Mobile Repeater

Murium Iqbal
Kevin Ngan
Grover Viernes
Sunil Khanolkar
Dan Mendat
Yuriy Shames

Rutgers, School of Engineering
Department of Electrical and Computer Engineering
332:428 – Capstone Design: Communications System, Wireless Communications

Senior Design Project Report

Advisor: Professor Christopher Rose

Introduction

With all the wireless systems in use today, getting a strong connection to a network has become a prevalent issue. Many times, frustrated users sit at their desktops, unable to get a wireless connection. Others pace about with their laptops, hoping to find a place where the signal is strong enough to allow them to gain access to the internet. There are systems in place already that try to alleviate these problems.

Some signals are propagated through wireless repeaters. If a user's laptop is too far away from its access point, a wireless repeater can be placed in between the computer and the router. The repeater will receive the signal from the router and retransmit it at a higher power. This will extend the range of the wireless network, giving the laptop access. These repeaters can be bought fairly cheaply at $40- $80 dollars, yet, they do have their drawbacks. Repeaters are stationary hardware; they are put in one place and left there to perform their duty. The environment they are in, though, is not stationary. Say a person has placed a wireless repeater outside their house. Now, perhaps, someone decides to build a huge wall blocking the side of the house with the repeater. The repeater has now become useless; it cannot receive the signal that it is supposed to repeat. The user must now take down the repeater and find a better place to put it.

Directional antennae are also a common attempt to forge solid networks. They have high gain and can transmit signal to a receiver by pointing to it. Yet these have similar issues to the wireless repeater; they are not adaptable to changing environments. For a directional antenna to be useful, the antenna has to know the placement of the receiver it is trying to transmit to. If the receiver moves, the antenna is no longer useful. Also, since most of the power of transmission is in one direction, if a similar wall was built in front of a directional antenna, it would not be able to perform its functions adequately.

Another possible solution to get the signal to receiver would be to just put more power into the signal at the transmitter, but the FCC regulates the power at which a signal can be transmitted at. Also, the higher the power in the signal being transmitted, the more power and battery life is used at the transmitter. If a laptop is communicating with a router which is transmitting at a higher power to send it a signal, when it tries to communicate back it will use up its battery life much faster since it would need to transmit using the same power. This would be very inefficient and inconvenient for the user.

This project proposes a solution that overcomes the limitations of the other systems. If the problem with a repeater and directional antenna is that they are not adaptable to their changing environments and are constantly needed to be repositioned to continue functioning, perhaps they should simply be made mobile. This would allow them to move around obstacles that are placed in their paths. A mobile platform could carry both a directional antenna and a repeater, allowing them to maneuver through their environments to relay a signal from a router to a laptop and back. The directional antenna could home in on the signal from the router while the repeater could repeat the signal back to the laptop. This would allow the laptop user to remain stationary and transmit at the same power as it would normally communicate at.

Thus, the end result of the project is to have a robot that can automatically find and transmit a signal between a router and a laptop. To implement this system, three major components are needed. The propagation of the wireless network must be performed by a directional antenna and a wireless repeater. These must be put onto a mobile platform. Yet, the mobile platform has to avoid obstacles and collisions in its environment, hence a collision detection system is also necessary for the overall robot to function.

Mobile Repeater

Communications Subgroup

Murium Iqbal
Kevin Ngan

Rutgers, School of Engineering
Department of Electrical and Computer Engineering
332:428 – Capstone Design: Communications System, Wireless Communications

Senior Design Project Report

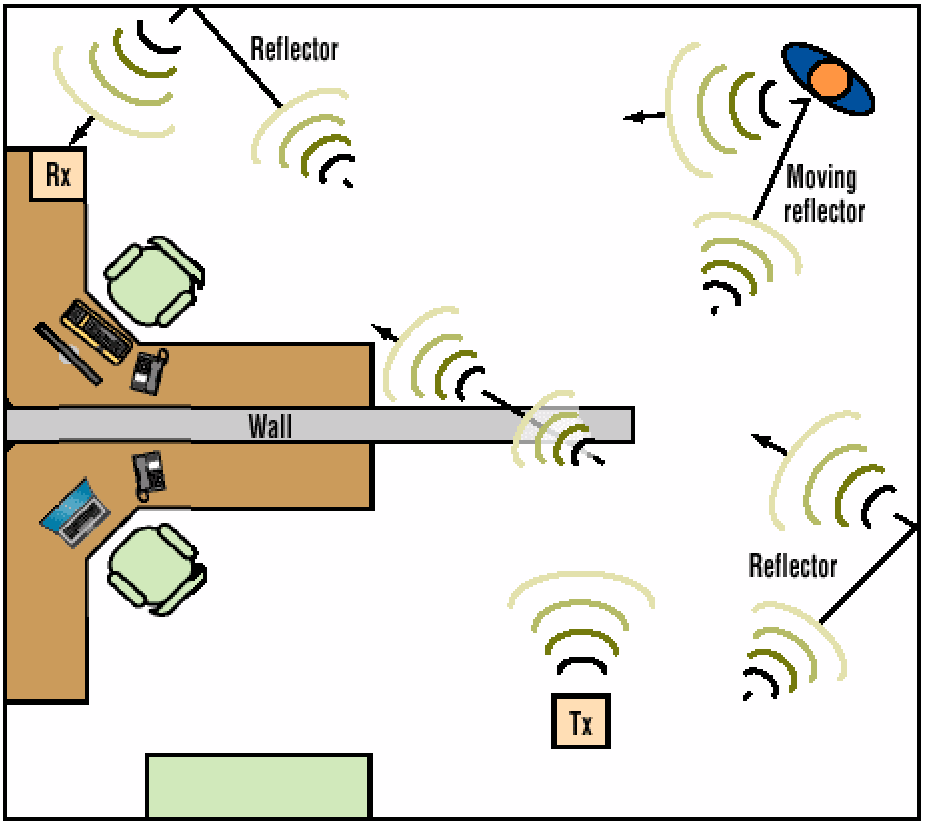
Advisor: Professor Christopher Rose

Abstract

The purpose of the overall project was to design and build a smart mobile repeater. The repeater will navigate an environment in search of the highest signal strength from a distant router. After it sufficiently acquires a high level signal, it will act as a relay so that other computers may connect to the router through it. The project was divided into three components: a mobile platform, a communications system, and a sensor network. This part of the paper focuses on the communications aspect of the project. It is responsible for isolating high signal strength and repeating the Wi-Fi signal.

1 Introduction

In a day to day setting, if a laptop has no direct line of sight to its router, how does it get signal? There are still bounce paths that can give it a signal, even if it has no direct path to the router. Bounce paths occur when a signal is propagated through an environment with obstacles; the signal collides with these obstacles and is either absorbed or reflected off in different directions. In this way, the signal can bounce from obstacle to obstacle around the area (the signal does lose signal strength as it moves through the environment).

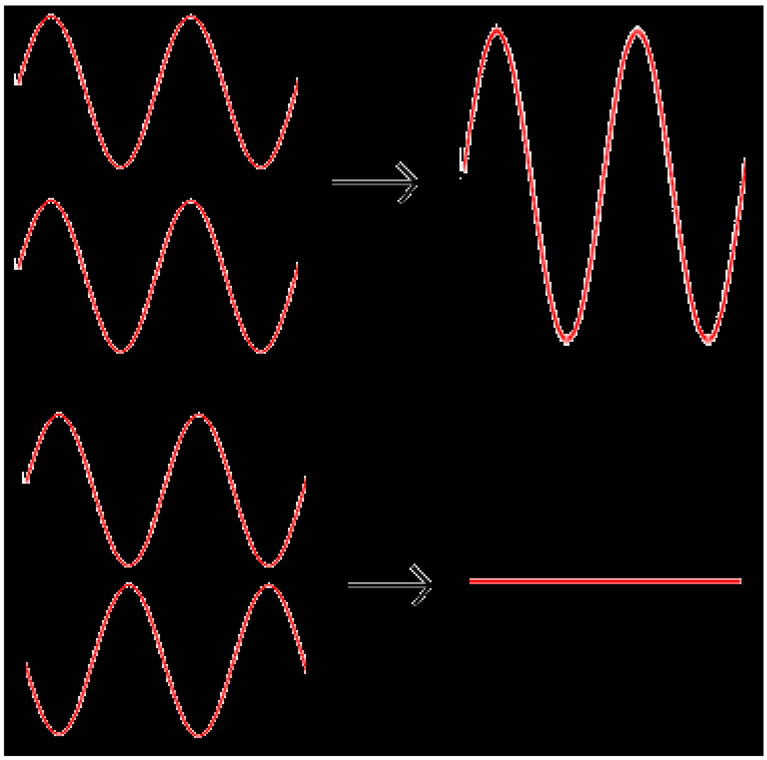


1. In real-world environments, signals find multiple paths from radio transmitter to radio receiver.

If the signal bounces around enough, it may eventually make its way to the receiver. One can think of the signal as a pinball, and the obstacles as the little objects in a pinball machine. The pinball bounces from object to object, until it reaches its destination, getting smaller and smaller with each collision. The path that it finds to the receiver is called a bounce path. Using these bounce paths, signals can make their way from the transmitter to the receiver, and, as long as it hasn't lost too much power bouncing around, give the receiver signal when it does not have a line of sight.

Since the point of the project is to allow the user to remain stationary while a robot hunts for the signal, the robot must be mounted with antennae to relay signals between the router and the laptop. One may intuitively think that if the robot were to just move in the direction of the router it should start getting better and better signal. Yet, due to the complex way in which signals travel through environments, this may not always be the most efficient solution. The many different waves that are occurring from the many different bounce paths interfere with one another. Some of the interference may be constructive or destructive.

Recall from physics that if a sinusoidal signal is superimposed onto another such signal which is out of phase with it by 180 degrees, then the two signals will interfere destructively and cancel each other out. If these two signals are actually in phase (0 degrees), then they will interfere constructively and the signal strength will increase. If the phase off-set of the waves is in-between these two values, then the signals will interfere partially constructively and partially destructively.



The many signals which are bouncing around the room interfere with one another. Thus, moving a remote controlled robot towards the door of a room in which a laptop is sitting may not be the most efficient way to get the laptop signal. There may be a spot in the room where the signal is stronger. Hence, this design uses a directional antenna to sweep the area around the robot for the best possible signal strength, which could, in fact, be resultant of a bounce path that isn't in the direction of the router.

This antenna system also allows the robot to be automatic, instead of having it be remotely directed towards a router. The sweeping functionality allows the robot to determine which direction to move in by itself, and since the direction of the best signal strength may not be intuitive or obvious, it may find a better signal strength faster and more efficiently than a user would.

2 Background

The idea to use directional antennae has many benefits. Primarily, directional antennae increase gain. While an omni-directional antenna transmits in all directions equally, a directional antenna focuses its power in a specific area. Also, as mentioned before, a directional antenna can help the robot localize the signal it is hunting. To do this, the robot can use the antenna to sweep the area around it. Sweeping implies that the antenna will first focus its power in one direction, then another, then another, rotating (physically or electronically) around in a circle and collecting data. Data from the antenna can then be compared to decide which direction has the strongest signal strength, and hence which direction to move in.

Originally, the robot was intended to have two directional antennas; this would allow the robot to localize both the router's signal and the receiver's signal. After realizing the complexity of having to coordinate two directional antennae at a time, it was decided that one antenna would be directional and the other would be a high gain omni-directional antenna. The directional antenna will focus on the signal from the router and the omni-directional antenna will transmit to the receiver(s).

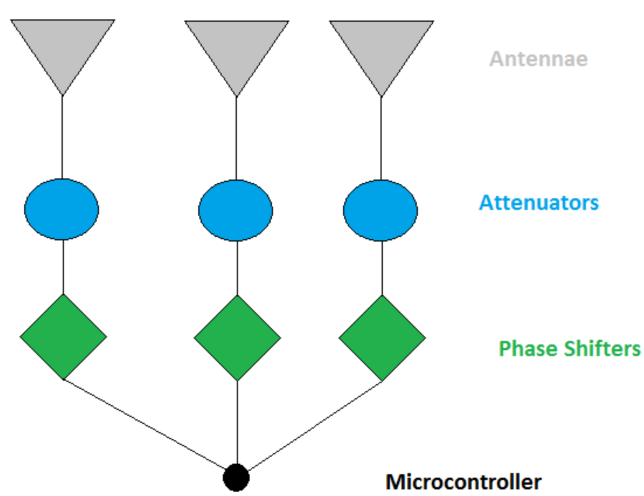
There are many different types of directional antennae to choose from, two of which were considered for the purposes of this project: a phased array antenna or a waveguide antenna (which shall be referred to as a cantenna from here onwards).

3 Technical

3.1 Directional Antenna

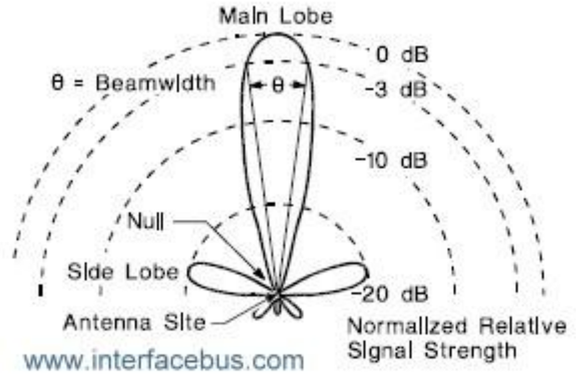
3.1.1 Phased Array

A phased array is actually a set of multiple antennae which act together to home in on signals. Each antenna is set on a track/rail with its own attenuator and phase shifter. The responses of each track are then added together to get the response of the whole system.



To transmit or receive to one particular point the phase of each antenna is offset, so that when the high frequency waves converge at that point, they interfere constructively. As can be imagined, very complex antenna plots can be generated depending on the pattern of attenuation and shifting on these antennae. If the antennae are off-set in a certain way, the phased array can be set in such a way that areas of high noise can even be attenuated!

For the purposes of this project, one could generate an antenna plot which resembles a beam.



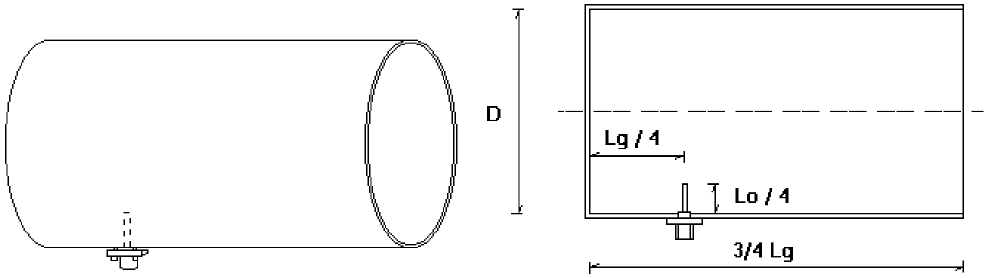
This fixed beam pattern could then be electronically swept in a circle in search of the signal which was to be received. To do this, the phase on the antennae could be changed so that the areas in which the HF waves constructively interfered were forming a similar beam as above, but in a different direction. The robot could then be steered in the direction which had the best signal strength.

Interested parties can view an applet on the following website that generates an antenna plot. It allows the user to change the number of radiating elements, the distance between them in tenths of wavelengths, and the phase shift between each through scroll bars: http://www.kyes.com/antenna/antennatypes/array/antenna.html

3.1.2 Cantenna

A cantenna is a cylindrical waveguide directional antenna. Waveguides are structures which guide waves (sound, electro-magnet, light) in different directions. Depending on the type and frequency of the wave, different materials must be used. This project is designed to work high-frequency (about 2.4 GHz) electro-magnetic waves. The reason that the project is centered round 2.4 GHz is because wi-fi signals propagate at this frequency (they also propagate at 3.6 GHz, 5 GHz, but 2.4 GHz is the most prevalent). A tin can can be used to direct the signals within this range and will act as a band-pass filter.

The tin can's cylindrical shape can direct the high-frequency signals into its structure, and reflect them off its back. The superposition of the reflection and the incoming signal results in a standing wave; the dimensions of the cantenna depend on this wave. The length of the can should be at least three-fourths of this wave length to properly direct the high-frequency signal into the can. Also, the first maximum of the standing wave occurs at one-fourth its wavelength in from the back of the can. This is the place in the can where the radiating element should be placed.



In the above images, Lg is the wavelength of the standing wave and Lo is the wavelength of the high frequency signal in open air

To find the wavelength of standing wave, the wavelength of the incoming high-frequency signal and the wavelength of the low cut-off frequency of the cantenna are needed. The wavelength of the high-frequency signal is 122 mm (for 2.4 GHz). To calculate the low cut-off frequency (and from this the wavelength) of the cantenna start with wave equation for E_z (since the z coordinate is going into the cantenna) in polar coordinates (since the opening of the can is a circle).

$$(\nabla^2 + k^2)E_z = 0$$

$$\left(\frac{\partial^2}{\partial\rho^2} + \frac{1}{\rho}\frac{\partial}{\partial\rho} + \frac{1}{\rho^2}\frac{\partial^2}{\partial\phi^2} + \frac{\partial^2}{\partial z^2} + k^2\right)E_z(\rho,\phi,z) = 0$$

If the equation for H_z is represented as such:

$$E_z(\rho,\phi,z) = R(\rho)P(\phi)e^{-j\beta z}$$

separation of variables can be executed by the following procedure:

$$\left[R''P + \frac{1}{\rho}R'P + \frac{1}{\rho^2}RP'' + \underbrace{(k^2 - \beta^2)}_{k_c^2}RP\right]e^{-j\beta z} = 0$$

Now dividing through on both sides by the H_z as RP gives the following:

$$\underbrace{\rho^2\frac{R''}{R} + \rho\frac{R'}{R} + \rho^2 k_c^2}_{\text{function of } \rho} + \underbrace{\frac{P''}{P}}_{\text{function of } \phi} = 0$$

As is taught in Advanced Calculus, the only way that the two separate functions can sum to a constant is if they both are themselves constants. Thus the follow assumption can be made:

$$\frac{P''}{P} = -k_\phi^2 \qquad \rightarrow \qquad P'' + k_\phi^2 P = 0$$

where $-k_\phi^2$ is an unknown constant

Solving the above differential equation gives:

$$P(\phi) = A_0 \sin(k_\phi \phi) + B_0 \cos(k_\phi \phi)$$

Unfortunately, the equation for R does not solve as neatly. The differential equation for R is a Bessel Differential equation:

$$\rho^2 R'' + \rho R' + (\rho^2 k_c^2 - k_\phi^2) R = 0$$

The solution to this differential equation is:

$$R(\rho) = C_0 J_{k_\phi}(k_c \rho) + D_0 N_{k_\phi}(k_c \rho)$$

Where $J_v(x)$ is a Bessel function of the first kind of order v and $N_v(x)$ is a Bessel function of the second kind of order v.

Recall now that E_z was represented as RP, thus now we can write:

$$E_z(\rho, \phi, z) = \left[C_0 J_{k_\phi}(k_c \rho) + D_0 N_{k_\phi}(k_c \rho)\right] \left[A_0 \sin(k_\phi \phi) + B_0 \cos(k_\phi \phi)\right] e^{-j\beta z}$$

Since the above is a sinusoidal wave, it must be that:

$$E_z(\rho, \phi, z) = E_z(\rho, \phi + 2\pi\ell, z)$$

Where l is an integer

Hence it can be deduced that k_ϕ must be an integer. This yields:

$$E_z(\rho, \phi, z) = \left[C_0 J_\nu(k_c \rho) + D_0 N_\nu(k_c \rho)\right] \left[A_0 \sin(\nu\phi) + B_0 \cos(\nu\phi)\right] e^{-j\beta z}$$

In the above equation, as ρ approaches 0 $N_v(k_c\rho)$ approaches -∞. Yet, ρ = 0 is in the domain of the waveguide and an infinite field intensity cannot occur at this point. Thus, D_0 must be zero to get rid of this term. This leaves the following terms in the equation:

$$E_z(\rho, \phi, z) = \left[A \sin(\nu\phi) + B \cos(\nu\phi)\right] J_\nu(k_c \rho) e^{-j\beta z}$$

Using the boundary condition that $E_z(a, \phi, z) = 0$ gives $J_\nu(k_c a) = 0$. This then leads to the following:

$$k_c a = p_{\nu n} \qquad \rightarrow \qquad k_c = \frac{p_{\nu n}}{a}$$

where p_{vn} is the nth zero of $J_v(x)$ and a is the radius of the can

The dominant mode of the waveguide is TE_{11} which gives a value of p at 3.8317. The cut-off frequency can then be calculated as:

$$f_{c,\nu n} = \frac{c}{2\pi} \frac{p_{\nu n}}{a}$$

Thus to find the wavelength of this wave one would take c/f and get λ is equal to 1.64 times the radius of the circle.

Together this wavelength, the high frequency wavelength and the standing wave wavelength form a right triangle:

$$(1/Lo)^2 = (1/Lc)^2 + (1/Lg)^2$$

In the above equation Lo is the wavelength of the high frequency signal, Lc is the wavelength of the low cut off frequency and Lg is the wavelength of the standing wave. Thus by knowing the former two wavelengths, one can solve for the latter. The can used as the waveguide must have dimensions that match the aforementioned parameters.

4 Approach

4.1 Phased Array

The original plan for the antenna was to use a phased array on top of a platform. Using a phased array would nullify any need for physically rotating the antenna, since it would electronically sweep the area for signal. It would also be highly adaptable to changes in the environment, since the antenna plot could be changed to match whatever was called for in the environment (i.e. attenuating a known source of noise, etc.).

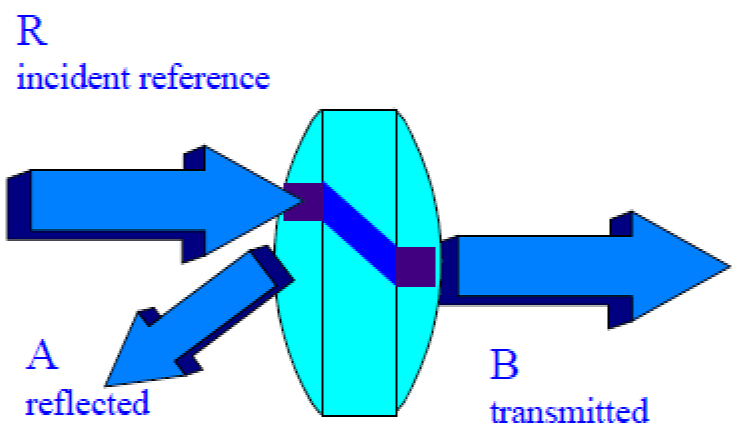
The parts that were needed for building the phased array were digital attenuators, digital phase shifters and the antennae. The idea was to build a rail for each antenna, with its components leading to it, and have the different attenuators and phase shifters controlled by a micro-controller. There would also need to be circuitry to sum the analog signals coming in from the different rails.

Although the idea of a phased array seems appropriate for the project, in practice it did not seem like a feasible plan. After trying to locate venders who sold attenuators and phase shifters in the wi-fi range, it was found that the parts were outside of the budget for the project. The voltage controlled attenuators that operate between 2-4 GHz were quoted at $1,098.00 each. Many of the vendors (such as DAICO and RFGlobalNet) did not respond to requests for quotes on their parts, and those that did required 6-8 weeks of processing time for the order. Also, the math involved for building and working the phased array is very complicated; it would have taken far more time than was available to make the phased array fully functional. What with the amount of time constraints and the budget for this project, the idea of using a phased array was discarded. Instead, the directional antenna would have to be implemented with a waveguide antenna in the form of a can (the cantenna).

4.2 Cantenna

The Cantenna was an easy enough idea to implement. Building one required very little work, and utilizing the cantenna was as simple as a "point and shoot" procedure. The parts were all very cheap and quickly accessible. Once the antennae had been constructed to specification, their directionality had to be tested. To do so, two cantennas were constructed. Both of them were then connected to a Vector Network Analyzer.

The VNA measures S-paramters or scattering parameters. When a signal enters a network through a transmission line, some of the waves are reflected and some are scattered.



The S-parameters measure the "voltage out versus voltage in" of a system. For the two-port system (each antenna being a port) there are four S-parameters. S_{11} and S_{22} measure the reflection of port 1 onto itself and the reflection of port 2 onto itself, respectively. The parameters S_{12} and S_{21} measure the transmission from port 2 to port 1 and the transmission from port 1 to port 2 respectively.

The VNA can be set to measure the transfer function for these 4 parameters for a wide range of frequencies. It sweeps a continuous sinusoidal signal through the desired frequencies at the transmitting antenna. (Note that a sweep here is not the same as the sweep mentioned previously in the paper. Here a sweep is referring to the fact that the frequency of the sinusoid will be increased from a starting frequency to an ending frequency and data will be gathered on this interval.) The signal is then picked up from the receiving antenna and the rate of transfer (voltage out versus voltage in) can be measured.

For the purposes of this lab, the sweep range was set to be from 1 to 4 GHz. To prove directionality, one antenna was held rigid pointing directly at the other. The other antenna started off facing directly back. It was then rotated counter clockwise in 45° increments and data from the VNA was collected each time.

Test set-up at WINLAB

For anyone wishing to continue with this project, a VNA can be found at WINLAB. The parts used to build the cantenna were:

- Baby Formula can
- Piece of wire (as the radiating element)
- N-type female connector
- Nuts and bolts
- Wi-fi adapter (USB to RP-SMA)

The hole was punched into the can at the proper distance from the back of the can, but initially the N-type connectors were held in place using massive amounts of tape. This was highly unstable, and could not support the heavy wires that were leading to the VNA. To make the cantenna more durable, the N-type connectors were then bolted into the can. A collection of connectors were used to go from the RP-SMA connector on the wi-fi card to the N-type female connector. These are all also readily available at WINLAB.

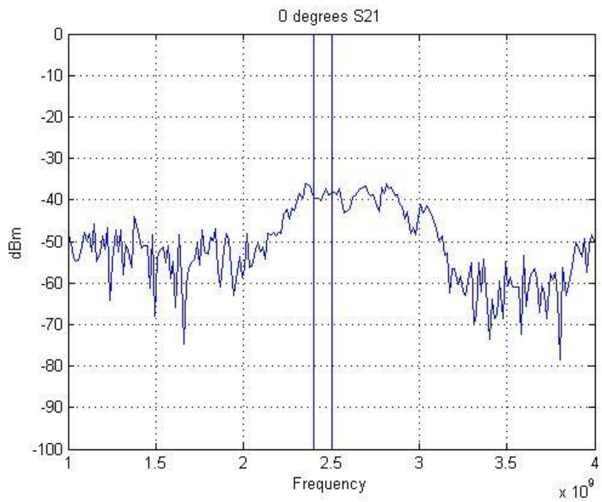
4.2 Repeating Function

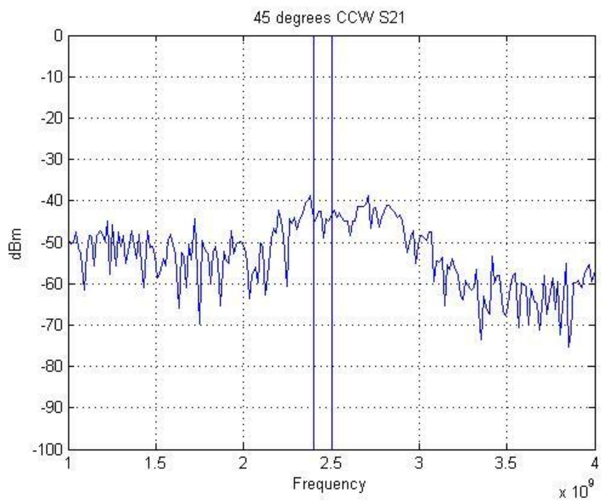
Once the antenna was constructed, tested, and mounted, the next step was to repeat the Wi-Fi signal. After researching online on ways to turn a laptop into a receiver, it was decided that an ad-hoc network using the internal wireless card would be created. The external USB Wi-Fi adapter was connected to the access point, which freed up the internal card. Once the ad-hoc network was created, the access point's hardware address was linked to the ad-hoc network. With both adapter's connected, another laptop was linked into the ad-hoc network. The new laptop was able to connect to the internet through the ad-hoc network.

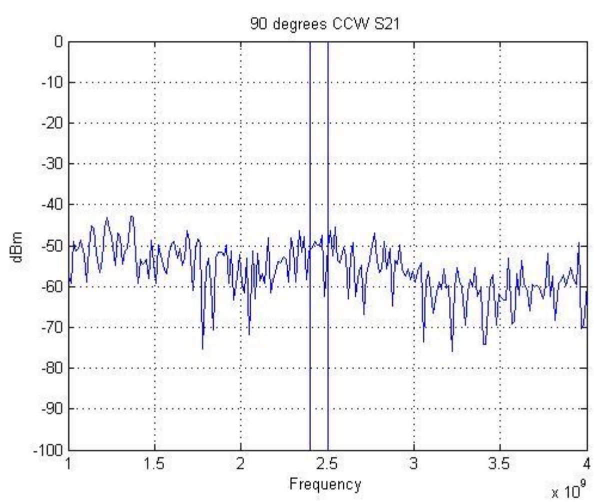
5 Results

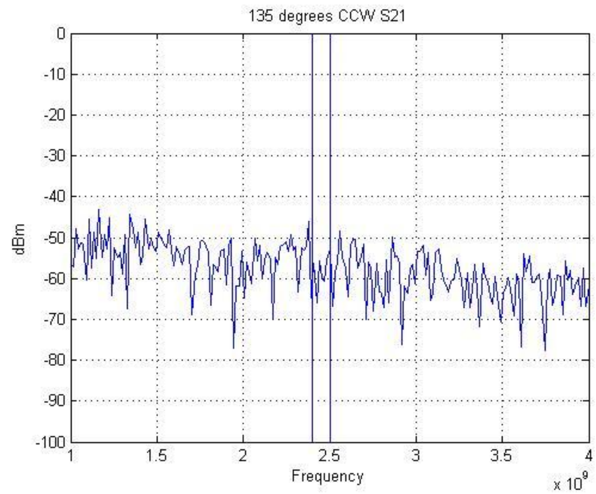
5.1 Cantenna

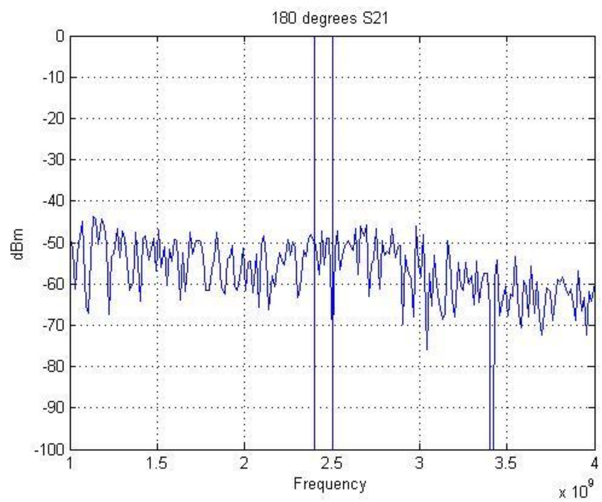
The following are the graphs from the VNA, they are only of the S_{21} parameters since, theoretically, the S_{21} and S_{12} should be the same (they were the same experimentally as well). The y-axis is in dBm and the x-axis is in GHz. The two vertical lines are surrounding the wi-fi frequency range, so that the response of the antennae within that range is outlined in each graph. When the two antennae are facing each other, the transfer function is that of a band-pass filter and the gain is relatively high in the wi-fi range. At a 45° deviation from the starting forward position, the gain is still relatively higher, but does drop significantly. As the antenna is rotated further away from the starting position the gain of the system drops, until it is again 45° away from the starting position, when it begins to pick back up.

