

IMPLEMENTACIÓN Y VALIDACIÓN DEL SISTEMA DE CONTROL  
EN LA ESTACIÓN TERRENA CUBESAT USCO PARA EL SEGUIMIENTO DE LA  
TRAYECTORIA DE UN PICO SATÉLITE

CAMILO ERNESTO GARAVIÑO VILLALBA  
COD. 2003100045

CRISTIAN SAUL ENDO PEREZ  
COD. 2003100283

UNIVERSIDAD SURCOLOMBIANA  
FACULTAD DE INGENIERÍA  
PROGRAMA DE INGENIERÍA ELECTRÓNICA  
NEIVA  
2008

IMPLEMENTACIÓN Y VALIDACIÓN DEL SISTEMA DE CONTROL  
EN LA ESTACIÓN TERRENA CUBESAT USCO PARA EL SEGUIMIENTO DE LA  
TRAYECTORIA DE UN PICO SATÉLITE

CAMILO ERNESTO GARAVIÑO VILLALBA  
COD. 2003100045

CRISTIAN SAUL ENDO PEREZ  
COD. 2003100283

Proyecto de grado presentado para optar  
al título de Ingeniero Electrónico

Director:  
JOSE DE JESUS SALGADO PATRON  
Ingeniero Electrónico Msc.

UNIVERSIDAD SURCOLOMBIANA  
FACULTAD DE INGENIERÍA  
PROGRAMA DE INGENIERÍA ELECTRÓNICA  
NEIVA  
2008

Nota de aceptación:

---

---

---

---

---

---

---

Firma del presidente del jurado

---

Firma del primer jurado

---

Firma del segundo jurado

Neiva, 12 de Septiembre de 2008

Dedico este trabajo a mi familia que con su gran apoyo he podido culminar una etapa mas de mi vida, en especial a mi madre que con su aprecio y preocupación me incentivaba a seguir adelante. A mis compañeros de estudio por ser coautores de muchos logros a lo largo de estos años. A mis amigos por la compañía y el apoyo brindado en los días buenos y en los difíciles. La lista se podría extender demasiado así que dedico este trabajo también a todos aquellos que estuvieron alrededor mío y me sirvieron de apoyo en algún momento de esta travesía.

**Camilo Ernesto Garaviño.**

A mi mamá y hermanos  
**Cristian Saul Endo.**

## CONTENIDO

	pág.
INTRODUCCIÓN	14
1. PLANTEAMIENTO DEL PROBLEMA	15
2. SOFTWARE UTILIZADO E INSTALACIÓN	17
2.1 SOFTWARE DE DESARROLLO	17
2.2 SOFTWARE DE PREDICCIÓN	19
2.2.1 Winorbit	19
2.2.2 Wxtrack	19
3. SOFTWARE DESARROLLADO PARA LA INTERFAZ ICUG5500	22
3.1 SOFTWARE PARA EL PIC18F 4550	22
3.1.1 Definición de variables, Calibración y lectura de rangos de calibración	25
3.1.2 Detección de PC conectado y envío datos de calibración al PC	26
3.1.3 Detección de rotor conectado, lecturas análogas, envío de datos actuales y recepción de resultados	30
3.2 SOFTWARE DESARROLLADO PARA EL PC	32
3.2.1 Evento form load	35
3.2.2 Evento on read	37
3.2.3 Evento Timer	39
4. INTERFAZ GS-232	41
4.1. CALIBRACION	42
4.2. SOFTWARE CONTROLADOR DE INTERFAZ GS-232	45
4.2.1. Evento form load	46
4.2.2 Evento Timer	48
4.2.3 Evento linknotify	48
4.2.4. Modo semiautomático	49
5. PRUEBAS REALIZADAS	50
6. VERIFICACION DE MÉTODOS DE DETECCIÓN DE BEACON	54
7. CONCLUSIONES	57
8. RECOMENDACIONES	59
BIBLIOGRAFIA	60
ANEXOS	62

## LISTA DE CUADROS

	Pág.
Cuadro 1. Valores de calibración	28
Cuadro 2. Estados de movimiento	31
Cuadro 3. Tareas realizadas	37

