Amateur Satellite Communication System

Final Version 4.0

Bo Jin *30jin@yahoo.com*
Junsung Cho *chojs@cs.utah.edu*
Vinh Pham *vpham@cs.utah.edu*
Suresh Subasinghe *subasing@eng.utah.edu*

A brief History of Amateur Satellites

The first amateur satellite, OSCAR I was launched December 12, 1961 piggyback with Discover 36, a United States Air Force satellite. Launch of OSCAR I led to the creation of The Amateur Satellite Corporation (AMSAT) in 1969. However the first remotely controlled amateur satellite was OSCAR 5 which was developed by electrical engineering students at the University of Melbourne in Australia. Amateur satellites continue to be designed and built in countries around the world. There are several new amateur satellites that will soon join the ranks above. Latest being the Amateur Radio onboard the International Space Station (ARISS) in 2004.

I. Objective Overview

Design an Antenna/Transmitter module to communicate with low orbital amateur Satellites in the sky. The purpose of this project is to allow users to transfer data using amateur Satellites as a medium. The complete system will contain following components:

- Two antennas for bidirectional communications.
- Antenna Position control system.
- UHF/VHF radio transmitter/receiver.
- Modem/Terminal Node Controller.
- Software to track and send/receive satellites data.

We will be using data obtained from NASA's J-Track database to locate different satellites at any given time. With J-Track information and Antenna control system, our software will track selected satellites in the sky. Once a communication link is established, the modem and the radio will be used to transfer data from the computer to the orbiting satellite. Users can then transfer/receive data from a satellite using GUI based software.

One or more satellites will be used ranging from AMSAT to OSCAR series. Our design will only allow communication with one satellite per session. The user will be able to switch satellites if they are within the range. Two antennas will be built for sending and receiving data. Modem circuit and TNC box will allow communication between the UHF/VHF radio and the computer. We will build both units using components available from the department's laboratory and electronic parts vendors. We have already acquired a UHF/VHF radio unit for use in this project. We will also build the rotator circuit for controlling the antenna position. In order to obtain motion in xy-plane an off the shelf rotator will be used. If we are unable to find a rotator, we will build a rotator using two motors.

A software package that will track the satellite will be implemented in GUI based environment using .NET architecture. Another GUI based package will be for user to interact with the antenna control system. A third software package will be available for sending/receiving data from the selected satellite. This software will also let the user select a satellite from a list of available operational satellites using information from J-Track database. Depending on the time available we will extend the user interface to the Internet, allowing users to login remotely and use the system to transmit data. This will require developing server software to handle remote control thus allowing users to communicate with the closest satellite. Security is a concern; however we will handle this from the server side. The system can be further developed in terms of security if this system is to be used as a stand alone node.

II. The Process of Satellite Digital Communication

To implement store-and-forward communications and use space-borne resources efficiently, the spacecraft needs lots of brainpower and it must be in control of all links. If a ground station wants to upload a file, or download a file, the ground station must first register with the satellite by sending a short message using UDP or satellite specific protocol. If this request is received, the satellite acknowledges and assigns the ground station to a queue where it awaits its turn to either downlink or uplink. The satellite then works its way through the queues one station at a time, allocating each station a small time slice to accomplish the task. If a station doesn't complete its task in the assigned time, the request goes to the bottom of the queue to wait its next turn, during which it can attempt to complete the task. This process continues until either a timeout or the completion of data transfer.

The messages are broken down into small packets, the packets then get assigned identification tags. If the satellite or ground station misses part of a message, a fill can be requested. Note that the ground station operator never directly activates the transmitter. Directions are given to the computer, which can request that a specific file be downloaded. From this point on the ground station is under the control of ground station software that is responding to on site conditions and satellite software. The ground station software decides when to transmit the request to enter the downlink queue, to avoid colliding with other requests. The ground station software recognizes if the initial request is unsuccessful, it then decides when to try again. The ground station software captures the packets containing the message. This process may take several rotations through the queue and additional messages may have to be uploaded to the satellite to request fills for holes (packets that have been missed). Finally, the computer strips off all the packaging material (headers, packet identification labels, etc.) and reconstructs the packets in the correct order to produce the desired file. During all these transmission and reception periods, which may take a few minutes, the operator at the ground station is just an observer watching these steps take place on the computer screen.

