VLA/VLBA Interference Memo # 19

Radio Frequency Interference Environmental Monitoring Station (EMS)

New Mexico Tech
May 5, 2001

Group Members:
Anh Nguyen
Thu Nguyen
Kerry Shores

Customer:
NRAO,
Raul Armendariz

Project Advisor:
Dr. Robert Bond

Radio Frequency Interference Environmental Monitoring Station (EMS)

New Mexico Tech
May 5, 2001

Group Members:
Anh Nguyen
Thu Nguyen
Kerry Shores

Customer:
NRAO,
Raul Armendariz

Project Advisor:
Dr. Robert Bond

Table of Contents

Table of Figures

Abstract

For Senior Design project, our group improved and calibrated an existing electromagnetic interference system (EMS) at the Very Large Array (VLA). The data produced by this system is used for design of the Expanded VLA and determine the frequency range where RFI interferes with astronomical observation. This report discusses our approach to fixing and enhancing the system.

Introduction

The Radio Frequency Interference Environmental Monitoring Station (EMS) is a dedicated system that utilizes calibrated small antennas with an azimuth-elevation rotator, ambient low noise amplifiers, and a receiver to monitors the radio frequency (RF) environment at the Very Large Array (VLA) outside of Magdalena, NM.

This system continuously logs data, 24 hours a day, 7 days a week. The output of this system is in the form of 3D grayscale plots, which can then be analyzed for potential interference to the VLA's observations. This is a necessary function because of the increase in terrestrial emissions near the radio frequencies favored for observation by the scientists at the VLA. If emissions get too close to a frequency of interest to the scientist, the emissions can wash out any data the scientist might hope to receive. Also, with an expansion project planned for the VLA, new (more sensitive) receivers and a correlator will be built, necessitating a thorough knowledge of strong emissions that could hurt equipment or degrade data. A working radio frequency environmental monitor has now become critical. It was the purpose of our Senior Design project to take the old EMS (which was non-operational) and deliver a working system that was automated, remote controllable, calibrated and reliable.
Note: Due to the demand of our customer, this project was divided in two phases. First, perform survey of the RFI signals at the VLA from 1 to 2 GHz and identify the strongest signals in each frequency band using the current set up (See Figure 1.) Second, implement a lightning protection scheme for the system and implement the band-switching capability, both remotely and manually, in the front end.

This report includes three sections. The first section discusses the process of characterize the existing system (for clarity, we will call this system, System I) so that data can be collected reliably for the survey. The second section discusses the process of designing the front end and a lightning protection system. The last section includes all the software implementation of the system

Part I: Characterize the System

Figure 1 shows a simplified diagram of the old EMS system (System I). System I consists of an Omni-directional Discone antenna mounted on top of a 50 ft. Rohn Tower. The signal is amplified by a JCA amplifier connects at the input to the AILTech NM67 receiver. The PC "Snow" contains software that controls both the receiver and the CCI-7 Controller. We will use this set up to collect the data for our survey

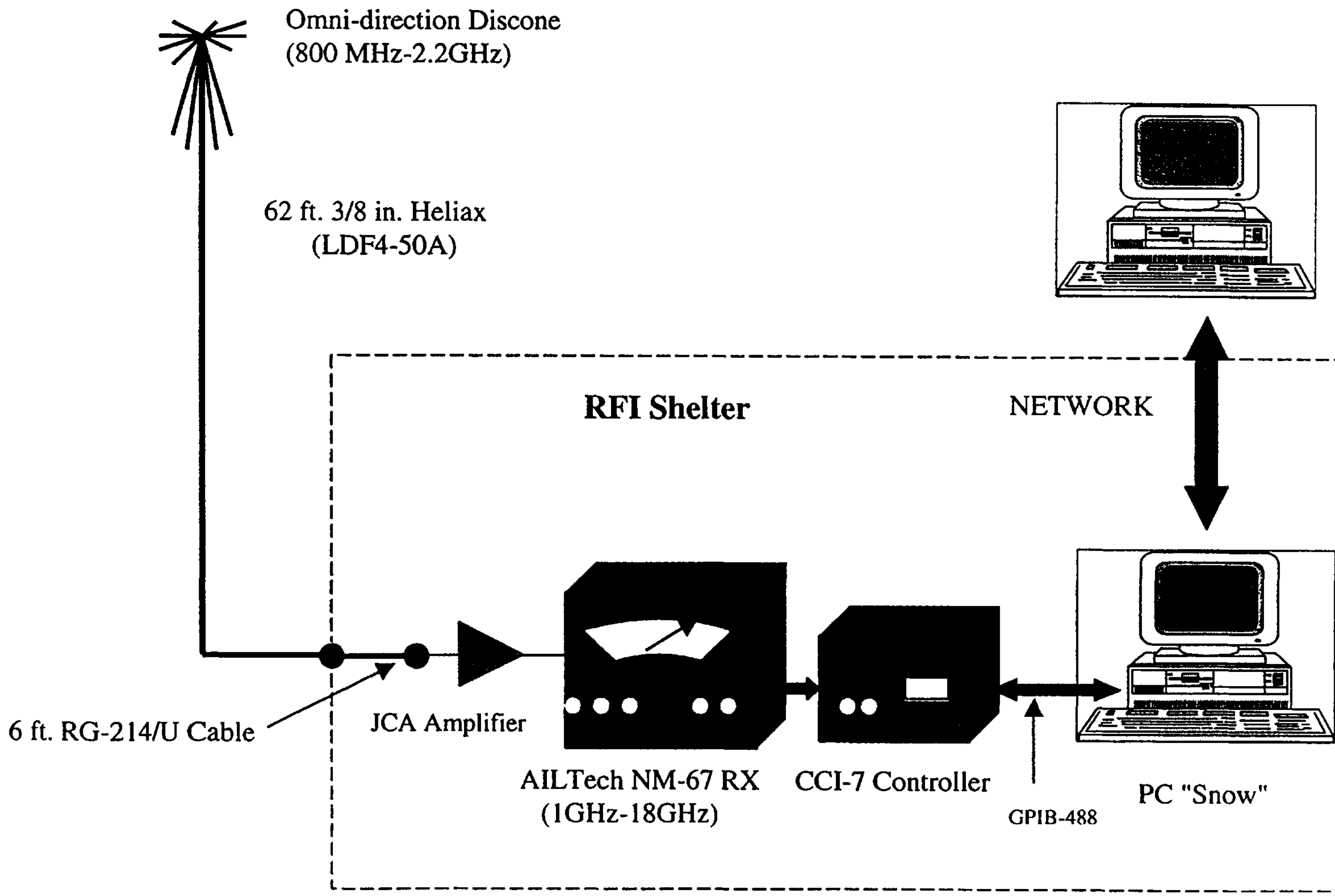


Figure 1: EMS System Set up

In order to determine the absolute power flux density of the RFI incidence at the VLA, we needed to characterize the gain pattern of the Discone antenna, transmission line loss, and amplifier gain.

Note: The result of the following antenna and amplifiers characterization will only be used in the first phase of this project for the purpose of the survey. For the final system (see Appendix A for a diagram of the system,) we will use two different antennas and also different amplifiers in the front-end.

1. Characterize the transmission line

Two different types of RF cables are used: the 3/8' Heliax cable and the RG 123-U cable. Using the line loss values provided for each type of cable, we were able to estimate the theoretical line loss for the two transmission lines. The values are shown in Table 1.

The actual line loss was found by injecting a signal of a known amplitude and frequency to one end of the line and measure the output at the other end of the line using a power meter. The difference between the two values is the approximate line loss. The measurement is repeated as the frequency varies from 1 to 12 GHz.

The result of the measurement shows a very high value for line loss (i.e., a loss of about 30 dB at 10 GHz.) The reason for this loss might due to bad cable, and or bad connectors. Using a reflectometer, we found that there were two bad connectors used to connect the cables. These bad connecters account for the high line loss. After replacing the bad connectors, we repeated the measurement for the line loss. The results are shown in Table 1.

Table 1: Line Loss Characterization

Frequency (GHz)	Theoretical (dB)	Actual Loss (dB)	Difference
1.0	(1) (2) (3) 2.12 + 0.52 + 1 = 3.64	3.88	0.19
1.5	2.7 + 0.68 + 1 = 4.38	4.33	0.05
2.0	3.16 + 0.82 + 1 = 4.98	5.19	0.21
3.0		6.78	
4.0	4.8 + 1.3 + 1 = 7.1	10.67	3.57
5.0	6.1 + 1.76 + 1 = 8.86	13.72	~~3.6~~ 4.86
6.0		12.46	
7.0		14.72	
8.0		14.91	
9.0		14.64	
10.0	8.4 + 2.55 + 1 = 11.95	14.99	3.04
11.0		14.57	
12.0		14.73	

Notes:

(1): 62 feet of the 3/8' Heliax cable

(2): 6 feet of the RG124-U Cable

(3): Estimated loss due to the connectors.

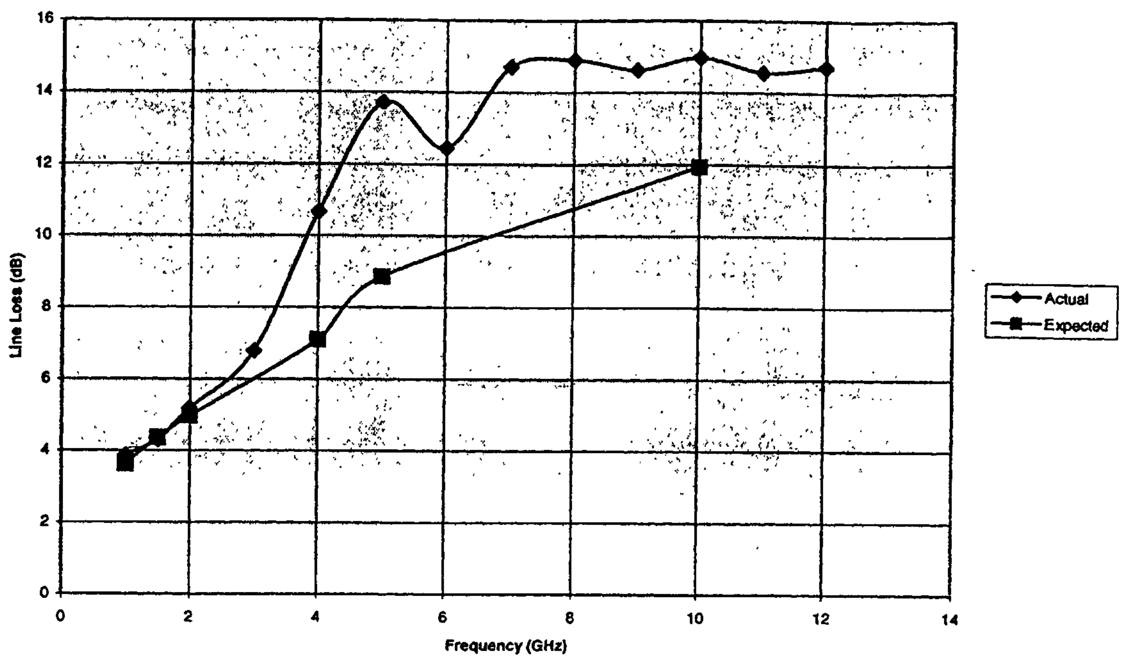


Figure 2: Plot of the Actual and Expected Line Loss

2. Characterize amplifier gain

Low noise amplifiers (LNA) are placed at the input to the receiver to amplify the signal. In order to properly calculate the power flux density incidence on the antenna we need to find out the gain of these amplifiers.

The amplifier characterization was done using a logic analyzer and a function generator. First, inject a signal at a known frequency (1.5 GHz) and amplitude to the amplifier and measure the power displayed on the logic analyzer. The difference between the power displayed on the logic analyzer and the power of the signal inject to the amplifier is the gain of the amplifier. Next, change the function generator to sweeping mode from (1GHz to 3 GHz) to find the operation frequency range of the amplifier.

The above procedures were used to measure the gain of 3 amplifiers: a JCA and two Miteq amplifiers. The manufacture specs sheet on the Miteq amplifier shows that it operates in the frequency range between 800MHz to 2 GHz with a gain of 18 dB. Table 2 shows the result of the measurement.

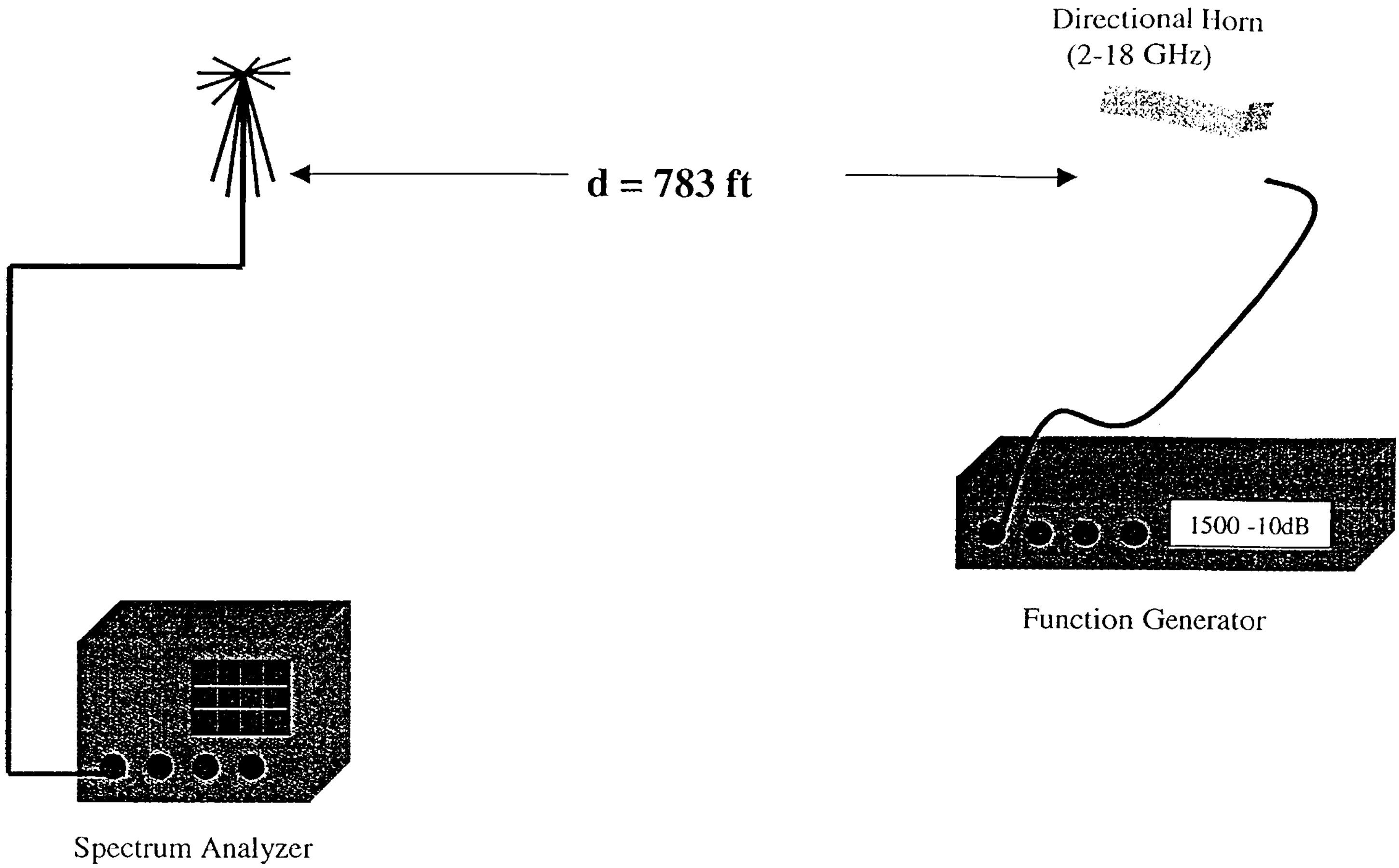
Table 2: Amplifier Gain at 1.5 GHz

Set up	Gain (dB)
1 Miteq Serial No. 39037	16
1 Miteq Serial No. 43599	17
2 Miteq in series (43599-39037)	34
2 Miteq in series (39037-43599)	36
JCA amp. Serial No. 134	28

Although the JCA provides a lower gain than the two Miteq amplifiers, we decided to use the JCA amplifier because when using the two Miteq amplifiers together we get a higher noise floor. Also, when putting the two Miteq amplifiers together, the second amplifier goes into saturation causing broadband noises.