## LISTA DE FIGURAS

	pág.
Figura 1. Rotor G-5500 y control G-5500	15
Figura 2. Diferencia horaria	20
Figura 3. Generación del archivo de texto	21
Figura 4. Diagrama de flujo interfaz ICU G5500	23
Figura 5. Detección de PC conectado	27
Figura 6. Gráficas de las dispersiones	29
Figura 7. Detección de rotor conectado	30
Figura 8. Interfaz de usuario	33
Figura 9. Modo semiautomático	34
Figura 10. Modo manual	34
Figura 11. Letrero de conexión del rotor	35
Figura 12. Primer mensaje de advertencia	36
Figura 13. Segundo mensaje de advertencia	36
Figura 14. GS-232	41
Figura 15. Conexión del GS-232	42
Figura 16. Controlador Dual G5500	43
Figura 17. Propiedades de COM1	43
Figura 18. Configuración ASCII	44
Figura 19. Interfaz de usuario GS-232	46
Figura 20. Modo semiautomático	49
Figura 21. GS-232 Azimut	51
Figura 22. GS-232 Elevación	51
Figura 23. ICU G5500 Azimut	52
Figura 24. ICU G5500 Elevación	52
Figura 25. Modulación PSK	55

## LISTA DE ANEXOS

	pág.
Anexo A. Cuadros de las pruebas	62
Anexo B. Diagrama UML de casos de uso del programa del PC	69
Anexo C. Diagrama UML de secuencia verificar programa abierto	70
Anexo D. Diagrama UML de secuencia seleccionar modo de uso	71
Anexo E. Diagrama UML de secuencia ejecutar evento On Read	72
Anexo F. Diagrama UML de secuencia ejecutar evento Timer	73

## GLOSARIO

**AMSAT:** (The Radio Amateur Satellite Corporation) o amateur satélite - Satélite aficionado, es la asociación mundial de satélites de radioaficionados, dividiéndose correspondientemente por cada país.

**ASCII:** (American Standard Code for Information Interchange — Código Estadounidense Estándar para el Intercambio de Información), es un código de caracteres basado en el alfabeto latino tal como se usa en inglés moderno y en otras lenguas occidentales.

**AUTOCAD:** es un programa de diseño asistido por ordenador (CAD "Computer Aided Design", Diseño asistido por computador) para dibujo en 2D y 3D. Actualmente es desarrollado y comercializado por la empresa Autodesk.

**AZIMUT:** es el ángulo medido sobre el horizonte que forman el punto cardinal Norte y la proyección vertical del satélite sobre el horizonte.

**BOOM:** elemento que soporta el dipolo y los directores de una antena.

**BUFFER:** es una ubicación de la memoria en una computadora o en un instrumento digital reservada para el almacenamiento temporal de información digital, mientras que está esperando ser procesada.

**CODIGO MORSE:** es un sistema de representación de letras y números mediante señales emitidas de forma intermitente (puntos y rayas).

**CORRIMIENTO EN DOPPLER:** consiste en la variación de la longitud de onda de cualquier tipo de onda emitida o recibida por un objeto en movimiento, es decir que cambia la frecuencia de las ondas percibidas, confirmando que el tono de un sonido emitido por una fuente que se aproxima al observador es más agudo que si la fuente se aleja.

**CUBESAT:** es un programa que se creó con el propósito de desarrollar una tecnología basada en pico-satélites de órbita baja (LEO) con fines educativos. Así también se denominan los dispositivos cuyos parámetros son: una masa no mayor a un KG y un tamaño que no supere al de un cubo de 10 cm. de arista.

**CW:** (del inglés Continuous Wave) es la telegrafía con designación A1A, es decir, es un sistema binario que consiste en emitir e interrumpir la señal portadora de acuerdo al Código Morse, conformado por elementos de tiempo denominados puntos y rayas, cuya duración así como la de los espacios entre ellos, deben mantener una relación constante. La CW se caracteriza por ocupar poco ancho de banda y por tener una relación señal/ruido muy alta; todo ello permite la comunicación a larga distancia aun en condiciones desfavorables de la propagación.

**DB-9:** (originalmente DE-9) es un conector analógico de 9 clavijas de la familia de conectores D-Subminiature (D-Sub o Sub-D). El conector DB9 se utiliza principalmente para conexiones en serie, ya que permite una transmisión asíncrona de datos según lo establecido en la norma RS-232 (RS-232C).