III. Antenna for Space Communication

Generally, there are five kinds of antennas; Yagi, quad, Quagi, loop Yagi, and delta loop. This project will use the Yagi style. The Yagi has a number of positive attributes including its simple structure, light weight, and low wind load for a given gain. As a result, most commercial manufacturers favor it over other types of beams at 2m and 70cm. Until 1980, most published Yagi designs and operated satisfactorily only over a very narrow bandwidth (often 1-2% of operating frequency). As a result, home-brews antennas had to be extremely careful to replicate all dimensions and materials exactly as described to duplicate the performance of the original. The narrow bandwidth also made the antenna very susceptible to detuning affects from the mounting, nearby antennas, and environmental condition. Back in the 1970s attempts to conquer the bandwidth limitation focused on the use of log-periodic feeds. This approach was successful but the extra elements added to weight and wind load.

However, in the late 1980s, Steve Powlishen, demonstrated that it is possible to design high performance arrays with bandwidths the order of 8% having extremely clean patterns (very low power in side lobes) with computer analysis tools.

This project requires the use of two antennas for communication with satellites at two different frequencies. A 2m Yagi will be used to downlink at 146MHz while another 70cm Yagi will be used to uplink at 435MHz. They will be mounted on same stand. In order to obtain precise measurements for the antenna design we need to use Computer

Aided Design (CAD) tools for simulation.

IV. TNC and Modem

Two components are required at minimum for interaction between a satellite and a computer, TNC (Terminal Node Controller) and a modem. The TNC and modem can be separate packages or integrated into a single box. They usually contain multiple digital-to-analog and analog-to-digital converters. TNC and modem are connected to the computer through an RS-232 serial link and to the transmitter, receiver and rotator signal through two signal lines and several control lines. The signal lines consist of an audio signal from modem to transmitter called TD (transmit data) and an audio signal from receiver to modem called RD (receive data). The control lines include transmitter PTT (push-to-talk), receiver AFC (automatic frequency control), transmitter AFC, antenna rotator azimuth and elevation. Depending on the satellites, they operate at two speeds; 1200 and 9600 Baud Modem. We will proceeds with constructing a 9600 baud rate Modem and if time permits, we will construct the second 1200 baud rate Modem.

V. Tracking Program

Many stations require a tracking program that automatically aims the antenna, compensates for frequency offsets and decodes telemetry. Some software can even be configured to turn a station on/off at the correct time and automatically collect data. However, a tracking program usually consists of eight common modules. In our design we will design software that implements the following eight modules.

- Distance and Bearing (DISTBEAR)
- Spiderweb construction data (SWEB)
- Ground Track data for circular orbit (GTCIRC)
- Ground Track data for elliptical orbit (GTEL)
- Average daily access time for circular orbit (ADATCIRC)
- Average daily access time for elliptical orbit (ADATEL)
- Basic Orbit Data (SATPFL)
- Antenna Control System (ACS)