3. Characterize the gain of the Discone antenna

The following setup was used to characterize the antenna gain. The testing antenna is mounted on top of a 50 ft tower with an azimuth-elevation rotator. A Heliax coaxial transmission line, with known line loss, connects the output of the antenna under test with a spectrum analyzer. The transmitting antenna is a directional horn antenna with a known gain and is placed on top of a 30-ft. height building 783-ft form the tower pointing toward the testing antenna. We injected a continuous wave (CW) signal of power 0dBm into the transmitting antenna and adjust its position until we get a maximum reading on the spectrum analyzer (SA.) We record the values shown on the SA, then use these values together with the transmitting antenna gain and the distance between the transmitting and receiving antenna to compute the land loss, the power spectrum density at the receiving antenna, and finally the gain of the antenna.



We measure the gain pattern of the antenna at different angles and frequency.

- First, we transmit the signal at various frequency. Record the power (on the SA) received at the receiving antenna. Table 3 shows the frequency response of Discone antenna at a single angle of incidence.
- Second, we transmit the signal at a single frequency, but this time, use the azimuth rotator to rotate the test antenna 360 degree with an increment of 30 degree to find the antenna response in azimuth. (Table 4)
- Finally, we transmit the signal at a single frequency while moving the test antenna in elevation and record the value every 30 degrees from horizon to horizon. (Table 5)

The gain (dB) of the test antenna is calculated using the following formula

$$G = \frac{4\pi}{\lambda^2} A_e \qquad (1)$$

where,

G: the measured gain of the test antenna (Discone antenna)

A_e: effective aperture

$$A_e = \frac{P}{S} \qquad (2)$$

$$P = 10^{(P_{SA}\ -\ \text{line loss})/10}$$

$$S = 10^{(0\ \text{dB}\ +\ G_T\ -\ \text{Propagation Loss})/10}$$

$$\text{Propagation Loss} = 10\log\left(\frac{1}{4\pi d^2}\right) = -58.55\ \text{dB}$$

where,

P: power deliver to the load (W)

P_{SA}: power reading from the Spectrum Analyzer (dB)

S: Power Flux Density (W/m^2)

G_T: Gain of the transmitting antenna (dB)

d : distance between the transmitting and receiving antenna

d=783 ft. = 238.66 m

Table 3: Frequency Response at a single angle of incidence

Transmit Freq. (GHz)	Reading on SA (dB)	Transmit Antenna Gain	Line Loss	Gain (dB)
0.8	-74.83	10.6	3.3	-14.05
1.0	-77.50	11.55	3.38	-5.67
1.2	-82.00	12.70	3.6	-9.52
1.4	-73.33	13.80	3.9	2.69
1.6	-79.50	14.70	4.5	-5.29
1.8	-74.83	15.45	4.9	-0.28
2.0	-82.17	16.10	5.19	-3.54
2.2	-86.00	16.55	5.5	-10.20
2.4	-96.00	16.65	6.1	-18.95
2.6	-89.33	16.00	7.2	-9.83
2.8	-85.33	15.50	7.8	-4.01

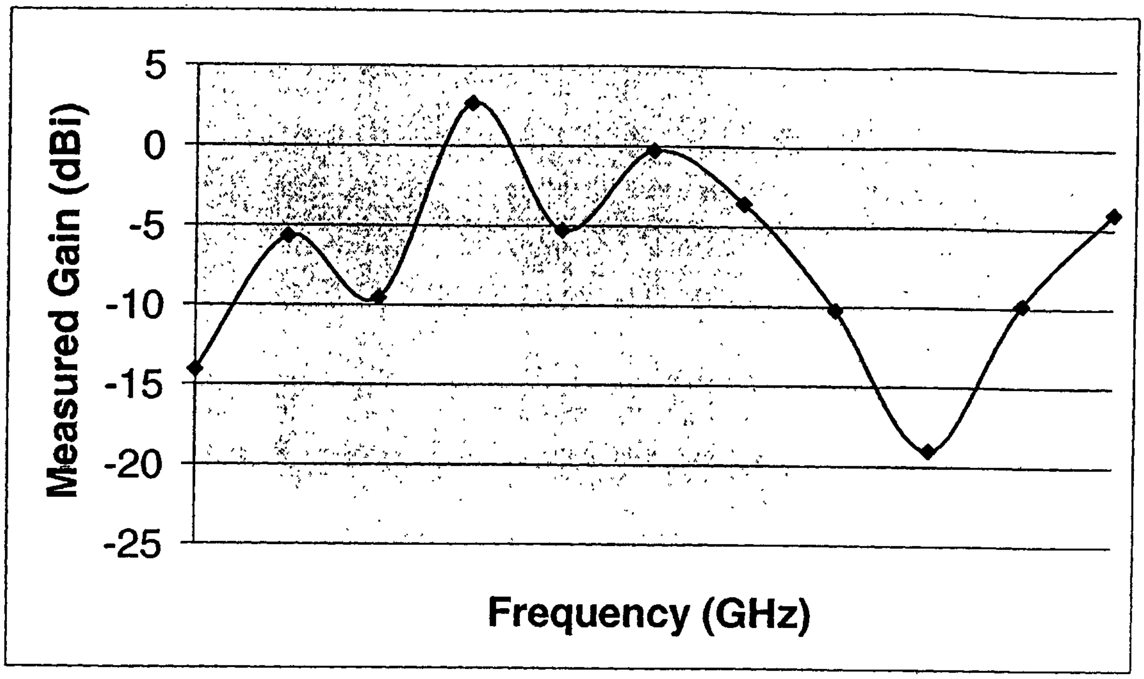


Figure 3: Plot of the gain at a single azimuth angle of incidence for the 800 MHz-2.2 GHz omni-directional Discone antenna

Table 4: Angular Response (Azimuth) at a Single Frequency: 1.5 GHz

Angle (degree)	Reading on SA (dB)	Measured Gain (dB)
0	-78	-4.40
30	-76.83	-3.23
60	-76.67	-3.07
90	-77.83	-4.23
120	-76.17	-2.57
150	-75.17	-1.57
180	-77.33	-3.73
210	-78.67	-5.07
240	-75.00	-1.40
270	-75.83	-2.23
300	-77.50	-3.90
330	-79.17	-5.57

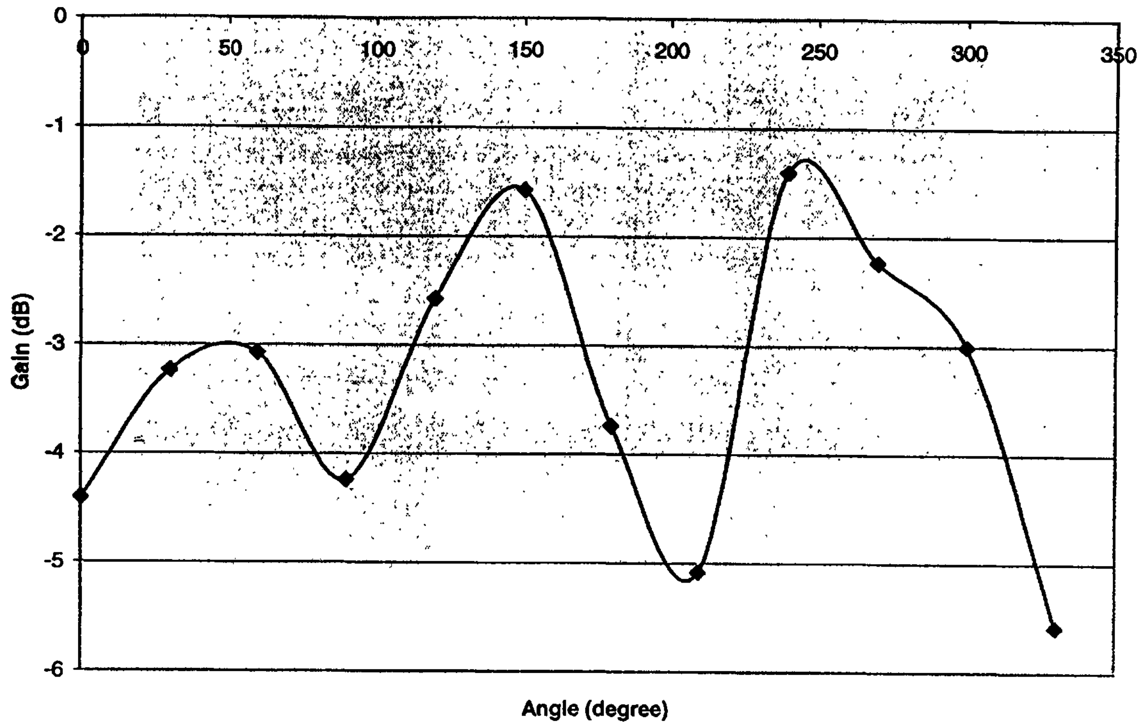


Figure 4: Plot of the Angular Response (Azimuth) at a Single Frequency

Table 5: Angular Response (Elevation)

Angle (degree)	Reading on SA (dB)	Measured Gain (dB)
0	-87.3	-13.73
30	-80	-6.4
60	-77	-3.4
90	-77.33	-3.73
120	-72.67	0.91
150	-71.83	1.77
180	-91	-17.40

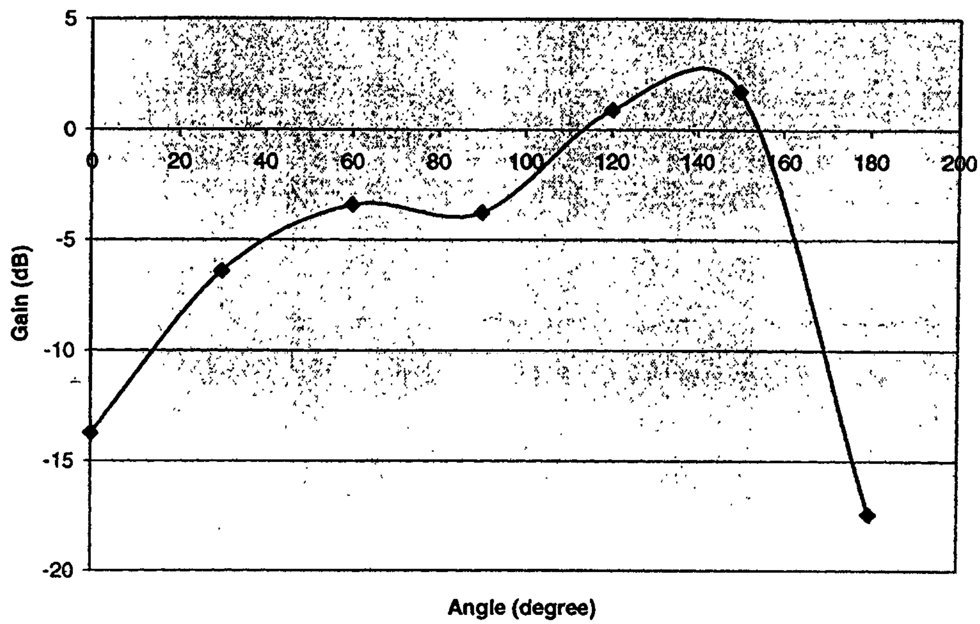
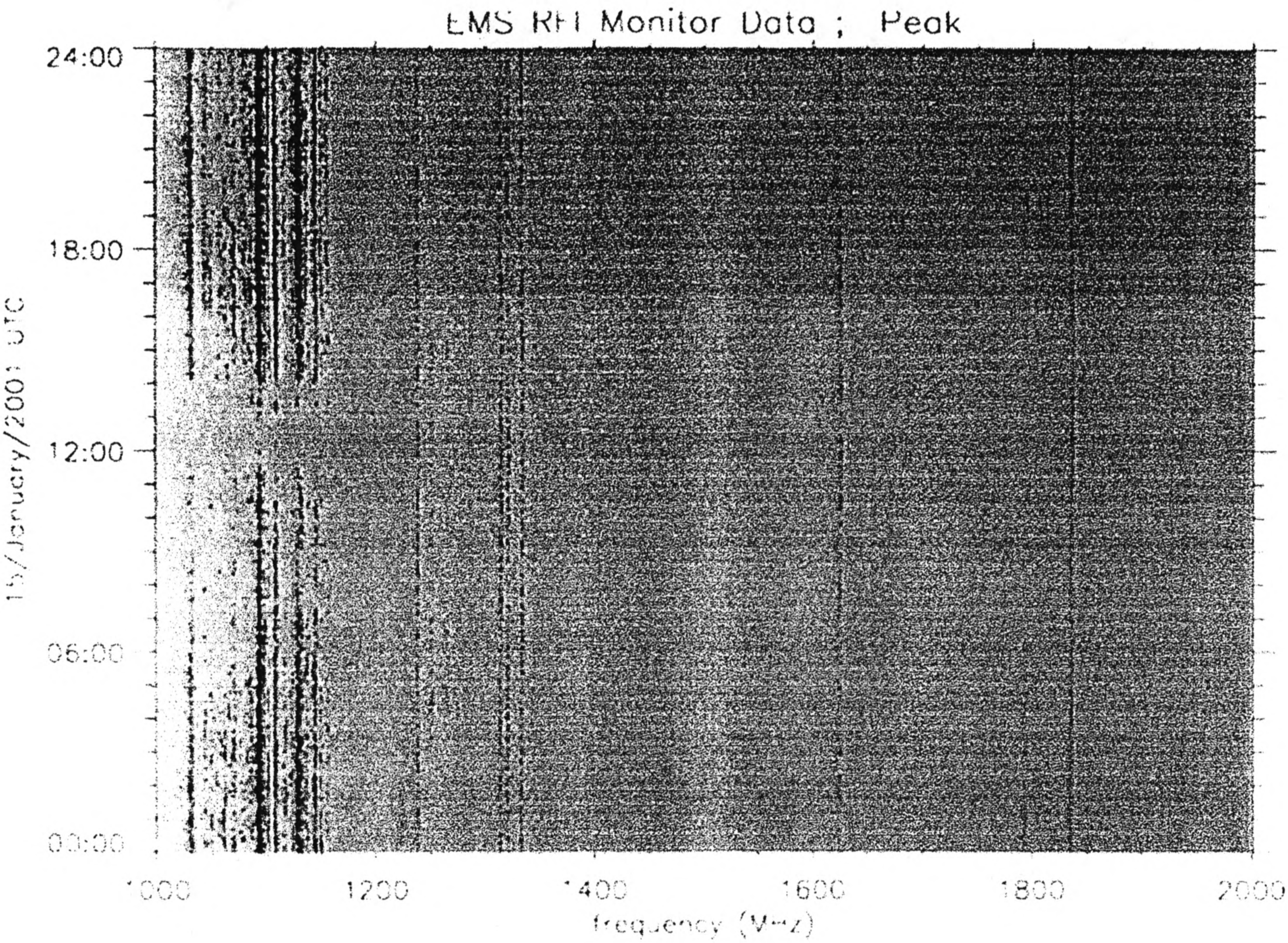
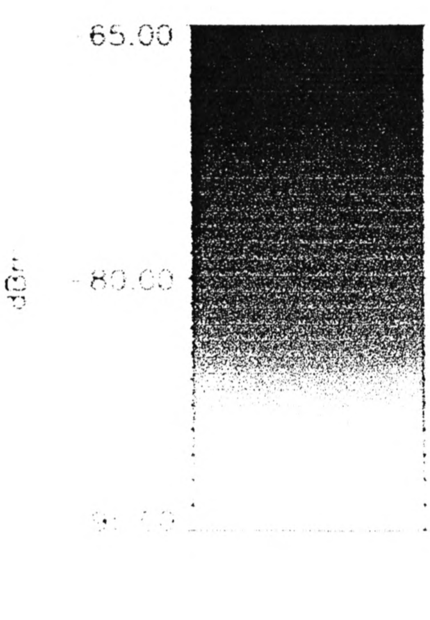