**DIN:** es un conector. Existen estándares DIN para una gran cantidad de diferentes conectores, pero por regla general se refiere a los conectores con extremo circular que fueron los que primero se estandarizaron por DIN para ser empleados en las señales de audio analógicas. Algunos de estos conectores fueron usados posteriormente en la transmisión de vídeo analógico y en interfaces digitales.

**DUAL 5-PIN:** son dos conectores, planos y de 5 pines cada uno.

**EASYHID USB WIZARD:** es una herramienta de software del MicroCode Studio que otorga el código básico y necesario para establecer comunicación USB entre el microcontrolador y el PC.

**EEPROM:** son las siglas de electrically-erasable programmable read-only memory (ROM programable y borrable eléctricamente).

**ELEMENTOS KEPLERIANOS:** los elementos keplerianos son números o datos que sirven para fijar la órbita y por consiguiente, determinar la posición del satélite en el espacio en un momento dado. Atendiendo puramente a la mecánica orbital y las leyes de Kepler seis son los "números" que se necesitan para definir la órbita de un satélite.

**ELEVACIÓN:** ángulo entre el plano del horizonte y el satélite.

**HOST:** Aquel dispositivo de la red que ofrece servicios a otros ordenadores conectados a dicha red.

**HYPERTERMINAL:** es una aplicación que se puede utilizar con el fin de conectar su ordenador a otros sistemas remotos. Estos sistemas incluyen otros ordenadores, los sistemas de tablón de anuncios, servidores, sitios Telnet, y los servicios en línea.

**ISIS PROTEUS:** es un entorno integrado diseñado para la realización completa de proyectos de construcción de equipos electrónicos en todas sus etapas: diseño, simulación, depuración y construcción.

**LCD:** es una pantalla de cristal líquido (acrónimo del inglés Liquid crystal display), pantalla delgada y plana formada por un número de píxeles en color o monocromos colocados delante de una fuente de luz o reflectora.

**LEO:** son las siglas en inglés de órbita terrestre baja (Low Earth Orbit).

**MACROMEDIA FLASH:** Es una aplicación en forma de estudio de animación que trabaja sobre "Fotogramas" destinado a la producción y entrega de contenido interactivo para diferentes audiencias alrededor del mundo sin importar la plataforma.

**MICROCODE STUDIO – PICBASIC PRO:** es una poderosa herramienta, con un ambiente integrado de desarrollo visual (IDE), con la capacidad de ser ejecutado línea por línea en tiempo de programación, diseñado específicamente por microEngineering Labs. Emplea los compiladores PICBASIC™ y PICBASIC PRO™.

**MID:** es una instrucción del visual Basic que permite tomar de una cadena ciertos caracteres, tomando como base la posición del carácter inicial y la cantidad que se quiere tomar.

**MPLAB ID:** es una herramienta de software desarrollada por Microchip para programar y compilar programas para microcontroladores.

**OVERLAYS:** son grandes cantidades de código en áreas muy pequeñas de memoria, auxiliadas por algunas llamadas al sistema operativo.

**PBP:** es la abreviatura empleada para el programa PICBASIC-PRO

**PLUG-IN:** Un Complemento (o plug-in en inglés) es una aplicación que se relaciona con otra para aportarle una función nueva y generalmente muy específica. Esta aplicación adicional es ejecutada por la aplicación principal e interactúan por medio de la API.

**PSK:** Es el caso de modulación donde tanto las señales de transmisión como las señales de datos son analógicas. Es un tipo de modulación exponencial al igual que la modulación de frecuencia. Se caracteriza porque la fase de la onda portadora varía directamente de acuerdo con la señal modulante, resultando una señal de modulación en fase.

**SELECT CASE:** es una instrucción que permite seleccionar entre varias alternativas de una variable de control.

**STRING:** es una cadena de caracteres, palabra, ristra de caracteres o frase. Es una secuencia ordenada de longitud arbitraria (aunque finita) de elementos que pertenecen a un cierto alfabeto. En general, una cadena de caracteres es una sucesión de caracteres (letras, números u otros signos o símbolos).

**USB:** Universal Serial Bus (bus universal en serie) o Conductor Universal en Serie es un puerto que sirve para conectar periféricos a una computadora.