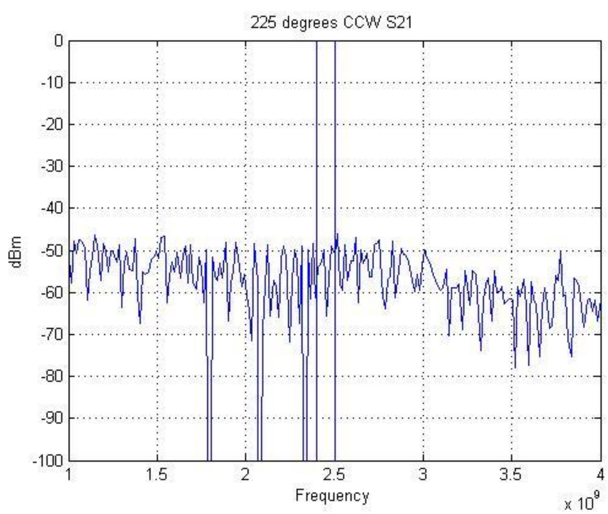

0 degrees S21
dBm
Frequency
x 10^9
0
-10
-20
-30
-40
-50
-60
-70
-80
-90
-100
1
1.5
2
2.5
3
3.5
4

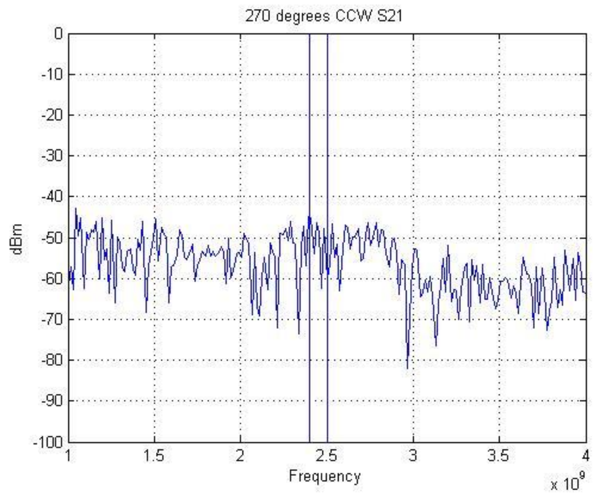

45 degrees CCW S21
dBm
Frequency
x 10^9
0
-10
-20
-30
-40
-50
-60
-70
-80
-90
-100
1
1.5
2
2.5
3
3.5
4

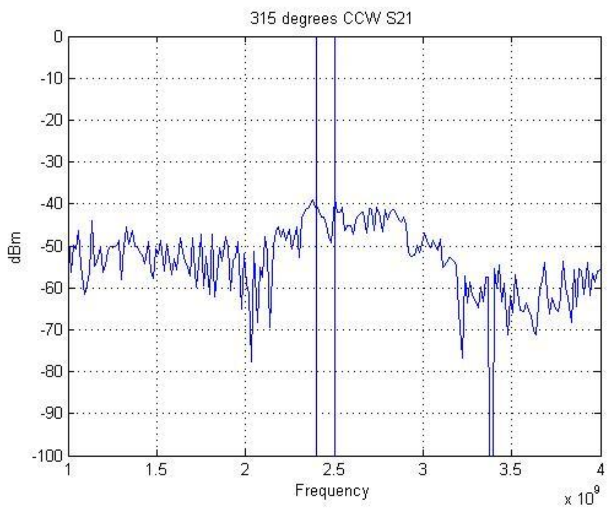

90 degrees CCW S21
dBm
Frequency
x 10^9
0
-10
-20
-30
-40
-50
-60
-70
-80
-90
-100
1
1.5
2
2.5
3
3.5
4


135 degrees CCW S21
dBm
Frequency
x 10^9


180 degrees S21
dBm
Frequency
x 10^9


225 degrees CCW S21
dBm
Frequency
x 10^9





5.2 Ad-hoc Network

Creating the ad-hoc network only required a simple procedure; it was established as follows:

- Start up netbook
- Connect to access point
- Open terminal and run command iwconfig to get the MAC address of the access point
- Disconnect from the access point
- Create ad-hoc network using the internal adapter
- Open up the properties of the newly created network
- Enter the MAC address of the access point into the BSSID field
- Connect external usb wireless adapter
- Connect to access point

Though the ad-hoc network functioned on some occasions, there were also times that it failed to work. These failures occurred sporadically and, due to a lack of knowledge on ad-hoc networks, there was not enough time to debug the issues. One possible issue was identified, though: setting up the network on RUWireless caused some issues. To gain internet access, both the robot and the laptop had to be logged onto a Rutgers account.

This could lead to some problems if both laptop and robot tried to use the same account. The above process did work consistently until integration with the robot, but, as was previously stated, there was not enough time to debug the issue.

5.3 Future Work

For anyone hoping to further this project, a few improvements are suggested here. First, the ad-hoc network can be debugged and made to work consistently. More research can be put into a phased array antenna and perhaps one can be built for the future. Another possibility is to have two directional antennae, one for the laptop and one for the router. Also, if multiple robots are used, a system can be put in place for them to pass the signal between one another until it reaches its destination. Since our current method does not actually repeat the signal, this functionality can also be added instead of creating an ad-hoc network.

6 Appendices

6.1 Code

This is the code written by Kevin Ngan to grab the signal strength from the directional antenna. It was later modified by Dan Mendat as part of integration.

antenna.h

```
/**
* This class is used to control an antenna found on the mobile repeater.
* It can be used to get the signal strength of the signal the antenna is
* using.
*
* Authors: Dan Mendat and Kevin Ngan
* Date: Spring 2010
*/
class antenna
{
public:
	/**
	* Get the signal strength the antenna is receiving in dBm.
	*/
	float getSignalStrength(int);
};
```

antenna.cpp:

```
/**
* Please see antenna.h for an explanation of this class and its functions.
*
* Authors: Dan Mendat and Kevin Ngan
* Date: Spring 2010
*/

#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include "antenna.h"

float antenna::getSignalStrength(int numSamples)
{
	float average = 0.0;
	int sum = 0;
	int value = 0;
	int k;

	for(k = 0; k < numSamples; k++){
		//system(); - C code to run UNIX commands
		//grep - prints to the terminal the line specified

		//This grabs the Signal Level phrase and outputs it to text file
temp.txt
		system("iwconfig ra0 | grep -o 'Signal level:-[0-9][0-9]' >
signal_strength.txt");

		FILE* stf;	//stf = signal text file
		stf = fopen("signal_strength.txt", "rwx");		//opens text
file with full access

		if (stf == NULL){
			fprintf(stderr, "Can't open wireless  file\n");
			exit(1);
		}

		char signal[20];
		while (fscanf(stf, "%s", signal) != EOF);

		fclose(stf);
```

```
                value = atoi(signal + 6);
                sum = sum + value;
                average = (float)sum / (k + 1);

                if (k != numSamples - 1)
                {
                        usleep(750000);
                }
        }
                remove("signal_strength.txt");
                return average;
}
```

6.2 Data

The data taken from the network analyzer during the antenna testing is attached here.

S11

Frequency	dBm
1.00E+09	-4.68
1.02E+09	-4.55
1.03E+09	-4.33
1.05E+09	-4.18
1.06E+09	-4.20
1.08E+09	-4.71
1.09E+09	-4.88
1.11E+09	-4.76
1.12E+09	-4.27
1.14E+09	-4.35
1.15E+09	-4.48
1.17E+09	-5.21
1.18E+09	-5.13
1.20E+09	-4.54
1.21E+09	-4.35
1.23E+09	-4.60
1.24E+09	-5.13
1.26E+09	-5.41
1.27E+09	-4.92
1.29E+09	-4.64
1.30E+09	-4.61
1.32E+09	-5.27
1.33E+09	-5.70
1.35E+09	-5.22
1.36E+09	-4.92
1.38E+09	-4.70
1.39E+09	-4.95
1.41E+09	-5.80
1.42E+09	-5.80
1.44E+09	-5.01
1.45E+09	-4.73
1.47E+09	-5.10
1.48E+09	-5.83
1.50E+09	-5.99
1.51E+09	-5.50
1.53E+09	-4.95
1.54E+09	-4.99
1.56E+09	-5.67
1.57E+09	-6.23
1.59E+09	-6.00
1.60E+09	-5.22
1.62E+09	-4.98
1.63E+09	-5.53
1.65E+09	-6.19
1.66E+09	-6.24
1.68E+09	-5.65
1.69E+09	-5.08
1.71E+09	-5.35
1.72E+09	-6.20
1.74E+09	-6.57
1.75E+09	-6.12
1.77E+09	-5.49
1.78E+09	-5.23
1.80E+09	-5.78
1.81E+09	-6.58
1.83E+09	-6.79
1.84E+09	-5.91
1.86E+09	-5.33
1.87E+09	-5.44
1.89E+09	-6.37
1.90E+09	-7.22
1.92E+09	-6.80
1.93E+09	-5.99
1.95E+09	-5.46
1.96E+09	-5.63
1.98E+09	-6.49
1.99E+09	-7.28
2.01E+09	-7.37
2.02E+09	-6.56
2.04E+09	-5.99
2.05E+09	-6.07
2.07E+09	-6.76
2.08E+09	-8.14
2.10E+09	-8.92
2.11E+09	-8.28
2.13E+09	-7.30
2.14E+09	-7.23
2.16E+09	-7.72
2.17E+09	-9.39
2.19E+09	-11.07
2.20E+09	-11.29
2.22E+09	-9.67
2.23E+09	-8.94
2.25E+09	-10.27
2.26E+09	-12.74
2.28E+09	-16.32
2.29E+09	-16.26
2.31E+09	-14.26
2.32E+09	-14.47
2.34E+09	-15.86
2.35E+09	-24.44
2.37E+09	-28.05
2.38E+09	-19.54
2.40E+09	-18.32
2.41E+09	-20.56
2.43E+09	-33.89
2.44E+09	-21.06
2.46E+09	-17.37
2.47E+09	-14.59
2.49E+09	-15.98
2.50E+09	-20.35
2.52E+09	-20.46
2.53E+09	-15.55
2.55E+09	-13.74
2.56E+09	-13.62
2.58E+09	-17.51
2.59E+09	-20.30
2.61E+09	-22.74
2.62E+09	-15.59
2.64E+09	-14.52
2.65E+09	-15.52
2.67E+09	-21.98
2.68E+09	-25.95
2.70E+09	-19.25
2.71E+09	-15.84
2.73E+09	-17.05
2.74E+09	-25.55
2.76E+09	-29.37
2.77E+09	-19.14
2.79E+09	-21.00
2.80E+09	-22.42
2.82E+09	-27.45
2.83E+09	-21.18
2.85E+09	-16.27
2.86E+09	-15.72
2.88E+09	-16.45
2.89E+09	-20.98
2.91E+09	-20.14
2.92E+09	-13.92
2.94E+09	-11.75
2.95E+09	-11.47
2.97E+09	-15.03
2.98E+09	-22.66
3.00E+09	-18.95
3.01E+09	-14.55
3.03E+09	-14.08
3.04E+09	-15.25
3.06E+09	-18.74
3.07E+09	-14.18
3.09E+09	-10.36
3.10E+09	-9.74
3.12E+09	-10.68
3.13E+09	-13.10
3.15E+09	-13.24
3.16E+09	-10.36
3.18E+09	-8.52
3.19E+09	-8.55
3.21E+09	-10.13
3.22E+09	-12.17
3.24E+09	-11.47
3.25E+09	-9.21
3.27E+09	-8.18
3.28E+09	-9.05
3.30E+09	-12.78
3.31E+09	-13.34
3.33E+09	-10.22
3.34E+09	-8.14
3.36E+09	-8.23
3.37E+09	-9.58
3.39E+09	-11.03
3.40E+09	-9.59
3.42E+09	-7.59
3.43E+09	-7.14
3.45E+09	-7.90
3.46E+09	-10.13
3.48E+09	-10.11
3.49E+09	-8.19
3.51E+09	-7.04
3.52E+09	-7.19
3.54E+09	-9.49
3.55E+09	-11.26
3.57E+09	-9.56
3.58E+09	-8.11
3.60E+09	-8.35
3.61E+09	-10.44
3.63E+09	-15.65
3.64E+09	-15.03
3.66E+09	-10.40
3.67E+09	-8.78
3.69E+09	-10.07
3.70E+09	-12.96
3.72E+09	-13.09
3.73E+09	-9.51
3.75E+09	-7.89
3.76E+09	-8.50
3.78E+09	-11.22
3.79E+09	-13.84
3.81E+09	-10.77
3.82E+09	-8.94
3.84E+09	-8.60
3.85E+09	-10.34
3.87E+09	-15.05
3.88E+09	-13.98
3.90E+09	-10.36
3.91E+09	-9.42
3.93E+09	-11.30
3.94E+09	-14.95
3.96E+09	-16.19
3.97E+09	-11.45
3.99E+09	-9.52
4.00E+09	-10.00

S22

Frequency	dBm
1.00E+09	-1.20
1.02E+09	-1.28
1.03E+09	-1.31
1.05E+09	-1.23
1.06E+09	-1.32
1.08E+09	-1.31
1.09E+09	-1.35
1.11E+09	-1.27
1.12E+09	-1.47
1.14E+09	-1.36
1.15E+09	-1.27
1.17E+09	-1.38
1.18E+09	-1.47
1.20E+09	-1.24
1.21E+09	-1.27
1.23E+09	-1.49
1.24E+09	-1.42
1.26E+09	-1.26
1.27E+09	-1.39
1.29E+09	-1.49
1.30E+09	-1.41
1.32E+09	-1.41
1.33E+09	-1.39
1.35E+09	-1.50
1.36E+09	-1.42
1.38E+09	-1.48
1.39E+09	-1.50
1.41E+09	-1.46
1.42E+09	-1.43
1.44E+09	-1.47
1.45E+09	-1.50
1.47E+09	-1.62
1.48E+09	-1.54
1.50E+09	-1.60
1.51E+09	-1.52
1.53E+09	-1.45
1.54E+09	-1.48
1.56E+09	-1.65
1.57E+09	-1.57
1.59E+09	-1.47
1.60E+09	-1.60
1.62E+09	-1.66
1.63E+09	-1.48
1.65E+09	-1.49
1.66E+09	-1.75
1.68E+09	-1.57
1.69E+09	-1.44
1.71E+09	-1.60
1.72E+09	-1.76
1.74E+09	-1.61
1.75E+09	-1.46
1.77E+09	-1.68
1.78E+09	-1.69
1.80E+09	-1.51
1.81E+09	-1.61
1.83E+09	-1.71
1.84E+09	-1.54
1.86E+09	-1.47
1.87E+09	-1.83
1.89E+09	-1.69
1.90E+09	-1.58
1.92E+09	-1.88
1.93E+09	-1.92
1.95E+09	-1.73
1.96E+09	-2.00
1.98E+09	-2.23
1.99E+09	-2.04
2.01E+09	-2.18
2.02E+09	-2.56
2.04E+09	-2.35
2.05E+09	-2.27
2.07E+09	-2.77
2.08E+09	-2.77
2.10E+09	-2.54
2.11E+09	-2.94
2.13E+09	-3.28
2.14E+09	-2.95
2.16E+09	-3.01
2.17E+09	-3.76
2.19E+09	-3.71
2.20E+09	-3.52
2.22E+09	-4.14
2.23E+09	-4.87
2.25E+09	-4.69
2.26E+09	-4.95
2.28E+09	-6.36
2.29E+09	-6.96
2.31E+09	-6.67
2.32E+09	-7.88
2.34E+09	-9.60
2.35E+09	-9.58
2.37E+09	-9.46
2.38E+09	-11.65
2.40E+09	-12.34
2.41E+09	-9.87
2.43E+09	-9.04
2.44E+09	-10.48
2.46E+09	-10.13
2.47E+09	-8.02
2.49E+09	-7.73
2.50E+09	-9.19
2.52E+09	-8.38
2.53E+09	-7.02
2.55E+09	-7.52
2.56E+09	-8.58
2.58E+09	-7.28
2.59E+09	-6.93
2.61E+09	-8.67
2.62E+09	-9.08
2.64E+09	-7.63
2.65E+09	-8.49
2.67E+09	-10.77
2.68E+09	-9.41
2.70E+09	-8.90
2.71E+09	-11.39
2.73E+09	-14.10
2.74E+09	-11.72
2.76E+09	-11.36
2.77E+09	-12.64
2.79E+09	-17.85
2.80E+09	-12.22
2.82E+09	-10.31
2.83E+09	-11.00
2.85E+09	-11.65
2.86E+09	-8.62
2.88E+09	-7.13
2.89E+09	-7.54
2.91E+09	-6.58
2.92E+09	-5.02
2.94E+09	-5.40
2.95E+09	-6.74
2.97E+09	-7.84
2.98E+09	-9.78
3.00E+09	-9.42
3.01E+09	-8.31
3.03E+09	-5.48
3.04E+09	-4.49
3.06E+09	-5.12
3.07E+09	-4.74
3.09E+09	-3.48
3.10E+09	-3.36
3.12E+09	-3.92
3.13E+09	-3.49
3.15E+09	-2.90
3.16E+09	-3.25
3.18E+09	-3.56
3.19E+09	-2.91
3.21E+09	-2.71
3.22E+09	-3.42
3.24E+09	-3.24
3.25E+09	-2.64
3.27E+09	-3.08
3.28E+09	-4.09
3.30E+09	-3.39
3.31E+09	-2.77
3.33E+09	-3.37
3.34E+09	-3.63
3.36E+09	-2.56
3.37E+09	-2.38
3.39E+09	-3.17
3.40E+09	-2.99
3.42E+09	-2.26
3.43E+09	-2.56
3.45E+09	-3.27
3.46E+09	-2.61
3.48E+09	-2.35
3.49E+09	-2.97
3.51E+09	-3.12
3.52E+09	-2.41
3.54E+09	-2.57
3.55E+09	-3.47
3.57E+09	-3.08
3.58E+09	-2.65
3.60E+09	-3.61
3.61E+09	-5.46
3.63E+09	-4.97
3.64E+09	-4.42
3.66E+09	-5.04
3.67E+09	-5.51
3.69E+09	-4.01
3.70E+09	-3.50
3.72E+09	-4.33
3.73E+09	-4.58
3.75E+09	-3.56
3.76E+09	-3.52
3.78E+09	-4.28
3.79E+09	-4.14
3.81E+09	-3.66
3.82E+09	-4.34
3.84E+09	-5.05
3.85E+09	-4.36
3.87E+09	-4.31
3.88E+09	-5.27
3.90E+09	-5.47
3.91E+09	-4.81
3.93E+09	-5.33
3.94E+09	-6.69
3.96E+09	-5.89
3.97E+09	-4.82
3.99E+09	-5.52
4.00E+09	-6.20

S21
0°

Frequency	dBm
1.00E+09	-47.56
1.02E+09	-50.51
1.03E+09	-53.58
1.05E+09	-54.85
1.06E+09	-54.46
1.08E+09	-52.51
1.09E+09	-47.78
1.11E+09	-49.75
1.12E+09	-48.24
1.14E+09	-52.37
1.15E+09	-45.86
1.17E+09	-54.69
1.18E+09	-52.68
1.20E+09	-48.65
1.21E+09	-51.84
1.23E+09	-46.68
1.24E+09	-64.14
1.26E+09	-54.82
1.27E+09	-47.17
1.29E+09	-52.53
1.30E+09	-47.14
1.32E+09	-58.07
1.33E+09	-50.63
1.35E+09	-52.25
1.36E+09	-56.34
1.38E+09	-43.93
1.39E+09	-46.05
1.41E+09	-48.82
1.42E+09	-51.50
1.44E+09	-51.02
1.45E+09	-51.02
1.47E+09	-61.13
1.48E+09	-48.51
1.50E+09	-68.12
1.51E+09	-53.89
1.53E+09	-52.31
1.54E+09	-51.19
1.56E+09	-55.03
1.57E+09	-50.96
1.59E+09	-59.02
1.60E+09	-54.95
1.62E+09	-66.00
1.63E+09	-48.28
1.65E+09	-58.70
1.66E+09	-74.83
1.68E+09	-60.35
1.69E+09	-56.16
1.71E+09	-55.04
1.72E+09	-54.15
1.74E+09	-50.53
1.75E+09	-55.68
1.77E+09	-47.16
1.78E+09	-52.49
1.80E+09	-52.95
1.81E+09	-49.04
1.83E+09	-49.87
1.84E+09	-46.92
1.86E+09	-56.77
1.87E+09	-60.88
1.89E+09	-51.48
1.90E+09	-47.96
1.92E+09	-49.31
1.93E+09	-56.59
1.95E+09	-62.97
1.96E+09	-59.13
1.98E+09	-53.69
1.99E+09	-58.56
2.01E+09	-56.72
2.02E+09	-48.05
2.04E+09	-56.34
2.05E+09	-56.01
2.07E+09	-51.88
2.08E+09	-50.31
2.10E+09	-52.43
2.11E+09	-51.21
2.13E+09	-54.20
2.14E+09	-48.04
2.16E+09	-48.32
2.17E+09	-47.69
2.19E+09	-48.63
2.20E+09	-47.94
2.22E+09	-48.20
2.23E+09	-43.47
2.25E+09	-42.17
2.26E+09	-44.65
2.28E+09	-41.96
2.29E+09	-42.45
2.31E+09	-40.46
2.32E+09	-38.45
2.34E+09	-39.70
2.35E+09	-36.07
2.37E+09	-36.44
2.38E+09	-36.56
2.40E+09	-38.84
2.41E+09	-39.59
2.43E+09	-39.62
2.44E+09	-40.21
2.46E+09	-38.50
2.47E+09	-37.35
2.49E+09	-38.88
2.50E+09	-38.54
2.52E+09	-38.26
2.53E+09	-38.69
2.55E+09	-37.38
2.56E+09	-40.29
2.58E+09	-43.01
2.59E+09	-42.55
2.61E+09	-41.87
2.62E+09	-39.33
2.64E+09	-38.98
2.65E+09	-38.08
2.67E+09	-37.22
2.68E+09	-37.06
2.70E+09	-36.57
2.71E+09	-38.51
2.73E+09	-38.98
2.74E+09	-38.60
2.76E+09	-40.58
2.77E+09	-42.74
2.79E+09	-37.12
2.80E+09	-38.41
2.82E+09	-36.01
2.83E+09	-37.14
2.85E+09	-37.07
2.86E+09	-39.02
2.88E+09	-38.71
2.89E+09	-41.01
2.91E+09	-41.61
2.92E+09	-44.50
2.94E+09	-42.74
2.95E+09	-48.08
2.97E+09	-46.73
2.98E+09	-48.52
3.00E+09	-44.25
3.01E+09	-41.08
3.03E+09	-43.13
3.04E+09	-41.41
3.06E+09	-42.33
3.07E+09	-43.31
3.09E+09	-45.28
3.10E+09	-46.84
3.12E+09	-49.93
3.13E+09	-48.55
3.15E+09	-53.35
3.16E+09	-52.69
3.18E+09	-62.45
3.19E+09	-56.65
3.21E+09	-56.58
3.22E+09	-60.24
3.24E+09	-58.48
3.25E+09	-62.05
3.27E+09	-63.13
3.28E+09	-59.46
3.30E+09	-55.00
3.31E+09	-70.09
3.33E+09	-67.26
3.34E+09	-55.06
3.36E+09	-62.03
3.37E+09	-54.33
3.39E+09	-65.25
3.40E+09	-73.58
3.42E+09	-63.82
3.43E+09	-68.52
3.45E+09	-67.61
3.46E+09	-59.56
3.48E+09	-68.54
3.49E+09	-54.90
3.51E+09	-60.79
3.52E+09	-58.63
3.54E+09	-61.09
3.55E+09	-60.96
3.57E+09	-61.06
3.58E+09	-72.40
3.60E+09	-53.40
3.61E+09	-65.61
3.63E+09	-60.66
3.64E+09	-57.70
3.66E+09	-56.54
3.67E+09	-59.16
3.69E+09	-67.12
3.70E+09	-60.61
3.72E+09	-62.32
3.73E+09	-68.45
3.75E+09	-57.72
3.76E+09	-59.27
3.78E+09	-57.46
3.79E+09	-63.66
3.81E+09	-78.47
3.82E+09	-56.34
3.84E+09	-63.00
3.85E+09	-60.31
3.87E+09	-57.36
3.88E+09	-54.00
3.90E+09	-50.03
3.91E+09	-51.25
3.93E+09	-53.60
3.94E+09	-49.36
3.96E+09	-57.42
3.97E+09	-50.95
3.99E+09	-48.54
4.00E+09	-50.15

S21
45° CCW

Frequency	dBm
1.00E+09	-48.84
1.02E+09	-50.34
1.03E+09	-50.19
1.05E+09	-47.61
1.06E+09	-51.27
1.08E+09	-52.72
1.09E+09	-61.55
1.11E+09	-52.35
1.12E+09	-48.29
1.14E+09	-48.70
1.15E+09	-55.77
1.17E+09	-48.32
1.18E+09	-47.28
1.20E+09	-48.05
1.21E+09	-49.71
1.23E+09	-44.86
1.24E+09	-57.63
1.26E+09	-45.89
1.27E+09	-55.64
1.29E+09	-47.55
1.30E+09	-50.95
1.32E+09	-48.31
1.33E+09	-55.00
1.35E+09	-52.15
1.36E+09	-47.35
1.38E+09	-50.80
1.39E+09	-53.95
1.41E+09	-47.96
1.42E+09	-55.38
1.44E+09	-60.85
1.45E+09	-47.12
1.47E+09	-51.21
1.48E+09	-51.04
1.50E+09	-53.07
1.51E+09	-58.61
1.53E+09	-57.58
1.54E+09	-53.94
1.56E+09	-55.68
1.57E+09	-49.41
1.59E+09	-48.06
1.60E+09	-50.46
1.62E+09	-52.94
1.63E+09	-65.87
1.65E+09	-52.54
1.66E+09	-53.51
1.68E+09	-61.01
1.69E+09	-50.35
1.71E+09	-54.37
1.72E+09	-44.22
1.74E+09	-57.25
1.75E+09	-69.67
1.77E+09	-49.50
1.78E+09	-51.66
1.80E+09	-52.77
1.81E+09	-60.01
1.83E+09	-52.63
1.84E+09	-50.95
1.86E+09	-54.45
1.87E+09	-65.35
1.89E+09	-52.21
1.90E+09	-54.52
1.92E+09	-55.25
1.93E+09	-47.37
1.95E+09	-52.64
1.96E+09	-50.33
1.98E+09	-49.80
1.99E+09	-50.18
2.01E+09	-51.33
2.02E+09	-54.56
2.04E+09	-63.68
2.05E+09	-57.70
2.07E+09	-55.86
2.08E+09	-59.84
2.10E+09	-50.02
2.11E+09	-51.16
2.13E+09	-62.86
2.14E+09	-57.94
2.16E+09	-49.14
2.17E+09	-46.63
2.19E+09	-47.68
2.20E+09	-42.31
2.22E+09	-44.68
2.23E+09	-48.58
2.25E+09	-60.78
2.26E+09	-44.35
2.28E+09	-45.34
2.29E+09	-44.00
2.31E+09	-46.85
2.32E+09	-44.53
2.34E+09	-42.94
2.35E+09	-40.41
2.37E+09	-39.89
2.38E+09	-38.72
2.40E+09	-42.70
2.41E+09	-45.01
2.43E+09	-42.57
2.44E+09	-42.65
2.46E+09	-49.11
2.47E+09	-44.44
2.49E+09	-44.81
2.50E+09	-43.04
2.52E+09	-42.39
2.53E+09	-44.08
2.55E+09	-42.86
2.56E+09	-44.08
2.58E+09	-44.82
2.59E+09	-44.90
2.61E+09	-48.36
2.62E+09	-44.98
2.64E+09	-44.83
2.65E+09	-41.38
2.67E+09	-41.40
2.68E+09	-41.48
2.70E+09	-40.92
2.71E+09	-38.60
2.73E+09	-46.68
2.74E+09	-42.03
2.76E+09	-41.72
2.77E+09	-46.44
2.79E+09	-43.59
2.80E+09	-42.63
2.82E+09	-41.07
2.83E+09	-41.03
2.85E+09	-42.09
2.86E+09	-42.98
2.88E+09	-43.98
2.89E+09	-43.39
2.91E+09	-44.97
2.92E+09	-48.59
2.94E+09	-52.55
2.95E+09	-48.81
2.97E+09	-47.33
2.98E+09	-55.09
3.00E+09	-50.48
3.01E+09	-48.34
3.03E+09	-48.63
3.04E+09	-49.48
3.06E+09	-47.64
3.07E+09	-47.52
3.09E+09	-59.56
3.10E+09	-54.40
3.12E+09	-54.74
3.13E+09	-53.75
3.15E+09	-65.32
3.16E+09	-55.39
3.18E+09	-56.47
3.19E+09	-54.09
3.21E+09	-55.97
3.22E+09	-58.16
3.24E+09	-62.85
3.25E+09	-63.96
3.27E+09	-59.26
3.28E+09	-60.27
3.30E+09	-61.67
3.31E+09	-60.93
3.33E+09	-56.53
3.34E+09	-66.60
3.36E+09	-73.63
3.37E+09	-62.87
3.39E+09	-66.52
3.40E+09	-67.46
3.42E+09	-53.51
3.43E+09	-59.70
3.45E+09	-57.92
3.46E+09	-57.94
3.48E+09	-66.38
3.49E+09	-67.65
3.51E+09	-65.46
3.52E+09	-60.25
3.54E+09	-58.07
3.55E+09	-57.35
3.57E+09	-70.69
3.58E+09	-59.68
3.60E+09	-60.48
3.61E+09	-60.58
3.63E+09	-70.06
3.64E+09	-61.31
3.66E+09	-64.36
3.67E+09	-64.85
3.69E+09	-71.24
3.70E+09	-65.68
3.72E+09	-58.10
3.73E+09	-67.31
3.75E+09	-61.70
3.76E+09	-58.69
3.78E+09	-66.29
3.79E+09	-59.43
3.81E+09	-72.46
3.82E+09	-65.72
3.84E+09	-55.03
3.85E+09	-75.34
3.87E+09	-71.59
3.88E+09	-59.76
3.90E+09	-59.85
3.91E+09	-59.19
3.93E+09	-60.99
3.94E+09	-57.53
3.96E+09	-56.40
3.97E+09	-55.45
3.99E+09	-59.60
4.00E+09	-56.75