Figure 5: Plot of the Angular Response (Elevation) at a Single Frequency

4. Survey 1-2 GHz

There are two parts to the survey: a general survey using the NM67 receiver, and a detail survey using a spectrum analyzer. The general survey done using the current setup collect and plot the power level of RFI signal as gray-scale plots. Figure 6 shows a gray-scale plot of the power level of RFI signal from 1 GHz to 2 GHz. The detail survey allow us to zoom in to any signal at any frequency and find its bandwidth.





plot center frequency = 1500.5 MHz

Pk Hold period = 5 min

frequency span = 1001.0 MHz

resolution bandwidth = 100.0 kHz

scan time = 1.5 sec.

input attenuation = 0 dB

1.2 GHz to 1.4 GHz Survey

In our general survey of this using the EMS's systems, we discovered 4 strong signals in this band. Upon doing a detailed survey of this band using a spectrum analyzer, we discovered that there were six signals present. All six signals are consistent in frequency, behavior, and appearance with aircraft radio navigation (Radar). The specific sites attributed to each signal have yet to be tracked down, but the list below summarizes their characteristics.

1.4 GHz to 2.0 GHz Survey:

The initial survey shows that there are three strong signals from 1.4 GHz to 2.0 GHz. We know that the signal at 1624 is from the Iridium Satellites.

Table 5: Summary of 1-2GHz Survey

Frequency (MHz)	Bandwidth	Max. Power (dBm)	RBW
1030	6 MHz	-40	30 kHz
1025-1150	?	-55	30 kHz
1090	6MHz	-40	30 kHz
1150-1215	?	-55	30 kHz
1288.8	1.20 MHz	-50	1.0 MHz
1246.3	600 kHz	-74	100 kHz
1261.2	600 kHz	-88	3 kHz
1310.0	200 kHz	-71	30 kHz
1316.8	1.20 MHz	-65	30 kHz
1330.0	200 kHz	-70	30 kHz
1624	?	-83	1 kHz
1796	10 kHz	-82	100 kHz
1839	1 kHz	-82	3 kHz

Part II: Design LPS and Front-end

1. Lightning Protection System

a. Problems

Lightning, whether a direct strike, a cloud to cloud discharge or earth current transient from a strike nearby, can damage or destroy electronic equipments in a fraction of a second. In the past, engineers who worked on EMS reported that they had to change the amplifiers in the front end about once a week. Apparently, lightning is one of the reasons that cause the amplifier to blow out. Thus, it is important that we provide a good lightning protection system for EMS.

When EMS was first installed, basic cares were taken to protect the system against lightning.

- Guy Tower
- 3 lightning rods on top of the tower
- 16 ground rods connected together forming a "single point" ground system.
- A lightning protection circuit*
  - *Note: The lightning protection circuit has a corona detector that detects potential difference in the air during the storm. This circuit controls a relay inside the front-end box that open the signal path in the event of lightning. The problem with this circuit is the corona detector. It is very unreliable and often gives false alarm.*

One major problem that we found with this system is that it didn't have any types of voltage surge protector for the power supply. Also, many cables enter the building were improperly installed and had no protection. (A diagram of all the cables enter the RFI shelter is included in the Appendix.) Table 6 gives a summary of all the unprotected components in the system.

Table 6: Summary of Unprotected Components in the system

Description	Initial Status
Rotator Control Cable (EMS)	No protection, cable enters through a hole on the floor
Serial Cable to Front-end (EMS)	No protection, cable enters through a hole on the floor
Phone Line	Cable enters through a hole on the floor.
RF Cable and Control Cable (STS)	Cables enter through a hole at the bulkhead
AC Main Power Supply	No surge protection
AC Power to Front-end	No surge protection

b. Solutions

To improve the old lightning protection circuit, we looked for other lightning detectors that are more reliable than the corona detector. We found that an Electric Field Mills (EFM) (made by the Langmuir Lab at NMT) or a commercial TSS 924 Thunderstorm Sensor (manufactured by Global Atmospherics, Inc.) would be the best lightning detector. However, these devices are very costly (4000 - 8000 dollar); and with the budget allocated to us on this project, we can't afford them.

The absolute protection for any electronic system is to enclose it in an electrically conducting box called a Faraday Cage. With a Faraday shield, unwanted current will flow around the outside of the box, thus, will not generate any potential difference inside. Also, this Faraday shield will conduct the lightning-current safely to ground and minimize the danger of induction to RF cable. Based on this knowledge, we attempted to create a system as close to a Faraday cage as possible. The way to achieve this is to run all the cables in metal conduit and properly terminate the conduit to a ground.

To do this, we need to consider all the lines (cables) that come in to and leave the shelter. There are 3 lines that run from the shelter to the tower: the RF transmission line, the serial line from the PC to the front-end, and the control line for the rotator. The RF and control cables for the STS system run underground from the satellite disk to the shelter. The power line for the front-end comes directly underground at the base of the tower. The phone line runs underground from the main control building to a hole on the floor in the shelter. All the communication lines are run in fiber optics; thus are protected from lightning. (See Figure 16 in the Appendix)

- We run the RF transmission line and serial line in one metal conduit (conduit #1) and terminate the conduit at the shelter and at the front-end box on top of the tower.
- The power.line (comes from the ground) is run in a separate conduit (conduit #2) and is terminated at the front end.
- The control line for the rotator is run in a separate conduit (conduit #3). The conduit is terminated at the shelter and at a steel box attach below the triangle plate on top of the tower. There are twelve conductors inside the control lines: six are used to control the azimuth and six are for the elevation. Inside the box, we put MOV (Metal Oxide Varistors) at each end of the 12 conductors. If the voltage exceeds the threshold of the MOV, the current will be shunt to ground. See Figure 7 for a wiring diagram for the MOV box.

<u>Notes:</u>

The following color code applies to the wiring of the azimuth and elevation control lines

There are 6 twisted pairs of wire at the controller box: 3 pairs (Red, White, Green) for the azimuth, 3 pairs (Blue, Brown, Yellow) for the Elevation.

<u>Azimuth:</u>

Red => 1	White => 2	Green => 3
Black => 4	Black =>5	Black => 6

<u>Elevation</u>

Blue => 1	Brown => 2	Yellow => 3
Black => 4	Black =>5	Black => 6

At the Elevation and Azimuth Rotator, two 3 pairs (Red, White, Green) were used and the wiring is the same

Red => 1	White => 2	Green => 3
Black => 4	Black =>5	Black => 6

At the AC main power supply inside the shelter, we install a Hybrid surge protector (ZONEMASTER). We also install a surge voltage arrester (SYSTEMTRAB) in the front-end box on top of the tower to protect the power supply uses to run the front-end. The voltage surge protector will divert excessive surge energy to ground if a surge arrives via the main power supply. Both type of surge protector have two to three levels of protection: MOV for fast response, gas tube for slower response, and thermal fusing. See Figure 8 for a simplified diagram of lightning protection for EMS.

c. Recommendation

We tried to install a good lightning protection for EMS to the best of our ability and resources available. The pre-existing system can not support some additional protection that we wanted to install for EMS. In addition, some lightning protection components needed to be handle by professional electrician and engineer and are out of the scope of this project. We recommend that the following items be added to the system to provide complete protection against lightning.

- Run all STS cables underground in metal conduit (instead of PVC). If possible, replace with fiber optics.
- Install an isolation transformer at the main AC power supply.
- Place a catenary system around the 3 existing lightning rods
- Place a stainless steel mesh on the ground around the area of the tower to help disperse the current (and protect personnel working around the area during a storm)

Figure 7: Wiring Diagram for MOV Box

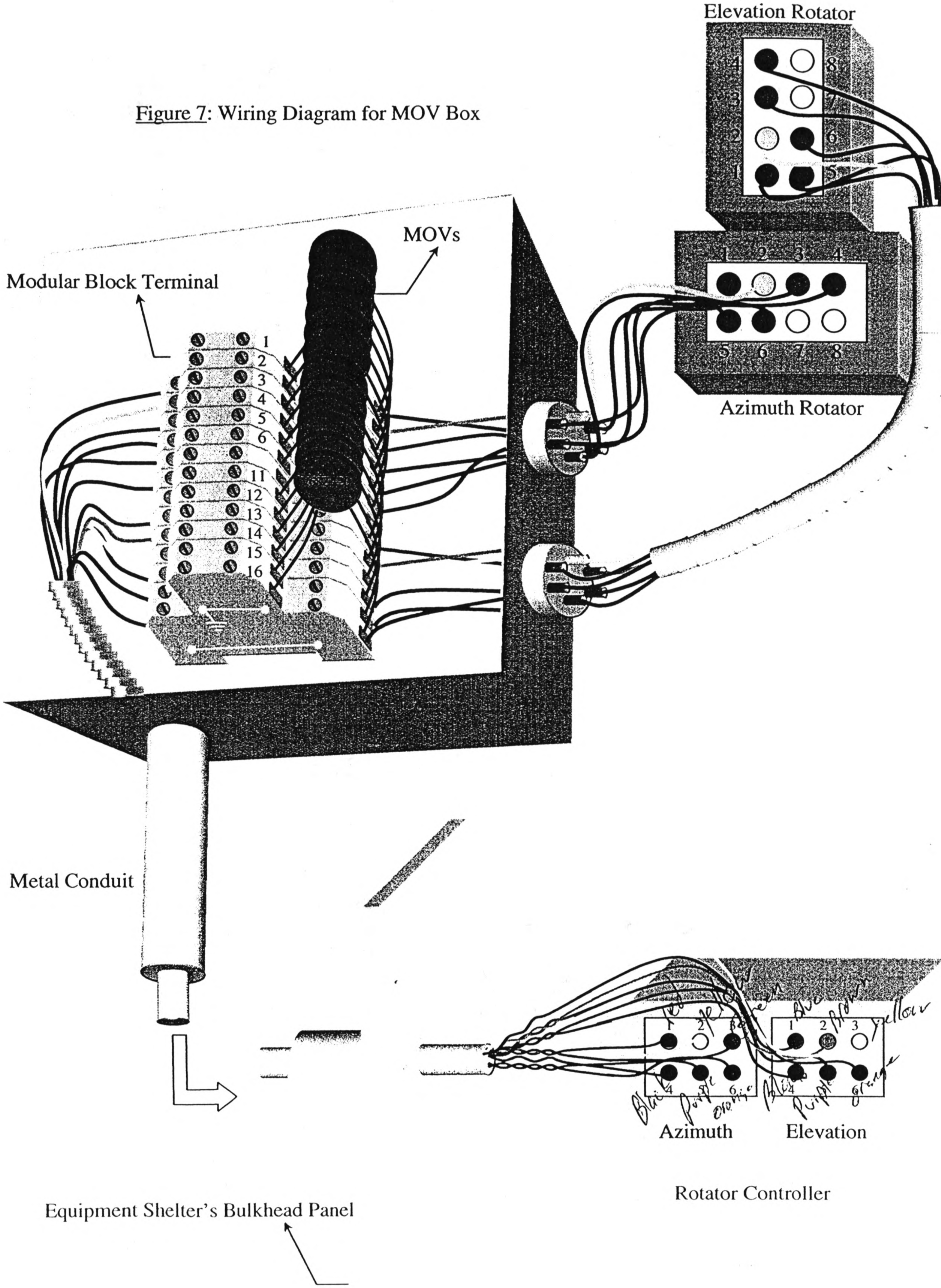
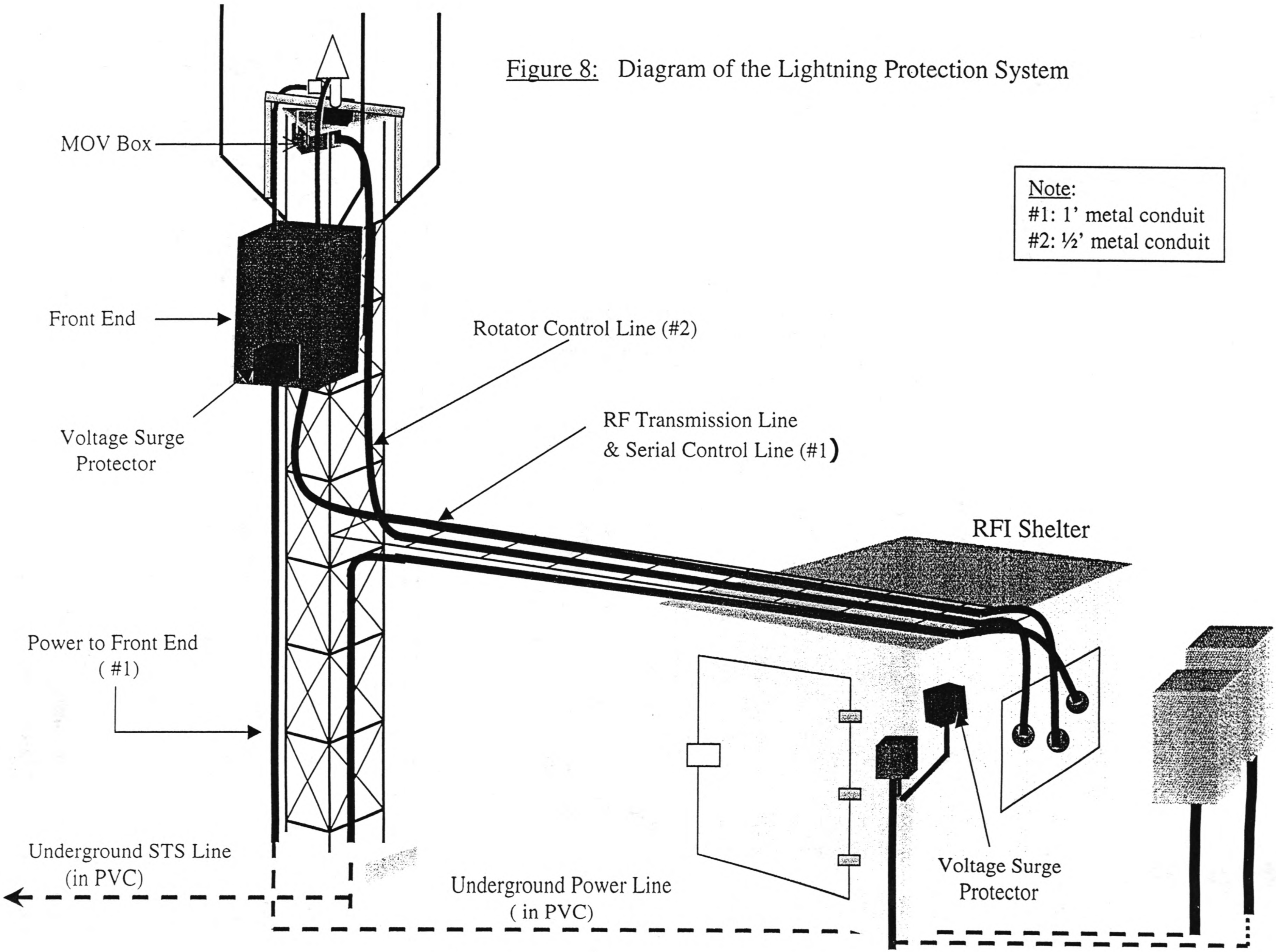