**UTC:** El tiempo universal coordinado. También conocido como tiempo civil, es la zona horaria de referencia respecto a la cual se calculan todas las otras zonas del mundo.

**.TXT:** es la extensión que suelen tener los documentos contienen texto plano, o texto sin formato. La aplicación en la cual se debe correr un archivo de este formato es el bloc de notas.

## **RESUMEN**

Un CUBESAT es un micro satélite en forma de cubo de 10 cm. De arista, y de peso menor o igual a 1Kg., que fue creado para fines educativos, experimentales y de investigación. Son satélites no comerciales de órbita baja, que viajan alrededor de la tierra a una velocidad mayor que la velocidad de rotación del planeta. Debido a eso, es que una estación terrena, que desee establecer un enlace con satélites de este tipo, debe realizar un seguimiento, para lo cual requiere de un sistema de control.

Este proyecto tiene como finalidad implementar, optimizar y validar el sistema de control existente en la Estación Terrena del proyecto CUBESAT de la Universidad Surcolombiana y se ha dividido en seis capítulos, en los cuales se explica el proceso que se llevo a cabo, los inconvenientes y las soluciones obtenidas.

En el segundo capítulo, se mencionan los requisitos de instalación y configuración que deben cumplir las herramientas de software utilizadas, debido a la cantidad de problemas que presentaron durante su uso.

En el tercero, se describen los programas desarrollados para la interfaz ICU G5500, tanto el de PicBasic para PIC18F4550 como el de Visual Basic para el PC, los inconvenientes que presentaron y sus correspondientes soluciones.

En el cuarto capítulo, se desarrollo un programa en Visual Basic con características similares al de la ICU G5500, pero utilizando la interfaz GS-232, la cual es propia para el rotor YAESU G5500 utilizado en el proyecto y no estaba disponible en los proyectos desarrollados anteriormente.

Finalizando el libro, se explican los dos últimos capítulos que en su orden describen las pruebas realizadas con los sistemas desarrollados y los métodos estudiados para la detección de una señal Beacon con los resultados obtenidos.

## ABSTRACT

A CUBESAT is a micro satellite with shape of cube, with 10cm edge, and its weight is smaller or similar than 1Kg. It was created with educational, experimental and investigational purposes. They are non commercial satellites of low orbit that travel around the Earth. Its rotation speed is bigger than the planet's. A Ground Station that wants to establish a connection with this type of satellites should carry out a pursuit, which requires a control system.

This project has as purpose to implement, optimize and validate the control system which exists in the Ground Station of the Surcolombiana University's CUBESAT project. It has been divided in six chapters, in which explains the process that was carried out, the inconveniences and solutions.

The second chapter mentions the installation requirements and configuration that software tools should complete, because of the problems that it presented during its use.

In the third one, the developed programs are described for the interface ICU G5500, so much PicBasic for PIC18F4550 as Visual Basic for the PC, also the inconveniences that presented and their corresponding solutions.

In the fourth chapter, a program in Visual Basic was developed with similar characteristics of the ICU G5500, but using the interface GS-232, which is own for the rotor YAESU G5500 used in the project and it was not available in projects developed previously.

At the end of the book, the last two chapters describe the tests carried out with the developed systems and the studied methods for the detection of a Beacon sign with the obtained results.

## INTRODUCCIÓN

El concepto de comunicaciones satelitales surgió desde antes de los años 50, en el siglo XX a partir de la idea de establecer comunicación mediante el uso de satélites artificiales. En los años 90 empezaron a desarrollarse proyectos para colocar satélites a altitudes bajas, llamados satélites de órbita baja (LEO). Estos por estar mas cerca de la tierra, poseen menor retardo de la señal, presentan menos pérdidas y utilizan menor potencia para transmisión, sin embargo no permanecen en un punto fijo en el espacio para un observador terrestre, aumentando la complejidad de la estación terrena si se quiere realizar un buen enlace de comunicaciones.

La contribución principal de la Universidad Surcolombiana en el macro-proyecto CUBESAT- UD – USCO, el cual propone poner en órbita baja (LEO) un pico satélite, es el montaje y puesta en funcionamiento de una estación terrena, que permita por medio de un sistema trasmisor-receptor realizar un enlace con el picosatélite.