S21
90° CCW

Frequency	dBm
1.00E+09	-56.92
1.02E+09	-59.12
1.03E+09	-49.04
1.05E+09	-51.26
1.06E+09	-50.85
1.08E+09	-48.79
1.09E+09	-52.37
1.11E+09	-59.06
1.12E+09	-50.47
1.14E+09	-45.27
1.15E+09	-45.71
1.17E+09	-52.03
1.18E+09	-56.68
1.20E+09	-53.36
1.21E+09	-45.63
1.23E+09	-43.26
1.24E+09	-46.12
1.26E+09	-47.67
1.27E+09	-54.72
1.29E+09	-46.95
1.30E+09	-47.78
1.32E+09	-54.55
1.33E+09	-51.29
1.35E+09	-50.78
1.36E+09	-42.75
1.38E+09	-43.14
1.39E+09	-52.22
1.41E+09	-59.26
1.42E+09	-53.49
1.44E+09	-54.52
1.45E+09	-53.23
1.47E+09	-58.19
1.48E+09	-48.56
1.50E+09	-54.25
1.51E+09	-59.08
1.53E+09	-49.89
1.54E+09	-53.28
1.56E+09	-55.67
1.57E+09	-56.84
1.59E+09	-51.53
1.60E+09	-49.47
1.62E+09	-49.00
1.63E+09	-53.19
1.65E+09	-50.67
1.66E+09	-54.91
1.68E+09	-53.17
1.69E+09	-46.29
1.71E+09	-48.83
1.72E+09	-60.86
1.74E+09	-51.02
1.75E+09	-48.54
1.77E+09	-49.21
1.78E+09	-75.43
1.80E+09	-55.36
1.81E+09	-53.15
1.83E+09	-59.59
1.84E+09	-70.50
1.86E+09	-52.93
1.87E+09	-51.65
1.89E+09	-51.64
1.90E+09	-52.61
1.92E+09	-49.90
1.93E+09	-59.31
1.95E+09	-51.25
1.96E+09	-63.20
1.98E+09	-57.48
1.99E+09	-52.10
2.01E+09	-59.33
2.02E+09	-61.58
2.04E+09	-53.03
2.05E+09	-71.73
2.07E+09	-51.43
2.08E+09	-62.95
2.10E+09	-52.02
2.11E+09	-58.68
2.13E+09	-56.74
2.14E+09	-59.11
2.16E+09	-48.08
2.17E+09	-61.62
2.19E+09	-58.29
2.20E+09	-56.50
2.22E+09	-53.92
2.23E+09	-54.23
2.25E+09	-52.84
2.26E+09	-53.33
2.28E+09	-59.04
2.29E+09	-48.06
2.31E+09	-50.92
2.32E+09	-58.92
2.34E+09	-46.20
2.35E+09	-51.58
2.37E+09	-47.88
2.38E+09	-58.40
2.40E+09	-51.24
2.41E+09	-50.87
2.43E+09	-49.03
2.44E+09	-49.88
2.46E+09	-50.18
2.47E+09	-47.39
2.49E+09	-62.55
2.50E+09	-50.33
2.52E+09	-46.49
2.53E+09	-52.51
2.55E+09	-45.53
2.56E+09	-53.73
2.58E+09	-54.24
2.59E+09	-50.58
2.61E+09	-52.49
2.62E+09	-63.02
2.64E+09	-52.57
2.65E+09	-49.37
2.67E+09	-62.42
2.68E+09	-56.20
2.70E+09	-54.91
2.71E+09	-66.76
2.73E+09	-56.93
2.74E+09	-54.88
2.76E+09	-51.45
2.77E+09	-46.92
2.79E+09	-53.71
2.80E+09	-56.51
2.82E+09	-55.72
2.83E+09	-48.94
2.85E+09	-55.73
2.86E+09	-50.85
2.88E+09	-64.89
2.89E+09	-53.68
2.91E+09	-54.47
2.92E+09	-49.91
2.94E+09	-56.08
2.95E+09	-57.39
2.97E+09	-55.77
2.98E+09	-58.46
3.00E+09	-56.29
3.01E+09	-58.86
3.03E+09	-56.26
3.04E+09	-54.33
3.06E+09	-73.64
3.07E+09	-59.05
3.09E+09	-56.16
3.10E+09	-61.22
3.12E+09	-66.48
3.13E+09	-61.63
3.15E+09	-58.90
3.16E+09	-60.67
3.18E+09	-55.74
3.19E+09	-60.48
3.21E+09	-59.80
3.22E+09	-75.79
3.24E+09	-60.23
3.25E+09	-55.40
3.27E+09	-58.55
3.28E+09	-60.03
3.30E+09	-69.31
3.31E+09	-61.22
3.33E+09	-58.22
3.34E+09	-60.05
3.36E+09	-55.47
3.37E+09	-62.31
3.39E+09	-60.50
3.40E+09	-74.12
3.42E+09	-74.05
3.43E+09	-65.01
3.45E+09	-58.41
3.46E+09	-57.13
3.48E+09	-69.58
3.49E+09	-61.98
3.51E+09	-62.82
3.52E+09	-63.27
3.54E+09	-63.31
3.55E+09	-53.07
3.57E+09	-69.07
3.58E+09	-68.49
3.60E+09	-56.82
3.61E+09	-62.01
3.63E+09	-53.53
3.64E+09	-59.22
3.66E+09	-61.40
3.67E+09	-66.03
3.69E+09	-59.43
3.70E+09	-59.95
3.72E+09	-59.87
3.73E+09	-60.25
3.75E+09	-62.94
3.76E+09	-59.92
3.78E+09	-52.02
3.79E+09	-62.54
3.81E+09	-57.49
3.82E+09	-68.38
3.84E+09	-59.27
3.85E+09	-58.73
3.87E+09	-57.25
3.88E+09	-59.69
3.90E+09	-57.81
3.91E+09	-55.39
3.93E+09	-58.73
3.94E+09	-59.44
3.96E+09	-49.22
3.97E+09	-70.28
3.99E+09	-69.39
4.00E+09	-59.61

S21
135° CCW

Frequency	dBm
1.00E+09	-56.45
1.02E+09	-57.23
1.03E+09	-47.89
1.05E+09	-52.85
1.06E+09	-51.18
1.08E+09	-51.74
1.09E+09	-60.49
1.11E+09	-45.60
1.12E+09	-57.34
1.14E+09	-48.76
1.15E+09	-54.98
1.17E+09	-43.00
1.18E+09	-54.83
1.20E+09	-49.17
1.21E+09	-52.77
1.23E+09	-45.12
1.24E+09	-64.17
1.26E+09	-52.43
1.27E+09	-54.70
1.29E+09	-53.68
1.30E+09	-58.79
1.32E+09	-49.30
1.33E+09	-67.52
1.35E+09	-44.18
1.36E+09	-48.24
1.38E+09	-52.61
1.39E+09	-48.81
1.41E+09	-56.69
1.42E+09	-54.27
1.44E+09	-45.46
1.45E+09	-52.77
1.47E+09	-50.08
1.48E+09	-52.53
1.50E+09	-53.30
1.51E+09	-48.59
1.53E+09	-49.98
1.54E+09	-51.80
1.56E+09	-52.38
1.57E+09	-47.98
1.59E+09	-53.14
1.60E+09	-56.99
1.62E+09	-52.10
1.63E+09	-54.10
1.65E+09	-56.49
1.66E+09	-53.10
1.68E+09	-52.49
1.69E+09	-52.25
1.71E+09	-68.76
1.72E+09	-59.03
1.74E+09	-56.80
1.75E+09	-51.03
1.77E+09	-50.74
1.78E+09	-51.30
1.80E+09	-53.64
1.81E+09	-66.62
1.83E+09	-56.62
1.84E+09	-57.20
1.86E+09	-58.31
1.87E+09	-53.24
1.89E+09	-52.68
1.90E+09	-61.56
1.92E+09	-52.01
1.93E+09	-50.07
1.95E+09	-76.92
1.96E+09	-61.70
1.98E+09	-61.80
1.99E+09	-53.36
2.01E+09	-64.66
2.02E+09	-56.07
2.04E+09	-58.64
2.05E+09	-61.23
2.07E+09	-50.08
2.08E+09	-55.75
2.10E+09	-51.86
2.11E+09	-60.16
2.13E+09	-55.33
2.14E+09	-53.69
2.16E+09	-55.08
2.17E+09	-69.86
2.19E+09	-54.86
2.20E+09	-56.47
2.22E+09	-52.33
2.23E+09	-51.97
2.25E+09	-51.14
2.26E+09	-53.39
2.28E+09	-49.36
2.29E+09	-53.04
2.31E+09	-52.22
2.32E+09	-63.29
2.34E+09	-52.55
2.35E+09	-53.21
2.37E+09	-51.89
2.38E+09	-46.02
2.40E+09	-64.83
2.41E+09	-56.35
2.43E+09	-65.88
2.44E+09	-55.63
2.46E+09	-59.70
2.47E+09	-60.69
2.49E+09	-55.43
2.50E+09	-52.88
2.52E+09	-66.68
2.53E+09	-58.76
2.55E+09	-56.08
2.56E+09	-48.46
2.58E+09	-54.74
2.59E+09	-57.59
2.61E+09	-64.59
2.62E+09	-51.57
2.64E+09	-50.56
2.65E+09	-50.45
2.67E+09	-57.55
2.68E+09	-54.56
2.70E+09	-51.55
2.71E+09	-70.07
2.73E+09	-56.13
2.74E+09	-57.39
2.76E+09	-68.05
2.77E+09	-56.16
2.79E+09	-63.24
2.80E+09	-63.73
2.82E+09	-67.08
2.83E+09	-55.37
2.85E+09	-65.97
2.86E+09	-49.92
2.88E+09	-54.79
2.89E+09	-54.40
2.91E+09	-56.40
2.92E+09	-76.17
2.94E+09	-61.97
2.95E+09	-63.72
2.97E+09	-57.54
2.98E+09	-56.58
3.00E+09	-61.35
3.01E+09	-53.29
3.03E+09	-53.42
3.04E+09	-52.01
3.06E+09	-59.17
3.07E+09	-53.55
3.09E+09	-65.36
3.10E+09	-63.55
3.12E+09	-53.79
3.13E+09	-52.69
3.15E+09	-57.79
3.16E+09	-60.47
3.18E+09	-61.47
3.19E+09	-63.26
3.21E+09	-60.63
3.22E+09	-60.15
3.24E+09	-55.23
3.25E+09	-57.71
3.27E+09	-60.13
3.28E+09	-67.19
3.30E+09	-58.59
3.31E+09	-67.09
3.33E+09	-62.44
3.34E+09	-56.65
3.36E+09	-62.62
3.37E+09	-71.70
3.39E+09	-56.25
3.40E+09	-59.70
3.42E+09	-60.57
3.43E+09	-64.52
3.45E+09	-66.12
3.46E+09	-59.62
3.48E+09	-66.42
3.49E+09	-70.68
3.51E+09	-63.17
3.52E+09	-62.56
3.54E+09	-59.89
3.55E+09	-62.61
3.57E+09	-69.09
3.58E+09	-58.89
3.60E+09	-60.99
3.61E+09	-76.74
3.63E+09	-53.88
3.64E+09	-58.40
3.66E+09	-54.53
3.67E+09	-60.85
3.69E+09	-60.86
3.70E+09	-59.55
3.72E+09	-58.85
3.73E+09	-64.50
3.75E+09	-77.54
3.76E+09	-61.07
3.78E+09	-57.84
3.79E+09	-66.45
3.81E+09	-64.46
3.82E+09	-59.27
3.84E+09	-59.45
3.85E+09	-68.79
3.87E+09	-55.63
3.88E+09	-60.51
3.90E+09	-58.06
3.91E+09	-63.97
3.93E+09	-60.99
3.94E+09	-59.69
3.96E+09	-66.70
3.97E+09	-57.37
3.99E+09	-66.94
4.00E+09	-61.92

S21
180° CCW

Frequency	dBm
1.00E+09	-48.32
1.02E+09	-51.55
1.03E+09	-61.33
1.05E+09	-52.59
1.06E+09	-48.28
1.08E+09	-44.84
1.09E+09	-65.43
1.11E+09	-67.06
1.12E+09	-53.07
1.14E+09	-43.86
1.15E+09	-44.34
1.17E+09	-50.43
1.18E+09	-44.27
1.20E+09	-46.17
1.21E+09	-50.41
1.23E+09	-67.40
1.24E+09	-53.23
1.26E+09	-52.76
1.27E+09	-46.75
1.29E+09	-53.67
1.30E+09	-47.32
1.32E+09	-48.97
1.33E+09	-55.23
1.35E+09	-61.41
1.36E+09	-59.59
1.38E+09	-47.56
1.39E+09	-55.07
1.41E+09	-64.15
1.42E+09	-49.08
1.44E+09	-48.32
1.45E+09	-54.11
1.47E+09	-51.32
1.48E+09	-49.08
1.50E+09	-57.00
1.51E+09	-47.00
1.53E+09	-55.89
1.54E+09	-51.14
1.56E+09	-60.36
1.57E+09	-51.64
1.59E+09	-54.63
1.60E+09	-49.35
1.62E+09	-49.51
1.63E+09	-63.76
1.65E+09	-52.41
1.66E+09	-61.81
1.68E+09	-55.57
1.69E+09	-47.63
1.71E+09	-52.83
1.72E+09	-49.69
1.74E+09	-49.58
1.75E+09	-50.00
1.77E+09	-53.40
1.78E+09	-61.52
1.80E+09	-61.51
1.81E+09	-56.62
1.83E+09	-53.87
1.84E+09	-47.51
1.86E+09	-52.99
1.87E+09	-60.97
1.89E+09	-62.39
1.90E+09	-53.92
1.92E+09	-53.72
1.93E+09	-50.66
1.95E+09	-60.37
1.96E+09	-61.56
1.98E+09	-56.91
1.99E+09	-51.21
2.01E+09	-61.65
2.02E+09	-54.81
2.04E+09	-54.66
2.05E+09	-62.12
2.07E+09	-52.67
2.08E+09	-65.64
2.10E+09	-50.39
2.11E+09	-48.50
2.13E+09	-53.75
2.14E+09	-66.33
2.16E+09	-58.06
2.17E+09	-60.55
2.19E+09	-56.14
2.20E+09	-52.39
2.22E+09	-54.34
2.23E+09	-55.51
2.25E+09	-49.30
2.26E+09	-53.19
2.28E+09	-49.99
2.29E+09	-51.27
2.31E+09	-63.21
2.32E+09	-61.34
2.34E+09	-51.86
2.35E+09	-53.26
2.37E+09	-48.84
2.38E+09	-48.16
2.40E+09	-49.19
2.41E+09	-50.63
2.43E+09	-57.84
2.44E+09	-47.20
2.46E+09	-55.45
2.47E+09	-49.08
2.49E+09	-48.89
2.50E+09	-72.71
2.52E+09	-51.68
2.53E+09	-47.17
2.55E+09	-56.49
2.56E+09	-53.00
2.58E+09	-50.62
2.59E+09	-49.62
2.61E+09	-50.78
2.62E+09	-51.97
2.64E+09	-46.55
2.65E+09	-57.86
2.67E+09	-45.91
2.68E+09	-48.39
2.70E+09	-45.85
2.71E+09	-63.05
2.73E+09	-55.16
2.74E+09	-53.95
2.76E+09	-46.67
2.77E+09	-61.15
2.79E+09	-49.42
2.80E+09	-53.13
2.82E+09	-52.23
2.83E+09	-52.90
2.85E+09	-46.67
2.86E+09	-53.59
2.88E+09	-50.92
2.89E+09	-50.82
2.91E+09	-69.85
2.92E+09	-53.18
2.94E+09	-56.65
2.95E+09	-58.92
2.97E+09	-68.07
2.98E+09	-46.14
3.00E+09	-55.43
3.01E+09	-58.40
3.03E+09	-48.21
3.04E+09	-75.76
3.06E+09	-57.52
3.07E+09	-65.32
3.09E+09	-52.95
3.10E+09	-60.02
3.12E+09	-64.47
3.13E+09	-68.62
3.15E+09	-66.96
3.16E+09	-49.69
3.18E+09	-53.97
3.19E+09	-63.96
3.21E+09	-68.02
3.22E+09	-54.81
3.24E+09	-60.61
3.25E+09	-61.66
3.27E+09	-54.82
3.28E+09	-59.86
3.30E+09	-57.91
3.31E+09	-64.95
3.33E+09	-54.42
3.34E+09	-64.59
3.36E+09	-60.01
3.37E+09	-57.53
3.39E+09	-57.52
3.40E+09	-65.98
3.42E+09	-200.00
3.43E+09	-58.50
3.45E+09	-54.31
3.46E+09	-67.76
3.48E+09	-63.42
3.49E+09	-60.74
3.51E+09	-67.67
3.52E+09	-62.32
3.54E+09	-63.12
3.55E+09	-53.47
3.57E+09	-67.20
3.58E+09	-70.51
3.60E+09	-59.25
3.61E+09	-60.77
3.63E+09	-68.04
3.64E+09	-55.81
3.66E+09	-67.16
3.67E+09	-59.37
3.69E+09	-68.14
3.70E+09	-72.24
3.72E+09	-66.52
3.73E+09	-60.97
3.75E+09	-61.95
3.76E+09	-68.90
3.78E+09	-63.97
3.79E+09	-59.05
3.81E+09	-59.45
3.82E+09	-58.32
3.84E+09	-61.30
3.85E+09	-62.62
3.87E+09	-61.40
3.88E+09	-64.81
3.90E+09	-68.93
3.91E+09	-56.98
3.93E+09	-66.49
3.94E+09	-63.40
3.96E+09	-72.48
3.97E+09	-61.72
3.99E+09	-64.10
4.00E+09	-59.68

S21
225° CCW

Frequency	dBm
1.00E+09	-50.64
1.02E+09	-57.75
1.03E+09	-47.83
1.05E+09	-50.64
1.06E+09	-47.46
1.08E+09	-47.81
1.09E+09	-49.14
1.11E+09	-61.97
1.12E+09	-54.92
1.14E+09	-51.57
1.15E+09	-46.40
1.17E+09	-48.59
1.18E+09	-57.29
1.20E+09	-48.41
1.21E+09	-50.09
1.23E+09	-55.13
1.24E+09	-50.00
1.26E+09	-50.50
1.27E+09	-52.91
1.29E+09	-48.61
1.30E+09	-63.48
1.32E+09	-52.68
1.33E+09	-50.39
1.35E+09	-54.49
1.36E+09	-54.96
1.38E+09	-47.13
1.39E+09	-60.27
1.41E+09	-67.27
1.42E+09	-55.19
1.44E+09	-55.61
1.45E+09	-55.12
1.47E+09	-52.11
1.48E+09	-51.49
1.50E+09	-50.30
1.51E+09	-51.92
1.53E+09	-46.91
1.54E+09	-46.70
1.56E+09	-62.39
1.57E+09	-56.11
1.59E+09	-52.68
1.60E+09	-55.11
1.62E+09	-50.46
1.63E+09	-56.57
1.65E+09	-51.10
1.66E+09	-48.98
1.68E+09	-57.82
1.69E+09	-48.55
1.71E+09	-57.63
1.72E+09	-59.33
1.74E+09	-51.67
1.75E+09	-53.81
1.77E+09	-62.53
1.78E+09	-49.90
1.80E+09	-192.89
1.81E+09	-51.46
1.83E+09	-56.03
1.84E+09	-57.16
1.86E+09	-53.06
1.87E+09	-56.50
1.89E+09	-48.00
1.90E+09	-66.73
1.92E+09	-57.76
1.93E+09	-54.97
1.95E+09	-48.02
1.96E+09	-51.53
1.98E+09	-58.77
1.99E+09	-55.62
2.01E+09	-60.74
2.02E+09	-64.61
2.04E+09	-71.62
2.05E+09	-48.41
2.07E+09	-57.17
2.08E+09	-195.32
2.10E+09	-69.04
2.11E+09	-58.17
2.13E+09	-48.61
2.14E+09	-65.55
2.16E+09	-57.11
2.17E+09	-58.69
2.19E+09	-65.95
2.20E+09	-51.77
2.22E+09	-49.08
2.23E+09	-51.27
2.25E+09	-71.89
2.26E+09	-57.07
2.28E+09	-49.87
2.29E+09	-52.15
2.31E+09	-67.28
2.32E+09	-49.02
2.34E+09	-189.88
2.35E+09	-49.82
2.37E+09	-61.59
2.38E+09	-48.29
2.40E+09	-61.27
2.41E+09	-54.07
2.43E+09	-52.85
2.44E+09	-49.74
2.46E+09	-65.69
2.47E+09	-56.54
2.49E+09	-48.83
2.50E+09	-50.67
2.52E+09	-46.03
2.53E+09	-58.03
2.55E+09	-59.56
2.56E+09	-48.66
2.58E+09	-57.08
2.59E+09	-52.53
2.61E+09	-52.59
2.62E+09	-46.90
2.64E+09	-62.40
2.65E+09	-49.74
2.67E+09	-52.94
2.68E+09	-50.85
2.70E+09	-56.28
2.71E+09	-56.35
2.73E+09	-48.60
2.74E+09	-48.31
2.76E+09	-47.43
2.77E+09	-61.09
2.79E+09	-63.77
2.80E+09	-56.75
2.82E+09	-56.22
2.83E+09	-47.88
2.85E+09	-61.40
2.86E+09	-54.78
2.88E+09	-49.48
2.89E+09	-51.09
2.91E+09	-51.37
2.92E+09	-52.80
2.94E+09	-55.85
2.95E+09	-59.94
2.97E+09	-55.78
2.98E+09	-59.58
3.00E+09	-53.70
3.01E+09	-50.00
3.03E+09	-51.96
3.04E+09	-53.39
3.06E+09	-55.68
3.07E+09	-57.09
3.09E+09	-58.39
3.10E+09	-59.63
3.12E+09	-58.92
3.13E+09	-54.39
3.15E+09	-70.39
3.16E+09	-58.83
3.18E+09	-58.93
3.19E+09	-58.80
3.21E+09	-58.15
3.22E+09	-68.76
3.24E+09	-54.76
3.25E+09	-58.38
3.27E+09	-63.12
3.28E+09	-54.96
3.30E+09	-55.56
3.31E+09	-65.17
3.33E+09	-73.73
3.34E+09	-61.95
3.36E+09	-57.95
3.37E+09	-56.44
3.39E+09	-66.90
3.40E+09	-54.91
3.42E+09	-60.04
3.43E+09	-59.88
3.45E+09	-58.50
3.46E+09	-62.72
3.48E+09	-61.92
3.49E+09	-61.59
3.51E+09	-61.71
3.52E+09	-77.83
3.54E+09	-58.63
3.55E+09	-60.99
3.57E+09	-56.77
3.58E+09	-63.45
3.60E+09	-77.48
3.61E+09	-57.58
3.63E+09	-61.63
3.64E+09	-62.73
3.66E+09	-75.36
3.67E+09	-63.36
3.69E+09	-58.90
3.70E+09	-62.66
3.72E+09	-68.65
3.73E+09	-67.90
3.75E+09	-56.08
3.76E+09	-57.18
3.78E+09	-50.53
3.79E+09	-61.03
3.81E+09	-60.78
3.82E+09	-71.33
3.84E+09	-75.24
3.85E+09	-56.69
3.87E+09	-57.16
3.88E+09	-58.52
3.90E+09	-63.59
3.91E+09	-68.35
3.93E+09	-61.73
3.94E+09	-61.66
3.96E+09	-64.37
3.97E+09	-61.92
3.99E+09	-66.70
4.00E+09	-61.44

S21
270° CCW

Frequency	dBm
1.00E+09	-62.80
1.02E+09	-57.09
1.03E+09	-62.62
1.05E+09	-42.88
1.06E+09	-49.68
1.08E+09	-45.17
1.09E+09	-62.51
1.11E+09	-48.69
1.12E+09	-50.39
1.14E+09	-48.22
1.15E+09	-48.54
1.17E+09	-46.18
1.18E+09	-60.17
1.20E+09	-45.31
1.21E+09	-55.55
1.23E+09	-52.90
1.24E+09	-63.73
1.26E+09	-45.66
1.27E+09	-65.86
1.29E+09	-49.81
1.30E+09	-51.33
1.32E+09	-57.45
1.33E+09	-58.48
1.35E+09	-53.40
1.36E+09	-52.65
1.38E+09	-57.61
1.39E+09	-59.22
1.41E+09	-50.33
1.42E+09	-52.96
1.44E+09	-46.16
1.45E+09	-68.14
1.47E+09	-56.95
1.48E+09	-53.59
1.50E+09	-49.78
1.51E+09	-45.57
1.53E+09	-55.73
1.54E+09	-47.42
1.56E+09	-49.13
1.57E+09	-49.97
1.59E+09	-66.02
1.60E+09	-57.29
1.62E+09	-56.90
1.63E+09	-54.27
1.65E+09	-47.99
1.66E+09	-49.28
1.68E+09	-54.94
1.69E+09	-55.50
1.71E+09	-53.84
1.72E+09	-51.71
1.74E+09	-60.77
1.75E+09	-56.67
1.77E+09	-55.53
1.78E+09	-53.39
1.80E+09	-54.44
1.81E+09	-51.76
1.83E+09	-54.41
1.84E+09	-53.27
1.86E+09	-54.44
1.87E+09	-53.82
1.89E+09	-52.11
1.90E+09	-53.27
1.92E+09	-61.23
1.93E+09	-50.14
1.95E+09	-59.69
1.96E+09	-58.16
1.98E+09	-53.90
1.99E+09	-53.65
2.01E+09	-54.76
2.02E+09	-49.05
2.04E+09	-50.19
2.05E+09	-51.41
2.07E+09	-68.82
2.08E+09	-53.55
2.10E+09	-66.94
2.11E+09	-69.16
2.13E+09	-59.39
2.14E+09	-54.92
2.16E+09	-62.68
2.17E+09	-51.28
2.19E+09	-53.38
2.20E+09	-54.82
2.22E+09	-72.02
2.23E+09	-49.06
2.25E+09	-49.28
2.26E+09	-47.94
2.28E+09	-50.67
2.29E+09	-46.02
2.31E+09	-51.29
2.32E+09	-51.51
2.34E+09	-73.46
2.35E+09	-51.88
2.37E+09	-47.14
2.38E+09	-56.95
2.40E+09	-44.95
2.41E+09	-45.12
2.43E+09	-54.04
2.44E+09	-46.47
2.46E+09	-48.85
2.47E+09	-62.45
2.49E+09	-47.84
2.50E+09	-59.73
2.52E+09	-56.55
2.53E+09	-46.68
2.55E+09	-54.70
2.56E+09	-51.72
2.58E+09	-62.89
2.59E+09	-52.63
2.61E+09	-47.07
2.62E+09	-47.65
2.64E+09	-52.86
2.65E+09	-49.97
2.67E+09	-49.71
2.68E+09	-47.87
2.70E+09	-55.79
2.71E+09	-55.16
2.73E+09	-49.28
2.74E+09	-46.33
2.76E+09	-51.95
2.77E+09	-49.01
2.79E+09	-46.45
2.80E+09	-52.34
2.82E+09	-48.07
2.83E+09	-48.42
2.85E+09	-53.99
2.86E+09	-57.30
2.88E+09	-50.74
2.89E+09	-50.24
2.91E+09	-53.58
2.92E+09	-64.00
2.94E+09	-55.44
2.95E+09	-56.67
2.97E+09	-82.00
2.98E+09	-69.12
3.00E+09	-57.30
3.01E+09	-52.91
3.03E+09	-53.19
3.04E+09	-64.57
3.06E+09	-63.35
3.07E+09	-60.36
3.09E+09	-63.23
3.10E+09	-59.44
3.12E+09	-64.25
3.13E+09	-76.55
3.15E+09	-65.92
3.16E+09	-61.74
3.18E+09	-55.02
3.19E+09	-63.44
3.21E+09	-51.98
3.22E+09	-65.63
3.24E+09	-62.69
3.25E+09	-63.00
3.27E+09	-70.12
3.28E+09	-62.33
3.30E+09	-57.17
3.31E+09	-56.37
3.33E+09	-70.73
3.34E+09	-55.42
3.36E+09	-60.48
3.37E+09	-56.80
3.39E+09	-65.81
3.40E+09	-56.56
3.42E+09	-64.92
3.43E+09	-65.04
3.45E+09	-59.78
3.46E+09	-63.35
3.48E+09	-67.41
3.49E+09	-65.08
3.51E+09	-60.77
3.52E+09	-60.63
3.54E+09	-59.84
3.55E+09	-60.45
3.57E+09	-64.64
3.58E+09	-61.95
3.60E+09	-63.49
3.61E+09	-68.62
3.63E+09	-61.07
3.64E+09	-54.75
3.66E+09	-58.78
3.67E+09	-59.28
3.69E+09	-61.57
3.70E+09	-71.95
3.72E+09	-57.31
3.73E+09	-68.43
3.75E+09	-57.36
3.76E+09	-63.24
3.78E+09	-72.56
3.79E+09	-68.04
3.81E+09	-65.40
3.82E+09	-54.91
3.84E+09	-67.12
3.85E+09	-62.63
3.87E+09	-66.45
3.88E+09	-53.05
3.90E+09	-57.68
3.91E+09	-63.05
3.93E+09	-55.09
3.94E+09	-65.39
3.96E+09	-53.58
3.97E+09	-56.72
3.99E+09	-63.30
4.00E+09	-63.54

