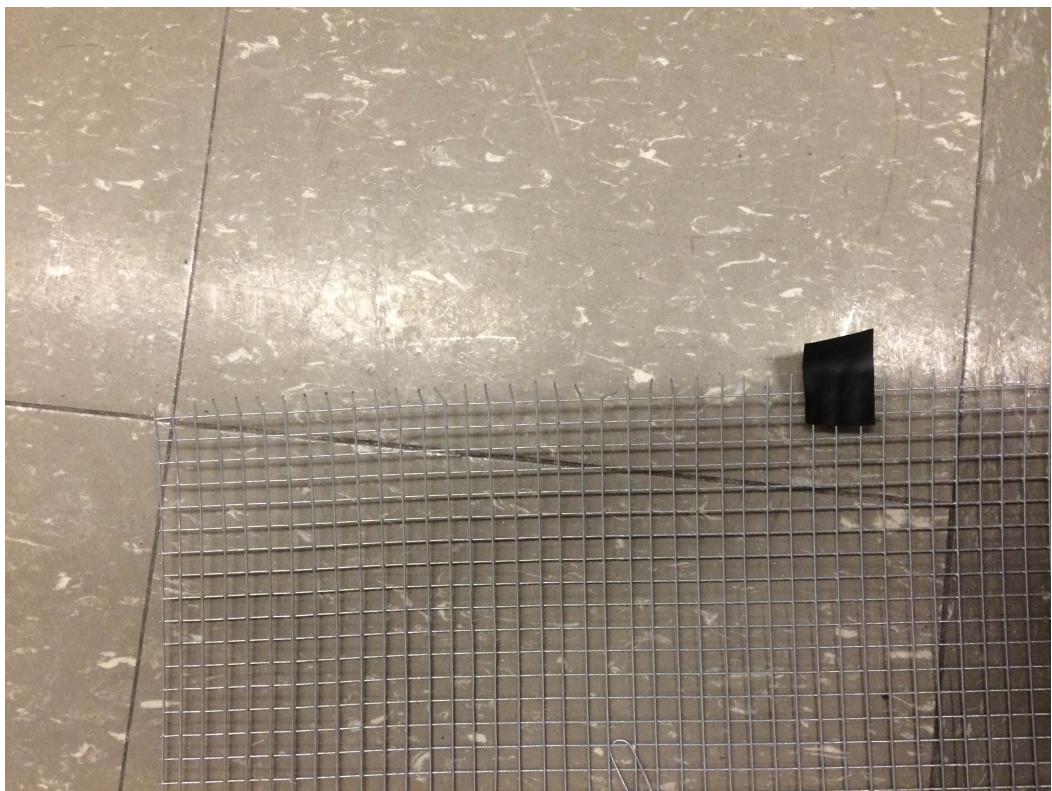
Make sure all the ends of the support rings are together. We did not crimp them, since they seemed secure enough so that they would not fall out.



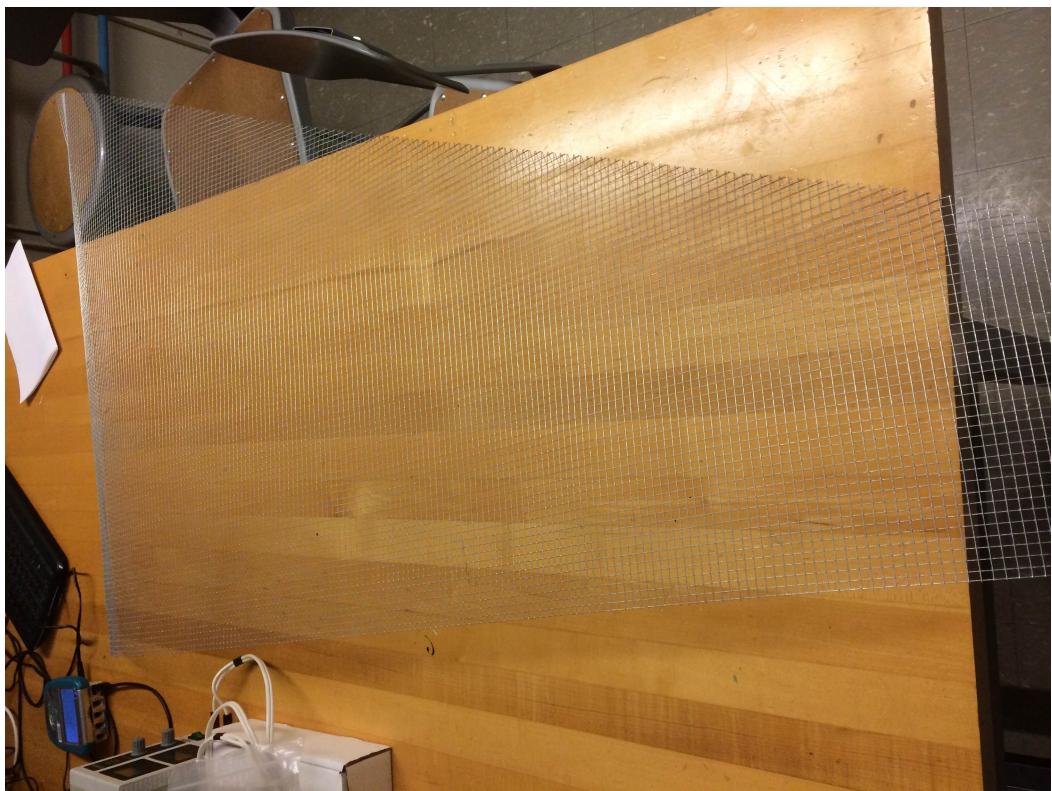
Cutting the mesh!