Esta estación consiste en un hardware y software que permite una recepción y transmisión de datos confiables. Sus componentes incluyen un sistema de transmisión y recepción de antenas, un computador programado con un software que prediga la posición del satélite, compatible con el hardware de auto seguimiento y el sistema transmisor-receptor que permita transmitir y recibir datos.

En la Universidad Surcolombiana este proyecto se ha dividido en dos grupos. El primero encargado de la parte de comunicaciones de la estación terrena y dentro del cual se han desarrollado tres trabajos de grado.

El segundo dentro del cual se desarrolla este proyecto, es el encargado de la parte de control y dentro del cual ya se adelantó dos trabajos de grado que consistieron en: diseño del sistema de control físico para el posicionamiento de la antena de la estación terrena, en el cual se realizó un modulo encargado de transferir datos entre el PC y los actuadores encargados del movimiento de la antena y el diseño del sistema de control del desplazamiento y software de monitoreo de la antena de la estación terrena, donde se desarrolló un programa para la verificación de parámetros, monitoreo y control de posición de antena de la estación terrena.

Como complemento a los trabajos anteriores, el presente proyecto tiene como objetivo realizar la implementación, integración y la validación del sistema de control desarrollado para la estación terrena, a través de pruebas de calibración terrestres y de seguimiento a satélites, con el fin de que el movimiento de las antenas en esta base realice el seguimiento óptimo a un pico satélite y así poder detectar correctamente la señal Beacon que emite.

## 1. PLANTEAMIENTO DEL PROBLEMA

El programa de Ingeniería Electrónica, en nombre de la Universidad Surcolombiana, ha expresado su interés en convenio con la Universidad Distrital Francisco José De Caldas, en el proyecto de lanzamiento, puesta en órbita y monitoreo de un pico satélite espacial de tipo CUBESAT, para la investigación en las tecnologías espaciales y comunicaciones satelitales.

El papel asumido por el programa de Ingeniería Electrónica de la Universidad Surcolombiana en el presente proyecto, hace referencia al montaje y puesta en funcionamiento de la estación terrena situada en la ciudad de Neiva. Esta parte del proyecto se ha dividido en dos grupos, el primero se refiere al área de comunicaciones y el segundo se relaciona con el sistema de control de la estación para el seguimiento adecuado del satélite, dentro del cual se desarrolla el presente proyecto en el que ya se han desarrollado dos trabajos de grados: diseño del sistema de control del desplazamiento y software de monitoreo de la antena de la estación terrena de la ciudad de Neiva y diseño del sistema de control físico para el posicionamiento de la antena de la estación terrena del proyecto CUBESAT en la Universidad Surcolombiana.

El presente proyecto, tiene como tarea realizar la integración y validación del sistema de control desarrollado en trabajos anteriores para la estación terrena, siendo de gran importancia su depuración y sobre todo integración de la parte dura (hardware) con la parte blanda (software) debido al desarrollo por separado que se le ha dado, ya que se debe dar un posicionamiento preciso del ROTOR YAESU G-5500, que cuenta con su control dual G-5500, apreciados en la figura 1.

**Figura 1. Rotor G-5500 y control G-5500**



*YAESU G-5500 Instruction Manual*

De acuerdo al planteamiento del problema, se han establecido los siguientes objetivos, que enmarcan el camino a seguir en el desarrollo del presente proyecto:

### **Objetivo general**

Implementación, validación y optimización del sistema de control existente en la estación terrena CUBESAT USCO para el seguimiento de la trayectoria de un pico satélite.

### **Objetivos específicos**

- Ajustar, integrar y validar el funcionamiento de los proyectos ya desarrollados dentro del proyecto de la Estación Terrena CUBESAT USCO para el área de control, mediante la realización de pruebas con pico-satélites.
- Depurar, optimizar e integrar el software desarrollado para el control del sistema de antenas de la estación terrena CUBESAT USCO.
- Desarrollar una metodología de calibración del sistema de control a través de pruebas terrestres.
- Evaluar distintos métodos de detección de la señal Beacon de un satélite para ser empleado en la base terrena de la Universidad Surcolombiana.

Para cumplir con estos objetivos se clarifican e integran los estudios y sistemas previamente desarrollados, teniendo en cuenta patrones y recomendaciones hechos en dichos trabajos, se realizan pruebas con satélites que permitan verificar el funcionamiento del sistema de control, y a su vez se realizan mediciones de la posición del rotor en un eje coordenado.