S21
315° CCW

Frequency	dBm
1.00E+09	-45.23
1.02E+09	-56.23
1.03E+09	-49.76
1.05E+09	-50.84
1.06E+09	-46.34
1.08E+09	-54.51
1.09E+09	-61.57
1.11E+09	-59.26
1.12E+09	-56.36
1.14E+09	-44.14
1.15E+09	-54.89
1.17E+09	-54.03
1.18E+09	-50.09
1.20E+09	-56.34
1.21E+09	-54.53
1.23E+09	-50.90
1.24E+09	-50.09
1.26E+09	-50.54
1.27E+09	-49.73
1.29E+09	-48.62
1.30E+09	-58.00
1.32E+09	-50.12
1.33E+09	-45.44
1.35E+09	-49.67
1.36E+09	-46.23
1.38E+09	-50.18
1.39E+09	-50.15
1.41E+09	-51.73
1.42E+09	-52.14
1.44E+09	-54.36
1.45E+09	-49.08
1.47E+09	-55.97
1.48E+09	-57.70
1.50E+09	-50.60
1.51E+09	-51.80
1.53E+09	-48.65
1.54E+09	-55.93
1.56E+09	-49.69
1.57E+09	-52.95
1.59E+09	-56.84
1.60E+09	-53.11
1.62E+09	-55.78
1.63E+09	-47.98
1.65E+09	-51.23
1.66E+09	-53.76
1.68E+09	-54.87
1.69E+09	-57.04
1.71E+09	-48.04
1.72E+09	-59.77
1.74E+09	-54.55
1.75E+09	-47.36
1.77E+09	-59.51
1.78E+09	-52.56
1.80E+09	-61.53
1.81E+09	-47.31
1.83E+09	-62.14
1.84E+09	-56.97
1.86E+09	-50.39
1.87E+09	-53.56
1.89E+09	-47.83
1.90E+09	-50.00
1.92E+09	-60.63
1.93E+09	-53.39
1.95E+09	-52.72
1.96E+09	-48.84
1.98E+09	-65.16
1.99E+09	-51.98
2.01E+09	-59.05
2.02E+09	-61.48
2.04E+09	-77.60
2.05E+09	-51.25
2.07E+09	-68.26
2.08E+09	-55.01
2.10E+09	-57.39
2.11E+09	-47.90
2.13E+09	-51.62
2.14E+09	-69.53
2.16E+09	-49.74
2.17E+09	-46.35
2.19E+09	-45.51
2.20E+09	-47.89
2.22E+09	-45.92
2.23E+09	-48.72
2.25E+09	-45.96
2.26E+09	-50.61
2.28E+09	-47.70
2.29E+09	-45.47
2.31E+09	-52.65
2.32E+09	-43.42
2.34E+09	-41.28
2.35E+09	-41.04
2.37E+09	-39.96
2.38E+09	-39.15
2.40E+09	-40.68
2.41E+09	-40.78
2.43E+09	-43.21
2.44E+09	-43.26
2.46E+09	-44.97
2.47E+09	-47.78
2.49E+09	-49.27
2.50E+09	-38.62
2.52E+09	-42.08
2.53E+09	-41.90
2.55E+09	-40.73
2.56E+09	-46.33
2.58E+09	-45.15
2.59E+09	-45.19
2.61E+09	-47.30
2.62E+09	-43.75
2.64E+09	-42.88
2.65E+09	-42.36
2.67E+09	-42.08
2.68E+09	-46.98
2.70E+09	-41.17
2.71E+09	-41.48
2.73E+09	-46.27
2.74E+09	-40.69
2.76E+09	-41.85
2.77E+09	-46.66
2.79E+09	-41.31
2.80E+09	-43.60
2.82E+09	-41.93
2.83E+09	-41.31
2.85E+09	-41.68
2.86E+09	-43.75
2.88E+09	-44.35
2.89E+09	-43.10
2.91E+09	-44.95
2.92E+09	-51.69
2.94E+09	-52.42
2.95E+09	-52.05
2.97E+09	-49.79
2.98E+09	-51.60
3.00E+09	-49.46
3.01E+09	-47.07
3.03E+09	-48.58
3.04E+09	-50.36
3.06E+09	-48.75
3.07E+09	-49.85
3.09E+09	-50.97
3.10E+09	-48.52
3.12E+09	-55.20
3.13E+09	-54.06
3.15E+09	-52.76
3.16E+09	-53.09
3.18E+09	-53.52
3.19E+09	-54.50
3.21E+09	-65.73
3.22E+09	-76.70
3.24E+09	-57.29
3.25E+09	-63.47
3.27E+09	-58.75
3.28E+09	-61.70
3.30E+09	-63.25
3.31E+09	-64.70
3.33E+09	-59.93
3.34E+09	-63.35
3.36E+09	-57.32
3.37E+09	-57.52
3.39E+09	-200.00
3.40E+09	-55.47
3.42E+09	-60.19
3.43E+09	-54.67
3.45E+09	-63.22
3.46E+09	-58.90
3.48E+09	-71.22
3.49E+09	-61.77
3.51E+09	-65.91
3.52E+09	-56.86
3.54E+09	-61.02
3.55E+09	-65.30
3.57E+09	-65.57
3.58E+09	-63.48
3.60E+09	-65.47
3.61E+09	-66.50
3.63E+09	-70.02
3.64E+09	-71.28
3.66E+09	-60.03
3.67E+09	-58.24
3.69E+09	-54.04
3.70E+09	-61.21
3.72E+09	-66.19
3.73E+09	-62.48
3.75E+09	-65.01
3.76E+09	-65.50
3.78E+09	-62.02
3.79E+09	-53.64
3.81E+09	-59.75
3.82E+09	-62.48
3.84E+09	-68.26
3.85E+09	-55.72
3.87E+09	-64.56
3.88E+09	-55.44
3.90E+09	-55.99
3.91E+09	-61.65
3.93E+09	-53.90
3.94E+09	-61.82
3.96E+09	-56.66
3.97E+09	-58.65
3.99E+09	-55.98
4.00E+09	-55.64

S12
45° CCW

Frequency	dBm
1.00E+09	-48.42
1.02E+09	-54.13
1.03E+09	-66.97
1.05E+09	-52.70
1.06E+09	-54.73
1.08E+09	-52.50
1.09E+09	-58.78
1.11E+09	-43.95
1.12E+09	-49.08
1.14E+09	-57.71
1.15E+09	-54.55
1.17E+09	-47.92
1.18E+09	-57.36
1.20E+09	-65.35
1.21E+09	-53.58
1.23E+09	-73.23
1.24E+09	-53.77
1.26E+09	-45.52
1.27E+09	-54.96
1.29E+09	-57.79
1.30E+09	-45.87
1.32E+09	-48.18
1.33E+09	-48.84
1.35E+09	-48.68
1.36E+09	-50.40
1.38E+09	-52.47
1.39E+09	-52.10
1.41E+09	-60.51
1.42E+09	-45.83
1.44E+09	-51.19
1.45E+09	-53.65
1.47E+09	-53.27
1.48E+09	-49.02
1.50E+09	-53.84
1.51E+09	-48.30
1.53E+09	-50.53
1.54E+09	-55.01
1.56E+09	-54.61
1.57E+09	-51.40
1.59E+09	-56.29
1.60E+09	-62.17
1.62E+09	-46.62
1.63E+09	-61.98
1.65E+09	-52.15
1.66E+09	-59.95
1.68E+09	-53.79
1.69E+09	-51.31
1.71E+09	-51.66
1.72E+09	-51.52
1.74E+09	-49.93
1.75E+09	-70.00
1.77E+09	-58.26
1.78E+09	-61.35
1.80E+09	-52.95
1.81E+09	-48.95
1.83E+09	-48.76
1.84E+09	-58.21
1.86E+09	-63.42
1.87E+09	-49.10
1.89E+09	-56.00
1.90E+09	-53.35
1.92E+09	-59.03
1.93E+09	-57.74
1.95E+09	-55.11
1.96E+09	-57.04
1.98E+09	-63.77
1.99E+09	-53.58
2.01E+09	-52.76
2.02E+09	-52.42
2.04E+09	-51.69
2.05E+09	-52.16
2.07E+09	-62.02
2.08E+09	-51.10
2.10E+09	-53.16
2.11E+09	-55.44
2.13E+09	-51.02
2.14E+09	-48.72
2.16E+09	-58.35
2.17E+09	-47.71
2.19E+09	-52.32
2.20E+09	-49.53
2.22E+09	-50.27
2.23E+09	-51.91
2.25E+09	-51.83
2.26E+09	-44.77
2.28E+09	-47.81
2.29E+09	-44.30
2.31E+09	-47.57
2.32E+09	-44.40
2.34E+09	-45.93
2.35E+09	-39.57
2.37E+09	-39.86
2.38E+09	-37.57
2.40E+09	-41.34
2.41E+09	-42.89
2.43E+09	-40.81
2.44E+09	-43.06
2.46E+09	-46.74
2.47E+09	-49.02
2.49E+09	-45.90
2.50E+09	-43.96
2.52E+09	-43.02
2.53E+09	-47.87
2.55E+09	-44.07
2.56E+09	-42.02
2.58E+09	-44.50
2.59E+09	-44.75
2.61E+09	-44.33
2.62E+09	-48.35
2.64E+09	-42.83
2.65E+09	-43.04
2.67E+09	-42.93
2.68E+09	-44.14
2.70E+09	-41.59
2.71E+09	-42.95
2.73E+09	-48.99
2.74E+09	-46.42
2.76E+09	-46.94
2.77E+09	-43.98
2.79E+09	-43.64
2.80E+09	-41.14
2.82E+09	-41.78
2.83E+09	-40.38
2.85E+09	-42.37
2.86E+09	-42.58
2.88E+09	-43.59
2.89E+09	-42.78
2.91E+09	-46.29
2.92E+09	-45.17
2.94E+09	-48.44
2.95E+09	-48.01
2.97E+09	-45.64
2.98E+09	-56.39
3.00E+09	-51.19
3.01E+09	-47.56
3.03E+09	-44.48
3.04E+09	-52.13
3.06E+09	-55.73
3.07E+09	-59.58
3.09E+09	-48.14
3.10E+09	-49.98
3.12E+09	-52.61
3.13E+09	-66.30
3.15E+09	-54.82
3.16E+09	-53.27
3.18E+09	-56.57
3.19E+09	-56.97
3.21E+09	-55.08
3.22E+09	-54.24
3.24E+09	-61.60
3.25E+09	-64.52
3.27E+09	-56.30
3.28E+09	-67.93
3.30E+09	-54.13
3.31E+09	-57.60
3.33E+09	-56.84
3.34E+09	-57.68
3.36E+09	-69.37
3.37E+09	-64.25
3.39E+09	-67.16
3.40E+09	-64.22
3.42E+09	-61.86
3.43E+09	-61.44
3.45E+09	-59.17
3.46E+09	-66.52
3.48E+09	-61.39
3.49E+09	-54.22
3.51E+09	-64.72
3.52E+09	-53.73
3.54E+09	-63.27
3.55E+09	-57.43
3.57E+09	-83.49
3.58E+09	-62.33
3.60E+09	-68.68
3.61E+09	-59.58
3.63E+09	-54.14
3.64E+09	-59.05
3.66E+09	-61.83
3.67E+09	-68.73
3.69E+09	-56.10
3.70E+09	-55.90
3.72E+09	-56.56
3.73E+09	-65.30
3.75E+09	-55.54
3.76E+09	-59.89
3.78E+09	-59.12
3.79E+09	-62.86
3.81E+09	-67.35
3.82E+09	-58.18
3.84E+09	-65.35
3.85E+09	-75.79
3.87E+09	-53.51
3.88E+09	-54.17
3.90E+09	-54.37
3.91E+09	-57.10
3.93E+09	-52.89
3.94E+09	-54.08
3.96E+09	-57.49
3.97E+09	-60.78
3.99E+09	-57.02
4.00E+09	-60.50

S12
90° CCW

Frequency	dBm
1.00E+09	-58.02
1.02E+09	-43.26
1.03E+09	-49.42
1.05E+09	-62.90
1.06E+09	-49.12
1.08E+09	-46.81
1.09E+09	-50.05
1.11E+09	-52.02
1.12E+09	-67.30
1.14E+09	-54.01
1.15E+09	-51.40
1.17E+09	-52.96
1.18E+09	-67.59
1.20E+09	-63.52
1.21E+09	-43.69
1.23E+09	-44.19
1.24E+09	-51.73
1.26E+09	-66.19
1.27E+09	-50.30
1.29E+09	-45.34
1.30E+09	-51.79
1.32E+09	-48.39
1.33E+09	-52.39
1.35E+09	-55.48
1.36E+09	-53.42
1.38E+09	-55.87
1.39E+09	-50.68
1.41E+09	-60.58
1.42E+09	-66.08
1.44E+09	-60.95
1.45E+09	-54.48
1.47E+09	-58.42
1.48E+09	-57.11
1.50E+09	-55.93
1.51E+09	-50.08
1.53E+09	-55.51
1.54E+09	-52.35
1.56E+09	-58.12
1.57E+09	-54.49
1.59E+09	-53.21
1.60E+09	-55.14
1.62E+09	-54.33
1.63E+09	-45.69
1.65E+09	-50.91
1.66E+09	-55.57
1.68E+09	-47.82
1.69E+09	-50.10
1.71E+09	-59.78
1.72E+09	-52.75
1.74E+09	-47.85
1.75E+09	-58.81
1.77E+09	-51.82
1.78E+09	-49.68
1.80E+09	-63.07
1.81E+09	-50.83
1.83E+09	-60.34
1.84E+09	-54.38
1.86E+09	-55.17
1.87E+09	-53.42
1.89E+09	-51.41
1.90E+09	-60.26
1.92E+09	-52.85
1.93E+09	-49.17
1.95E+09	-52.17
1.96E+09	-61.71
1.98E+09	-50.64
1.99E+09	-60.18
2.01E+09	-60.10
2.02E+09	-50.99
2.04E+09	-52.67
2.05E+09	-49.28
2.07E+09	-49.42
2.08E+09	-60.80
2.10E+09	-52.28
2.11E+09	-65.18
2.13E+09	-65.82
2.14E+09	-52.93
2.16E+09	-54.55
2.17E+09	-53.14
2.19E+09	-50.91
2.20E+09	-50.24
2.22E+09	-56.91
2.23E+09	-46.19
2.25E+09	-47.19
2.26E+09	-48.45
2.28E+09	-51.72
2.29E+09	-54.84
2.31E+09	-47.96
2.32E+09	-50.50
2.34E+09	-49.08
2.35E+09	-53.53
2.37E+09	-53.98
2.38E+09	-51.45
2.40E+09	-49.68
2.41E+09	-49.67
2.43E+09	-53.29
2.44E+09	-51.76
2.46E+09	-48.99
2.47E+09	-57.93
2.49E+09	-50.80
2.50E+09	-57.29
2.52E+09	-46.27
2.53E+09	-64.81
2.55E+09	-47.35
2.56E+09	-47.06
2.58E+09	-51.41
2.59E+09	-51.56
2.61E+09	-53.15
2.62E+09	-72.59
2.64E+09	-51.49
2.65E+09	-45.03
2.67E+09	-51.92
2.68E+09	-51.27
2.70E+09	-54.00
2.71E+09	-50.80
2.73E+09	-56.75
2.74E+09	-68.41
2.76E+09	-46.18
2.77E+09	-49.25
2.79E+09	-52.29
2.80E+09	-55.05
2.82E+09	-50.50
2.83E+09	-47.15
2.85E+09	-53.59
2.86E+09	-62.41
2.88E+09	-69.45
2.89E+09	-49.40
2.91E+09	-54.39
2.92E+09	-49.46
2.94E+09	-51.11
2.95E+09	-54.97
2.97E+09	-54.33
2.98E+09	-58.08
3.00E+09	-57.16
3.01E+09	-58.75
3.03E+09	-52.72
3.04E+09	-53.89
3.06E+09	-73.90
3.07E+09	-61.13
3.09E+09	-60.70
3.10E+09	-61.83
3.12E+09	-75.49
3.13E+09	-61.62
3.15E+09	-58.43
3.16E+09	-57.62
3.18E+09	-56.57
3.19E+09	-61.66
3.21E+09	-66.67
3.22E+09	-59.71
3.24E+09	-57.63
3.25E+09	-59.80
3.27E+09	-60.85
3.28E+09	-70.47
3.30E+09	-64.71
3.31E+09	-60.63
3.33E+09	-49.51
3.34E+09	-55.41
3.36E+09	-54.25
3.37E+09	-68.88
3.39E+09	-55.04
3.40E+09	-63.20
3.42E+09	-74.41
3.43E+09	-51.68
3.45E+09	-55.14
3.46E+09	-69.03
3.48E+09	-56.70
3.49E+09	-72.30
3.51E+09	-57.20
3.52E+09	-55.82
3.54E+09	-57.60
3.55E+09	-56.47
3.57E+09	-60.94
3.58E+09	-62.74
3.60E+09	-58.21
3.61E+09	-65.12
3.63E+09	-60.85
3.64E+09	-62.73
3.66E+09	-59.96
3.67E+09	-58.43
3.69E+09	-64.07
3.70E+09	-84.32
3.72E+09	-64.18
3.73E+09	-68.52
3.75E+09	-68.55
3.76E+09	-67.86
3.78E+09	-66.38
3.79E+09	-58.89
3.81E+09	-64.50
3.82E+09	-62.58
3.84E+09	-65.28
3.85E+09	-66.21
3.87E+09	-64.60
3.88E+09	-53.41
3.90E+09	-55.61
3.91E+09	-58.52
3.93E+09	-60.91
3.94E+09	-56.18
3.96E+09	-57.74
3.97E+09	-56.31
3.99E+09	-60.13
4.00E+09	-61.00

S12
135° CCW

Frequency	**dBm**
1.00E+09	-53.03
1.02E+09	-48.64
1.03E+09	-53.84
1.05E+09	-54.15
1.06E+09	-57.08
1.08E+09	-59.69
1.09E+09	-53.86
1.11E+09	-53.67
1.12E+09	-45.69
1.14E+09	-50.88
1.15E+09	-63.11
1.17E+09	-49.44
1.18E+09	-48.33
1.20E+09	-59.44
1.21E+09	-53.72
1.23E+09	-52.19
1.24E+09	-49.22
1.26E+09	-50.22
1.27E+09	-50.06
1.29E+09	-59.39
1.30E+09	-48.70
1.32E+09	-47.85
1.33E+09	-50.39
1.35E+09	-61.18
1.36E+09	-53.35
1.38E+09	-43.71
1.39E+09	-51.61
1.41E+09	-52.28
1.42E+09	-51.21
1.44E+09	-50.45
1.45E+09	-59.75
1.47E+09	-49.08
1.48E+09	-52.62
1.50E+09	-47.97
1.51E+09	-46.72
1.53E+09	-55.73
1.54E+09	-50.10
1.56E+09	-53.54
1.57E+09	-48.95
1.59E+09	-59.03
1.60E+09	-52.89
1.62E+09	-50.33
1.63E+09	-49.68
1.65E+09	-57.46
1.66E+09	-49.81
1.68E+09	-51.39
1.69E+09	-49.26
1.71E+09	-60.32
1.72E+09	-50.78
1.74E+09	-56.21
1.75E+09	-54.36
1.77E+09	-53.03
1.78E+09	-61.40
1.80E+09	-58.90
1.81E+09	-48.93
1.83E+09	-64.06
1.84E+09	-52.94
1.86E+09	-53.41
1.87E+09	-53.45
1.89E+09	-55.44
1.90E+09	-56.47
1.92E+09	-56.01
1.93E+09	-61.35
1.95E+09	-56.95
1.96E+09	-60.26
1.98E+09	-58.07
1.99E+09	-60.20
2.01E+09	-54.81
2.02E+09	-54.50
2.04E+09	-48.97
2.05E+09	-54.20
2.07E+09	-49.30
2.08E+09	-57.03
2.10E+09	-54.75
2.11E+09	-53.21
2.13E+09	-62.48
2.14E+09	-52.27
2.16E+09	-53.12
2.17E+09	-62.31
2.19E+09	-57.33
2.20E+09	-52.74
2.22E+09	-62.06
2.23E+09	-50.49
2.25E+09	-51.94
2.26E+09	-71.60
2.28E+09	-52.41
2.29E+09	-63.67
2.31E+09	-45.34
2.32E+09	-54.27
2.34E+09	-56.40
2.35E+09	-51.28
2.37E+09	-53.32
2.38E+09	-48.78
2.40E+09	-53.01
2.41E+09	-53.13
2.43E+09	-54.14
2.44E+09	-46.81
2.46E+09	-53.36
2.47E+09	-61.98
2.49E+09	-58.80
2.50E+09	-49.92
2.52E+09	-53.52
2.53E+09	-54.18
2.55E+09	-56.79
2.56E+09	-49.31
2.58E+09	-50.79
2.59E+09	-45.90
2.61E+09	-53.51
2.62E+09	-49.19
2.64E+09	-54.71
2.65E+09	-61.63
2.67E+09	-55.28
2.68E+09	-54.33
2.70E+09	-59.20
2.71E+09	-59.47
2.73E+09	-52.81
2.74E+09	-53.92
2.76E+09	-62.27
2.77E+09	-50.95
2.79E+09	-49.87
2.80E+09	-53.47
2.82E+09	-51.19
2.83E+09	-55.95
2.85E+09	-53.04
2.86E+09	-47.81
2.88E+09	-61.95
2.89E+09	-54.29
2.91E+09	-53.55
2.92E+09	-56.71
2.94E+09	-58.57
2.95E+09	-59.45
2.97E+09	-58.52
2.98E+09	-56.39
3.00E+09	-48.95
3.01E+09	-50.65
3.03E+09	-64.09
3.04E+09	-64.03
3.06E+09	-59.59
3.07E+09	-59.33
3.09E+09	-66.99
3.10E+09	-51.54
3.12E+09	-69.48
3.13E+09	-62.49
3.15E+09	-66.92
3.16E+09	-59.95
3.18E+09	-66.04
3.19E+09	-55.30
3.21E+09	-64.16
3.22E+09	-67.05
3.24E+09	-63.21
3.25E+09	-61.28
3.27E+09	-74.50
3.28E+09	-57.75
3.30E+09	-61.87
3.31E+09	-62.51
3.33E+09	-61.97
3.34E+09	-61.87
3.36E+09	-70.34
3.37E+09	-83.53
3.39E+09	-73.69
3.40E+09	-58.15
3.42E+09	-54.66
3.43E+09	-61.42
3.45E+09	-60.76
3.46E+09	-62.28
3.48E+09	-70.56
3.49E+09	-78.34
3.51E+09	-64.72
3.52E+09	-67.57
3.54E+09	-59.94
3.55E+09	-65.24
3.57E+09	-55.14
3.58E+09	-69.13
3.60E+09	-61.08
3.61E+09	-62.43
3.63E+09	-64.77
3.64E+09	-57.33
3.66E+09	-49.56
3.67E+09	-64.11
3.69E+09	-78.61
3.70E+09	-66.20
3.72E+09	-52.67
3.73E+09	-60.43
3.75E+09	-61.38
3.76E+09	-62.00
3.78E+09	-63.82
3.79E+09	-61.00
3.81E+09	-55.39
3.82E+09	-60.38
3.84E+09	-67.27
3.85E+09	-60.98
3.87E+09	-61.23
3.88E+09	-59.04
3.90E+09	-56.84
3.91E+09	-54.64
3.93E+09	-67.27
3.94E+09	-59.51
3.96E+09	-62.38
3.97E+09	-58.95
3.99E+09	-61.66
4.00E+09	-60.84

S12
180° CCW

Frequency	dBm
1.00E+09	-55.31
1.02E+09	-48.81
1.03E+09	-51.75
1.05E+09	-47.03
1.06E+09	-55.32
1.08E+09	-54.40
1.09E+09	-49.97
1.11E+09	-53.89
1.12E+09	-49.41
1.14E+09	-46.73
1.15E+09	-59.85
1.17E+09	-49.98
1.18E+09	-51.90
1.20E+09	-53.66
1.21E+09	-57.22
1.23E+09	-49.31
1.24E+09	-56.45
1.26E+09	-57.28
1.27E+09	-55.07
1.29E+09	-42.51
1.30E+09	-48.82
1.32E+09	-49.20
1.33E+09	-42.93
1.35E+09	-59.50
1.36E+09	-53.32
1.38E+09	-49.76
1.39E+09	-48.15
1.41E+09	-52.71
1.42E+09	-46.75
1.44E+09	-50.04
1.45E+09	-51.72
1.47E+09	-52.17
1.48E+09	-50.22
1.50E+09	-51.04
1.51E+09	-51.31
1.53E+09	-47.05
1.54E+09	-49.44
1.56E+09	-55.94
1.57E+09	-52.35
1.59E+09	-53.55
1.60E+09	-61.47
1.62E+09	-57.46
1.63E+09	-58.33
1.65E+09	-55.82
1.66E+09	-55.56
1.68E+09	-56.11
1.69E+09	-50.24
1.71E+09	-52.62
1.72E+09	-60.10
1.74E+09	-54.34
1.75E+09	-53.43
1.77E+09	-54.78
1.78E+09	-51.10
1.80E+09	-62.06
1.81E+09	-46.02
1.83E+09	-57.04
1.84E+09	-51.40
1.86E+09	-52.11
1.87E+09	-55.99
1.89E+09	-48.37
1.90E+09	-61.29
1.92E+09	-51.82
1.93E+09	-47.01
1.95E+09	-53.77
1.96E+09	-53.88
1.98E+09	-50.60
1.99E+09	-71.31
2.01E+09	-52.08
2.02E+09	-50.81
2.04E+09	-51.98
2.05E+09	-56.64
2.07E+09	-51.21
2.08E+09	-68.40
2.10E+09	-67.99
2.11E+09	-60.89
2.13E+09	-57.43
2.14E+09	-60.98
2.16E+09	-56.61
2.17E+09	-57.58
2.19E+09	-60.11
2.20E+09	-56.52
2.22E+09	-52.46
2.23E+09	-51.69
2.25E+09	-48.63
2.26E+09	-50.87
2.28E+09	-51.77
2.29E+09	-54.40
2.31E+09	-51.00
2.32E+09	-47.26
2.34E+09	-50.44
2.35E+09	-51.16
2.37E+09	-49.83
2.38E+09	-44.54
2.40E+09	-49.04
2.41E+09	-60.50
2.43E+09	-49.93
2.44E+09	-50.39
2.46E+09	-62.41
2.47E+09	-49.06
2.49E+09	-43.53
2.50E+09	-61.59
2.52E+09	-51.90
2.53E+09	-57.64
2.55E+09	-52.26
2.56E+09	-53.84
2.58E+09	-51.51
2.59E+09	-49.86
2.61E+09	-57.29
2.62E+09	-53.33
2.64E+09	-49.52
2.65E+09	-51.35
2.67E+09	-47.73
2.68E+09	-56.06
2.70E+09	-46.82
2.71E+09	-47.57
2.73E+09	-49.74
2.74E+09	-49.22
2.76E+09	-46.64
2.77E+09	-63.90
2.79E+09	-53.27
2.80E+09	-51.71
2.82E+09	-60.74
2.83E+09	-50.63
2.85E+09	-49.52
2.86E+09	-54.38
2.88E+09	-49.19
2.89E+09	-52.58
2.91E+09	-51.18
2.92E+09	-70.23
2.94E+09	-61.11
2.95E+09	-51.24
2.97E+09	-58.89
2.98E+09	-49.80
3.00E+09	-52.68
3.01E+09	-56.58
3.03E+09	-69.15
3.04E+09	-58.43
3.06E+09	-64.35
3.07E+09	-53.10
3.09E+09	-54.20
3.10E+09	-54.29
3.12E+09	-63.86
3.13E+09	-59.90
3.15E+09	-61.25
3.16E+09	-58.57
3.18E+09	-66.41
3.19E+09	-56.98
3.21E+09	-67.00
3.22E+09	-67.57
3.24E+09	-57.60
3.25E+09	-59.81
3.27E+09	-65.36
3.28E+09	-59.33
3.30E+09	-56.15
3.31E+09	-76.53
3.33E+09	-62.62
3.34E+09	-61.06
3.36E+09	-64.97
3.37E+09	-63.11
3.39E+09	-63.37
3.40E+09	-69.23
3.42E+09	-67.72
3.43E+09	-62.43
3.45E+09	-61.86
3.46E+09	-52.46
3.48E+09	-60.46
3.49E+09	-58.01
3.51E+09	-61.66
3.52E+09	-67.86
3.54E+09	-67.30
3.55E+09	-57.70
3.57E+09	-58.32
3.58E+09	-71.61
3.60E+09	-64.91
3.61E+09	-61.99
3.63E+09	-59.42
3.64E+09	-60.49
3.66E+09	-69.45
3.67E+09	-66.56
3.69E+09	-62.97
3.70E+09	-56.94
3.72E+09	-66.96
3.73E+09	-64.52
3.75E+09	-65.43
3.76E+09	-58.58
3.78E+09	-61.23
3.79E+09	-67.86
3.81E+09	-64.49
3.82E+09	-66.28
3.84E+09	-56.15
3.85E+09	-65.80
3.87E+09	-55.75
3.88E+09	-68.71
3.90E+09	-69.22
3.91E+09	-59.07
3.93E+09	-65.84
3.94E+09	-57.13
3.96E+09	-64.20
3.97E+09	-66.47
3.99E+09	-61.05
4.00E+09	-72.11