Antenna System

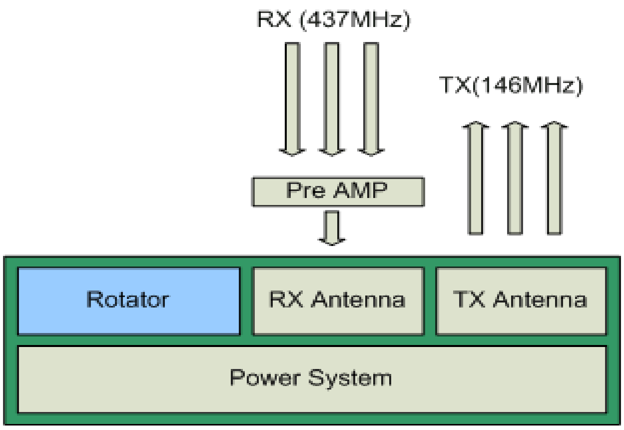


Fig 1: Overall Antenna System Design

The Antenna system is determined by five main characteristics

1. *Directional properties (gain and pattern)* − we will be using directional antenna. This required less power and better range of communication.
2. *Transmission and reception properties* − the transmission gain pattern of an antenna is the same as its receptions pattern. In other words, the efficiency of an antenna is same in both directions of communication.
3. *Antenna efficiency* − Antenna with 100% efficiency means the antenna radiates all power reaching its input terminal. Reflected transmission of greater then 20% (only 80% is transmitted) will cause possible equipment malfunction.
4. *Polarization* − Radio waves consist of electric and magnetic fields, both of which are always present and inseparable. When a radio wave passes a point in space, the electric field at that point varies cyclically at the frequency of the wave. When we discuss the '*polarization of a radio wave'* we're focusing on how the electric field varies. Most amateur antennas are designed to respond primarily to the electric field.
5. *Link effect (spin modulation, Faraday rotation, etc.)* − Building antennas to match these characteristic can be difficult. However, there are software packages both commercial and free (i.e. NEC-Win Synth, MMANA) that will aid us in designing our antenna.

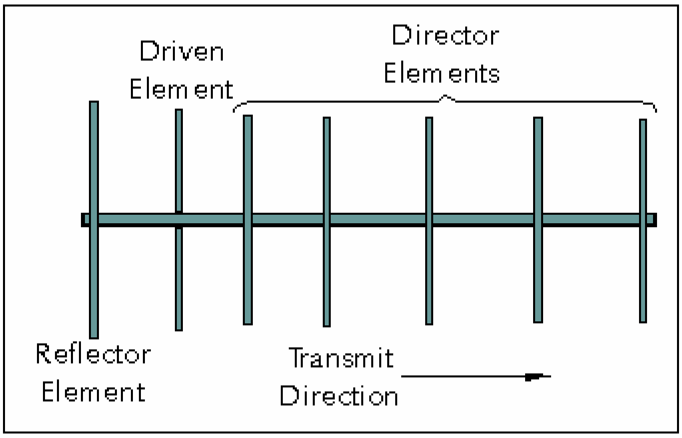


Fig 2: Sample YAGI antenna with small number of elements

We will build two antennas. The receiving antenna will be directional with a length of 70 centimeter. The transmitting antenna length will be 2 meters. Material cost for the two antennas is estimated at $60.

Pre Amplification

Signals from satellites are often weak due to atmospheric affect. Pre amplification unit will aid in restoring signal to it original state and maintain signal consistency. Consistency in signal level is very important during the A/D and D/A conversion process.

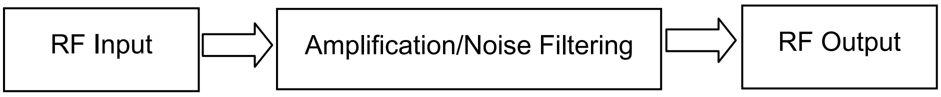


Fig 3: Amplifier stage high level circuit design

Rotator Controller

The software that tracks satellites must be able to control the orientation of the Antenna System. For this, we will design a rotator controller which in turn controls the antenna's position (see Figure 4). The rotator controller consists of a microcontroller (Motorola M68HC11), MAX232 (for serial communication), D/A converter, motor driver circuits and 2 motors. The microcontroller's main task is to interpret four types of signals received from the tracking software (x-axis +/- and y-axis +/-) and generate the appropriate signal to the driving circuit of each motor. Only two degrees of movement is required to track an object in our design. Estimated cost of construction for this module is $125.