Figure 8: Diagram of the Lightning Protection System



2. Front-end Design

The EMS system had an existing Front-end (FE) with band switching capability controlled by the Motorola HC11 microprocessor. However, the band switching was not working properly. The following section discusses the design process of the new FE.

The hardware design of the front-end and its controlling software are implemented to meet the following system specifications: remote controlled band switching capability, two separate antenna inputs, self-calibrated in terms of system's gain and noise temperature, continuous amplification from 1-12 GHz, and a robust design.

a. Hardware Design

The front-end consists of low noise amplifiers (LNA) to amplify the signals, switching logic to select input source and amplifiers of desired frequency range, and a noise diode to act as a calibration source.

LNAs Set-up

One specification for this FE design requires continuous amplification from 1-12 GHz. We use multiple LNA to cover this entire frequency range because we do not have one low noise amplifier with this frequency range available. Besides, based on the survey done during the first half of the project, there are some strong signals detected below 1 GHz and in L band (1-2 GHz). Therefore, we need to have both a low gain and a high gain path for these frequency bands. Low gain signal paths are used to observe strong signals; and high gain paths are used to observe weak signals.

There are seven possible signal paths in the front-end as shown in figure 9. From 10 MHz-1 GHz and 1-2 GHz, there are two paths for each frequency band. For 2-4 GHz, 4-10 GHz, and above 10 GHz configurations, there is only one signal path for each band, since RF signals in these frequency bands are mostly weak.

b. Remote Switching Logic

There are four possible input sources to the system: one 1-12 GHz omni-directional conical spiral antenna, one 2-18 GHz directional horn, one noise diode, and one spare input for additional antenna in the future. Also, there are seven possible signal paths configured for different frequency ranges as discussed in the last section. We use RF coaxial switches to select one input and one signal path at a time and use a Universal Asynchronous Receiver/Transmitter (UART AY-5-1013A) to control them. Figure 10 shows the simplified wiring diagram of the switching logic.

The UART receive data from the Serial Port and output the data on parallel bus at a baud rate of 2400. We configured the UART to receive one start bit, eight data bits, and one stop bit. There are eight data bits that the UART puts out on the parallel bus.

- The MSB of the data byte (bit RD8) turns the noise diode on or off.
- Bits RD7 and RD6 control the RF relay, through which either the high gain or the low gain path of the 10 MHz-1 GHz frequency band is selected.
- The next two bits, RD5 and RD4, control the 2:4 de-multiplexer used for input (antennas or noise diode) selection.
- The last three bits (RD3-1) control the 3:8 de-multiplexer used for band selection.

When an entire character has been received, the Data Available (DAV) line go high (logic 1). After this line is asserted, the received data need to be placed onto the output lines by de-asserted (logic 0 or low) the Received Data Enable Line (/RDE). This is done by connecting the /RDE line with the inverted output of the DAV line

The two multiplexers need to be enabled after the received data has been placed onto the output lines (after /RDE goes low) to ensure proper switching in the RF coaxial switches. Since this UART has an output propagation delay of 500 ns, we need to delay the signal on the /RDE 500 ns before using this line to enable these two multiplexers. The 500 ns delay is generated by an 8-bit shift register.

See Appendix for a detail schematics of the band switching and antenna selection logic.

c. Noise Diode

Like the two antennas, the noise diode is treated as a source and can be selected or deselected using the switching logic. Once the noise diode is selected, we can turn it on or off to determine the system's gain and noise temperature.

The noise diode has a frequency range of about 50 MHz-18 GHz and, when turned on, will act as a signal generator across its entire frequency range. The power levels that the noise diode puts out for each Resolution Band Width (RBW) setting of the receiver are calculated in the Noise Diode code (will discuss later.) Thus, by connecting the output of the noise diode to a signal path to the receiver, and comparing the values from the receiver to the calculated value, we will know the system's gain. Similarly, we can measure the system's noise temperature with the noise diode off.

Figure 9: RF Band setting in the Front-end

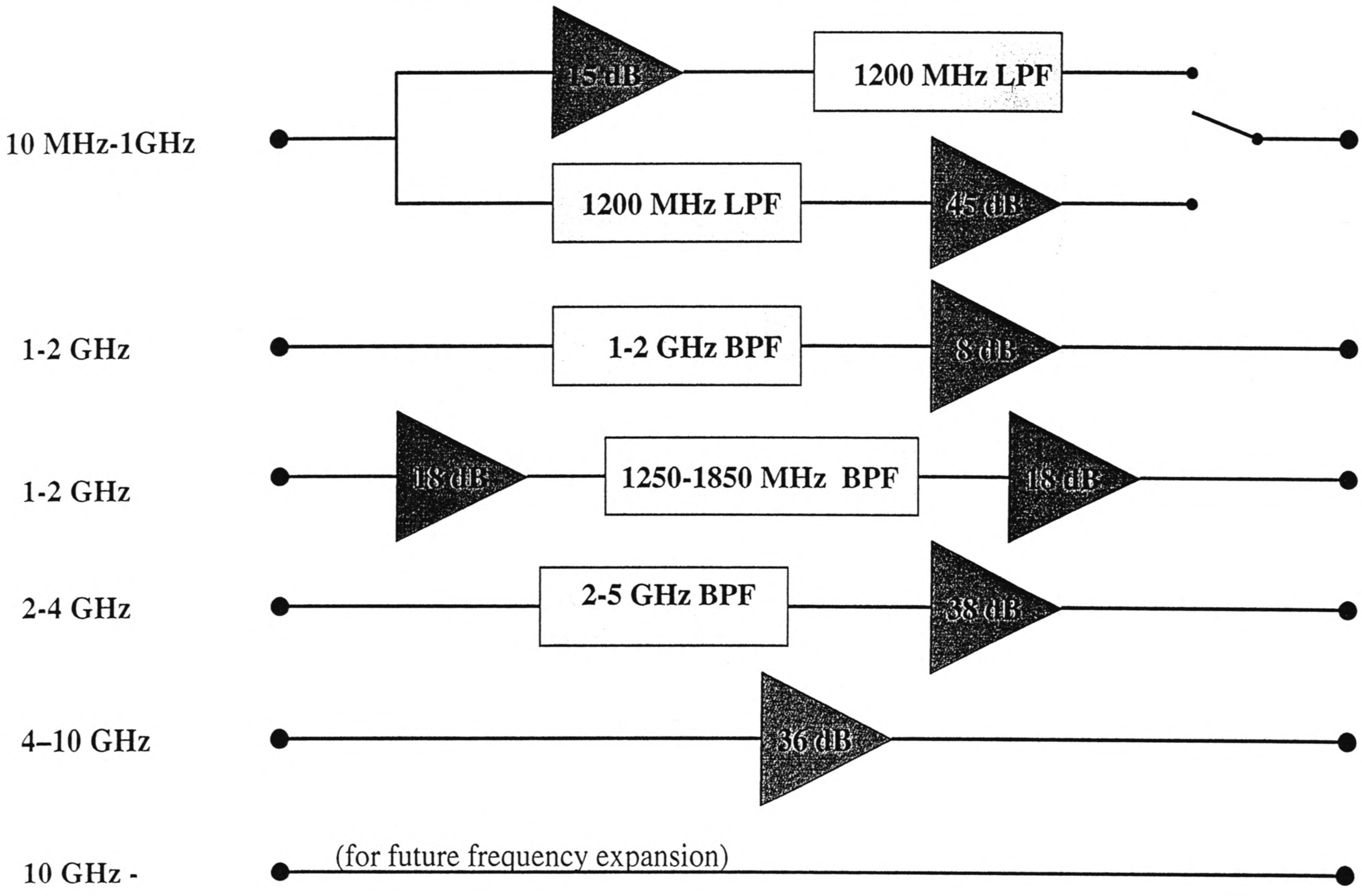
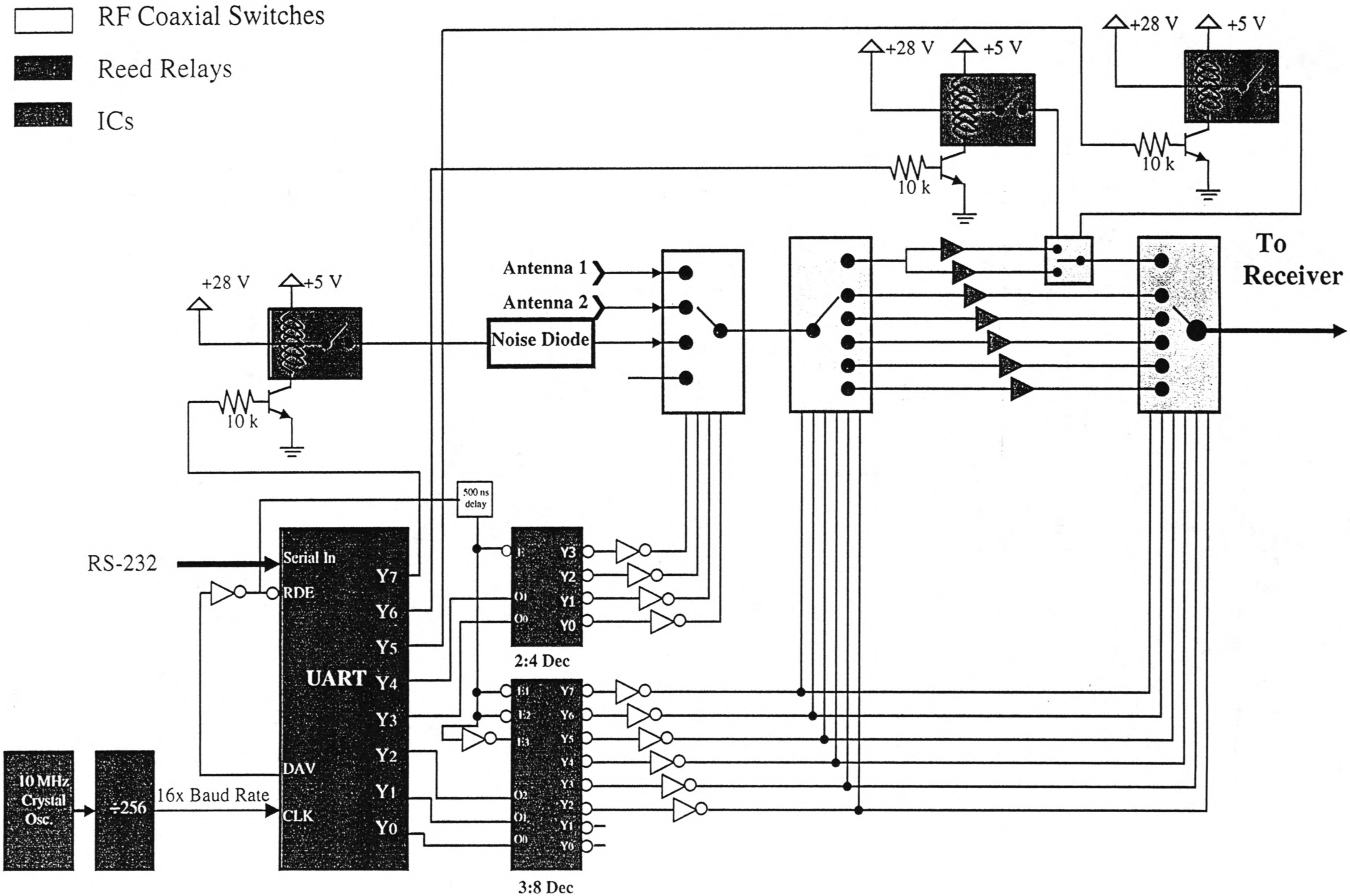


Figure 10: Front-end Remote Switching Logic



Part III. Software Implementation

The software side of the EMS system had two major flaws: it was not user-friendly, and it did not produce some of the necessary data. The software was not user-friendly in that it required constant manual updating in order to keep it running. These were often simple changes, but required editing a table that could quite often be confusing. The solution to this problem was to automate the code so that it updated itself and performed the majority of its tasks without prompting. The second major defect was that the system did not produce some very desirable data. The system can only measure power levels at the input to the receiver. While this raw data does give one an idea of relative strengths of signals, it is not in a form useful to the NRAO community. This useful data would be in the form of power flux density, which can be derived from the measured-power data if enough of the EMS system parameters are known. This problem was solved by creating code to model the pertinent EMS parameters in a way so that power flux density could be modeled from the measured-power data. The solutions to the EMS software problems were both implemented with severe time restrictions that forced functionality and speed of development to be the prime considerations.