S12
225° CCW

Frequency	dBm
1.00E+09	-48.26
1.02E+09	-48.30
1.03E+09	-47.81
1.05E+09	-49.36
1.06E+09	-53.33
1.08E+09	-73.87
1.09E+09	-48.28
1.11E+09	-51.13
1.12E+09	-48.77
1.14E+09	-56.37
1.15E+09	-63.80
1.17E+09	-42.93
1.18E+09	-51.65
1.20E+09	-54.60
1.21E+09	-46.93
1.23E+09	-50.75
1.24E+09	-50.68
1.26E+09	-52.04
1.27E+09	-60.47
1.29E+09	-54.46
1.30E+09	-57.55
1.32E+09	-55.20
1.33E+09	-55.36
1.35E+09	-49.25
1.36E+09	-51.39
1.38E+09	-45.00
1.39E+09	-48.34
1.41E+09	-61.65
1.42E+09	-51.15
1.44E+09	-52.65
1.45E+09	-52.88
1.47E+09	-67.52
1.48E+09	-51.37
1.50E+09	-67.04
1.51E+09	-53.32
1.53E+09	-62.44
1.54E+09	-59.09
1.56E+09	-61.89
1.57E+09	-49.99
1.59E+09	-58.56
1.60E+09	-53.89
1.62E+09	-49.37
1.63E+09	-49.19
1.65E+09	-63.63
1.66E+09	-47.94
1.68E+09	-61.45
1.69E+09	-49.54
1.71E+09	-68.85
1.72E+09	-50.94
1.74E+09	-58.51
1.75E+09	-54.27
1.77E+09	-58.26
1.78E+09	-47.85
1.80E+09	-54.44
1.81E+09	-55.57
1.83E+09	-70.32
1.84E+09	-57.75
1.86E+09	-52.45
1.87E+09	-54.90
1.89E+09	-55.32
1.90E+09	-58.10
1.92E+09	-54.04
1.93E+09	-67.32
1.95E+09	-46.70
1.96E+09	-57.35
1.98E+09	-54.95
1.99E+09	-54.13
2.01E+09	-49.56
2.02E+09	-50.57
2.04E+09	-58.60
2.05E+09	-48.57
2.07E+09	-61.91
2.08E+09	-52.65
2.10E+09	-52.47
2.11E+09	-57.50
2.13E+09	-52.44
2.14E+09	-56.78
2.16E+09	-53.92
2.17E+09	-60.17
2.19E+09	-50.63
2.20E+09	-66.78
2.22E+09	-58.47
2.23E+09	-52.71
2.25E+09	-55.71
2.26E+09	-56.24
2.28E+09	-56.80
2.29E+09	-52.00
2.31E+09	-53.49
2.32E+09	-52.47
2.34E+09	-50.15
2.35E+09	-48.31
2.37E+09	-56.67
2.38E+09	-51.09
2.40E+09	-49.32
2.41E+09	-51.16
2.43E+09	-52.92
2.44E+09	-49.58
2.46E+09	-55.47
2.47E+09	-54.64
2.49E+09	-49.75
2.50E+09	-50.51
2.52E+09	-58.12
2.53E+09	-61.09
2.55E+09	-58.68
2.56E+09	-48.50
2.58E+09	-52.00
2.59E+09	-58.74
2.61E+09	-51.64
2.62E+09	-48.62
2.64E+09	-56.51
2.65E+09	-49.33
2.67E+09	-66.57
2.68E+09	-50.71
2.70E+09	-52.21
2.71E+09	-50.58
2.73E+09	-53.98
2.74E+09	-52.84
2.76E+09	-47.13
2.77E+09	-53.09
2.79E+09	-49.01
2.80E+09	-59.79
2.82E+09	-55.44
2.83E+09	-54.66
2.85E+09	-56.19
2.86E+09	-53.91
2.88E+09	-46.94
2.89E+09	-49.42
2.91E+09	-62.36
2.92E+09	-53.00
2.94E+09	-58.71
2.95E+09	-60.00
2.97E+09	-55.51
2.98E+09	-55.67
3.00E+09	-54.76
3.01E+09	-55.82
3.03E+09	-52.89
3.04E+09	-58.05
3.06E+09	-54.60
3.07E+09	-56.68
3.09E+09	-54.19
3.10E+09	-56.50
3.12E+09	-59.26
3.13E+09	-65.11
3.15E+09	-55.36
3.16E+09	-56.94
3.18E+09	-59.01
3.19E+09	-55.79
3.21E+09	-62.62
3.22E+09	-65.00
3.24E+09	-58.52
3.25E+09	-57.68
3.27E+09	-63.46
3.28E+09	-57.76
3.30E+09	-67.23
3.31E+09	-63.02
3.33E+09	-64.14
3.34E+09	-58.22
3.36E+09	-61.62
3.37E+09	-67.47
3.39E+09	-61.58
3.40E+09	-59.69
3.42E+09	-56.80
3.43E+09	-58.25
3.45E+09	-62.37
3.46E+09	-59.72
3.48E+09	-74.45
3.49E+09	-67.06
3.51E+09	-58.18
3.52E+09	-63.35
3.54E+09	-68.01
3.55E+09	-65.96
3.57E+09	-62.08
3.58E+09	-59.10
3.60E+09	-61.09
3.61E+09	-67.59
3.63E+09	-59.40
3.64E+09	-58.02
3.66E+09	-64.57
3.67E+09	-59.53
3.69E+09	-62.83
3.70E+09	-62.66
3.72E+09	-59.04
3.73E+09	-65.21
3.75E+09	-57.82
3.76E+09	-60.01
3.78E+09	-77.40
3.79E+09	-63.01
3.81E+09	-62.97
3.82E+09	-84.41
3.84E+09	-61.42
3.85E+09	-62.37
3.87E+09	-58.98
3.88E+09	-63.76
3.90E+09	-63.63
3.91E+09	-63.07
3.93E+09	-64.29
3.94E+09	-58.11
3.96E+09	-69.24
3.97E+09	-58.06
3.99E+09	-56.57
4.00E+09	-60.53

S12
270° CCW

Frequency	dBm
1.00E+09	-51.67
1.02E+09	-54.40
1.03E+09	-54.41
1.05E+09	-53.00
1.06E+09	-48.70
1.08E+09	-56.10
1.09E+09	-52.68
1.11E+09	-51.82
1.12E+09	-54.48
1.14E+09	-54.22
1.15E+09	-46.00
1.17E+09	-50.09
1.18E+09	-53.68
1.20E+09	-63.16
1.21E+09	-50.15
1.23E+09	-49.38
1.24E+09	-48.51
1.26E+09	-48.89
1.27E+09	-62.34
1.29E+09	-47.29
1.30E+09	-49.98
1.32E+09	-61.04
1.33E+09	-44.88
1.35E+09	-50.10
1.36E+09	-56.39
1.38E+09	-50.43
1.39E+09	-44.59
1.41E+09	-52.31
1.42E+09	-52.07
1.44E+09	-46.98
1.45E+09	-60.72
1.47E+09	-64.50
1.48E+09	-52.67
1.50E+09	-50.42
1.51E+09	-57.08
1.53E+09	-48.28
1.54E+09	-55.98
1.56E+09	-50.79
1.57E+09	-53.70
1.59E+09	-49.54
1.60E+09	-44.35
1.62E+09	-60.22
1.63E+09	-55.00
1.65E+09	-49.70
1.66E+09	-64.45
1.68E+09	-46.62
1.69E+09	-51.34
1.71E+09	-51.19
1.72E+09	-46.84
1.74E+09	-51.56
1.75E+09	-51.67
1.77E+09	-53.35
1.78E+09	-50.08
1.80E+09	-50.27
1.81E+09	-54.53
1.83E+09	-48.81
1.84E+09	-52.87
1.86E+09	-57.01
1.87E+09	-51.44
1.89E+09	-59.55
1.90E+09	-56.60
1.92E+09	-71.08
1.93E+09	-64.05
1.95E+09	-62.77
1.96E+09	-57.14
1.98E+09	-54.73
1.99E+09	-50.95
2.01E+09	-64.23
2.02E+09	-70.00
2.04E+09	-68.12
2.05E+09	-52.56
2.07E+09	-56.90
2.08E+09	-53.30
2.10E+09	-50.59
2.11E+09	-52.90
2.13E+09	-64.08
2.14E+09	-51.00
2.16E+09	-53.07
2.17E+09	-49.73
2.19E+09	-53.66
2.20E+09	-61.92
2.22E+09	-54.61
2.23E+09	-51.78
2.25E+09	-54.18
2.26E+09	-49.12
2.28E+09	-50.67
2.29E+09	-58.86
2.31E+09	-65.55
2.32E+09	-52.78
2.34E+09	-56.27
2.35E+09	-50.72
2.37E+09	-47.75
2.38E+09	-60.49
2.40E+09	-44.81
2.41E+09	-46.67
2.43E+09	-51.24
2.44E+09	-45.83
2.46E+09	-45.23
2.47E+09	-61.60
2.49E+09	-45.58
2.50E+09	-52.64
2.52E+09	-62.49
2.53E+09	-46.70
2.55E+09	-57.92
2.56E+09	-52.85
2.58E+09	-55.03
2.59E+09	-53.09
2.61E+09	-49.80
2.62E+09	-47.04
2.64E+09	-50.13
2.65E+09	-49.67
2.67E+09	-57.87
2.68E+09	-46.01
2.70E+09	-57.55
2.71E+09	-51.33
2.73E+09	-50.32
2.74E+09	-46.56
2.76E+09	-58.82
2.77E+09	-48.15
2.79E+09	-48.99
2.80E+09	-47.56
2.82E+09	-48.23
2.83E+09	-52.10
2.85E+09	-57.47
2.86E+09	-61.57
2.88E+09	-60.72
2.89E+09	-54.11
2.91E+09	-60.63
2.92E+09	-65.39
2.94E+09	-63.44
2.95E+09	-58.25
2.97E+09	-57.75
2.98E+09	-56.63
3.00E+09	-59.26
3.01E+09	-54.97
3.03E+09	-51.81
3.04E+09	-60.12
3.06E+09	-66.46
3.07E+09	-63.10
3.09E+09	-60.52
3.10E+09	-61.25
3.12E+09	-57.01
3.13E+09	-62.28
3.15E+09	-62.30
3.16E+09	-60.85
3.18E+09	-62.33
3.19E+09	-61.36
3.21E+09	-66.67
3.22E+09	-70.11
3.24E+09	-59.36
3.25E+09	-68.16
3.27E+09	-61.20
3.28E+09	-57.92
3.30E+09	-65.65
3.31E+09	-60.35
3.33E+09	-59.28
3.34E+09	-63.37
3.36E+09	-69.30
3.37E+09	-61.87
3.39E+09	-60.00
3.40E+09	-74.16
3.42E+09	-61.69
3.43E+09	-64.04
3.45E+09	-65.62
3.46E+09	-58.94
3.48E+09	-63.96
3.49E+09	-67.40
3.51E+09	-61.73
3.52E+09	-58.39
3.54E+09	-66.22
3.55E+09	-61.13
3.57E+09	-58.78
3.58E+09	-64.60
3.60E+09	-78.16
3.61E+09	-59.47
3.63E+09	-56.06
3.64E+09	-58.55
3.66E+09	-58.74
3.67E+09	-61.82
3.69E+09	-56.86
3.70E+09	-56.58
3.72E+09	-63.84
3.73E+09	-53.35
3.75E+09	-66.46
3.76E+09	-57.10
3.78E+09	-77.44
3.79E+09	-66.24
3.81E+09	-61.27
3.82E+09	-51.86
3.84E+09	-57.68
3.85E+09	-65.22
3.87E+09	-70.63
3.88E+09	-60.72
3.90E+09	-59.91
3.91E+09	-63.19
3.93E+09	-57.93
3.94E+09	-61.73
3.96E+09	-56.54
3.97E+09	-63.02
3.99E+09	-59.39
4.00E+09	-67.96

S12
315° CCW

Frequency	dBm
1.00E+09	-59.30
1.02E+09	-48.27
1.03E+09	-52.43
1.05E+09	-56.26
1.06E+09	-51.50
1.08E+09	-46.55
1.09E+09	-56.55
1.11E+09	-54.17
1.12E+09	-51.43
1.14E+09	-44.71
1.15E+09	-52.86
1.17E+09	-55.13
1.18E+09	-47.88
1.20E+09	-49.06
1.21E+09	-49.53
1.23E+09	-61.17
1.24E+09	-46.27
1.26E+09	-52.40
1.27E+09	-48.22
1.29E+09	-49.67
1.30E+09	-49.74
1.32E+09	-56.10
1.33E+09	-46.68
1.35E+09	-49.44
1.36E+09	-46.85
1.38E+09	-66.72
1.39E+09	-49.99
1.41E+09	-50.72
1.42E+09	-50.18
1.44E+09	-58.98
1.45E+09	-52.37
1.47E+09	-49.14
1.48E+09	-57.86
1.50E+09	-48.70
1.51E+09	-59.27
1.53E+09	-53.04
1.54E+09	-51.95
1.56E+09	-62.09
1.57E+09	-52.44
1.59E+09	-50.88
1.60E+09	-60.44
1.62E+09	-46.31
1.63E+09	-55.27
1.65E+09	-58.80
1.66E+09	-53.63
1.68E+09	-47.14
1.69E+09	-51.97
1.71E+09	-52.83
1.72E+09	-48.41
1.74E+09	-46.62
1.75E+09	-54.65
1.77E+09	-50.27
1.78E+09	-54.95
1.80E+09	-70.02
1.81E+09	-59.07
1.83E+09	-59.83
1.84E+09	-56.04
1.86E+09	-49.95
1.87E+09	-55.35
1.89E+09	-48.44
1.90E+09	-52.25
1.92E+09	-61.10
1.93E+09	-52.46
1.95E+09	-55.36
1.96E+09	-51.32
1.98E+09	-56.24
1.99E+09	-49.01
2.01E+09	-58.23
2.02E+09	-56.82
2.04E+09	-48.83
2.05E+09	-45.85
2.07E+09	-51.98
2.08E+09	-71.92
2.10E+09	-64.56
2.11E+09	-62.62
2.13E+09	-65.81
2.14E+09	-59.74
2.16E+09	-51.50
2.17E+09	-48.09
2.19E+09	-49.95
2.20E+09	-49.42
2.22E+09	-48.62
2.23E+09	-45.89
2.25E+09	-46.13
2.26E+09	-54.19
2.28E+09	-46.16
2.29E+09	-48.44
2.31E+09	-45.40
2.32E+09	-45.19
2.34E+09	-42.19
2.35E+09	-40.72
2.37E+09	-39.77
2.38E+09	-38.75
2.40E+09	-41.29
2.41E+09	-41.59
2.43E+09	-42.06
2.44E+09	-46.21
2.46E+09	-44.62
2.47E+09	-46.52
2.49E+09	-45.12
2.50E+09	-41.61
2.52E+09	-43.58
2.53E+09	-43.29
2.55E+09	-42.90
2.56E+09	-47.37
2.58E+09	-42.31
2.59E+09	-44.03
2.61E+09	-46.20
2.62E+09	-41.48
2.64E+09	-46.34
2.65E+09	-42.36
2.67E+09	-39.55
2.68E+09	-43.38
2.70E+09	-38.36
2.71E+09	-42.82
2.73E+09	-42.97
2.74E+09	-41.67
2.76E+09	-42.02
2.77E+09	-46.41
2.79E+09	-42.41
2.80E+09	-41.77
2.82E+09	-41.44
2.83E+09	-42.81
2.85E+09	-41.22
2.86E+09	-44.14
2.88E+09	-42.98
2.89E+09	-44.91
2.91E+09	-44.49
2.92E+09	-50.32
2.94E+09	-50.19
2.95E+09	-51.46
2.97E+09	-47.35
2.98E+09	-47.78
3.00E+09	-48.32
3.01E+09	-48.67
3.03E+09	-47.84
3.04E+09	-57.87
3.06E+09	-51.24
3.07E+09	-45.97
3.09E+09	-50.36
3.10E+09	-53.21
3.12E+09	-49.62
3.13E+09	-54.39
3.15E+09	-53.07
3.16E+09	-60.71
3.18E+09	-54.65
3.19E+09	-58.82
3.21E+09	-56.29
3.22E+09	-65.07
3.24E+09	-57.93
3.25E+09	-57.29
3.27E+09	-59.14
3.28E+09	-59.75
3.30E+09	-59.67
3.31E+09	-55.40
3.33E+09	-62.65
3.34E+09	-60.60
3.36E+09	-51.75
3.37E+09	-61.41
3.39E+09	-178.97
3.40E+09	-70.19
3.42E+09	-59.74
3.43E+09	-61.32
3.45E+09	-58.41
3.46E+09	-62.97
3.48E+09	-67.44
3.49E+09	-62.91
3.51E+09	-61.64
3.52E+09	-63.53
3.54E+09	-60.87
3.55E+09	-57.90
3.57E+09	-66.95
3.58E+09	-61.05
3.60E+09	-60.10
3.61E+09	-63.85
3.63E+09	-58.14
3.64E+09	-72.49
3.66E+09	-56.90
3.67E+09	-59.35
3.69E+09	-67.30
3.70E+09	-64.09
3.72E+09	-61.36
3.73E+09	-62.23
3.75E+09	-61.95
3.76E+09	-63.02
3.78E+09	-60.86
3.79E+09	-65.34
3.81E+09	-65.27
3.82E+09	-65.79
3.84E+09	-57.82
3.85E+09	-53.93
3.87E+09	-58.31
3.88E+09	-58.15
3.90E+09	-52.72
3.91E+09	-55.49
3.93E+09	-54.73
3.94E+09	-55.68
3.96E+09	-51.45
3.97E+09	-56.19
3.99E+09	-55.90
4.00E+09	-53.16

6.2 Quotes

Attenuator The PDF quote is attached

MCLI MICROWAVE COMMUNICATIONS LABORATORIES, INC.

7255 30th Ave. North
Saint Petersburg, Florida 33710
U.S.A.
PHONE: 727-344-6254
FAX: 727-381-6116
http://www.MCLI.com

QUOTE #: 001741

PAGE NO.	: 1
CHANGE ORDER	: 0
DATE ORDERED	: 02/22/2010
RELATED DOC #	:
SALESMAN	: PETER N.
CUSTOMER PH #	: (732)742-1978
CUSTOMER PO #	:
FOB	: ST. PETERSBURG, FL

Rutgers University
Kevin Ngan
Rutgers University, Dept. of ECE
5 Bradford Rd
Edison, 08820

SHIP TO ADDR

Rutgers University
Kevin Ngan
Rutgers University, Dept. of ECE
5 Bradford Rd
Edison, 08820

SHIP VIA: MCLI GROUND TERMS: PRE-PAY/CC ACCT: RUT002

ITEM	PARTNO	DESCRIPT	DATE REQ	QTY	UNIT	PRICE	TOTAL PRICE
0001	VC-3	Attenuator, Voltage Cont. 2-4 GHz	/ /	1	ea	1098.00	1098.00

6 WEEKS DELIVERY

NOTES:

MCLI QUALITY ASSURANCE SYSTEM: [illegible]
DESTINATION DISCLOSURE STATEMENT: [illegible]
[illegible] accordance with the U.S. [illegible] U.S. Export Administration Regulations and/or the U.S. Department of State may require an export license prior to exporting these goods. [illegible]
[illegible]
OVERSEAS PAYMENT TERMS: [illegible]
[illegible]
PRICE, PRODUCT, AND DESIGN CHANGES: [illegible]
WARRANTY: [illegible]

Quality System
Quality Endorsed Company
LEAD FREE Pb RoHS COMPLIANT

Phase Shifter

Taken from email response from Pulsar Microwave Corporation:

DST-13-480/1S (2.0-4.0 GHz)

1-3 pcs.: $1775.00 each - Delivery 6-8 Weeks ARO

Best regards,

Patrick Claudio
Sales & Applications Engineer
Pulsar Microwave Corporation
Phone: (973) 779-6262
Fax: (973) 779-2727
E-mail: sales@pulsarmicrowave.com
website: www.pulsarmicrowave.com

7 References

***picture of boune paths from: http://archive.electronicdesign.com/files/29/12985/figure_01.gif**
***picture of interference from:**
http://www.chem.wisc.edu/~newtrad/CurrRef/BDGTopic/BDGFigs/3_13cdfere.gif
***picture of transmitted and reflected waves:**
http://www.ece.unh.edu/courses/ece711/refrense_material/s_parameters/1SparBasics_1.pdf
***equations for waveguide from:**
http://www.nhn.ou.edu/~johnson/Education/Juniorlab/Microwave/CylindricalWaveguide.pdf
***diagram of cantenna from: http://www.saunalahti.fi/elepal/antenna2.html**

Mobile Repeater
Platform Subgroup

Dan Mendat
Yuriy Shames

Rutgers, School of Engineering
Department of Electrical and Computer Engineering
332:428 – Capstone Design: Communications System, Wireless Communications

Senior Design Project Report

Advisor: Professor Christopher Rose

Introduction

The platform is the cornerstone of the mobile repeater. It greatly enhances the capabilities of the repeater itself by ensuring that the repeater can get out of signal fade zones and better maintain a connection with the Wi-Fi router. In addition, repeater mobility is important in dynamic environments because the robot can then adapt to changes in its surroundings that affect signal paths. Yuriy and Dan worked very closely throughout the whole semester to make the platform work the way they wanted with each person shouldering his fair share of the workload and responsibility.

Choosing Components and Early Development

The original intent was to use a preexisting solution for the platform with the hope that developing movement algorithms for it would be simpler than creating everything from scratch. Thus, the existing ESRP robots at WINLAB seemed to be a viable option, and they were pursued first. Linux was the logical choice for the operating system to use because of the wide-ranging support among hobbyists. A laptop running Linux was not immediately available, so the initial attempt to set up the robots using Linux was done through a virtual machine with VMWare Server running on a laptop using 64-bit Windows 7. Difficulties were encountered using VMWare Server, so VirtualBox was tried next for virtualization. With VirtualBox up and running, a 32-bit Ubuntu distribution was running virtually, and the environment was all set.

However, problems were encountered while setting up the software needed to control the ESRP robot. Students at WINLAB had previously encountered those same problems, so their solutions to the errors were attempted, but the commands were unsuccessful. Therefore, the next step was trying Windows because the ESRP software was also available for that operating system.

The Windows setup did not go well either, and it was eventually discovered that 64-bit Windows is unsupported by the ERSP software, which probably explains the issues that were occurring. The software installed but would not actually talk to the robot and get it moving.

At the same time the Windows setup was experimented with, another robot platform from Zagros Robotics was investigated as a possible alternative. The difference with the Zagros Robotics platform is that it does not come with any software to control the robot. This means that the onus was on the platform group to figure out how to control all the motors and come up with all the necessary supporting hardware to get the platform to reliably move. Below is a picture of the Zagros Robotics platform.

Using this new platform is beneficial because it allows for more freedom. In essence, one is free to do anything possible to change the voltage across each motor that controls the wheels of the robot. This also means that building the robot was a good learning experience because research was done to learn about how motors are controlled. Another attribute of the platform is that the code can be divorced as much as possible from the hardware it controls, which means that the hardware can be swapped out if something more capable, cheaper, powerful, etc. comes along. Thus, a goal from the beginning of software development was to ensure that the functionality of the robot itself is controlled in one main section of code and that it is easily swapped for different code that controls another physical platform.

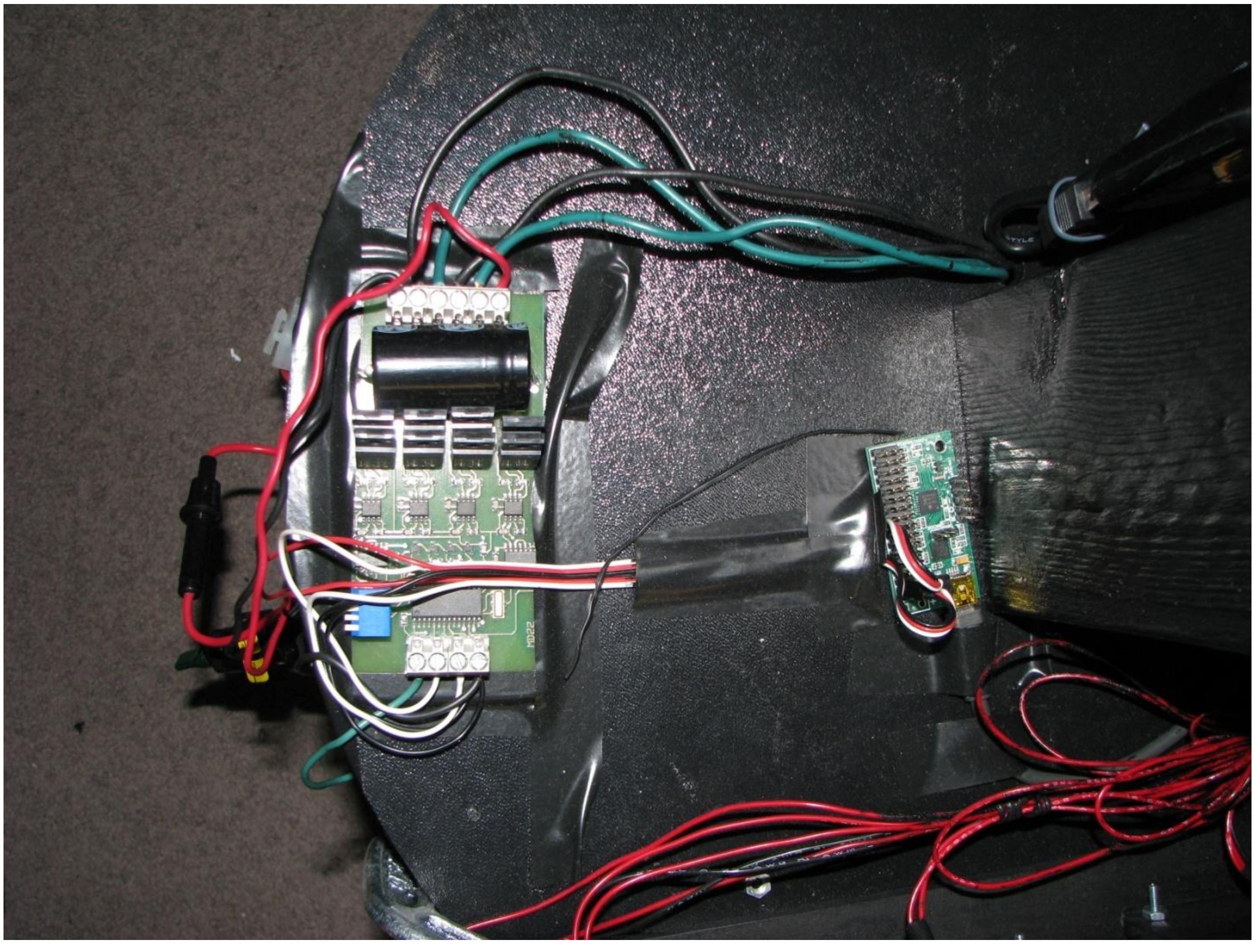
The next main decision was to determine how all the devices associated with the platform should be controlled. A microcontroller could have been used, but that would mean that the microcontroller would need to be programmed and run separately. Therefore, all the devices were chosen so that they would be connected to a laptop using its USB ports. This allows for much easier software development because of the advanced visual nature of the output medium as well as the ease with which code can be compiled and run.

Since a laptop was going to be used to control all the USB devices associated with the robot, two Acer Aspire One laptops were purchased. These netbooks were cheap (about $300) and small, making them ideal for use with the mobile platform. In addition, their battery life proved to be more than sufficient for testing the robots. The following is a picture of one of the netbooks.

There was already one Zagros Robotics platform set up that uses a motor driver that takes input from a servo controller, so it was clear that some of these components would be needed to make another robot. In addition, the servo controller that was already being used was a serial device, so it was time to upgrade to a USB servo controller for better compatibility with today's computers.

The main goal in choosing the servo controller was to ensure that it is easy to use. Therefore, since Pololu makes a USB servo controller that comes with documentation and example code, Pololu's USB 16-Servo Controller was chosen. This controller allows for simultaneous control of up to sixteen servos, so there was plenty of room to grow with the use of this controller. Since the interface is USB and the manner of controlling the device is simply sending commands to the controller over USB, no drivers were needed on Linux. This made communication with the controller theoretically very easy.

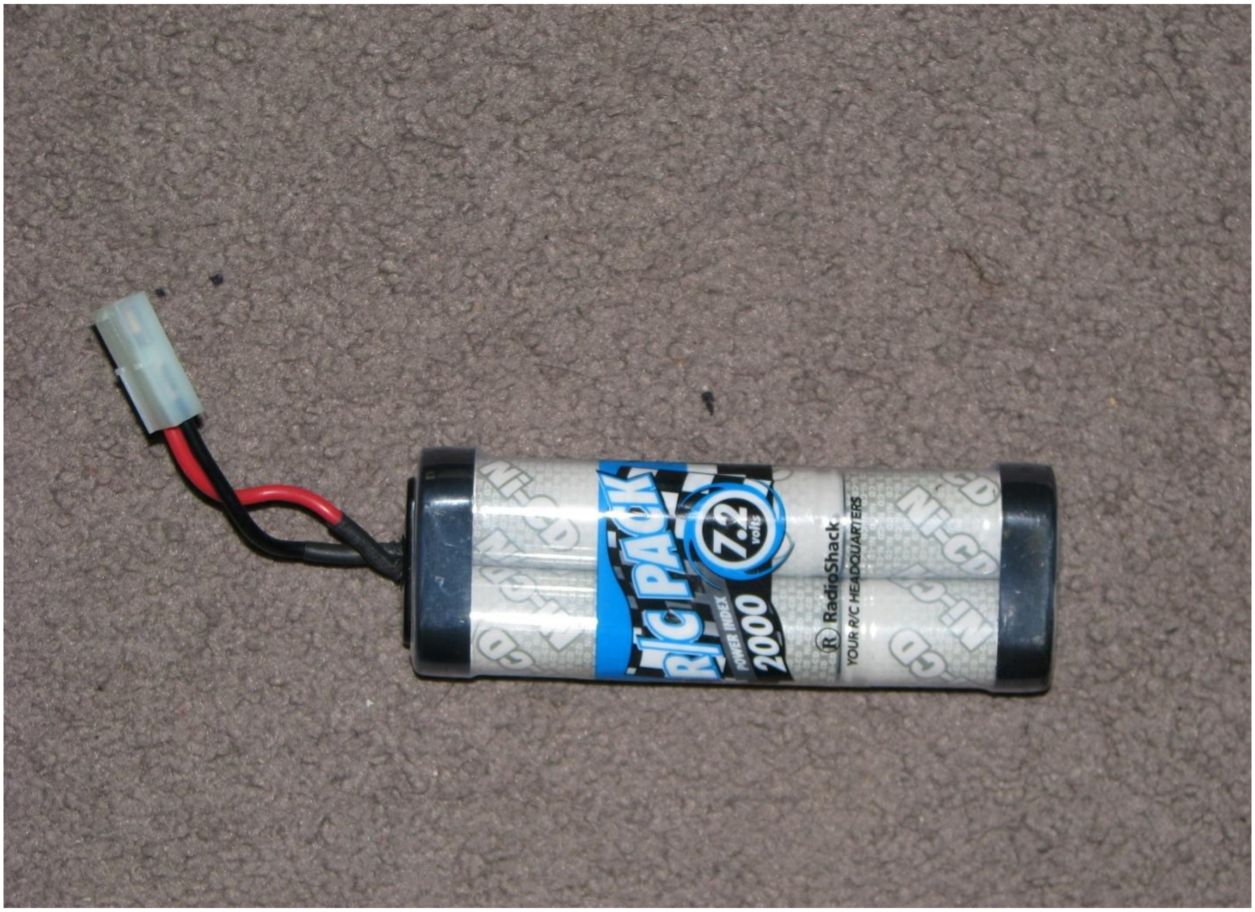
Since the motor controller for the other robot was used in the past and was also offered on Zagros Robotics' web site, it stayed with the robot for the duration of the project. It performs all the necessary functionality of controlling the voltages to the wheel motors of the robot in a manner that allows for turns as well as straight forward and backward motion. The motor controller bridges the gap between the servo controller and the wheels, since the motor controller translates signals the servo controller outputs into usable voltages that are placed across the wheel motors so the robot can move. In the picture below, the motor controller is the device to the left and the servo controller is the one to the right.