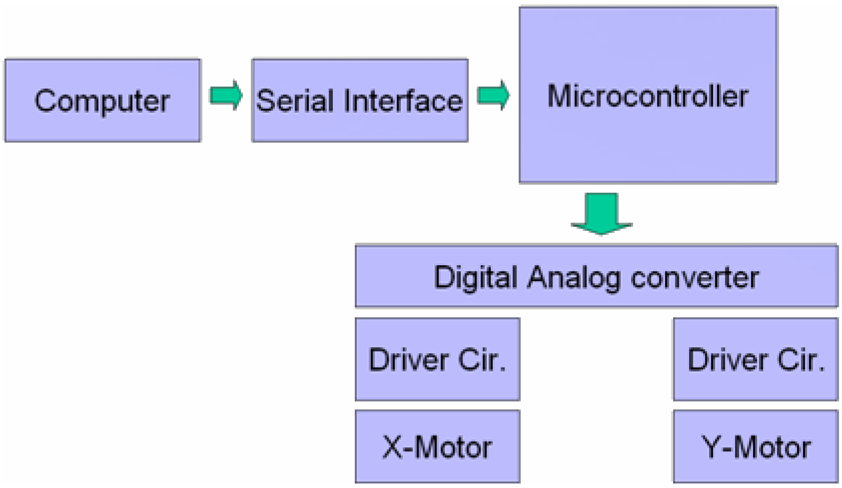


Fig 4: Overall design of the rotator controller system

Modem/Terminal Node Controller

The communication interface between analog (satellite transmission) and digital world (user's computer) will be the modem and TNC's main responsibility (see Figure 5). The modem will operate in FSK (Frequency Shift Key) mode. Implementation of the modem circuit was chosen for low cost and reliability. Estimated cost for construction of this module is $20. The Terminal Node Controller costs approximately $60.

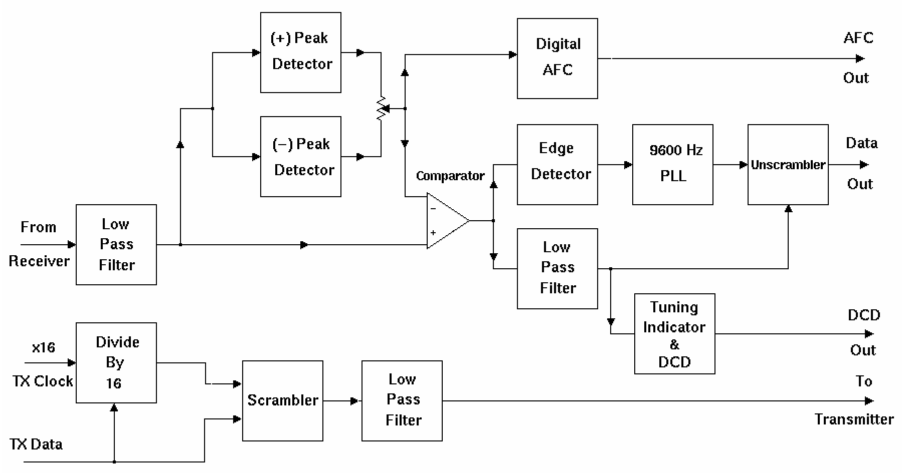


Fig 5: High-level schematic of 9600 baud rate modem circuit design

Power System

Transmission of data requires a significant amount of power. We plan to use standard 110V power outlets and a step-down transformer to power our equipment. If there is no power source available, for instance if our Antenna/equipment has to be setup outside, we will borrow power equipments from the Electrical Engineering laboratory.

Tracking Software

NASA provides real time satellite tracking data via a system called J-Track. We plan to use this data and incorporate it to our satellite tracking software. We will select satellites with paths that are closest to our geographical location and allow users to select from this list. Software package to track low orbiting satellite will be implemented on .NET platform. This program consists of 8 main modules to track a given satellite and control the antenna's orientation.

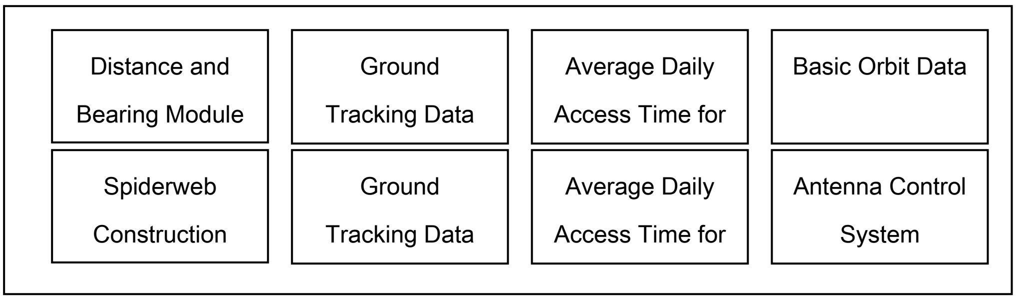


Fig 6: Tracking Program Modules

User Interface Software

User interface will be built on .NET platform. The initial interface will be implemented using a command line UI. Once this is completed it will be further extended to a Graphical environment similar to fig 7. This package would present GUI windowpanes allowing the user to select and communicate with the nearest amateur satellite.