1. Automation Software

Making the EMS system more user-friendly and automated was carried out using shell scripts. Shell scripts were used for a variety of reasons. The bulk of the data storage and processing is done on a SUN™ Sparc20. This computer uses a Unix operating system, and interfaces with the remote data-logging computer (which runs Linux) over NRAO's existing network. The necessary software tasks involved determining the day's date (at the end of each day, Greenwich Mean Time), locating all data files that were logged on that day, transferring them from the remote data-logging computer to the Sparc20, where the files were renamed, plotting programs were called, and the plotting programs output then stored in the proper directory with the proper filename. Most of these file-handling tasks are easily carried out in shell scripts. Also, shell scripts are easily modified (some of the minor details of the above process needed to be changeable as the system was developed more), as well as allowing for quicker development than C or IDL (the other programming languages on this platform that we were familiar with).

Our solution was to first reorganize the existing software, and then add some automation. The old software used the cron daemon to automatically transfer the files, then rename them, and call the plotting software. Each set of instructions had to be hand-entered for each day they were to be carried out. We created a shell script that when called, determined that day's date, and then called other shell scripts. These other shell scripts then located all files pertaining to that day's date on the data-logging computer, transfer those files to the Sparc20, rename them, and then call the plotting routines. This simplified the cron table significantly; it reduced the cron table to a single line of code that never had to be modified. These changes produced a system that would automatically retrieve data, and produce plots of the measure power from the receiver each day without intervention or prompting. This made the EMS system extremely user-friendly. The user never had to do anything other than look at the output (the plots) at the end of each day to get the desired information. These process produced plots of the power that the EMS receiver measured over the course of one day. Later, this code was modified to include calling code that converts the power data into power flux density data and plot this as well.

There were alternative ways to implement the software changes. We could have written one large piece of code to carry out all functions. Or we could have used other programming languages. However, at the time this code was being written, the system requirements were in a state of flux, and shell scripts seemed to offer the necessary quick development time as well as easy adaptability to any new needs. The organization was chosen in order to make each task modular, and easy to change without necessitating changes in other pieces of code.

2. Power Flux Density (PFD) Software

The EMS system's second major flaw was that it did not produce data in the form of power flux density. We solved this problem by creating a model that accomplished two things: it compensated for any gain or losses the EMS system introduced between the antenna and the receiver, and it modeled the antenna's effective aperture so that we could take the power coming out of the antenna and directly convert it into the power flux density that was incident on the antenna. Both parts of the model are critical. The first part of the model that compensates for gains or losses is necessary, in order to derive what the power coming out of the antenna is. The second part of the model can then derive the power flux density from the power derived in the first part of the model. This model was implemented using C on the EMS system's Sparc20 computer. C was used because this model required a capability to deal with various data structures as well as quite a bit of numerical computation. C seems superior to shell scripts in both respects. IDL could have been used, but since we have no experience with data structures in IDL, we thought C would be quicker and easier to implement on a Unix platform.

We created several 'device' files to implement this model. Each file represents a device, and describes its gain (or loss) across a specific frequency band. At present, this frequency band is 1 – 2 GHz. There are 1,024 equally-spaced data points across this frequency band. Each data point represents the gain or loss that device introduces (in dB) at the specific frequency that data point represents (the magnitude portion of a Bode plot). These data points form the file that represents each device in the EMS signal path. We can then derive a relationship between the power measured at the receiver and the power coming out of the antenna. By adding (or subtracting) the appropriate files, we convert the measured power at the receiver to the power coming out of the antenna (or at least a reasonable approximation). There is one more file to use at this point. This file is identical to the device files mentioned before, except that its data points are not gain or loss at specific frequencies. This file's data points represent the antenna's effective aperture at each specific frequency. This fact allows us to convert the power coming out of the antenna into the power flux density incident upon the antenna. This output is then stored in a data file that has the same form as the original (measured power) data file. Some minor modifications to the existing plotting programs then allow us to plot the data in a familiar form without much extra work.

Usually, the system is modeled as an antenna, with a long transmission line and one amplifier connected to the input of the receiver. Each one of these (antenna, transmission line, amp) has a device file that describes its behavior. The PFD (Power Flux Density) code takes the power measurements the receiver makes, uses the device files to work backward and compute what the power coming out of the antenna is, and then uses the antenna's device file (effective aperture) to convert that power into power flux density.

All units are in a logarithmic scale in order to simplify calculations (makes the operations addition and subtraction instead of multiplication and division). The measured data is transferred across the network from the data-logging computer to the Sparc20. The Sparc20 then applies the data to the model and produces the output, which is automatically plotted (one of the later modifications to the shell scripts).

There were several different ways to carry out this model. The specific set-up just described was used because the device files would take the same format as the data files and hence be easily combined. All of the calculations were carried out in terms of logarithmic units (10 Log XX, where XX would be in linear units, like a gain of 1000) to simplify computation, since multiplication and division in linear units becomes addition and subtraction in logarithmic units. This specific format seemed like the easiest to implement.

3. System Daily Routine

In order to understand how all the EMS software interacts, it's important to understand how the system operates over the course of a day. First, the system clock is based on the UCT (Universal Coordinated Time ,) which means that the system's day begins at 5:00 pm MST, or 6:00 pm MDT. When this new day begins, a directory is created in which the day's data is stored. Then the data is acquired, all day long. The data is stored in 5 minute intervals. At the end of the day, permissions on the data are changed so the data is readable to all network users. This data is then copied to a different machine across the network. This machine is the archive for the data. The data is then renamed and plotted. (See Figure 11 for a summary of how the system works)

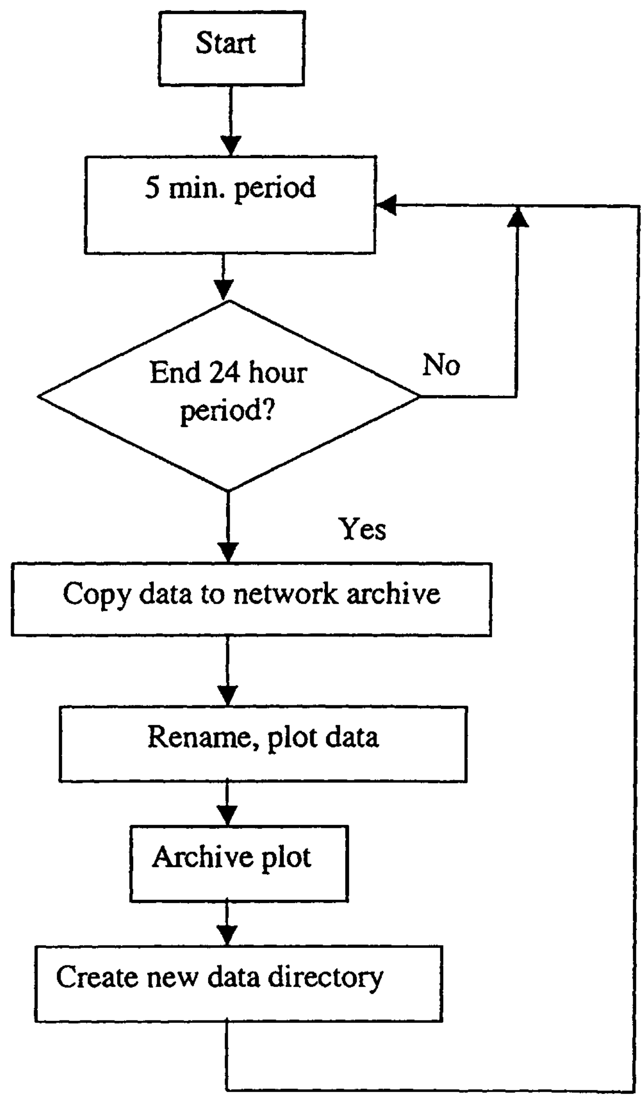


Figure 11. System Daily Routine

To understand this better, we need to know a few things about the hardware. There are two computers that are involved in this whole process. *Snow* is a PC located at the VLA site. It is running RedHat Linux as its operating system. Snow is the control and data logging computer for the system proper. The Java Servers, used to configure the system, are housed on this machine. Snow controls the receivers, front-end and rotator. Snow also logs the data and performs the conversion from the measured values (power) to the desired data (power flux density incident on the antenna). *Electra* is the other machine. One of its drives (electra2) is used to archive data. Electra renames the data and produces the plots that are the output of the system. Together, these two machines run all of the software used by the EMS. Copies of all source code is archived at **/home/electra2/ailmon/ver3** .

The 4 languages used in most of the software are Java, C, IDL, and the Bourne, again Shell Script (bash). The Java code is used to provide some control functions and a GUI for the system. This GUI is available (over the internet) at **snow.vla.nrao.edu/ailmon.html,** (available on NRAO's internal network only.) This GUI provides a means to configure the system as needed. The GUI source code on Snow is at **/usr/src/ailmon/ver4** and is named **client_box.java.** The executable version of this code is **client_box.class**, and must be in the */etc/htdocs* directory in order to be run automatically by this system.

All of the Java source code has a suffix of .java, whereas the compiled (executable) Java code will always have a suffix of .class. A copy of all Java code can be found in Appendix B. This GUI allows 3 major functions, to configure the background job, to view the background job, or to configure and run a single sweep. The background job is the 24 hour monitoring function that the EMS performs. A single sweep is just that, a single 1.5 second measurement that can be over any receiver band, and is squeezed in between successive scans of the background job. The single sweep does not have to have the same configuration as the background job, and the single sweep has all of the same options that the background job has. The main difference is that a single sweep is performed only on demand, while the background job is performed routinely without prompting. A user can change the background job's configuration (BGConfig), view the data from the background job in real time (BGView), or set up and perform a single sweep (Single). Each one of these functions is carried out by its own Java server.

All executable Java server code must be stored in **/usr/src/ailmon/class** on Snow in order to be run. There are three Java server

- **ailBGConfServer.class** is responsible for setting up the background job. This code builds up a configuration file for both the background job and any requested single sweep. The configuration file is named **ail.dfl** and is a single line of text and is explained in **/usr/src/ailmon/ver4/*getAilmKey.txt*** on Snow.
- **ailBGViewServer.class** allows the user to view the data the receivers take in real-time, on a display much like a spectrum analyzer's. This display is on the Java GUI used to configure the system.
- **ailMainServer.class** is the main control code. This is the code that is constantly running and calling all of the routines that keep the system running. The other 2 servers, **ailBGConfServer.class** and **ailBGViewServer.class** are both run only when **client_box.class** calls them. **ailMainServer.class** is always running in the background. It serves as the master control to the entire system, but does very little itself. It constantly checks the **ail.dfl** file to see if it has been updated. If the background job has been changed, the MainServer won't institute the change until the beginning of the next 5 minute interval. If a single sweep is requested, the MainServer executes the requested sweep and returns to performing the background job from before.

When a user accesses the Java GUI over the internet, **client_box.class** is called and handles all user interaction. When the user has set up a new background job, calls for a single sweep, or wants to view the background job data in real time, **client_box.class** calls the appropriate server to carry out the desired task. At the end of each 5 minute interval, MainServer forces the data logging code to write it's data to file. In this manner, an entire day of data is accumulated.

There are 3 pieces of code that actually handle setting up the system and taking data, these are: **getAilm, bandswitch, diode**. All C executable files will have no extension. All C source files will have a .c extension (**getAilm.c, bandswitch.c** ,

diode.c). All C source code can be found in **/usr/src/ailmon/ver4** on Snow. All executables are located in */bin* on Snow.

- **GetAilm.c** is the main program. It handles communication to the receivers, reads the data from them, and writes the data to file. This code also reads 3 system calibration files (**gain.cal, noise.cal,** and **antenna.cal**) and uses them to convert the measured power data into power flux density data.
- **Bandswitch.c** is the program that controls the front-end. It sets up the front end according to the desired system configuration through a serial cable.
- **Diode.c** is the program used to create the **gain.cal** and **noise.cal** files that allow the data to be converted to power flux density. All 3 calibration files **gain.cal, noise.cal** , and **antenna.cal** are in the same binary format as the 5 minute data files, and can be viewed using the same software. These 3 files can be found in */etc* on Snow.

A standard 5 minute interval for the C code works as follows (See Figure 12 for a summary of the 5-minute operation routine.)

- The Java MainServer reads the ail.dfl configuration file and calls diode, feeding it the configuration parameters.
- **diode** (which has a copy of **bandswitch** embedded in it) then sets up the front end (through the serial cable) and the receivers (through the GPIB interface to the CCI-7). Instead of using an antenna as a source, the noise diode is used as the source. One single sweep is performed with the noise diode (the selected source) turned on; one single sweep is performed with the noise diode turned off. **diode** can then use the data from these two sweeps to build the **gain.cal** and **noise. cal** files. These files are save to disk.
- **bandswitch** is called, and resets the front end to the proper settings for observation.
- **getAilm** is repeatedly called by MainServer for the rest of the 5 minute interval. It reads the 3 calibration files at the beginning of the 5 minute interval, and uses those files to convert the data it reads during the 5 minute period. **getAilm** also

performs a peak hold over this time period, so it's data represents the peak value at each given frequency point for 5 minutes.

- At the end of the 5 minute interval, **getAilm** then writes the peak values to disk, and this whole process (1-8) begins anew.