By the time the Zagros Robotics platform was chosen over the ERSP robot, it was clear that an alternative to a phased array would likely have to be used as the antenna that links with the router. Therefore, a servo was also needed for each platform in order to aim a directional antenna in an advantageous orientation. The project requires that the servo can move to specific positions, so a continuous rotation servo would not suffice. Most available servos rotate only in a 180 degree range, so for full rotation abilities of the antenna the platform itself would need to also move if it used one of those servos. However, a servo was found that rotates a full 360 degrees to avoid needing to move the platform for antenna orientation. This servo is the Acroname High Torque Full Turn Servo, and it was chosen for use in the project.

Accelerometers were deemed useful for honing in the robot's motion. Since the robots are imprecise, a compass and accelerometer could be used to ensure the robot really moves as expected. For example, if the robot is supposed to rotate ninety degrees clockwise and really only rotates 75 degrees, a compass could be used to discover that discrepancy and assist in correcting the difference. In addition, if an unexpected collision with an obstacle occurs, the accelerometer can inform the robot that the collision has occurred and help the robot get back on track and adjust to avoid the impediment. The search for combined compasses and accelerometers only yielded results that were very expensive, so they were not used. However, two accelerometers were bought in the hopes that they could be used to assist in making the platform's motion more precise. But as development moved forward, the accelerometers were not used because they would not afford much information about preciseness of movement. Movement mainly consists of three main events: starting, going, and stopping. Starting and stopping trigger the accelerometers, but the movement in between is smooth and unlikely to allow the accelerometers to provide extra useful information. Integration for detection of collisions that could go unnoticed by the ultrasonic rangefinders could prove to be useful though and is something that may be worth pursuing in the future.

The original platform had a 7.2 V hobby battery onboard that was stepped down using two diodes to a servo-acceptable 5.8 V. During early development, D cell batteries were used instead because of the ease with which they could be swapped as compared to having just one battery that needed to periodically be recharged. That way, no diodes were necessary because the voltage was low enough to work with. Unfortunately, those D batteries were unable to sufficiently power the platform. Whenever the robot was commanded to move quickly, the robot would stutter and struggle to move, but slower speeds worked correctly. Thus, rechargeable hobby batteries were used instead in the final design. These NiMH and NiCd batteries enabled the robot to reliably move at the robot's fastest speeds without even a hint of jittering. Below is a picture of one of the hobby batteries used.



Component Interaction

The laptop is the brain of the entire robot, and all the components are controlled through the laptop via USB cables. Everything involved with the platform itself is controlled through one USB cable. This USB cable is connected to the USB servo controller that can accomplish all the necessary movement tasks for the robot.

The way the servo controller works is that it takes commands from the laptop and translates them into analog pulse width modulated (PWM) signals. The width of these signal peaks determines how the connected servos move. The pulse width range for the Pololu servo controller is from 250 to 2750 µs, and the neutral position is 1500 µs (1.5 ms). At the neutral position the connected servos do not move. As the pulse width goes up, the servos move in one direction either farther or faster, depending on whether the servo is a continuous rotation servo. As the pulse width moves down, the behavior is the same except that the direction is opposite.

Three slots out of the available sixteen channels of the servo controller were used for the platform. One slot was for the servo that controls the orientation of the directional antenna, and the other two were plugged into the inputs to the motor controller.

The motor controller is used to take the PWM signals from the servo controller and translate them into usable analog DC voltage for the wheel motors. Those motors take -5 to 5 V DC, and that's what the motor controller outputs. When the voltage is inverted, the motors spin in the opposite direction. The motor controller can basically translate the neutral PWM signal (1.5 ms width) into 0 V. Larger widths translate to higher voltages, and lower widths translate to numerically lower voltages that have higher magnitude. There are different modes the motor controller can operate in though, so the process is not quite that simplified.

The motor controller was set in a mode called "RC Servo with Differential Drive," and that means that the motor controller can take PWM input signals with widths from 1 ms to 2 ms and translate them into DC voltages for the wheel motors. One channel is used to control steering and another channel is used to control movement forward and backward. This means that when a 2 ms PWM signal is sent on one channel, the robot moves forward at full speed. A 1 ms PWM signal on that channel moves the robot backwards at full speed. Those same values fed into the other channel rotate the robot clockwise and counterclockwise. Utilizing both channels at once creates the logical combination of forward/backward movement with simultaneous turning behavior similar to the way an RC car behaves with separate joysticks for straight and turning movements.

Commands are sent to the servo controller through the USB port by directly writing to the file corresponding to that port on Linux (see the code in servo.cpp for more details). Each command consists of five or six bytes. The first byte is a starting byte that has a fixed value, the second byte is the device ID that is fixed for the servo controller, and the third through sixth bytes specify the actual meat of the commands that tell the servos how to move. All the commands sent to the servo controller for this project used command four, which sets the absolute position of the servo. The fourth byte is the channel of the servo being sent the command. The next two bytes contain the upper and lower bits of the absolute position, where the range of values are mapped to the range from 500 to 5500.

The overall picture of the chain of command is that the laptop sends a position value to the servo controller corresponding to a specific channel, and that PWM signal is sent to whatever is connected to that channel of the servo controller. If the device is the servo for the directional antenna, the directional antenna moves to that position. If the device is one of the inputs to the motor controller, the motor controller translates the PWM signal into a DC voltage to control the motors in the manner described above.

Hardware and Software Development

The first robot platform was already set up, but it had an old servo controller with a serial interface. The USB servo controller was purchased for this project, so it was swapped in instead of using the serial servo controller. With the basic setup squared away, some test code was developed to get a feel for the robot and its movement capabilities.

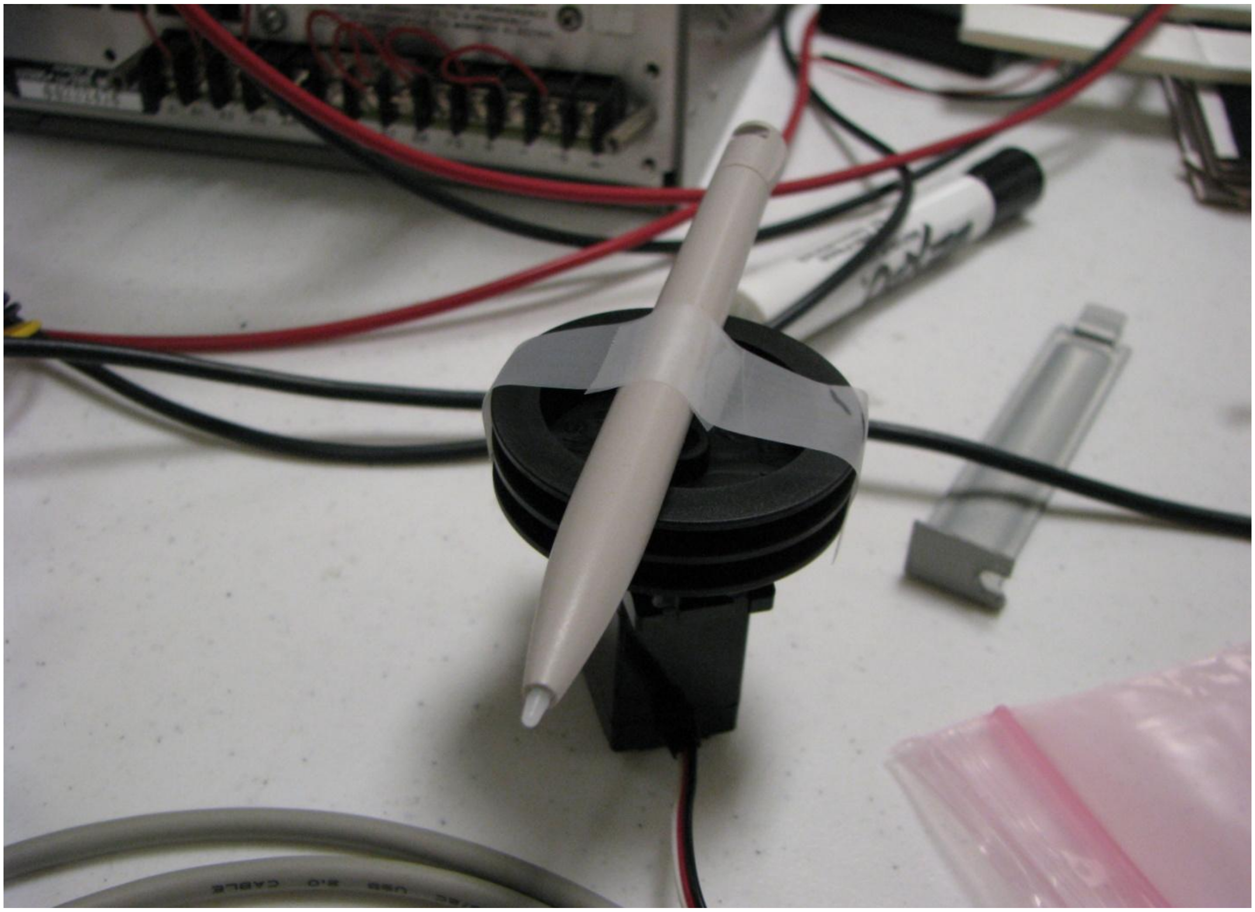
The first major hurdle was getting the laptop to communicate with the servo controller correctly. Pololu's website has some nice documentation that outlines the communication with the servo controller, so the commands weren't too difficult to get going once messages were traveling across the USB cable correctly. However, USB communication was initially a challenge. The platform group members were unused to writing to a serial or USB port, so research was done to determine how to accomplish that task. Once established, there was some trial and error coding done to make sure the parameters for the serial port were correctly set. One of the main problems was the baud rate. Initially, the baud rate was too low for the servo controller, but once it was increased the servo controller's status lights indicated that the commands were being received.

Once USB commands could be sent to the servo controller, the next step was powering the platform and trying it out. Four D batteries were initially used to power everything at 6 V, and basic commands were sent to the platform to test movement. Immediately a problem cropped up, though. The robot was acting erratically, only moving part of the time when it received commands. The problem turned out to be that there were faulty connections in the original wiring of the platform. When all the connections were checked and improved when necessary, the robot began to operate smoothly.

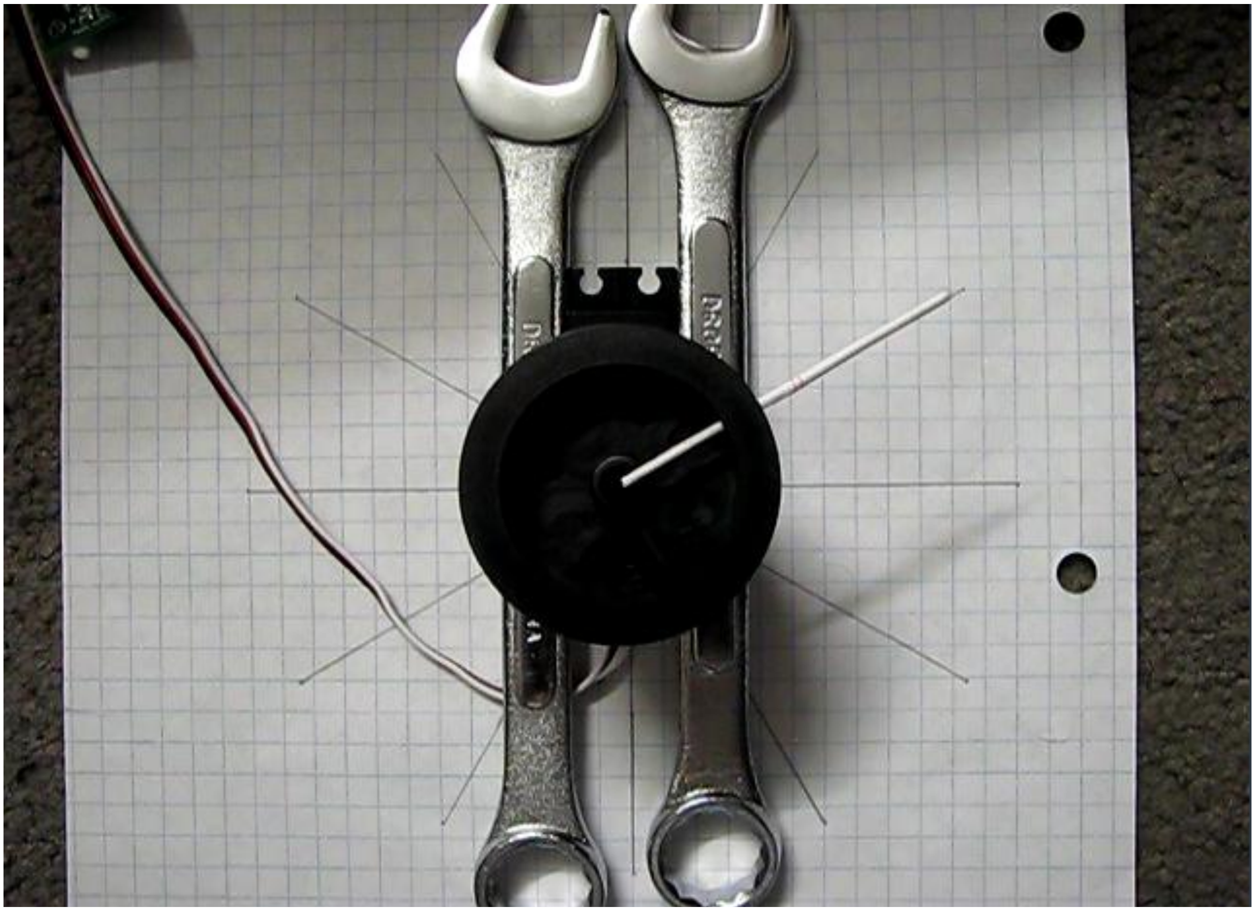
However, soon after the robot started reliably moving, it again was unpredictable and was not following commands. The robot would try to move when no command was issued, and it would move in completely different directions than what it was told to do. After swapping in the other servo controller it was determined that the first servo controller was burnt out. More were quickly purchased. The wiring was investigated, and a faulty ground was found. There were two separate grounds in the platform wiring. Once the grounds were connected the robot moved reliably again.

The software was starting to get slightly complicated, and it was clear that organization of code was going to be very important. Organized code would make things much easier for future groups of people working with the robot, and it also makes things easy for the first group who started development. Therefore, everything began to become modularized, and the code was broken down into classes in C++ to make everything be as logically separated as possible. Comments were included in the code from the very beginning to make referencing and building upon previous work simple. In order to prevent repetition and to make things clearer for people continuing to work on this project in the future, an explanation of the code organization is found in the "Code Organization" section of the report later on. This way, the "Using the Robot" section should be a useful quick start guide for working with the robot and understanding how the different components interact.

In order to make the directional antenna change its orientation, the two servos had to be calibrated. Initially, one servo was calibrated by taping a pen to the top of the servo and making sure that it could spin in approximately ninety degree increments (see below). However, this method was just an initial attempt and was quickly improved. Calibration involves altering the position values sent to the servo controller from the computer to see where the servo moves and adjusting those values until the servo rotates to the desired orientation. These calibration values can be found (and used) in the code in servo.cpp.



The goal of the improved calibration was to make the servos' positions be visually accurate in thirty degree increments from 0 to 360 degrees so that the antenna orientation could be changed as the robot moves (see below). Lines marking each of those angles were drawn on paper, and an outline of the servo was drawn in the middle to ensure that the servo was centered on the diagram correctly. Both servos were calibrated using this method.



As the calibration and other development were going on, work with the accelerometers was also started. A simple class was developed to get the acceleration values the accelerometers output. However, they proved to not be immediately useful for the project, so development for them was halted and instead focused on the movement of the platform itself.

The next step was making an interface to remotely control the robot. This was valuable for testing purposes because the robot could move based on user input instead of following a set of coded paths. Thus, every time the robot was controlled in this manner, a variety of tests could be performed on the fly to see how it operated. This interface is found in curseinterface.cpp, and more details about compiling and how all the software modules tie together is found in the "Integration" section later in this report. In essence, though, the remote control interface is set up so that a user can simply hit keys on the keyboard in an intuitive manner (similar to moving video game characters) and easily move and control the robot.

The controller interface to the robot is not natively separated from the robot itself, though, since it runs on the netbook sitting on the robot platform. Therefore, to control the robot remotely, an SSH server and client were established on each of the two netbooks to enable either one of them to be accessed from a standalone machine away from the robot. This has the added benefit of making debugging the robot's motion much easier because the person testing the robot can view the status of the robot without having to kneel down and view the laptop on the robot or follow the robot as it moves by carrying the laptop next to the moving robot. The SSH server and client were installed by going into the Ubuntu software repositories and installing them. More details about the SSH process can be found in the "Integration" section.

Since the platform's wiring was faulty at the beginning, work was done to solder everything to ensure that all the connections are solid. Along the way, battery pack connections were soldered so that the D cell batteries could all be replaced with the rechargeable RC car batteries. In addition, a power switch was added so that the robot could be shut off if it acts erratically. This also made testing the robot easier because the robot could simply be shut off to not waste battery power instead of requiring that the batteries must be removed to accomplish the same goal.

Throughout testing thus far, the robot was not being sent commands to move as quickly as it possibly could because it often jittered when sent commands to move quickly. However, once the rechargeable NiMH and NiCd batteries were utilized, the robot was able to reliably move at its fastest speeds. This was nice because the robot isn't very fast at its quickest pace, and it was even slower before the higher speeds were attained.

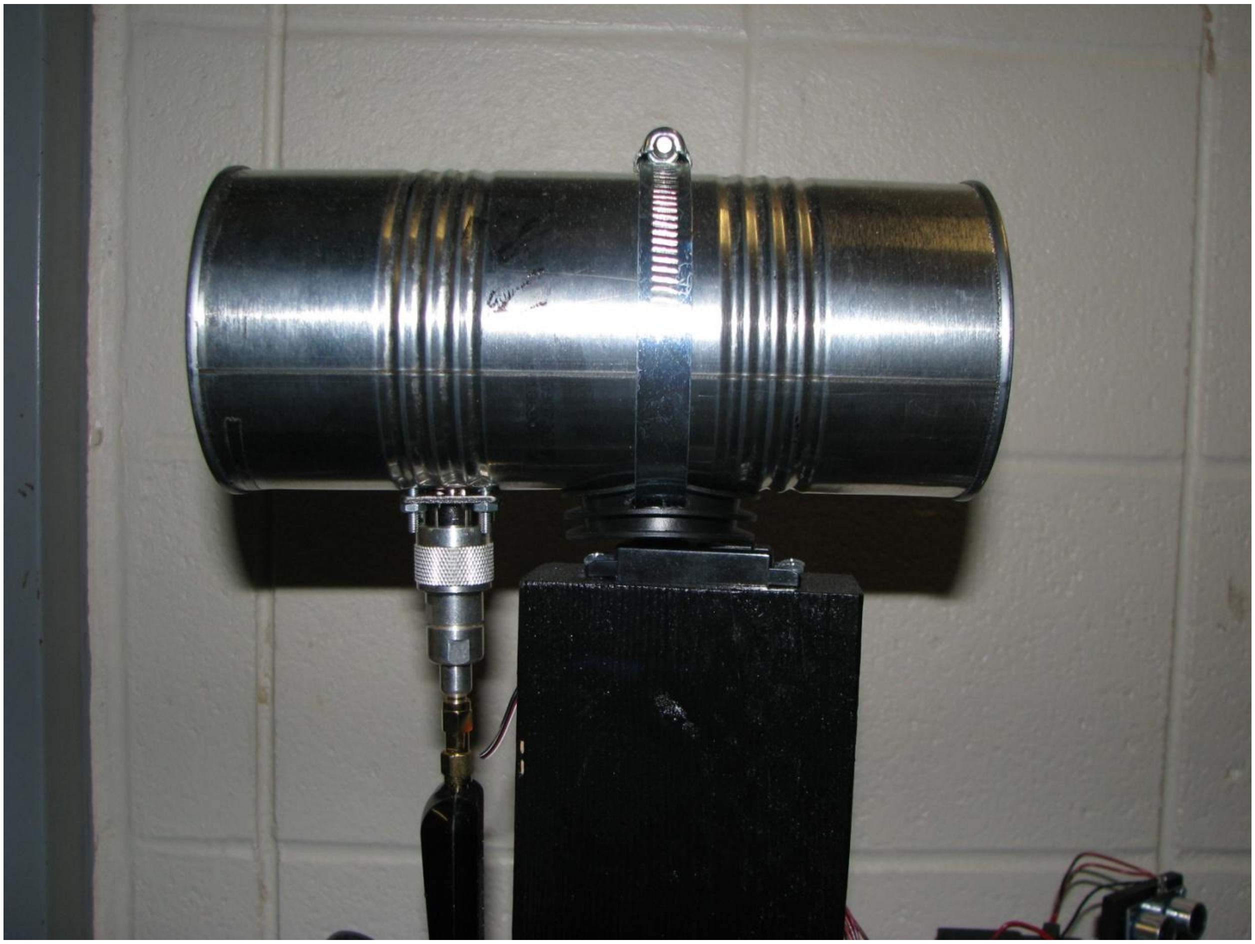
A goal of the project was to set up a second platform with the hopes that both platforms could work together to achieve better signal strength than one robot can provide. More information about this idea and others can be found in the "Future Work" section of the report. The second robot was soldered together the same way the first robot is set up, and a new motor controller of the same type as the first one was used. Unfortunately, though, when testing the movement of the new platform, most commands made the robot move forward instead of performing as expected. The wiring was double-checked, and everything seemed to be wired correctly. Swapping USB servo controllers did not change the behavior, so the problem was narrowed down to be the motor controller. It was set to the same mode as the old one, and it was wired up the same.

Swapping the first motor controller into the second robot made the second robot work correctly, though, and putting the new motor controller into the first robot made the first one work incorrectly. Therefore, the problem lies with the new motor controller rather than with the rest of the robot. This issue has not yet been resolved, but the old motor controller was placed back on the first robot and it works correctly again.

Throughout the whole platform development process, the platform group worked on keeping abreast of everyone else's progress and tried to ensure that the integration would go smoothly. This meant, for example, that when people wrote code, the platform group found out about the software's basic structure and made sure it accomplished what would be necessary for integration to work.

Integration

The first integration step involved beginning to get the directional antenna mounted to the servo so it could be used with the rest of the robot. This process was repeated for the second robot as well. The antenna group took care of connecting the antenna to the servo, but the platform group worked on creating a mount for the servo itself so the antenna would be perched away from the rest of the components in case its wire would hang down and be in danger of hitting the other components. The mount was made out of a vertical piece of two-by-four with an area drilled out at the top for the servo. Toward the back of the board a hole was drilled in the proper location for the servo cables to come out and be connected to the servo controller. The board was painted black to blend in with the platform. Holes were drilled through the board and the platform to use two screws to mount the board onto the top of the platform, and four more screws were used to attach the servo itself to the top of the board. Below is a picture of the mounted servo and antenna.



The antenna group wrote some code to determine the signal strength of the Wi-Fi connection the directional antenna receives, so the platform group integrated that code into the rest of the codebase. This integration involved converting the function created by the antenna group into a class and making some minor adjustments such as removing console prints and allowing the number of readings that are averaged to be variable.

The whole mobile repeater group tested the operation of the directional antenna’s signal strength value collecting. Using the code to rotate the servo, the platform group wrote simple code to point the antenna in specific directions and call the code started by the antenna group to get the signal strength in that orientation.

All was going productively, but suddenly the servo began behaving strangely the day before the Rutgers Day demonstration of the robot was to take place. Unfortunately, the servo broke because the cable attached to the antenna snagged on some other robot components. As a result, the servo began to spin without stopping when many position values were sent to the servo, and the motions were unpredictable. The tension in the cable wore out the servo.

The second servo was then put in place of the first servo, but the same problem occurred as the platform group was testing the operation of both the servo and the antenna itself. Since the Rutgers Day demonstration was the next day, the platform group decided to just keep the servo in a static orientation on top of the robot and simply have the robot rotate in place to point the antenna rather than separately rotate the antenna. The platform's motion was extremely reliable, so development was quickly started.

It was not long before code was written by the platform group that calibrated the platform's rotations so that it could rotate in approximately forty-five degree increments at a time. Then, the original working movement algorithm was coded. This algorithm worked by having the robot "sweep" for signal strengths and then move in the direction of best signal strength, sweep again and then move, and so on.

The sweep involves checking the signal strength at the original orientation, rotating forty-five degrees, checking the signal strength again, rotating another forty-five degrees, checking the signal strength, and so on until the full circle is completed. Whichever direction had the best signal strength was the direction the robot moved in before stopping and sweeping again before continuing movement.

In this original implementation, the signal strength was just one reading, so common fluctuations in signal strength in each direction were taken at face value and not averaged. However, in the final version of the movement algorithm, three readings are taken at each position and averaged to get a better picture of what the signal strengths at those various orientations are.

The original intent of the algorithm after consulting with Professor Rose was to have the robot constantly move forward and rotate the antenna checking for directions with better signal strength. However, due to broken servos, the robot had to be stopped and rotated in place so that the signal strength could be checked in a full circle's worth of orientations. This method may make more sense, though, since then the strengths are gathered from one set position rather than having them all taken at slightly different positions as the robot moves forward during readings.

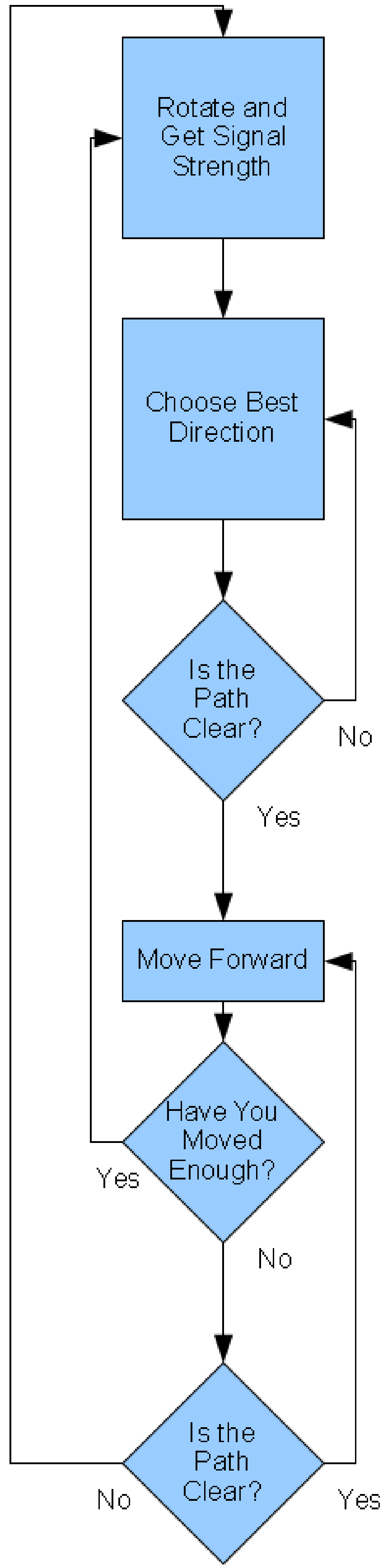
The first experiments involving tracking the signal down with the first movement algorithm were done in the EE building on Busch Campus at Rutgers. On the first floor the wireless router is located up high on the wall of the main hallway close to the front door. The platform group placed the robot at the other end of the hallway and let the robot move. Some tweaking was done to fix some small bugs involving the direction the robot chose to travel in, and once that was done the robot did travel toward the router. Unfortunately, no video was taken at that time. Sometimes the robot traveled directly toward walls, so presumably a strong bounce path came from those directions, so the robot favored moving toward the high signal strength created in those situations.

Soon after, the collision detection group was ready to integrate their portion of the project into the platform. The platform group took all the code the collision detection group wrote and again tweaked it to better fit into the framework of the overall project. The original implementation of the collision detection was to have a threshold for all the sensors. Whenever an object came within that distance of any of the sensors, a collision would be detected, and the robot would act appropriately.

While testing the threshold implementation, though, a problem became immediately apparent. The ultrasonic rangefinders were not updating their values very quickly inside the controlling program on the laptop. When a hand was waved in front of the sensors, the values did not decrease to a few inches until a few full seconds later, which was way too late for the robot to react. By the time the distance values got updated within the software application the robot would have hit the wall.

The fact that the values updated quickly when the collision detection group demoed their code meant that the problem must be in the implementation. After reviewing the changes with the collision detection group, it was clear that none of the alterations the platform group made would affect the rate at which the sensor values updated. The big difference in implementation is that the slow version only polled the serial port for data periodically whereas the fast version kept polling the serial port all the time. Therefore, the movement algorithm code was revamped by the platform group so that it became multithreaded. One thread constantly polled the sensors and updated some shared memory while the other thread performed the movement duties. This constant polling was effective in making the values change quickly in response to objects being placed in front of the sensors, since the values were recorded from the sensors all the time and only used periodically.

This was when the final version of the movement algorithm was established. It is very similar to the old version except that imminent collisions are checked and dealt with if necessary. When a seek occurs, the robot also makes sure that the path in each possible direction is clear. If an obstacle blocks the way, the robot will not go in that direction and instead will go in the next best signal strength direction. In addition, as the robot moves forward for its seven second interval before its next seek, the robot checks every quarter of a second to make sure that no collision is imminent. If a close obstacle is detected, the robot stops and performs another seek so that it does not run into the object. Below is a flowchart describing the final version of the algorithm.


Rotate and Get Signal Strength
Choose Best Direction
Is the Path Clear?
No
Yes
Move Forward
Have You Moved Enough?
Yes
No
Is the Path Clear?
No
Yes

While all the groups were performing initial testing of the final version of the movement algorithm together, it was clear that the thresholds for the robot needed to be changed. The set threshold of fifteen inches for each sensor was not working well when the robot came toward a wall at an angle. Sometimes the sensors did not pick up the fact that the wall was too close until it was too late. Therefore, a different threshold was set for the outside sensors. The final value for the outside sensors is set at twenty-two inches. This way, when the robot approaches a wall at an angle, the outside sensors pick up the wall in time and the robot can seek to move in an obstacle-free direction.

The first tests of this algorithm were done in a dormitory hallway where there isn't a lot of free space and the router was located right in the small hallway. Because of the proximity to the router and the lack of room to move, the full mobile repeater group expected the robot to follow lots of bounce paths and need to avoid obstacles frequently. After changing the thresholds for the rangefinders as described above, the robot performed admirably and avoided hitting all of the walls. In addition, when the router was around a corner, the robot did usually go toward that corner of the hallway in a meandering manner.