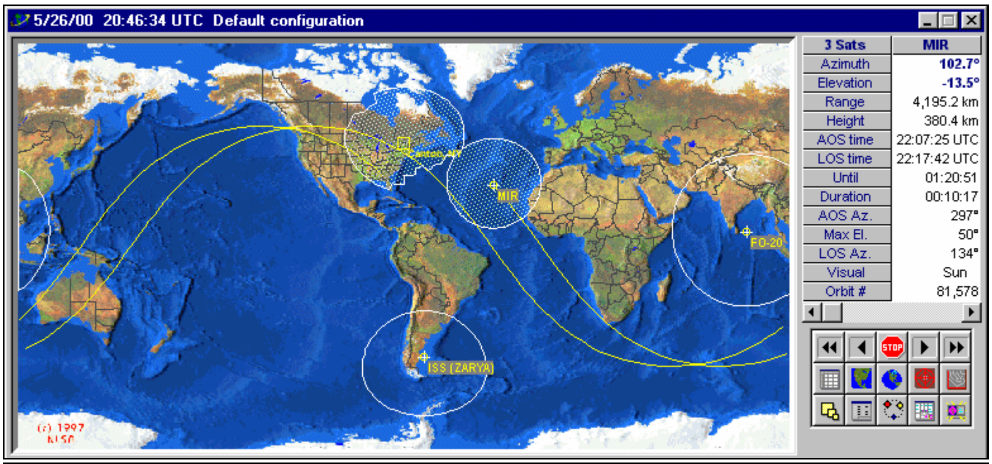


Fig 7: Sample graphical user interface

Primitive Server Software

Based on either .NET or Java, this software package will enable user to share information. Essentially, this server will be built on a queue structure. Remote users can access our server via the Internet. They can transmit files of limited size (in the order of several Kilobytes) to the server's queue for uploading to satellite. As soon as the system is available to access the specified satellite, these files are uploaded automatically. At the same time, gateway will receive files that are destined to it. All transmitting and receiving data will be publicly available to remote user. Limited security feature will be based on the platform that the server software runs on. For example, if the server is implemented on a Windows system, security and file permissions will be handled by the windows machine.

Note that amateur satellite system is open to the public domain. Therefore data transmission is not encrypted and users are not expected to transmit any sensitive material.

Testing Strategy

Development of individual components can be done separately. In other words, the modem modules do not need to wait for the antenna module during development. This allows our group to design and implement each module independently of each other. Regular group meeting will ensure that everyone is on the right page. Each module will be independently tested using lab equipments. The entire design can be tested in two stages. Send stage will only focus on data transmission and receiving stage will only focus on data reception. Final stage testing will combine the two and test the entire loop of communication, from sending to receiving data packets.

Backup Strategy

In the event of a problem that may occur due to reasons beyond our control during the project development process, we have developed the following as a backup plan for various possible problems that we may run into.

- Satellite availability/operational issues – There are several dozen satellites available; therefore we expect at least one satellite to be operational at any given time. In the worse case scenario we will communicate with amateur weather satellites.
- Satellite communication problems – We will try to communicate with satellites that are closest to us geographically, and those that follow a circular orbit versus an elliptical orbit.
- Equipment failure – Most equipment we use are available as off the shelf components. This reduces dependence on specialized equipment.

Bill of Materials

Antenna System

- Aluminum bars
- Wiring
- Power source

Estimated Cost of materials $60

Amplifier

- Capacitors
- Inductors
- Diodes
- RCA Jacks
- MES FET
- Resistors

Estimated Cost of materials $20

Rotator Controller

- M68HC11 Motorola Microcontroller
- Max232 chip for serial communication
- Breadboard, wires, capacitors, resistors etc
- D/A converter, connecters
- 2 step motors or a rotator (borrowed)

Estimated Cost of materials $25

9600 Baud rate Modem

- TL064 IC

- CD4538, CD4013
- LEDs, Zeners
- Breadboard, resistors and capacitors

Estimated Cost of materials $27

Total estimated cost of materials $132

Schedule

We have divided our project development in to two semesters. Following Gantt chart shows the division of modules among team members. During the summer we will spend majority of our time acquiring parts and components, running simulations and building necessary circuits. Almost all the software development will be carried out during the fall semester. There are 4 checkpoints setup in the schedule denoted by (MARKER) and a final Integrations test to see if we are meeting the project goals and deadlines.