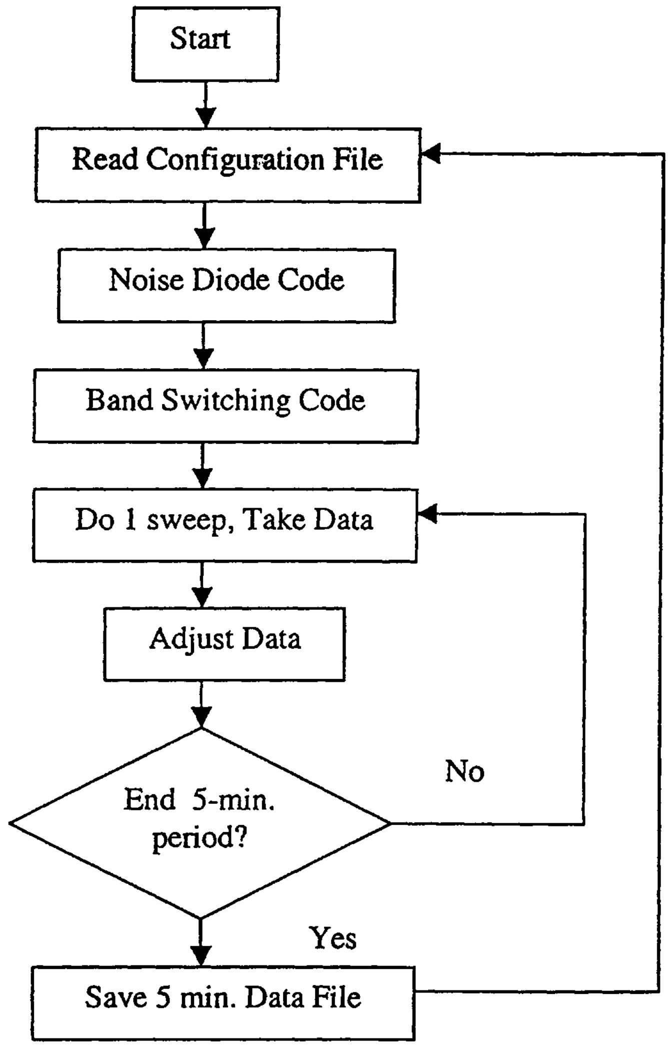


Figure 12: 5-minute Operation Routine

A result of this cycles is that the EMS updates its calibration files every 5 minutes. So if an amplifier is replaced, or the system's behavior varies according to temperature, the system is configured to adapt to it. By doing it every 5 minutes, we assure that at most 2 of the 5 minute data files contain inaccurate data.

The **noise. cal** files are used to remove the noise power added by the system from the data. This way, the data reflects the power levels caused by the environment. The **gain.cal** file is necessary to remove the system's gain from the data, to arrive at the power levels coming out of the antenna. Finally, the **antenna.cal** file represents the effective aperture of the (currently selected) antenna. This allows us to convert the power levels coming out of the antenna into the power flux density levels incident on the antenna.

This entire process of converting data and writing a data file every 5 minutes leads to a collection of 288 data files over the course of a day. At the end of each day (a UCT day,) this data (and the directory this data is in) has its permissions changed so that anybody can see and read these file. All 288 files are then copied to the archive on the Sparc 5 named electra (to its hard drive electra2). This data is then automatically renamed, and plotted. All of this is entirely automated.

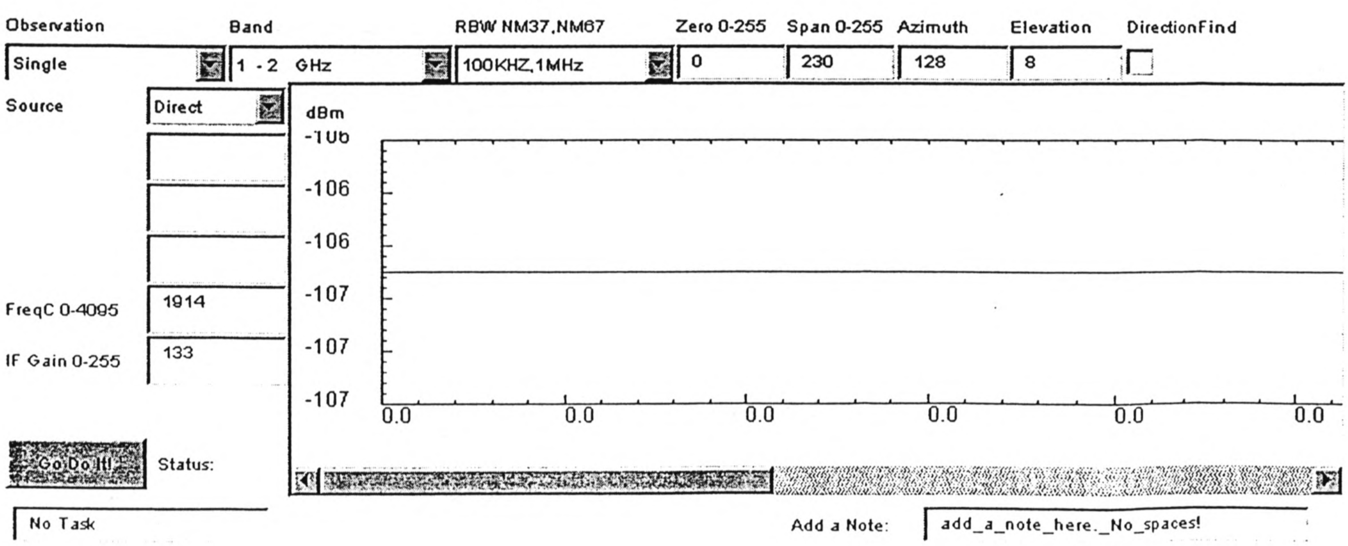
A cron job on Snow starts a bash shell script once a day, **kpermissions**. This script makes the data available to other computers. A cron job on electra starts a script once a day, **ktransfer_plot**. This shell script handles the copying of the data from Snow, archiving it on electra2. The shell script then renames all of the files to the format the plotting program expects. Finally, the script calls the plotting program **pgsail.pro** (written in IDL) to plot the day's data. A copy of the C code can be found in Appendix C, a copy of ktransfer_plot in Appendix D, and a copy of the IDL code is in Appendix E.

Appendix

A1. User Instructions

The following documentation provides instructions on how to control the Environmental Monitoring Station (EMS).

Controlling the EMS system has been simplified to the use of a Java Applet over NRAO's internal network. This applet can be accessed using a browser at a protected website available only to NRAO's network. The current system is configured such that the EMS is always running a background scan. This means that if left alone, the system will scan from the start frequency to the stop frequency every 1.5 seconds, continuously. This is called the background observation, and it is always running. In addition to the background observation, a single scan at any frequency band with any settings can be squeezed in-between the successive background scans. A third available option is to view the background observation in near real-time. This will display the result from each individual scan done by the background observation on the display of the Java applet at http://snow.vla.nrao.edu/ailmon.html. This display looks just like the output of a spectrum analyzer. See Figure 13. for a picture of the Java applet.



Calibrated Frequency Sweep Table

Frequency(GHz)	Band	RBW(kHz/MHz)	Zero	Span	FreqC	IFGain
1.000 - 2.001	1-2 GHz	100/1.0	0	231	1914	35
2.000 - 3.022	2-3.6 GHz	100/1.0	0	152	2808	36
3.014 - 3.601	2-3.6 GHz	100/1.0	151	89	2808	33
3.600 - 4.623	3.6-7.6 GHz	100/1.0	0	62	2964	45
4.592 - 5.600	3.6-7.6 GHz	100/1.0	63	61	2964	86
5.597 - 6.602	3.6-7.6 GHz	100/1.0	121	61	2964	35
6.593 - 7.601	3.6-7.6 GHz	100/1.0	180	62	2964	34
7.600 - 8.615	7.6-12 GHz	100/1.0	0	56	2872	32
8.608 - 9.620	7.6-12 GHz	100/1.0	56	56	2872	34
9.575 - 10.587	7.6-12 GHz	100/1.0	111	56	2872	35
10.580 - 11.601	7.6-12 GHz	100/1.0	163	57	2872	36
11.591 - 12.004	7.6-12 GHz	100/1.0	220	23	2872	37
12.000 - 12.994	12-18 GHz	100/1.0	0	41	3148	16
12.983 - 13.977	12-18 GHz	100/1.0	41	41	3148	22
13.962 - 14.956	12-18 GHz	100/1.0	80	41	3148	13
14.945 - 15.964	12-18 GHz	100/1.0	121	42	3148	22
15.956 - 16.974	12-18 GHz	100/1.0	160	42	3148	28
16.970 - 17.983	12-18 GHz	100/1.0	203	42	3148	25
17.516 - 18.000	12-18 GHz	100/1.0	224	20	3148	16

Figure 13: JAVA applet

Here's a step-by-step procedure to configure the Java applet and run a job:

Step 1. Decide what kind of job you want to run. Under "**Observations**" choose:

- Single, single scan.
- BG View, view the result of the background job.
- BG Config, reconfigure the background job.

Any changes you make will only apply to either Single or BG Config

Step 2. Choose the **Source**. There are three options here: Direct, Omni-D, or Noise.

- The Direct option allow the use of the directional horn at the top of the tower.
- The Omni-D option switch to the omni-directional antenna.
- Noise chooses the calibrated noise source (noise diode) as the input to the system. Currently, the Noise option is not available in software (turned off), but is expected to be made available soon.

Step 3. Choose the **Band**. The frequency range you want to observe must lie within one of the listed bands. The chart below the applet lists all of the available bands as well as other useful information.

Step 4. Select the **Resolution BandWidth (RBW)**.

- If you're observing below 1 GHz, your options are 10 kHz, 100 kHz, 1 MHz.
- If you're observing above 1 GHz, your options are 100 kHz, 1 MHz, and 10 MHz.

Step 5. Set the **Zero**. The Zero represents where (within the Band) you'd like to start the scan. The Zero needs to be an integer between 0 and 255. However, a correction factor also needs to be applied in order for this to work properly. Use this formula to calculate the proper Zero:

$$Zero = \frac{(Start - Low)}{BW} * 255 + CF$$

where,

Start is the frequency you'd like to start the scan at (say 1.25 GHz for example).
Low is the lowest frequency of whatever Band you're in (1 – 2 GHz Band, *Low* = 1 GHz).
BW is the bandwidth of the Band you're in (1 – 2 GHz Band, *BW* = 1 GHz).
CF is a correction factor that needs to be applied. This can be read off of the chart below the applet. Look on the row of the Band you're interested in, under the column Zero. This number is the correction factor (in our case the 1 – 2 GHz Band has a *CF*=0. So, our.Zero = (250 MHz)/1 GHz * 255 + 0 or
Zero = 64 (63.75 rounded to the nearest integer). Left-click in the Zero window, and enter the number you arrive at using the formula.
Note: Eventually, this formula will be imbedded in the applet so that only the desired Start frequency will be the input.

Step 6. Set the **Span**. Span is the width of the frequency range you wish to observe. In conjunction with Zero, Span will determine the exact frequency coverage of the observation. Use the following formula to calculate Span,

$$Span = \frac{desiredBW}{BW} * CF$$

where,

desiredBW is your desired bandwidth

BW is the bandwidth of the Band

Example: we want to observe 1.25 – 1.75 GHz.

Total width (*desiredBW*) is 500 MHz. This frequency range lies within the 1 – 2 GHz Band. So, 500 MHz / 1000 MHz = .5 . We read the correction factor from the chart below the applet. For the 1 – 2 GHz Band, the correction factor is 231. We multiply 231 * .5, to get 115.5, but you **must** round to the nearest integer, so we'll use a span of 116. Left-click in the Span window, and enter 116.

Note: Like the Zero window, eventually this formula will get imbedded in the applet software, and then you'll only need to enter the desired BW.

Step 7. Set the **Azimuth** and **Elevation.** (Have to select **Direction Find**) This is done only if you have selected the directional antenna as the source, otherwise these inputs are ignored. This allows you to position the directional antenna at most angles. The azimuth angle should be between 0° and 359°, with 0° indicating true North, East would be 90°. The elevation angle should be between 0° and 90°, with 0° indicating that the antenna's pattern is directed out towards the horizon, and 90° indicating that the antenna is scanning straight up.

Step 8. Enter the **IF Gain**. This is a correction factor used by the Receiver to keep the measurements properly calibrated. This factor changes according to the Band and specific frequencies observed. This factor is read from the chart below the Java Applet. In our example, we are observing in the 1-2 GHz range on the table, we enter an IF Gain of 35.

Step 9. Click on the **Go Do It!** button to send the parameters to the computer for execution.

Step 10 . Add a note in the **"add_a_note_here._No_spaces!"** window if desired. Spaces will cause several errors when the computer tries to create the data file, and may cause the data to be unusable.

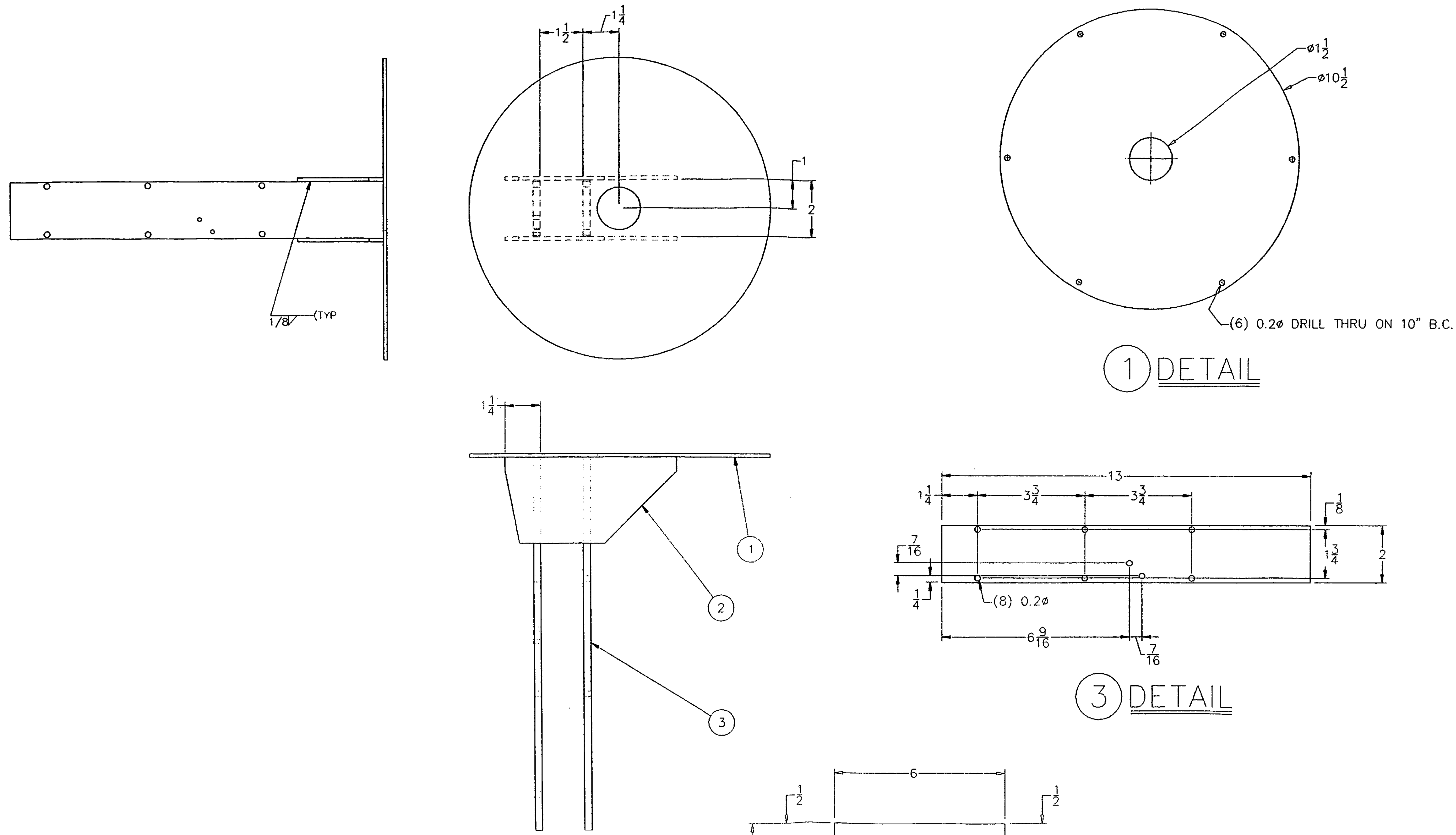
Note: The **FreqC** window is an artifacts from a previous software version, and is ignored in the code. Finally, the entire display of the won't fit on the applet as is, hence the slide bar underneath the data window. This will allow you to scroll across and view the entire data range.