The next test was to put the robot in a room off to the side of the hallway where some couches and chairs were scattered around. In this environment the robot was still able to avoid hitting those obstacles, and it positioned itself a little bit closer to the router which was outside the door to the room.

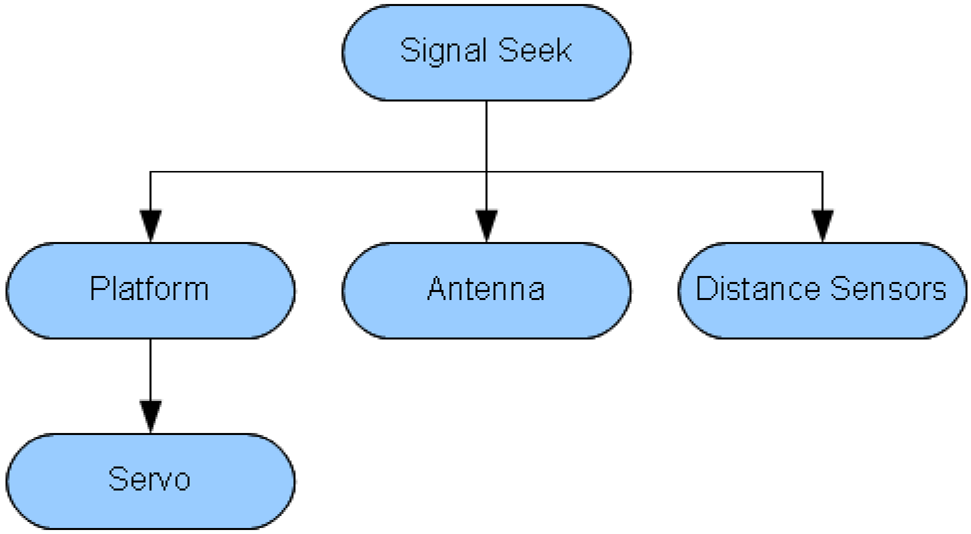
Excited by this success, the entire repeater group went over to WINLAB and tried out the robot in the large ORBIT room. The position of the router for the wireless network the robot was connected to in WINLAB is unconfirmed, but it looks like it is probably one of the few routers hung up by the front door to the building. When the robot was set loose in ORBIT, the robot consistently meandered toward that area of the building and could successfully avoid boxes, ladders, and other obstacles in its path.

The final test at WINLAB was placing the robot in a hallway. The router is suspected to be at the end of the hallway. Again, the robot worked as expected, traveling down the hallway toward the router. The robot tried to follow some paths toward walls, but the collision detection team's setup successfully kept the robot from hitting the walls and filing cabinets placed in the corridor.

Later, back in a large lounge located in the dormitory, another experiment was completed. The robot was placed in the center of the room and the router was placed in one corner. Then, the robot was placed back in the center of the room and the router was moved to a different corner. In both of those situations the robot moved toward the router.

Finally, a dynamic test was done. The robot was again placed in the middle of the room, and the router was placed off to one side. Once the robot started moving toward that location the router was quickly moved to the opposite side of the room. The robot turned to go in that direction, and the router was again returned to the original side. This was repeated until the robot stopped moving. This test shows that the robot can adjust to changing conditions as they occur and that it really is tracking the correct router signal.

Code Organization



- servo: This class contains code that directly accesses the USB port and talks to the servo controller. Calibrated position values for the two broken servos are in this class, but they're not useful because the servos broke. However, for the purposes of calibrating future servos, the framework is there to make such a task simple. In the current implementation, this class is not directly accessed because the servos broke. However, should someone want to use a servo in the future, directly accessing this class will be necessary.
- platform: This code is used to control the platform itself. It can be used to move the robot in various directions. This class encapsulates the servo class in order to control the wheel motors of the platform.
- antenna: This class is used to determine the signal strength of the Wi-Fi connection that the directional antenna uses.
- distanceSensors: This code controls the ultrasonic rangefinders and allows distance values to be gathered from them.
- signal_seek.cpp: This is the main driver for the operation of the mobile repeater. This class polls all the distance sensors, gets the signal strength for the directional antenna, and moves the robot. This code is multithreaded to ensure quick updating of the distance sensor values by constantly polling the sensors.
- remote_control.cpp: This driver allows the robot to be controlled remotely, much like an RC car.

The remote control application uses a library called "ncurses," which makes it easy to create an interface that gets cursor input to control a program. Therefore, to compile and run remote_control.cpp, the ncurses library must be installed.

To compile and run signal_seek.cpp, the pthread library must be set up. On the version of Ubuntu found on the netbooks, no installation was necessary because it was already included in the distribution.

Using the Robot

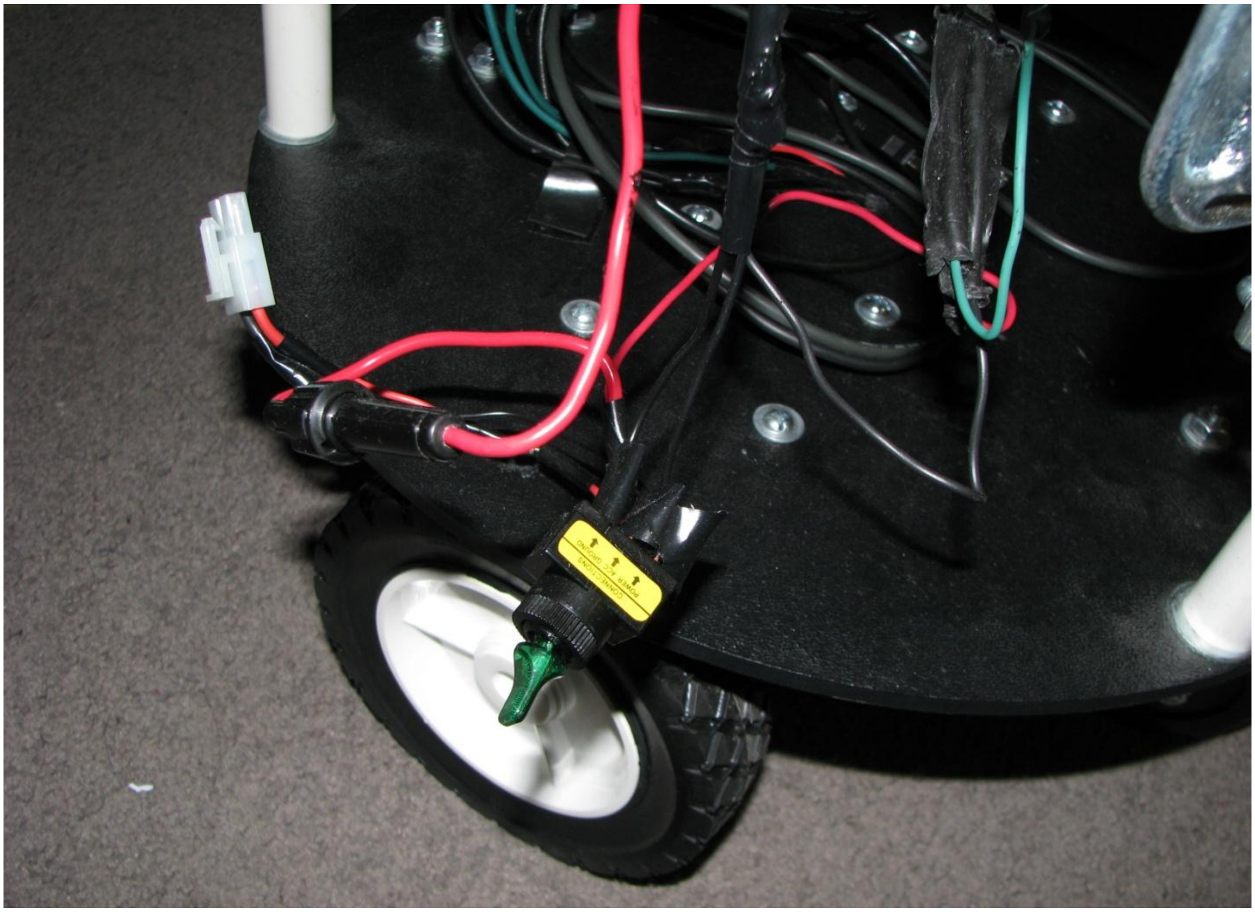
To log into the two netbooks, the username is “mobilerepeater” and the password is “signaltonoiseratio”. This password is used everywhere a password is required for this project.

To get the robot up and running, first plug one of the RC batteries in to the connector on the robot. Then, with the laptop turned on, the USB devices must be connected in a specific order. The servo controller must be connected first so that it is addressed as USB0 and then the arduino board must be connected so it’s USB1. Finally, the external directional wireless antenna can be connected anytime after that since it is accessed based on its name rather than its order of being connected.

Before running anything on the laptop to make the robot move, make sure that the servo controller has one yellow and one green light turned on. If this is not the case, disconnect the servo controller and reconnect it until those two lights are on. In addition, it is prudent to disconnect the servo controller and reconnect it periodically to reset the controller.

In addition, make sure that the directional antenna is pointed forward on the robot. If the antenna is not pointed forward, the robot will not move in the correct directions of best signal strength because it will be picking up the signals from directions other than straight ahead. It is not currently designed to work with other antenna orientations.

Once everything is set up as explained above, turn the robot on using the switch located in the wiring on the side or top part of the platform. Below is a picture of the switch.



The code to run the robot is already compiled for use on either of the two netbooks, and so is the code used to remotely control the robots. To run the code to seek the signal, simply navigate to the directory the executable is found in and type in "./demo" without the quotation marks. To run the code to control the robot remotely, type in "./remoteControl" without the quotation marks.

If any changes are made to the code or a different laptop is used on the robot, the code must be recompiled. To compile the signal seeking code, run the following command: "g++ signal_seek.cpp antenna.cpp servo.cpp platform.cpp distanceSensors.cpp –lpthread –o demo". "demo" can be replaced with anything else to change the name of the executable created. If everything after and including the "-o" part of the command is removed, the executable will revert to its default value of "a.out".

To recompile the remote control code, run the following command: "g++ remote_control.cpp antenna.cpp servo.cpp platform.cpp –lncurses –o remoteControl". Again, the output executable "remoteControl" can be renamed.

These applications are run directly on the laptop located on the platform itself. Therefore, to run them in this configuration means typing in the command to run the specific application and then either hold the laptop and walk with the robot or run the command and then step away. Neither of these scenarios is very convenient. However, this can be overcome by remotely controlling the robot through an SSH connection. As long as the two laptops can see each other (meaning that they can ping each other by being accessible and most likely on the same local network) the laptop external to the robot can log into the other machine. The command to be run on the external laptop is the following: “ssh –C mobilerepeater@<IP address of platform laptop>”. The –C flag means that the data sent across the link is compressed, which can help make the interface a little more responsive over slow connections. If prompted about unrecognized hosts, the user must also type in “yes” as a response. The IP address of the robot laptop can be determined by right clicking on the wireless icon in the upper right portion of the screen of that laptop and clicking on “Connection Information”. Then, look at the tab for the “ra0” connection which is the external directional antenna and what is used when connecting.

Videos of the robot can be found at http://www.youtube.com/user/RUmobilerepeater.

Future Work

Work toward using the accelerometers was halted during early development to focus on the rest of the platform, but accelerometers could be a useful backup for the collision detection system. They can be used to check if a sudden deceleration occurs, which would mean that the robot has collided with an object that the ultrasonic rangefinders missed.

More work can also be done toward integrating computer vision into the robot. The current system using ultrasonic rangefinders works well in testing so far, but it’s always nice to have a second way to avoid hitting obstacles. Also, computer vision can be used to map an environment as the robot moves. Using this map, the robot could make more informed movement decisions. For example, if the robot knows there are clear areas around an object blocking the way to a stronger signal, the robot can try to move through those clear areas instead of only knowing about objects that are in the robot’s immediate path.

Using a phased array antenna to pick up Wi-Fi signal is another extremely interesting path to eventually follow. The prices were high for phased array components, but if prices come down it may be worth pursuing this option. Another difficulty with phased arrays is that they are very difficult to get working properly because of their precise timing requirements. However, a correctly executed phased array would be a wonderful way to make the robot more reliable by reducing the number of moving parts, especially given the fact that the servos that control the directional antenna orientation both broke during development. The speed of the movement process can also be drastically improved because the phased array can very quickly “point” to different areas instead of having to slowly rotate as the robot currently does. This improved scenario could make the movement more fluid as well because the robot could again go back to picking up signal strengths as it moves forward instead of stopping along the way and sweeping a full circle.

A multihop network consisting of multiple mobile repeaters would drastically increase the usable Wi-Fi range of a single router. Instead of having one robot repeat the signal, multiple robots could be chained together to repeat the signal across multiple nodes. Each of those hops adds to the range of the network. This development would include code to make each node communicate together and determine which nodes would be the recipients of the packets they send along the chain.

If a multihop network is established, it could be useful to have all the robots communicate together to come up with a better map of the environment. With more than one robot roaming around, the environment can potentially be mapped quicker because they can all work on that task simultaneously and cover more ground at once.

Another idea is to develop a utility to measure the performance of the robot. This utility would determine how efficient the robot is at finding better signal by keeping track of how much movement the robot did to get to its final destination. At each stage of the signal seeking process, the current best signal strength could also be tracked to try to ensure that the robot consistently improves the signal strength as it moves.

Improving the runtime of the mobile repeater would be a useful accomplishment as well. One way to do this is to alter the algorithms that access the USB ports of the laptop. For example, sending movement commands to the robot less frequently would use less of the laptop's power. In addition, a different solution to the lag problem with the ultrasonic rangefinders could be found. It would save power to not constantly poll the sensors and to simply check their values periodically instead.

Since the laptop battery life is the limiting factor regarding the runtime of the robots, augmenting the laptop power would be another useful option. An external battery could enable the platform to keep running for a longer period of time.

Mobile Repeater

Collision Detection Subgroup

Grover Viernes
Sunil Khanolkar

Rutgers, School of Engineering
Department of Electrical and Computer Engineering
332:428 – Capstone Design Communications Systems, Wireless Communications

Senior Design Project Report

Advisor: Professor Christopher Rose

1. Abstract

The aggregate project goal was to design an intelligent mobile system that seeks the highest wireless signal strength within a given environment. This wireless mobile platform is a composite of three sub-projects: a mobile platform, a communications system and a collision detection system. Due to the uncertainty of its environment, the platform requires information that will allow it to identify objects and maneuver around them. Utilizing a networked ultrasonic system, it is possible to find the distance between obstacles and the platform's current position. This allows the robot to avoid obstacles while searching for wireless signals.

2. Introduction

The wireless mobile platform is designed to autonomously find the direction of strongest signal strength. However, without an effective obstacle avoidance system that will communicate with the signal-searching algorithm, the platform is susceptible to running into obstacle and incurring damage to itself or trapping itself into a corner.

In order to design the system, two types of sensor systems were considered for object detection: computer vision implemented through cameras and sonar implemented through ultra sonic range finders. The goal ultimately with the sensor system was to gather as much information about the environment and then use this data to confirm or deny where the robot would be allowed to go once it found a direction of strongest signal strength.

Computer vision has the advantage of producing more information about its environment but requires more intense data processing. For a camera to discern obstacles it must be first told what an obstacle is. If an image of a room is taken with a table in the middle, the robot must be told how to identify that the gradient shift in the image is a table. If a camera looks at a blank wall, computer vision cannot identify that it is a blank wall unless another object is there. The camera might assume that the data shows open space. Also, the camera would not be able to judge the distance to the wall, since it would not be able to find the two matching points needed for triangulation.

Ultrasonic range finders, on the other hand, offer a simpler solution for implementation. In effect, all the ultrasonic range finder needs to identify an object is the distance from an object to the receiver. Unfortunately, if the angle between the range finder and the object is too steep such that the sound wave produced from the transmitter does not come back directly to the receiver after it hits the object, the distance measured between the sonar and the object will be very inaccurate. Also, depending on the frequency of the sonar, it may be very difficult to detect very small objects. Ultimately, due to their ease of use, ultrasonic range finders were chosen to create the sensor system.

Design began with the analysis and integration of the senscomp/polaroid 6500, a range finder designed for cameras. It can detect objects up to 35 feet, can detect relatively small objects with a frequency of 50kHz, and has a wide beam width which allows it to detect objects in a broad area. Unfortunately, implementing the device and extracting the data was more difficult than expected and very cumbersome. Different microcontrollers were considered for interfacing: Motorola 68HC11, PIC18F series microcontrollers, and the arduino interface. The 68HC11 microcontroller was outdated and the PIC18F series offered a seemingly simpler and more inexpensive solution to the derivatives of the 68HC11 line. Unfortunately, integrating the PIC with the Polaroid 6500 was unsuccessful and progression moved along to the arduino. This time, however, the 2 amp current needed to drive the polaroid 6500 was too high causing its interface with the arduino to reset denying the output data needed to calculate the distance. Eventually, a decision was made to interface the arduino to a different sensor, the Parallax PING.

Interfacing, with the PING was more successful than earlier attempts with interfacing with the previous combinations. The PING sensors were then configured into a networked array that helped the robot maneuver around obstacles.

Before continuing with the design and technical details of the project, an overview behind the physics of sound will be presented.

3. Physics of Sound

Sound is a mechanical, pressure wave that travels through a medium by the vibration of particles: atoms or molecules. These vibrations are caused by the compression of particles together and its antithesis, the rarefaction, or separation, of these particles. Due to its mechanical nature, these compressions and rarefactions allow the particles to move from its rest position and exert a force on the neighboring molecules, consequently passing kinetic energy from particle to particle. In this way, sound from a given source, travels outward radially in all directions from that source.

In order for sound to travel, there must be enough particles for energy to be transferred. Mediums such as space or a vacuum contain very little or no particles, respectively, for sound to exist. Mediums such as solids allow energy to be transferred very easily. This is because sound travels faster in objects whose molecules are closer and are tightly bonded. This allows molecules to pass energy as soon as they receive it. Consequently, after solids, liquids generally are a much faster medium than gases.

Speed of Sound Through a Medium

$V=Cij\rho$

The velocity of sound through a medium is defined as the square root of its elasticity, Cij over its density, ρ. Elasticity is the tendency of a material to change its shape when a force is applied and to return to its original form when that force is removed. The higher the elasticity of an object the more rigid an object; conversely, the lower the elasticity of an object the more flexible an object is. Density is the mass of a substance per volume. If a material is denser because its molecules are larger it will take more energy to move the larger molecule and more time for the sound wave to travel through the medium assuming the elasticity of two materials are equal. Note, that elasticity has a greater effect on the velocity of sound in a medium than density. Thus, since solids are more rigid they are a faster medium than gases, which are more flexible.

For the speed of sound in air, an increase in temperature causes molecules to move faster as noted by the equation below.

$vair=331.3+0.606oC-1*T$

where T is the temperature in degrees Celsius. Hence at the normal room temperature of 20^{o} C, the speed of sound in air is 343 m/s or 1130 ft/s. Humidity and air pressure also have some influence in the velocity in air.

Doppler Effect

Differences in sound are caused by intensity, pitch, and tone. Intensity, measured in decibels, is the amount of sound over a given area. Pitch, as a function of frequency, is a measure of how low or how high something sounds; the higher the frequency, the higher the pitch and the lower the frequency, the lower the pitch. Frequency, measured in hertz, is the number of wavelengths per unit time. One hertz is equal to one cycle of compression and rarefaction. Typically, humans hear frequencies between 20 hertz and 20 kilohertz. Frequencies above the human threshold for hearing are called ultrasonic frequencies.

In a dynamic system where objects are moving around relative to one another, it is necessary to consider the Doppler Effect. The Doppler Effect is the change in frequency of a wave relative to the source of a wave. If you are moving toward a source or if the source is moving closer to you, the sound will seem to have a higher pitch. If you are moving away from a source or if the source is moving away from you, the sound will seem to have a lower pitch. Since the frequency has changed the amount of time for the sound wave to come back to the receiver has also changed. Thus, it is important to include the influence of the Doppler Effect if the design requires dynamic movement throughout the environment.

f=v+vrv+vsf0

f = observed frequency
f_0 = emitted frequency
v_r = velocity of receiver
v_s = velocity of source

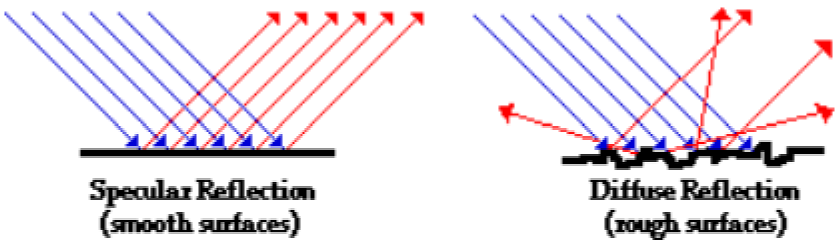
Constructive and Destructive Interference

In the event that multiple waves occur at the same time, the existing waves will interact with each other to form a new wave. Waves aligned in phase so that the compressions are in line with other compression regions and rarefactions are in line with other corresponding rarefactions create constructive interference. Waves that are out of phase create destructive interference. Therefore to avoid interference from each sonar in the design configuration (to be explained later) and consequently distortions in the sonar measurements, each sensor is to operate one after the other, where the next sensor in line waits for the previous sensor to receive back its echo before it sends out its own pulse.

Specular and Diffuse Reflection

Sound waves in gases and liquids are longitudinal waves, that is, the waves travel in the same direction as the particle movement. However, when the medium is a solid they can also be transverse waves. As a transverse wave, the wave moves perpendicular to the direction of the particle movement. As a longitudinal wave, sound waves can be simply represented by rays, which are arrows that indicate the path of the wave.

The surface of an object affects the way a sound wave is reflected off an object. When an object has a specular, or smooth surface, such that each ray of the wave has the same orientation, the incident angle is the same as the reflected angle for the entire group. However, when the surface of an object is a rough surface, each individual ray hits the surface with a different orientation. This causes the sound waves to bounce off or diffuse in different directions. Even though each ray may go off in different directions, each ray still observes the law of reflection relative to the surface they are interacting with.



4. Technical Details of the Parallax's (Ping Ultrasonic Sensor)

The parallax ultrasonic sensor functions by transmitting an ultrasonic burst and by simultaneously providing an output pulse that coincides with the time it takes for the burst echo to be received by the sensor. By measuring the pulse width of the echo pulse, the distance to the object/target can be easily calculated.

a. Key Specifications

The sensor has a 3 pin header that is used to supply ground, power, and signal.

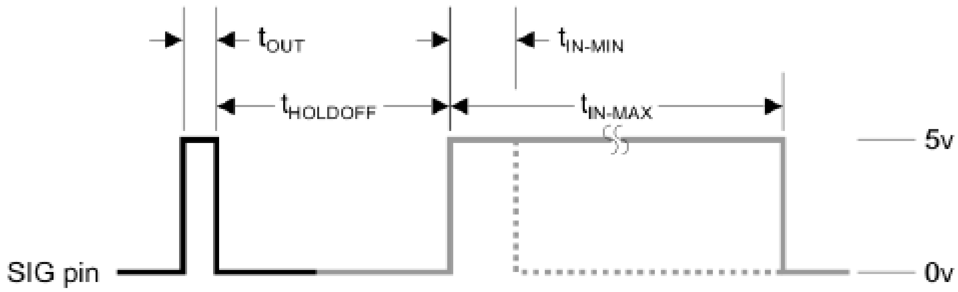
i. Ground: The ground pin should be connected to the common ground used for the entire system.

ii. Power: The ultrasonic sensor requires a supply voltage of 5 volts direct current and a supply current in the range of 30 milliamps to 35 milliamps.

iii. Signal: To transmit an ultrasonic ping, it is required to input a 5 volt pulse to the signal pin. Once the 5 volt pulse has been given, the signal pin will output a pulse that will terminate once an echo has been received.

b. Basic Communication:

The ultrasonic sensor transmits a short 40 kHz burst which is done by use of a microcontroller. The microcontroller will send out a 5 volt pulse to the signal pin. This will cause the ultrasonic sensor to transmit the 40 kHz burst. At that same moment, the ultrasonic sensor will output a 5 volt pulse from the signal pin to the microcontroller. This pulse will ultimately terminate once the ultrasonic sensor receives an echo.

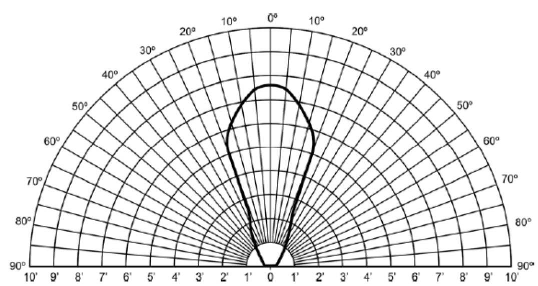


In the figure above, t_{out} represents the pulse width of the 40 kHz burst. $t_{HOLDOFF}$ represents the pulse width of the time after the burst has been sent and before the output pulse has been initialized. t_{IN-MIN} and t_{IN-MAX} define the minimum and maximum pulse widths of the time it takes the echo signal to be received.

c. Horizontal Beam Width

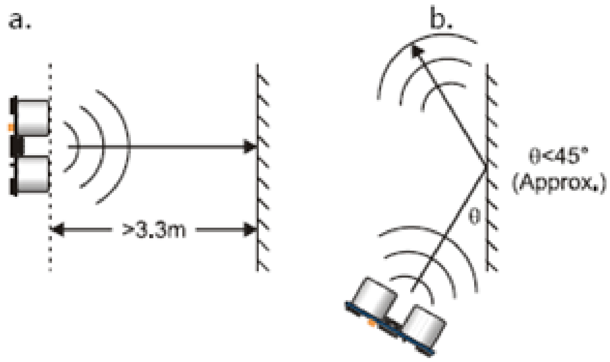
Ultrasonic sensors transmit waves outwards, within a specified beam width. This occurs because the transducer contains a think disk made of crystalline material that has piezoelectric properties, able to generate an electric field in response to mechanical stress. When electricity is applied to such materials, they begin to vibrate. In order to keep these waves from going backward into the transducer and interfering with echo waves, an absorptive material is layered behind the crystal to prevent such from happening. Therefore, these sound waves will only be able to travel outward.

The parallax ping has a beam width of roughly 45 degrees with objects within a foot of the sensor. For objects beyond the range of a foot from the sensor, the beam width of the sensor is approximately 40 degrees.



d. Object Positioning

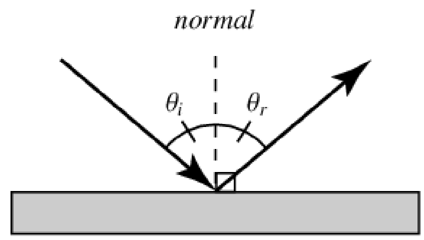
There are several situations where the ultrasonic sensor cannot accurately detect objects as demonstrated in the figure below.



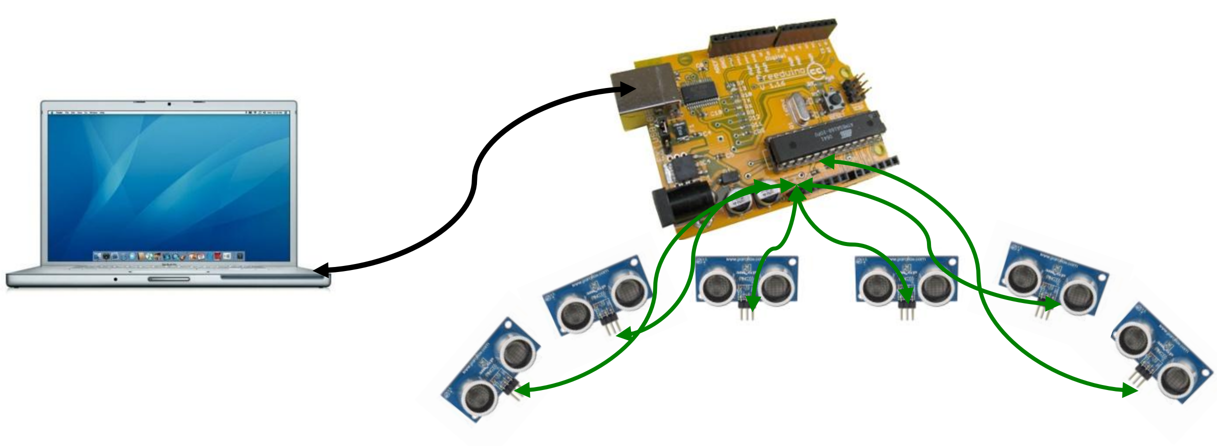
i. In figure a, the object is greater than 10.8 feet (3.3 meters) away from the sensor. This poses as a problem because the ultrasonic burst sent by the sensor decays in energy rather quickly. The farther the wave is sent out, the weaker the energy of the echo signal will be. If the echo signal is too low in energy, the sensor will not be able to receive the echo signal.

ii. In figure b, the wall is at an angle of 45 degrees to the ultrasonic sensor. Due to the law of reflection the angle of reflection θ_r is measured from the reflected sound wave to the surface normal. The angle of incidence θ_i is measured between the sound wave and a line normal to the surface that intersects the surface at the same point as the sound wave.

Since the beam width is roughly 40 degrees, objects that are angled less than 45 degrees from the sensor's straight path will not be able to be detected because the echo waves will all be reflected at an angle of θ_i normal to the object's surface.



5. Overview of the System



Basic Overview: The arduino board sends a pulse of 5 volts to each parallax ping's signal pin. Simultaneously, as the signal pins receive the pulse, the ultrasonic sensor transmits 40 kHz sound waves and the signal pin of the ultrasonic sensor provides an output pulse. This output pulse terminates once the given ultrasonic sensor receives an echo. Once this happens, the arduino board measures the signal pin's output pulse's width and calculates an object's distance. Each individual ultrasonic sensor is polled in sucession, from leftmost to rightmost sensor. In addition, once an object's distance is calculated, that value is instantly sent to the computer's serial port. By the use of C++, the data is stored onto the computer.

The arduino board can be viewed as the centralized unit in this system. Initially it is required that the arduino board is preprogrammed to send 5 volt pulses to the ultrasonic sensor's signal pin, time the signal pins' output pulses, calculate objects' distances, and send data to a computer's serial port. These processes can be viewed as three different operations.