Timeline: Summer 2004

Month	May				June				July				August			
	W1	W2	W3	W4	W1	W2	W3	W4	W1	W2	W3	W4	W1	W2	W3	W4
Vinh			Research		Design and Implementation **(Marker)**				Testing, Optimization and Integration				Project Integration and Testing in Live Environment **(Major Marker)**			
Junsang			Research		Design & Implementation		Testing **(Marker)**		Testing with Modem, Pre-Amp and rotator controller				Project Integration and Testing in Live Environment **(Major Marker)**			
Suresh		Research / Design		Circuit Simulation. Gather parts		Build circuit	Program Microcontroller **(Marker)**			Testing with motors	Integrate and test with Antenna. Calibration		Project Integration and Testing in Live Environment **(Major Marker)**			
Bo			Research and gather parts			Design and Implementation **(Marker)**			Testing				Project Integration and Testing in Live Environment **(Major Marker)**			

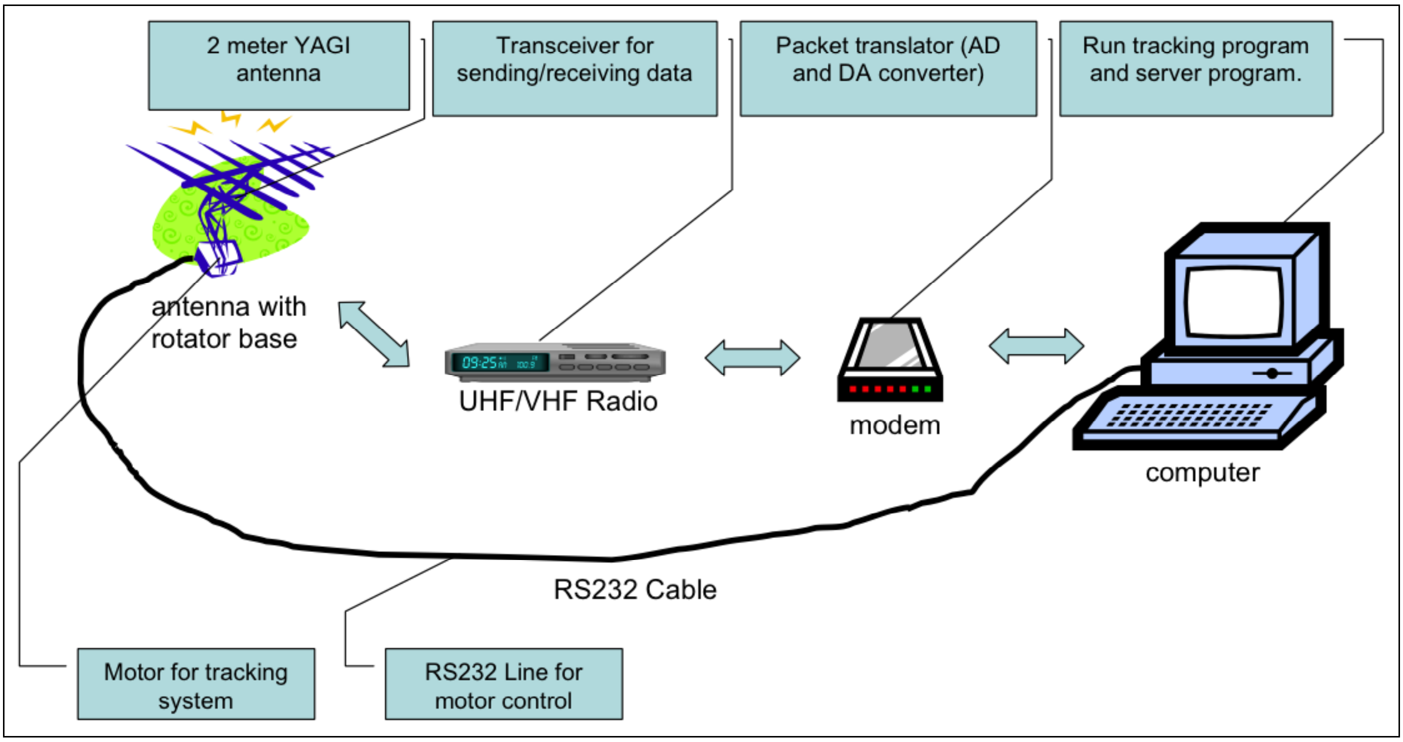
Timeline: Fall 2004

Month	September				October				November				December			
	W1	W2	W3	W4	W1	W2	W3	W4	W1	W2	W3	W4	W1	W2	W3	W4
Vinh	GUI Interface Design and development **(Marker)**				Testing and Debugging **(Marker)**				Final Testing of Overall Project **(Major Marker)**				Documentations and Project Submission			
Junsang			Research and Development for tracking modules			Integrating with Hardware **(Marker)**			Final Testing of Overall Project **(Major Marker)**				Documentations and Project Submission			