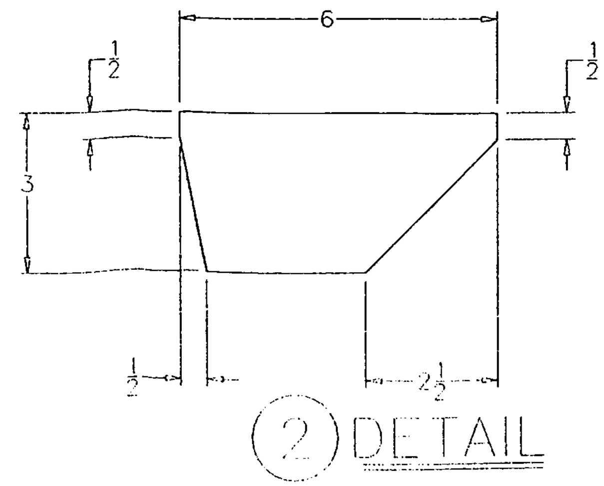
All other data plots are automatically produced and are available for viewing on NRAO's internal network as postscript files. They may be viewed at **/home/electra2/ailmon/data/plots.**

A2. Antenna Mounting Plate

In order to collect data up to 12 GHz, we need to find an antenna that cover the desired frequency range. (As noted earlier, the Discone antenna only covers from 800 MHz to 2.2 GHz frequency range) We decided to use two antennas: the conical-spiral antenna (1-12 GHz), and the horn (2-18 GHz) for direction finding. The task is to design a mounting plate that would support the two antennas. Figure 14 shows the antenna mount design

REV	BY	DESCRIPTION





No.	REQ	DESCRIPTION
3	2	3/16 x 2 x 13
2	2	1/8 x 3 x 6
1	1	1/8 x 10 1/2ø

DIMENSIONS ARE IN INCHES
UNLESS OTHERWISE SPECIFIED
TOLERANCES
3 PLACE DECIMALS (X.XXX) +/- 0.002
2 PLACE DECIMALS (X.XX) +/- 0.020
FRACTIONAL (X Y/Z) +/- 1/32
ANGLES +/- 30'