Below are some pictures that detail the mesh cutting process.





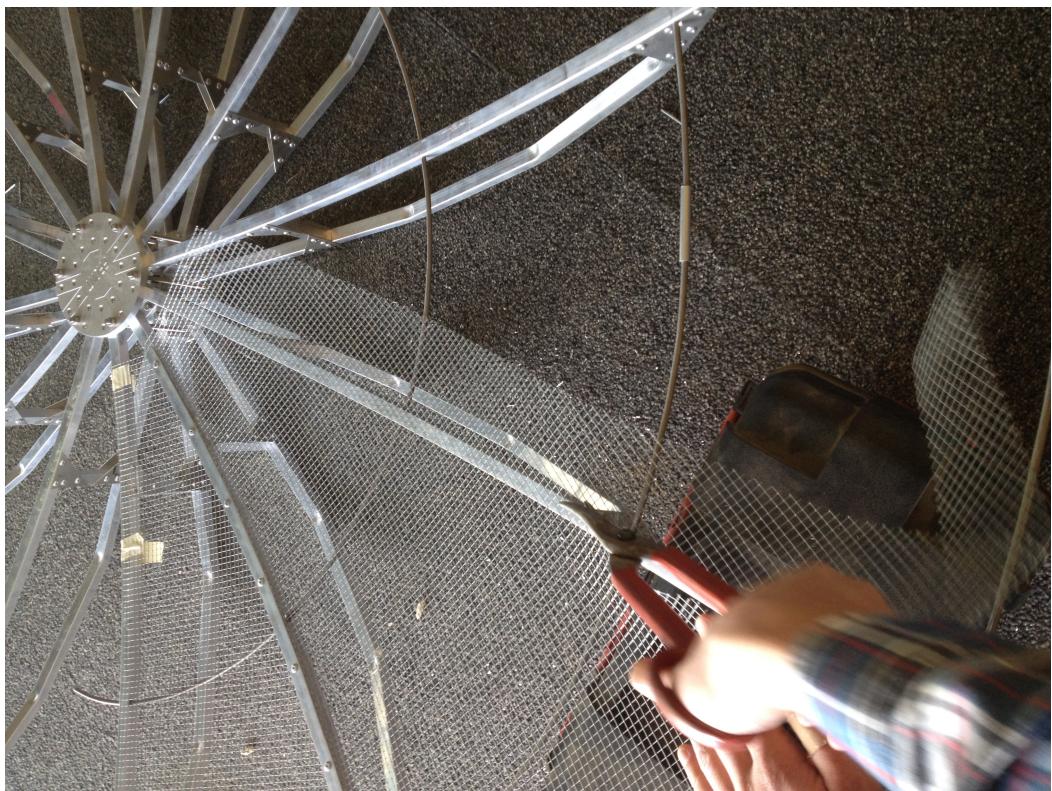




After the Mesh was cut to specifications, it was laid out on the skeleton of the dish, to make sure there was enough material and it would come together properly



After we were sure that we had enough material all cut to the right size to allow for some extra, all the sheets were removed except for one. One end of the sheet was secured to a rib using tape, while the other end was trimmed down to be flush with the next rib



Then, the next sheet is placed on top of the end of the first sheet so they are overlapping, and a strip with pre-drilled holes (which came with the kit) were lined up on top of the rib, and holes were drilled through the rib at the places where the holes in the strip were



The strip, with the overlapping pieces of mesh underneath, was then riveted to the rib.



This process was repeated for all 12 sections of the dish until it was complete.



Then, the edges were trimmed, leaving about 1/4 to 1/2 inch mesh past the outer band (enough to wrap around the band a bit)



It was then wrapped around the band



And secured using zipties (4 zipties per section * 12 sections = 48 zipties)