Suresh		Develop 2-D mapping for the tracking and bearing calculations			Integrate with J-track data **(Marker)**		Integrate with tracking and Network modules **(Marker)**		Final Testing of Overall Project **(Major Marker)**				Documentations and Project Submission			
Bo		Develop Networking module			Testing and Debugging **(Marker)**				Final Testing of Overall Project **(Major Marker)**				Documentations and Project Submission			

References

- *http://www.nlsa.com/index.html*
- *http://science.nasa.gov/RealTime/JTrack/*
- *http://www.arrl.org/*
- *http://www.sat-net.com/winorbit/index.html#winorbit*
- *http://www.tele-satellite.com/*
- *http://www.amsat.org/amsat/sats/n7hpr/satsum.html*
- *http://www.tbs-satellite.com/tse/online/*
- *http://www.amsat.org/amsat/ftpsoft.html*
- *http://www.saao.ac.za/~wpk/index.html*
- *http://home.nycap.rr.com/capcom/fsattrak.html*
- *http://home.hiwaay.net/~wintrak/*
- *http://www.riverland.net.au/~hutchja/tracking_antenna.htm*
- *http://web.usna.navy.mil/~bruninga/astars.html*
- *http://www.qsl.net/kd2bd/predict.html*
- *http://www.amsat.org/amsat/instanttrack/sattrack.html*
- *http://www.jpl.nasa.gov/releases/96/leoterm.html*
- *http://www.kantronics.com*
- *http://www.shpc.pe.kr/sbaycom.html*
- *http://users3.ev1.net/~medcalf/*

Appendix



Project Big Picture (overall view)

Interfaces:

Components	**Interface type**
Computer ↔ modem	Serial bus
Modem ↔ UHF/VHF radio	Serial port
UHF/VHF radio ↔ Antenna	wire
Antenna ↔ Satellite	Radio signal
Computer ↔ Motor control Unit	Serial/ Parallel
Computer ↔ Other networks	Ethernet