MATERIAL

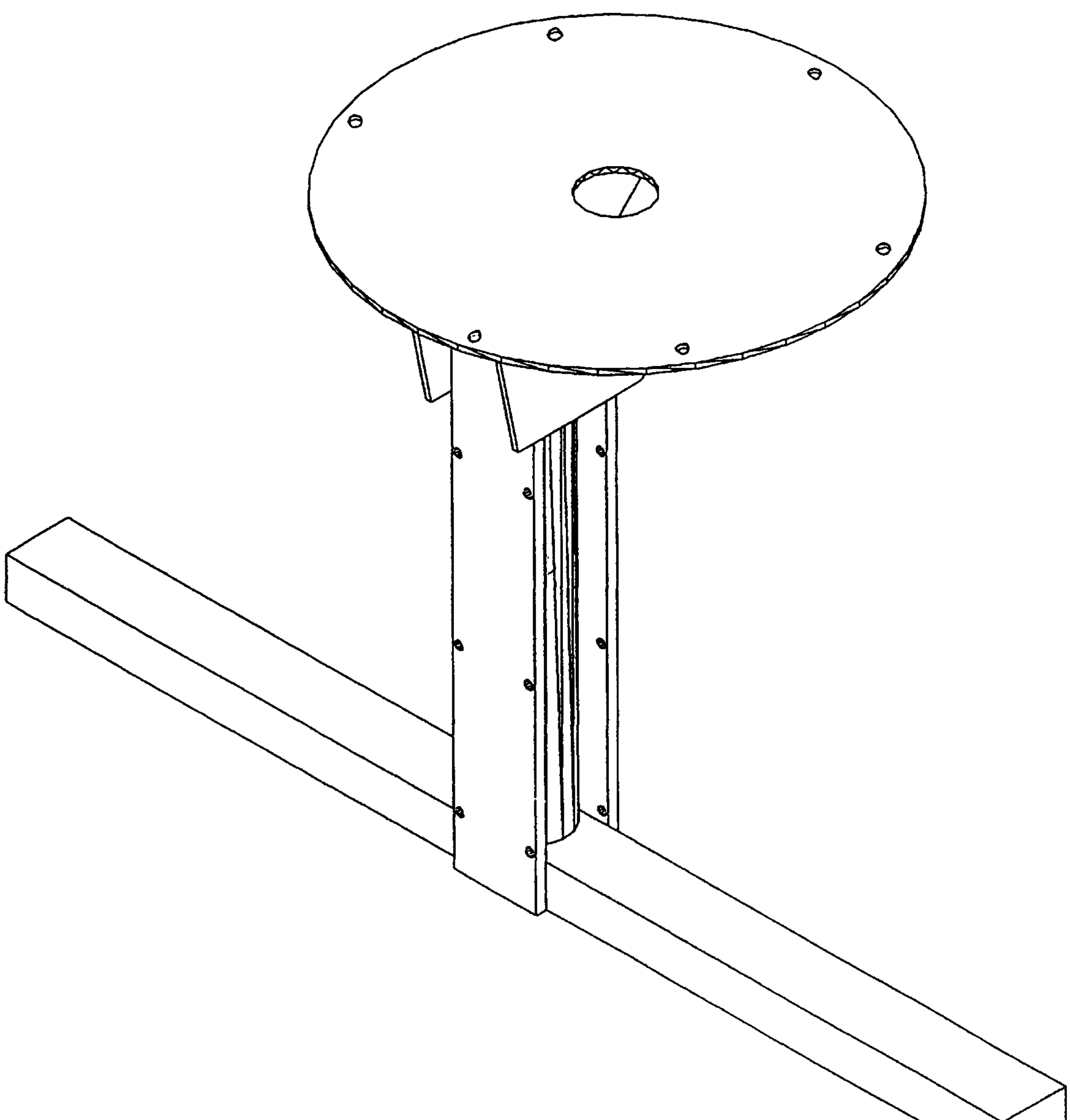
FINISH

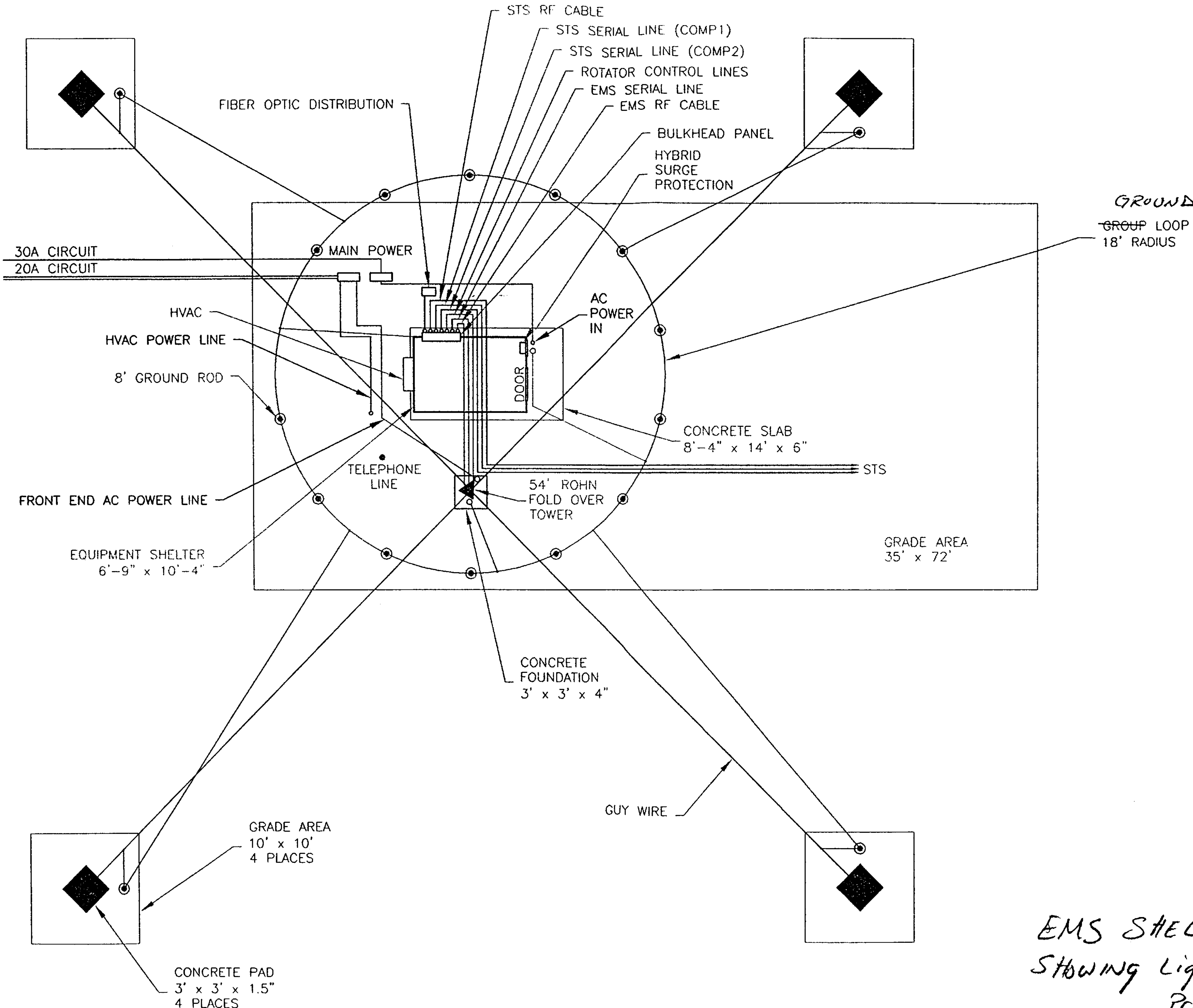
20010208.1635
ANH.DWG

VLA/VLBA ANTENNAS

ANTENNA MOUNT
ENVIRON. MONITORING SYS.
ASS'Y & DETAILS

SCALE NONE | REV | SHEET NUMBER 1 OF 1



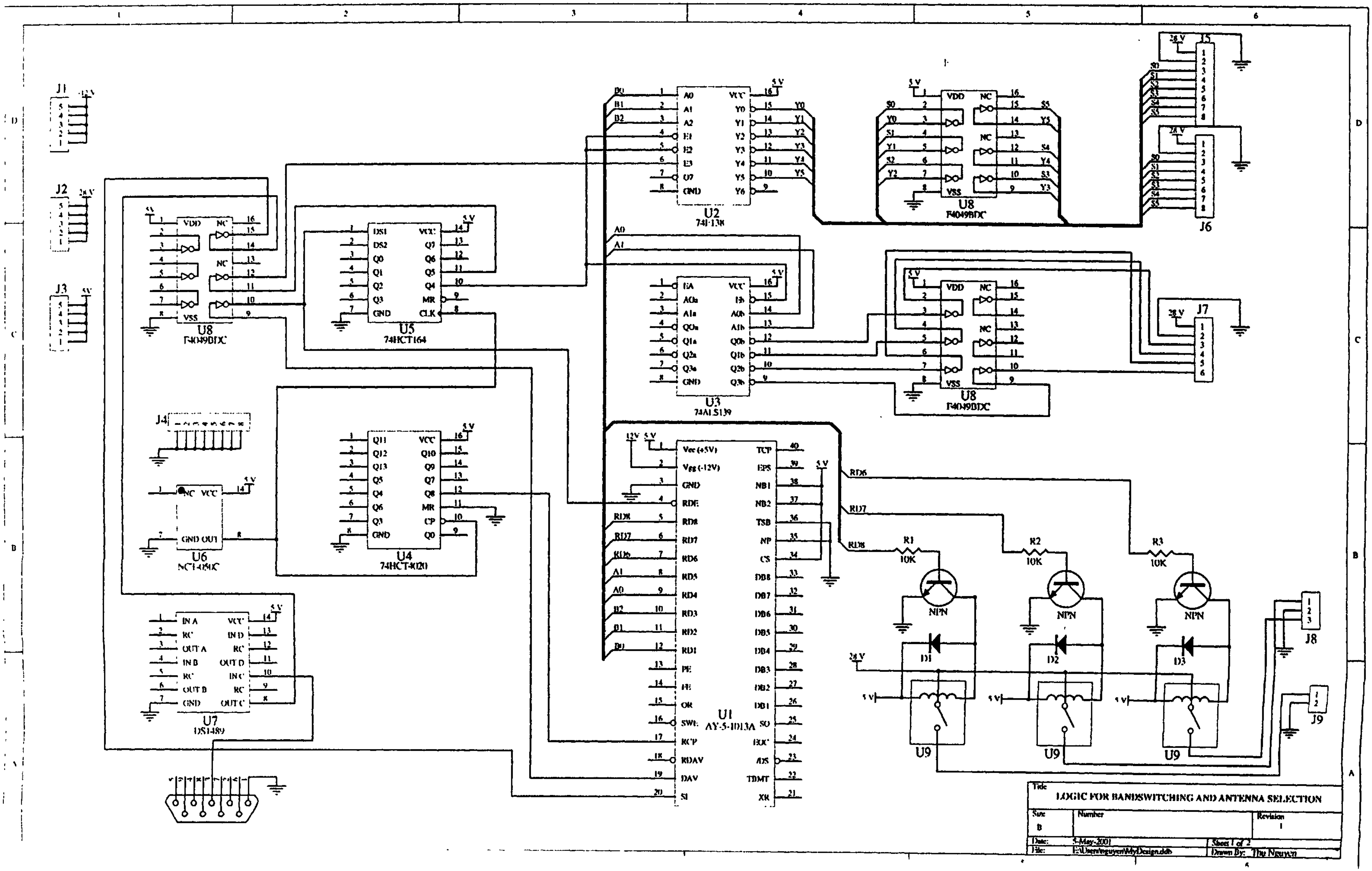


EMS SHELTER
Showing Lightning Protection

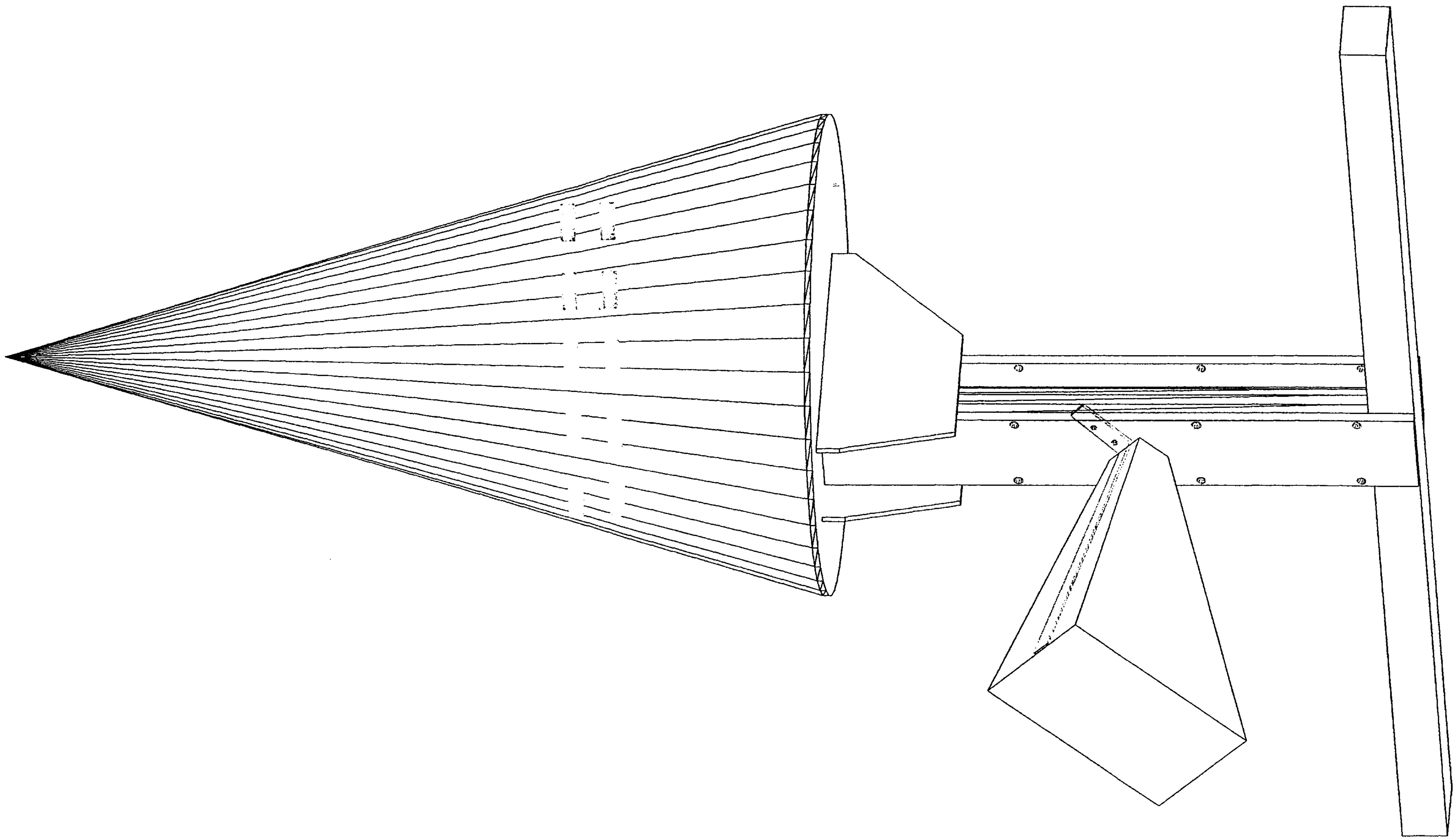
A3. Logic Schematic for Band Switching

Table 1: Part Name for the Logic Schematic

Designator	Part Name
U1	UART
U2	3:8 Decoder
U3	2:4 Decoder
U4	12 Stage Ripple Counter
U5	8 Bit Shift Register
U6	Crystal Clock Oscillator
U7	Quad Line Receiver (Driver)
U8	Hex-Inverting Buffer
U9	Reed Relay



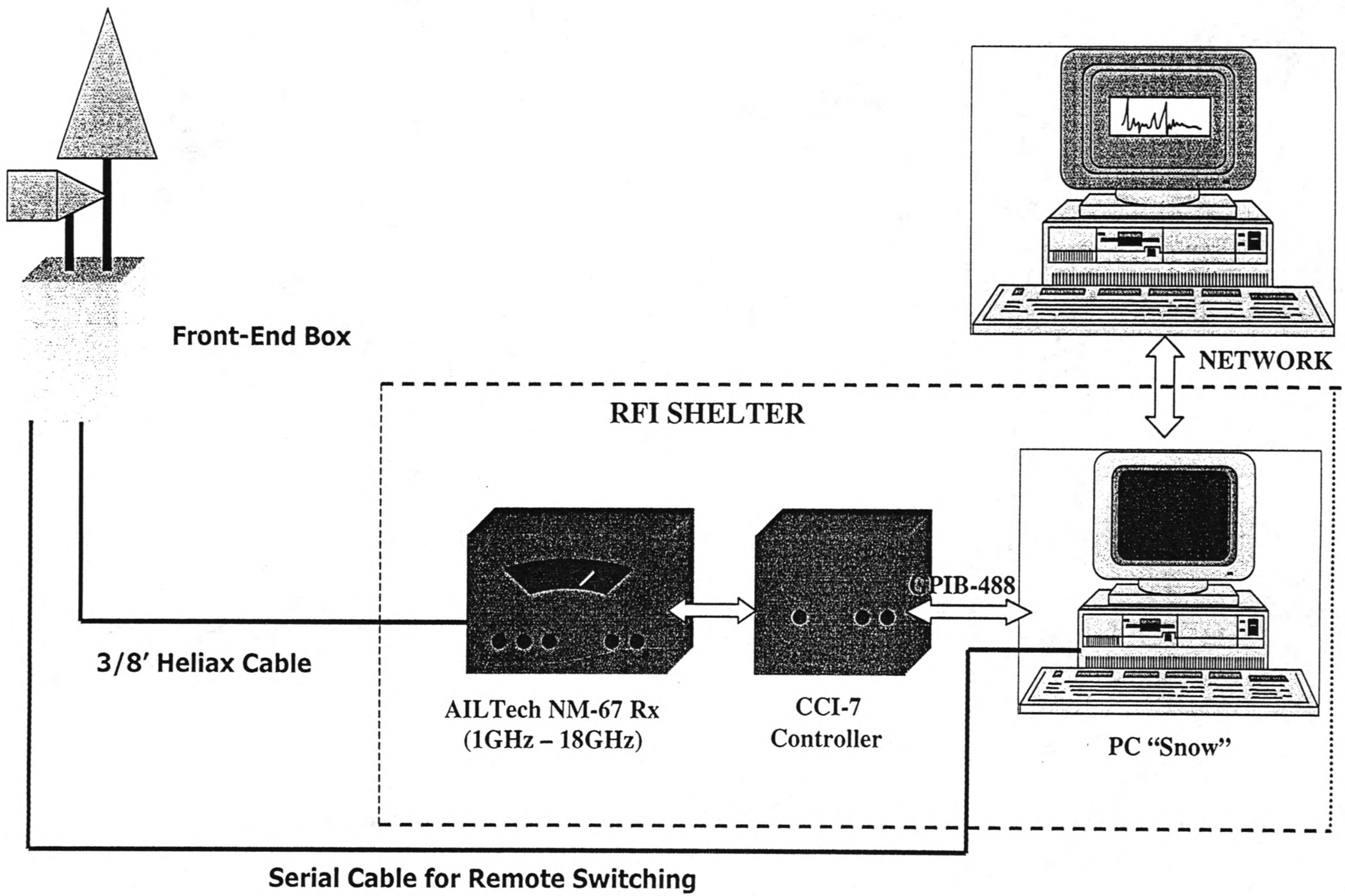
J1
J2
J3
J4
J5
J6
J7
J8
J9
U1
AY-5-1013A
U2
74F138
U3
74ALS139
U4
74HCT4020
U5
74HCT164
U6
U7
DS1489
U8
F4049BDC
U9
R1
R2
R3
10K
NPN
D1
D2
D3
28 V
5 V
12V
Title
LOGIC FOR BANDSWITCHING AND ANTENNA SELECTION
Size
B
Number
Revision
1
Date:
5-May-2001
Sheet 1 of 2
File:
Drawn By:
Thu Nguyen



A4. Maintenance for the Lightning Protection System

<u>Metal Conduit</u> (3 lines from the shelter to the front end box and the rotator)	-Frequent inspection and resistance measuring of connectors to assure continuity.
<u>Lightning Rod</u>	-Frequent inspection to assure no corrosion on the rod and the wire connecting the rod to ground.
<u>MOV Box for the rotator</u>	-Every three months, check the MOV (Metal Oxide Varistors) in the box at the rotator. -Replace MOV if they're burned out. (If possible, during lightning season, check more often) MOV type: **Little-Fuse V68ZA20**
<u>**ZoneMaster** Hybrid Surge Protector</u> (for the AC Power Supply inside the RFI shelter)	-The **ZoneMaster** has three phases. In case lightning strikes and one of the phases burnt out, disconnect the wire and connect it to one of the other phases. *Notes: Hybrid Surge Protector should be installed by the electricians at the VLA.*
<u>**SYSTEMTRAB** Surge Protector</u> (inside the front-end box)	-Each protection modules inside the surge protector is monitored by its own thermal disconnect circuit. All internal circuits are connected to a diagnostic light located on the front of the **SYSTEMTRAB** panel. When the light is ON, all modules are functioning properly and the circuit protection is assured. If the diagnostic light should go OFF, then one or more modules have sacrificed its life to protect the equipment. Identify which module has failed and replace the module. *Notes: Refer to PHOENIX CONTACT technical guideline for instructions on how to replace these modules*

Figure 15: Block Diagram of RFI Monitoring System



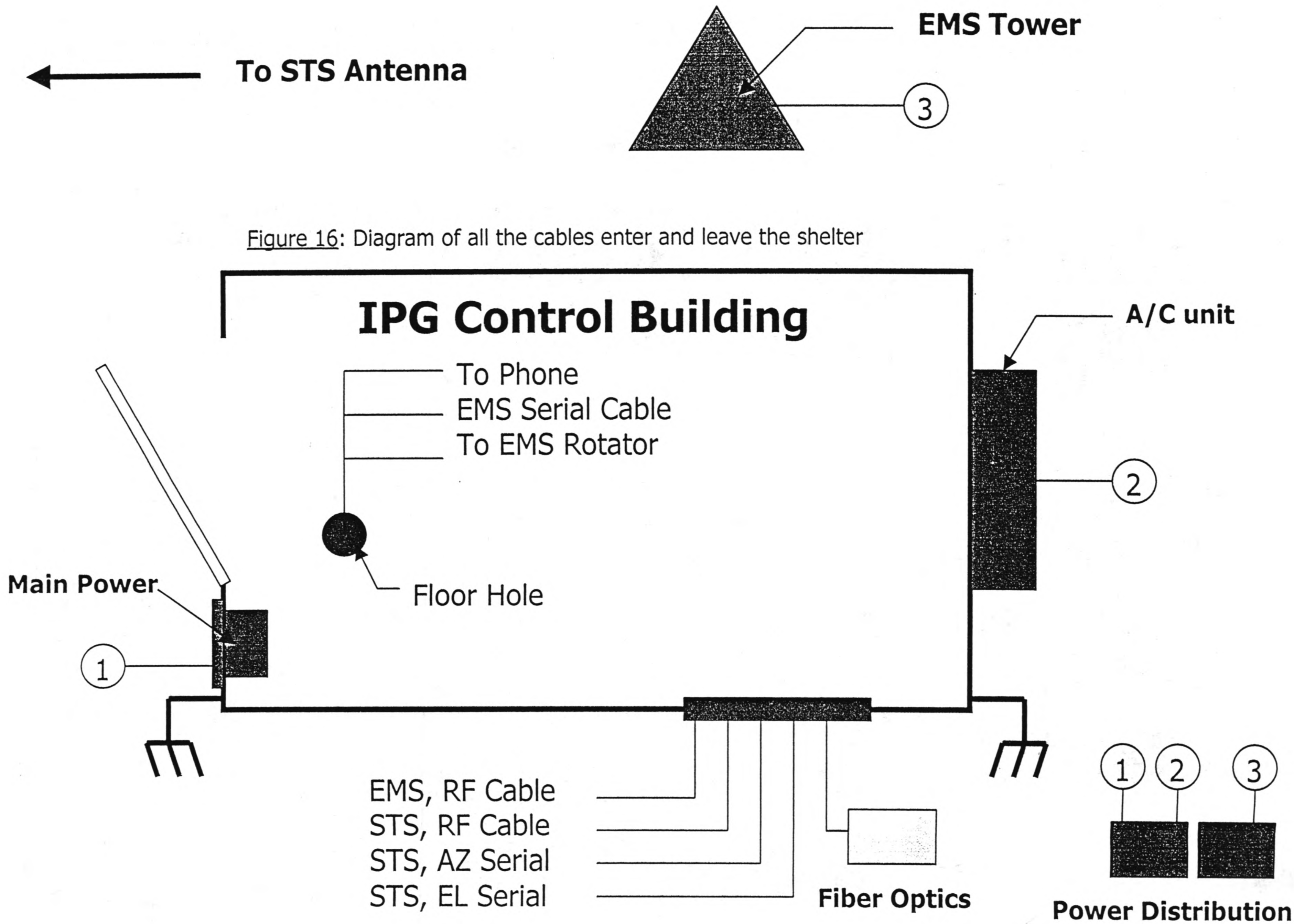


Figure 16: Diagram of all the cables enter and leave the shelter