- Sending and receiving 5 volt pulses
  - Voltage pulses are sent and recieved via input/output pins of the arduino board.
- Calculating object's distances
  - Done directly on the arduino board.
- Sending data to the computer
  - Sent from the arduino board to the computer via mini usb cable

Arduino boards are programmed in a language very similar to C++. Common programmable commands are to output 5 volt pulses, receive 5 volts pulses as inputs, print symbols to serial port, etc. Also, similar to how C++ programs have a main(), a program for the arduino board has a void loop(). The code written in the void loop() will loop infinitely. Once the code is uploaded to the arduino board, the program will run for an unlimited amount of time unless either specified in code.

a. Arduino Code Programmed for Project

i. Sending 5 Volt Pulses to Ultrasonic Sensor to Transmit 40 kHz sound waves

As stated earlier, the program written for the project will cause the arduino to send out 5 volt pulses to each ultrasonic sensor's signal pin. Inside void loop(), six ports are specified to send 5 volt pulses to the ultrasonic sensors. For a given sensor, the code used to send 5 volts is the following:

```
pinMode(1, OUTPUT);
digitalWrite(1, LOW);
delayMicroseconds(2);
digitalWrite(1, HIGH);
delayMicroseconds(5);
digitalWrite(1, LOW);
```

Here, the command pinMode(1, OUTPUT) specifies the first input/output port on the arduino board to become an output port. Generally, pinMODE takes two inputs, the first specifying the port number, and the second describing whether to turn the given port into an input or output. The second line of code, digtalWrite(1,LOW) is used to have port 1 output 0 volts. DigitalWrite also takes in two inputs, the first specifying the port number, and the second specifying whether to output 0 volts (LOW) or 5 volts (HIGH). Next, in the code digitalWrite(1,HIGH) is used to output a 5 volts. Then, a delay of 5 microseconds is utilized followed by digitalWrite(1,LOW). This is done to ensure that a 5 volt pulse was output by port 1.

ii. Calculating Objects' Distances

Since 5 volts has been output by the arduino board to port 1, which is connected to the signal pin of an ultrasonic sensor, the ultrasonic sensor will output 40 kHz pulses. Simultaneously, the ultrasonic sensor will output a pulse via its signal pin. This wave terminates once an echo is received by the ultrasonic sensor. By calculating the pulse's width, the object's distance can be calculated. This is done by the arduino by the following code:

```
pinMode(1, INPUT);
Duration1 = pulseIn(1, HIGH);
Inches1 = microsecondsToInches(Duration1);

…
long microsecondsToInches(long microseconds)
{
return microseconds / 74 / 2;
}
```

In this code, pinMode(1,INPUT), is first used. This switches port 1 on the arduino to an input port. This is done since, at this time, the ultrasonic sensor outputs a pulse wave to the arduino. PulseIn(1,HIGH) is then used. This function takes two inputs, the first being the port number, and the second either HIGH or LOW. The way PulseIn works is that when called, a timer initializes and will stop once either an incoming pulse goes from high to low or low to high depending on the second input. If HIGH is specified, the timer will terminate once the incoming wave goes low. Similarly, if LOW is specified, the timer will terminate once the incoming wave goes high. In this case, HIGH is utilized since the ultrasonic sensors outputs 5 volts until an echo is received.

Lastly, the user created function microsecondsToInches is used to calculate an object's distance. This function converts microseconds to inches. Since sound travels at 1130 feet per second, from calculation, it can be found that there are 73.746 microseconds per inch. Thus, once an ultrasonic's output pulse's width has been measured, by dividing the pulse's width by 74, the distance traveled by the ultrasonic burst can be obtained. By taking this value and diving by 2, the object's distance will then be found. If an echo is never recived by the ultrasoinc sensor, the object's distance for that sensor will be its maximum value, 144 inches.

iii. Communication with Computer's Serial Port via Arduino Board

Communication with a computer's serial port via arduino board is a simple process. By utilizing a few different commands, proper communication between the arduino board and the computer's serial port can be established. Initially, outside void loop(), the baud rate is specified by the function Serial.begin(9600). Baud rate, also known as symbol rate, is the number of symbol changes made to a transmission medium per second using either a digitally modulated signal or a line code. In the code written for the project, the baud rate is set to 9600, which is considered to be the standard baud rate for data transfer via serial by the arduino. Since the data rate is relatively low compared to the baud rate, there will be no issue of lag between the data calcualted by the arduino board and data sent to the computer's serial port.

For each ultrasonic sensor, the code `Serial.print(Inches1)` is used to print the object's distance to the computer's serial port. For the first five ultrasonic sensors, after each value is printed to the serial port, the next line of code is `Serial.print(",")`. After the sixth ultrasonic sensor's value is printed, the following code is `Serial.println()`. This function is equivalent to \n in C++. An example output to the arduino code would be the follow:

```
14,23,12,53,32,5
12,22,12,51,35,4
134,110, 23, 12,54,2
```

b. Accessing the Serial Port via C++ and Storing Data

As data is being sent from the arduino board to the computer's serial port via mini usb cable, the C++ program opens the device, in this case a com port, to receive the data and stores the objects' distances of each sensor in six separate arrays. The C++ program is written such that the past 20 object distances are stored for each sensor.

The basis of the C++ program is the scan() function. The scan() function stores incoming data in a character array. The different types of characters (symbols) that can be sent from the arduino board to the serial port are integer values, commas, and newline. The scan() function waits for six values, each separated by a comma and the last value followed by a newline. Once this occurs, each of the six values will be stored in the six different arrays with each array corresponding to a different sensor value. Each new line specifies a new iteration for storing sensor values. A technique known as water falling is used where the newest iteration of values will be stored, and the oldest (21^{th} iteration) will be discarded. This is done by shifting each array value for each sensor consistently for each iteration.

The arduino board is programmed to poll each sensor for an unlimited amount of time. Similarly the arduino board is also programmed to send data constantly to the computer's serial port. Thus, it is up to the user to use the function scan() within the given C++ code, to obtain and store objects' distances when it is desired.

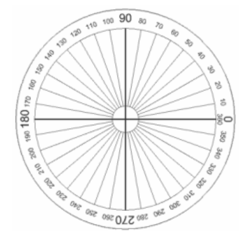
6. Configurations of the Ultrasonic Sensors

The main purpose of the robot is to find the best signal strength. The ultrasonic sensors aid the robot to avoiding objects. The ideal design is to have the robot search for the best signal strength while the ultrasonic sensors accurately notify the robot when the robot is too close to an object. Therefore, it is inefficient for the robot to move forward or rotate for the purpose of just verifying if an object exists or not. Thus, the configurations of the sensors were based off the robot's limitations, such as movement, rotation, memory, its circular platform, etc.

a. Analyzing Robot's Limitations

i. Circular Platform

Since the robot has a circular platform, sensors were placed at various degree marks to see what kind of objects would be able to be seen by the robot. It was found that objects that were angled too steeply were unable to be recognized by ultrasonic setup. Also, it is clear that the only difference in placing one sensor on the circumference of the surface verses multiple sensors on the circumference is the time required to rotate that platform. Simply put, the difference between rotating one sensor 360 degrees and six equally spread out sensors on the circumference is time (1/6 of the time required).



Robot's circular platform

ii. Forward Movement

The robot was limited to only move forward. This encouraged the design to only consider the robot's frontal vision. Objects that are angled in front of the robot pose as an issue since the ultrasonic sensors will not be able to receive an echo. In addition, if the robot is moving at an angle towards a wall, sensors attached in the front of the robot will not be able to recognize the wall.

Design with multiple sensors facing forward

iii. Rotational Speed

To fix this design, by simply rotating the robot several degrees clockwise and counter-clockwise, the robot can check its surroundings every few seconds. This eliminates the problem of the robot moving into walls or objects on its side when moving at an angle. This, however, creates new problems for the robot. With this design the robot is required to constantly check its surroundings. This ultimately will become time consuming. Considering all the other processes going on with the robot, the task of frequently rotating the robot for object detection seems inefficient. Also, this design would require the robot to keep track of its previous positions.

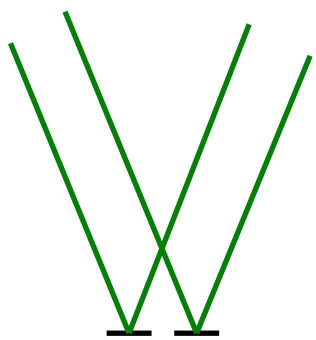
iv. Memory

Memory also plays a huge factor when considering a design for the ultrasonic sensors. Several issues may come into play. For example, if the robot is to rotate every few seconds, previous positions must be saved in memory. Accessing the robot's previous would then cause the robot to pause and waste time. Every rotation would also require that the robot also remember the corresponding sensor values. It was found, through testing, that accessing past data may lag the robot's processes, specifically the data transfer between the arduino board and the computer's serial port.

b. Design Solution

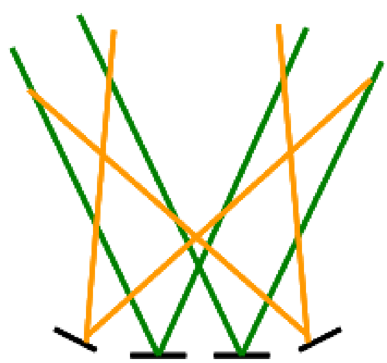
As stated earlier, the ideal design for the ultrasonic sensors would be for the robot to never have to rotate, for obstacle avoidance purposes, while still being able to receive accurate object distances. From a better understanding of the ultrasonic sensor's beam width, a feasible design solution was able to be devised. Initially, a test was done to understand what areas two ultrasonic sensors could see when placed two inches apart.

Two ultrasonic sensors' beam widths



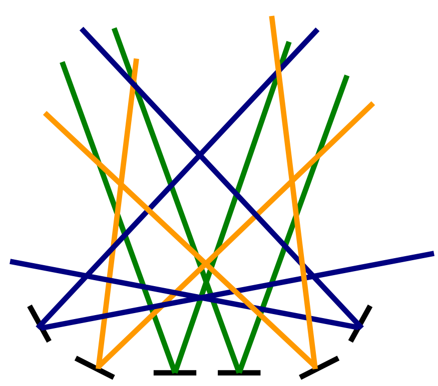
Clearly, if two sensors are placed next to each other in this way, objects to the left side and right would go undetected by the robot. It was found, from trial and error, that if an ultrasonic sensor was placed to the right of an ultrasonic sensor but rotated 20 degrees counter-clockwise, the robot would be able to gain more vision of objects located to the left side and right side of the robot.

Four ultrasonic sensors' beam widths



Furthermore, by adding another ultrasonic sensor, rotated 20 degrees counter-clockwise from the rightmost sensor, even more vision for the left and right sides were added for the robot.

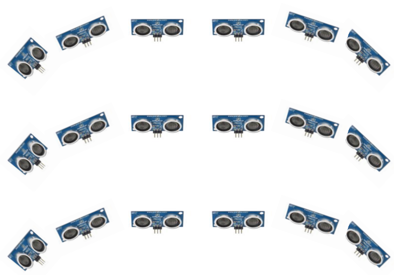
Final Design
Six ultrasonic sensors' beam widths



(Adding even more sensors to this design, rotated 20 more degrees would offer no more vision).

This design was found to be the most efficient design compared to the previous designs. By using this design, the robot would not have to rotate at all to see its left and right side. In addition, the robot would also be able to see angled objects. The most appealing factor of this design is that the robot gains a very high accuracy about close by objects in its surroundings without having to rotate or move forward. This eliminates the possibility of having the robot remember its previous locations. By using this design, the robot can entirely focus on finding the signal strength and be notified instantaneously when the robot is too close to an object.

7. Future Work



Despite, the effectiveness of the sonar system to avoid obstacles, it is by all means far from efficient. In its present form, the sensor network is sluggish and its level of intelligence is below the competency that was desired. Its current design contains discrepancies for objects of varying shapes and sizes to be unobserved and without a system to keep track of places it has gone before and obstacles it has encountered, the robot would be unable to optimize its movement algorithm.

Currently, the sensor network exists on the top of the robot platform. Since it is only on one plane it fails to detect objects that are farther above and below the sensor network. For example, if the robot moves in the direction of a table it will fail to detect the top of the table if the top is higher than what the sensor network can detect. This leaves the antennae very vulnerable to damage. If a soccer ball or an object shorter than the height of the platform comes into the robot's path, the robot is susceptible to collisions. Although the current placement of the sensors allows it to move around walls, practical environments are much more dynamic containing obstacles of various shapes and sizes.

One solution is to place multiple sensor networks along different planes of the platform. Unfortunately, this solution has discrepancies in and of itself. It is not possible to place ultrasonic range finders too close to the floor, otherwise, the bounce paths coming from the floor would distort the readings. Also, at a certain point it is too cumbersome and too inefficient in terms of cost to have multiple sensors on different planes. A more efficient design, however, would be to incorporate computer vision. Due to the wider field of view that computer vision has, it is possible to observe objects even on the floor without the threat of bounce paths interfering. Now objects oriented with steeper angles relative to the ultrasonic sensors can now be detected by computer vision. Conversely, situations where computer vision does not have enough information to discern objects, such as a blank wall, ultrasonic sensors should be able to identify the obstacle. In this way, the strengths of one system can complement the other.

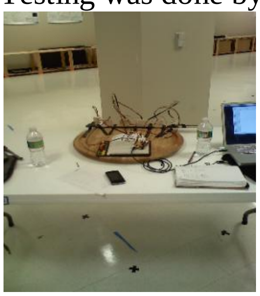
With a more efficient sensor system, it is now possible to create a map of the environment. Since the robot can keep track of places it has been and obstacles it has encountered, it can optimize its movement by making decisions before it reaches the vicinity of a known obstacle. Although, it has a rough sketch of the environment, updates must constantly be made in order to account for uncertainties and/or sudden changes. This can be done by constant polling of information of sensors as the robot moves. However, this method requires the consideration of the Doppler Effect to take into account the time it takes to receive back an echo while the robot is moving.

The current design is too inefficient for real world applications due to the dynamic nature of the robot's expected environment. Obstacles will come in many different shapes and sizes and the robot must be poised to maneuver around them all. It is hopeful that the integration of computer vision and a mapping system will increase the speed of the robot and make it more adverse to and more aware of objects in its path.

8. Appendices

DATA TESTING

Testing was done by placing the system on a turntable at WINLAB up against a 3 foot pillar.



a. Sensor Design Facing Wall Straight 2 Feet Away

i. 0 degrees turned

126	24	24	24	24	57
126	24	24	24	24	59
126	24	24	24	24	59
126	24	24	24	24	129
126	24	24	24	24	129
126	24	24	24	24	129
126	24	24	24	24	129
126	24	24	24	24	129
126	24	24	24	24	129
126	24	24	24	24	58
126	24	24	24	24	57
126	24	24	24	24	129
126	24	24	24	24	129
126	24	24	24	24	129
126	24	24	24	24	129
126	24	24	24	24	129
126	24	24	24	24	59
126	24	24	24	24	129
126	24	24	24	24	59
126	24	24	24	24	129

ii. 27.5 degrees turned counter clockwise

26	26	25	24	130	129
26	26	25	24	130	129
26	26	25	24	130	129
26	26	25	24	130	129
26	26	25	24	130	129
26	26	25	24	130	129
26	26	25	24	130	129
26	26	25	24	130	129
26	26	25	24	130	129
26	26	25	24	130	129
26	27	25	24	130	129
26	27	25	24	130	129
26	27	25	24	130	129
26	27	25	24	130	129
26	26	26	24	130	129
26	26	25	24	130	129
26	27	25	24	130	129
26	27	25	24	130	129
26	27	25	24	130	129
26	27	25	24	130	129

iii. 45 degrees turned counter clockwise

31	30	143	130	130	129
31	30	143	130	130	129
31	30	143	130	130	129
31	30	143	130	130	129
31	30	143	130	130	129
31	30	143	130	130	129
31	30	143	130	130	129
31	30	143	130	130	129
31	30	143	130	130	129
31	30	143	130	130	129
31	30	143	130	130	129
31	30	143	130	130	129
31	30	143	130	130	129
31	30	143	130	130	129
31	30	143	130	130	129
31	30	143	130	130	129
31	30	143	130	130	129
31	30	143	130	130	129
31	30	143	130	130	129
31	30	143	130	130	129

iv. 27.5 degrees turned clockwise

126	132	24	25	27	26
126	132	24	25	27	26
126	132	24	25	27	26
126	132	24	25	27	26
126	132	24	25	27	26
126	132	24	25	27	26
126	132	24	25	27	26
126	132	24	25	27	26
126	132	24	25	27	26
126	132	24	25	27	26
126	132	24	25	27	26
126	132	24	25	27	26
126	132	24	25	27	26
126	132	24	25	27	26
126	132	24	25	27	26
126	132	24	25	27	26
126	132	24	25	27	26
126	132	24	25	27	26
126	132	24	25	27	26
126	132	24	25	27	26

v. 45 degrees turned clockwise

126	132	24	25	27	26
126	132	24	25	27	26
126	132	24	25	27	26
126	132	143	130	30	30
126	132	143	130	30	31
126	132	143	130	30	31
126	132	143	130	30	31
126	132	143	130	30	31
126	132	143	130	30	31
126	132	143	130	30	30
126	132	143	130	30	30
126	132	143	130	30	31
126	132	143	130	30	31
126	132	143	130	30	30
126	132	143	130	30	30
126	132	143	130	30	31
126	132	143	130	30	31
126	132	143	130	30	30
126	132	143	130	30	30
126	132	143	130	30	31

b. Sensor Design Facing Wall Straight 14 Inches Away

i. 0 degrees turned

51	14	14	14	14	69
51	14	14	14	14	42
51	14	14	14	14	42
51	14	14	14	14	42
51	14	14	14	14	42
51	14	14	14	14	43
51	14	14	14	14	42
51	14	14	14	14	42
51	14	14	14	14	42
51	14	14	14	14	42
51	14	14	14	14	129
51	14	14	14	14	42
51	14	14	14	14	129
51	14	14	14	14	46
51	14	14	14	14	68
51	14	14	14	14	68
51	14	14	14	14	68
51	14	14	14	14	68
51	14	14	14	14	68
51	14	14	14	14	68

ii. 27.5 degrees turned counter clockwise

15	16	15	13	31	129
15	16	15	13	31	129
15	16	15	13	31	129
15	16	15	13	31	130
15	16	15	13	31	130
15	16	15	13	31	130
15	16	15	13	31	130
15	16	15	13	31	130
15	16	15	13	31	130
15	16	15	13	31	130
15	16	15	13	31	130
15	16	15	13	31	130
15	16	15	13	31	130
15	16	15	13	31	130
15	16	15	13	31	130
15	16	15	13	31	130
15	16	15	13	31	130
15	16	15	13	31	130
15	16	15	13	31	130
15	16	15	13	31	130

iii. 45 degrees turned counter clockwise

20	19	143	130	130	130
20	19	143	130	130	130
20	19	143	130	130	130
20	19	143	130	130	130
20	19	143	130	130	129
20	19	143	130	130	129
20	19	143	130	130	129
20	19	143	130	130	129
20	19	143	130	130	129
20	19	143	130	130	129
20	19	143	130	130	129
20	19	143	130	130	129
20	19	143	130	130	129
20	19	143	130	130	129
20	19	143	130	130	129
20	19	143	130	130	129
20	19	143	130	130	129
20	19	143	130	130	129
20	19	143	130	130	129
20	19	143	130	130	129

iv. 27.5 degrees turned clockwise

126	32	13	15	16	15
126	32	13	15	16	15
126	32	14	15	16	15
121	32	14	15	16	15
126	32	14	15	16	15
126	32	14	15	16	15
126	32	14	15	16	15
126	32	14	15	16	15
126	32	14	15	16	15
126	32	14	15	16	15
126	32	14	15	16	15
126	32	14	15	16	15
126	32	14	15	16	15
126	32	14	15	16	15
126	32	14	15	16	15
126	32	14	15	16	15
126	32	14	15	16	15
126	32	14	15	16	15
126	32	14	15	16	15
126	32	14	15	16	15

v. 45 degrees turned clockwise

126	132	143	130	20	20
126	132	143	130	20	20
126	132	143	130	20	20
126	132	143	130	20	20
126	132	143	130	20	20
126	132	143	130	20	20
126	132	143	130	20	20
126	132	143	130	20	20
126	132	143	130	20	20
126	132	143	130	20	20
126	132	143	130	20	20
126	132	143	130	20	20
126	132	143	130	20	20
126	132	143	130	20	20
126	132	143	130	20	20
126	132	143	130	20	20
126	132	143	130	20	20
126	132	143	130	20	20
126	132	143	130	20	20
126	132	143	130	20	20

c. Sensor Design Corner of Wall Facing

i. 0 degrees turned

126	132	143	130	130	130
126	132	143	130	130	130
126	132	143	130	130	130
126	132	143	130	130	130
126	132	143	130	130	130
126	132	143	130	130	130
126	132	143	130	130	130
126	132	143	130	130	130
126	132	143	130	130	130
126	132	143	130	130	130
126	132	143	130	130	130
126	132	143	130	130	130
126	132	143	130	130	130
126	132	143	130	130	130
126	132	143	130	130	130
126	132	143	130	130	130
126	132	143	130	130	130
126	132	143	130	130	130
126	132	143	130	130	130
126	132	143	130	130	130

i. 27.5 degrees turned counter clockwise

126	132	143	130	130	130
126	132	143	130	130	130
126	132	143	130	130	130
126	132	143	130	130	130
126	132	143	130	130	130
126	132	143	130	130	130
126	132	143	130	130	130
126	132	143	130	130	130
126	132	143	130	130	130
126	132	143	130	130	130
126	132	143	130	130	130
126	132	143	130	130	130
126	132	143	130	130	130
126	132	143	130	130	130
126	132	143	130	130	130
126	132	143	130	130	130
126	132	143	130	130	130
126	132	143	130	130	130
126	132	143	130	130	130
126	132	143	130	130	130

ii. 45 degrees turned counter clockwise

126	132	143	130	130	130
126	132	143	130	130	130
126	132	143	126	132	143
126	132	143	130	130	130
126	132	143	130	130	130
126	132	143	130	130	130
126	132	143	130	130	130
126	132	143	130	130	130
126	132	143	130	130	130
126	132	143	130	130	130
126	132	143	130	130	130
126	132	143	130	130	130
126	132	143	130	130	130
126	132	143	130	130	130
126	132	143	130	130	130
126	132	143	130	130	130
126	132	143	130	130	130
126	132	143	130	130	130
126	132	143	130	130	130
126	132	143	130	130	130

Arduino Code

```
/Position is in terms of the robot itself
const int firstPING = 7;
const int secondPING = 6;
const int thirdPING = 5;
const int fourthPING = 4;
const int fivePING = 3;
const int sixPING = 2;
void setup()
{
  // set communication data rate to 9600 bits per second
  Serial.begin(9600);
}

// 1) was recommended that you pass a quick square pulse to ensure that when the signal
// looks for an input it will ensure that it will receive a clean signal
// 2) pulseIN finds the time duration it takes for when signal is high until it receives something that
// make it return low. In this case "something" is echo.
// 3) convert time (ms) to distance (in)
void loop()
{
  long durONE, durTWO, durTHREE, durFOUR, durFIVE, durSIX;
  long inchONE, inchTWO, inchTHREE, inchFOUR, inchFIVE, inchSIX;

  //Set firstPING quickly to HIGH to ensure a clean read for pulseIN
  pinMode(firstPING, OUTPUT);
  digitalWrite(firstPING, LOW);
  delayMicroseconds(2);

  digitalWrite(firstPING, HIGH);
  delayMicroseconds(5);
  digitalWrite(firstPING, LOW);

  //measure pulseIN for the LEFT sonar
  //pulseIN measures the time taken from a pulse to a ECHO
  pinMode(firstPING, INPUT);
  durONE = pulseIn(firstPING, HIGH);
  inchONE = timetoDIST(durONE);
  Serial.print(inchONE);

  delay(100);
  Serial.print(",");

  //Set secondPING quickly to HIGH to ensure a clean read for pulseIN
  pinMode(secondPING, OUTPUT);
  digitalWrite(secondPING, LOW);
  delayMicroseconds(2);

  digitalWrite(secondPING, HIGH);
  delayMicroseconds(5);
  digitalWrite(secondPING, LOW);

  //measure pulseIN for the RIGHT sonar
  pinMode(secondPING, INPUT);
```

```
durTWO = pulseIn(secondPING, HIGH);
inchTWO = timetoDIST(durTWO);
Serial.print(inchTWO);

delay(100);
Serial.print(",");

pinMode(thirdPING, OUTPUT);
digitalWrite(thirdPING, LOW);
delayMicroseconds(2);

digitalWrite(thirdPING, HIGH);
delayMicroseconds(5);
digitalWrite(thirdPING, LOW);

//measure pulseIN for the LEFT sonar
//pulseIN measures the time taken from a pulse to a ECHO
pinMode(thirdPING, INPUT);
durTHREE = pulseIn(thirdPING, HIGH);
inchTHREE = timetoDIST(durTHREE);
Serial.print(inchTHREE);

delay(100);
Serial.print(",");

pinMode(fourthPING, OUTPUT);
digitalWrite(fourthPING, LOW);
delayMicroseconds(2);

digitalWrite(fourthPING, HIGH);
delayMicroseconds(5);
digitalWrite(fourthPING, LOW);

//measure pulseIN for the LEFT sonar
//pulseIN measures the time taken from a pulse to a ECHO
pinMode(fourthPING, INPUT);
durFOUR = pulseIn(fourthPING, HIGH);
inchFOUR = timetoDIST(durFOUR);
Serial.print(inchFOUR);

delay(100);
Serial.print(",");

pinMode(fivePING, OUTPUT);
digitalWrite(fivePING, LOW);
delayMicroseconds(2);

digitalWrite(fivePING, HIGH);
delayMicroseconds(5);
digitalWrite(fivePING, LOW);

//measure pulseIN for the LEFT sonar
//pulseIN measures the time taken from a pulse to a ECHO
pinMode(fivePING, INPUT);
durFIVE = pulseIn(fivePING, HIGH);
inchFIVE = timetoDIST(durFIVE);
```

```
  Serial.print(inchFIVE);

  delay(100);
  Serial.print(",");

  pinMode(sixPING, OUTPUT);
  digitalWrite(sixPING, LOW);
  delayMicroseconds(2);

  digitalWrite(sixPING, HIGH);
  delayMicroseconds(5);
  digitalWrite(sixPING, LOW);

  //measure pulseIN for the LEFT sonar
  //pulseIN measures the time taken from a pulse to a ECHO
  pinMode(sixPING, INPUT);
  durSIX = pulseIn(sixPING, HIGH);
  inchSIX = timetoDIST(durSIX);
  Serial.print(inchSIX);

  Serial.println();

  delay(200);
}

//converts time (ms) to distance (in)
long timetoDIST(long ms)
{
  return ms / 74 / 2;
}
```

SerialReader.h

```
#ifndef SERIALREADER_H
#define SERIALREADER_H

#include <termios.h>

class SerialReader
{
	public:
		int sensors[6][20];
		SerialReader(char* device);
		~SerialReader();
		bool openDevice();
		void closeDevice();
		void readLine();
		void readLines(int howMany);
		void scan();
	private:
		char* device;
		int fd;
		struct termios oldtio,newtio;

};

#endif
```

SerialReader.cpp

```
#include <termios.h>
#include <stdio.h>
#include <unistd.h>
#include <fcntl.h>
#include <sys/signal.h>
#include <sys/types.h>
#include <stdlib.h>
#include <string.h>
#include <iostream.h>
#include <fstream.h>
#include <iomanip.h>
#include "SerialReader.h"
#define _POSIX_SOURCE 1
#define BAUDRATE B9600

using namespace std;

int main() {
	int numSENSORS = 6;
	int numLINES = 20;
	int fileNUM = 1;

	SerialReader test = SerialReader("/dev/tty.usbserial-A6007Anv");
	if(!test.openDevice())
	{
		printf("Couldn't open device\n");
		return 0;
	}

	test.scan();
```

```
	for(int i = 0; i < 6; i++)
	{
		for(int j = 0; j < 5; j++)
	    {
			if (test.sensors[i][j] < 25)
			{
				return 0;
			}
	    }
	}
	test.closeDevice();

	return 0;
}

SerialReader::SerialReader(char* dev)
{
	device = (char*)malloc(sizeof(char)*(strlen(dev)+1));
	strcpy(device, dev);
}

SerialReader::~SerialReader()
{
	if(device != NULL)
		free(device);
}

bool SerialReader::openDevice()
{
	fd = open(device,  O_RDWR | O_NOCTTY | O_NONBLOCK);
	if(fd < 0)
	{
		perror("Could not open device.");
		return false;
	}
	tcgetattr(fd,&oldtio);
	newtio = oldtio;
	cfsetispeed(&newtio, B9600);
	cfsetospeed(&newtio, B9600);
	newtio.c_lflag &= ~(ICANON | ECHO | ECHOE | ISIG);
	newtio.c_oflag &= ~OPOST;
	newtio.c_cflag &= ~PARENB;
	newtio.c_cflag &= ~CSTOPB;
	newtio.c_cflag &= ~CSIZE;
	newtio.c_cflag |= CS8;
	newtio.c_cflag &= ~CRTSCTS;
	newtio.c_cflag |= CREAD | CLOCAL;
	newtio.c_iflag &= ~(IXON | IXOFF | IXANY);

	newtio.c_cc[VMIN]=0;
	newtio.c_cc[VTIME]=20;

	tcflush(fd, TCIFLUSH);
	tcsetattr(fd,TCSANOW,&newtio);

	return true;
```

```
}

void SerialReader::closeDevice()
{
	tcsetattr(fd,TCSANOW,&oldtio);
	close(fd);
}
//reads 20 lines of data from the 6 sonars, stores in 2D array
//data is read as an array of char and then stored as an int
void SerialReader::scan()
{
	int res, i =0, j = -1;
	char buffer_string[1024];
	char buffer[1];

	while(j < 20)
	{
		do {
			res = read(fd, buffer, 1);
			if(res <= 0) {
				usleep(500);
				continue;
			}
			buffer_string[i] = buffer[0];
			i++;
		} while(buffer[0] != '\n' && i < 1024);


		if(j == 5)
		{
			//
			for(int l = 0; l < 6; l++)
			{
				for(int k = 0; k < 19; k++)
					sensors[l][k] = sensors[l][k+1];
			}
		}
		else
		{
			j++;
		}
		printf("%s", buffer_string);

	if(sscanf(buffer_string,"%d,%d,%d,%d,%d,%d",&(sensors[0][j]),&(se
nsors
[1][j]),&(sensors[2][j]),&(sensors[3][j]),&(sensors[4][j]),&(sensors[5]
[j])) < 6) j--;
		memset(buffer_string,0,i);
		i = 0;
	}
}
```

9. References

http://www.ndt-ed.org/EducationResources/CommunityCollege/Ultrasonics/Introduction/description.htm

http://www.parallax.com/tabid/768/ProductID/92/Default.aspx

http://www.ladyada.net/learn/arduino/help.html

http://hyperphysics.phy-astr.gsu.edu/hbase/HFrame.html