Program

```
// Subject: DISTBEAR
// Date: Mar.05,2004
// Modifier: Junsang Cho (chojs@cs.utah.edu)
// Note: This code is converted from the codes that were made in Basic language.
//
//  DISTBEAR provides distance and bearing information for two stations, P1
//  and P2, on the surface of the Earth. With small modifications it can provide
//  azimuth and elevation antenna aiming information for a ground station at P1
```

```
//  and a subsatellite point at P2.

#include <iostream>
#include <string>
#include <math.h>
using namespace std;

float  fnarccos( float z );
int sgn( float a );   //Check the sign of a given value

int main(){
	string sLocation1 = "Baltimore";
	string sLocation2 = "Moscow";

	float fLatitude1 = 39.3;  //'deg. E**
	float fLongitude1 = -76.6;

	float fLatitude2 = 56.1;  //'deg. E**
	float fLongitude2 = 37.5;

	cout << "Starting... Program DISTBEAR ..." << " from "
	      << sLocation1 << " to " << sLocation2 << endl;

	float PI = 3.141593;
	float fRadians2Deg = 180.0/PI;
	float fDeg2Radians = PI/180.0;
	float fKm2mile = 0.6214;
	float fMeanEarthRadius = 6371.0; //killometer

	float fLat1 = fLatitude1 * fDeg2Radians;
	float fLng1 = fLongitude1 * fDeg2Radians;
	float fLat2 = fLatitude2 * fDeg2Radians;
	float fLng2 = fLongitude2 * fDeg2Radians;

	float dsp,dlp;

	if ( abs( fLatitude1 ) > 89.99 ) { // P1 is North or South Pole
		cout << "If P1 is North or South pole then azimuth is not defined" << endl;
		cout << "Point antenna along P2 longitude (short path) or" << endl;
```

```
		cout << "P2 longitude + 180 degree (long path)." << endl;
		if( sgn( fLat1 ) == sgn( fLat2 ))
			dsp = fMeanEarthRadius*( PI/2 - abs( fLat2 ));

		if( sgn( fLat1 ) != sgn( fLat2 ))
			dsp = fMeanEarthRadius*( PI/2 + abs( fLat2 ));

		cout << "Short path: " << dsp << "km" << endl;
		cout << "Long path: " << int( 2*PI*fMeanEarthRadius - dsp ) << "km" << endl;
		exit(0);
	}

	float fCosBeta = sin(fLat1) * sin(fLat2) *
				cos(fLat1) * cos(fLat2) * cos(fLng2-fLng1);

	if( fCosBeta > 0.99999 ){
		cout << "points coincide." << endl;
		exit(0);
	}

	if ( fCosBeta < -0.999999 ){ //Antipodes
		cout << "Antipodes, azimuth not defined, distance = "
			<< fMeanEarthRadius * PI << " km" << endl;
		exit(0);
	}

	float fBeta = fnarccos( fCosBeta );
	float fDistanceShortPath = fBeta * fMeanEarthRadius; //Distance short path
	float fDistanceLongPath = 2*PI*fMeanEarthRadius - fDistanceShortPath;	//Distance long
path

	float fCosAZ = (sin(fLat2)-sin(fLat1)*cos(fBeta))/(cos(fLat1)*sin(fBeta));

	float AZ = 0;
	if(fCosAZ > 0.999999){
		AZ = 0;
	} else if ( fCosAZ < -0.999999 ){
		AZ = 180;
```

```
	} else {
		AZ = fnarccos( fCosAZ ) * fRadians2Deg;
	}

	float AZSP = 0;
	float AZLP = 0;
	if ( sin( fLng2-fLng1 ) >= 0 ){
		AZSP = AZ;
		AZLP = 180 + AZ;
	}

	if( sin( fLng2-fLng1 ) < 0 ){

		AZSP = 360 - AZ;
		AZLP = 180 - AZ;
	}

	AZLP = int(AZLP*10 + 0.5)/10;    //round off
	AZSP = int(AZSP*10 + 0.5)/10;    //round off

	dsp = int(fDistanceShortPath + 0.5);
	dlp = int(fDistanceLongPath + 0.5);

	int dspmi = int(fDistanceShortPath * fKm2mile + 0.5 ); // km -> miles
	int dlpmi = int(fDistanceLongPath* fKm2mile + 0.5);

	cout << "Short Path: " << dsp << "km (" << dspmi
		<< "mi.) " << AZSP << " Deg. E of N" << endl;

	cout << "Long Path: " << dlp << "km (" << dlpmi
		<< "mi.) " << AZLP << " Deg. E of N" << endl;

	return 0;
}


float fnarccos( float z ){
	float PI = 3.141593;
	return (PI/2 - atan( z/sqrt(1-z*z)));
```

```
}

//Sign checker
int sgn( float a ){
	if( a > 0 ) return 1;
	if( a == 0) return 0;
	if( a < 0 ) return -1;
}

//END
```

DISTBEAR

This function provides distance and bearing information. It also provides azimuth and elevation antenna aiming information for a ground station at P1 and a sub-satellite point at P2. See the appendix.

SWEB

Provides the data needed to draw a spiderweb around any point on the surface of the Earth. It's designed so the number and spacing of the range circles and the number of bearings employed can be easily changed.

GTCIRC & GTEL

The two provides ground track data for circular and elliptical orbits. It will also use for long term predictions.

ADATCIRC & ADATEL

The two functions provide information on average daily access time to a satellite. It will be used to calculate the percentage of time the satellite spends at different elevations – information that's extremely useful for selecting a ground station antenna.

SATPFL

Calculated orbit data for satellite. It uses mean motion and eccentricity, form standard Keplerian elements, and to produce the parameters listed in the Spacecraft Profiles.

ACS

Makes signals for controlling physical tracking system from computed data.