## 2. SOFTWARE UTILIZADO E INSTALACIÓN

Durante el desarrollo del presente proyecto se utilizó una serie de programas necesarios para la realización de tareas específicas, estos componentes lógicos en su gran mayoría fueron seleccionados en anteriores trabajos de grado correspondientes al área de control del macroproyecto Cubesat y deben tener ciertas características específicas que aseguren su correcto funcionamiento. Por esta razón y debido a la cantidad de inconvenientes que presentaron, es que en esta sección se realiza una completa descripción sobre su funcionamiento y configuración.

### 2.1 SOFTWARE DE DESARROLLO

Comprende las herramientas de trabajo con las cuales se validaron y desarrollaron los programas, buscando soluciones óptimas que dieran cumplimiento a algunos objetivos del presente proyecto de grado.

El primer software que se instaló y que además se convirtió en el centro de desarrollo del proyecto, fue el **Microsoft Visual Basic 6.0**, elegido en el trabajo de grado del diseño del control físico<sup>1</sup> y que no presentó problemas durante su instalación.

Igualmente se seleccionó el **MicroCode Studio – PICBASIC PRO** y el **MPLAB ID**. Estos programas funcionan en conjunto, demandando de una serie de requisitos relacionados con su instalación y la versión utilizada para un funcionamiento exitoso. Este hecho es la razón por la cual a continuación, se da a conocer el procedimiento necesario hacia el correcto funcionamiento de estas aplicaciones, para facilitar su manejo en un futuro, teniendo en cuenta la exclusión de algunos pasos innecesarios y la corrección de algunos errores de configuración presentados en los trabajos de grado ya mencionados:

**Primero:** Instalar una versión del **MPLAB ID** igual o superior a la 7.1, ya que estas versiones incluyen las librerías correspondientes a los **PIC** de la familia 18f XXX (Unidad de proceso del ICU G5500<sup>2</sup>, el cual tiene como objetivo, servir de interfaz entre el PC y el rotor Yaesu-G5500), a diferencia del trabajo de grado del diseño del control de desplazamiento, en la que solo la versión 7.1<sup>3</sup> funcionó, en este proyecto se comprobó que versiones superiores a la 7.1 funcionan perfectamente por lo cual se utilizó la versión 7.40, que no presentó problemas

**Segundo:** Copiar la carpeta **PBP** mencionada en el trabajo de grado del diseño del control de desplazamiento, al disco local **C**, siguiendo posteriormente con la ejecución de la

---

<sup>1</sup> ANDRADE VIDARTE, Diego Mauricio y NUÑEZ ROMERO, Leonardo Ernesto. Diseño del sistema de control físico para el posicionamiento de la antena de la estación terrena del proyecto cubesat en la universidad Surcolombiana. 2007. p. 38. Trabajo de grado (Ingeniero Electrónico). Universidad Surcolombiana. Facultad de Ingeniería.

<sup>2</sup> ANDRADE VIDARTE y NUÑEZ ROMERO, Op. cit., P. 28-29

<sup>3</sup> GARCIA LOPEZ, Juan Gilberto y MONTEALEGRE VANEGAS, John Wilder. Diseño del sistema de control del desplazamiento y software de monitoreo de la estación terrena de la ciudad de Neiva del macroproyecto cubesat. Neiva. 2007. p. 61. Trabajo de grado (Ingeniero Electrónico). Universidad Surcolombiana. Facultad de Ingeniería

aplicación **P246A** y así actualizar la carpeta **PBP**, la cual contiene las librerías para el manejo del protocolo **USB** utilizadas en el compilador que se mencionara en el tercer paso. En esta parte se comprobó que son innecesarios los dos pasos siguientes que modificaban la configuración del sistema mencionado, con lo cual se optimizó el procedimiento de instalación y configuración al reducir en dos el número de pasos a seguir.

**Tercero:** Instalar el compilador **MicroCode Studio – PICBASIC PRO** y enlazarlo con el **MPLAB ID** en **view->compile and program options->** habilitando todo. Luego en el menú **Compiler** en **find manually** seleccionar la carpeta **PBP** ubicada en **C**, de igual forma en el menú **Assembler** habilitar **Use MPASM** y en **find manually** seleccionar la carpeta **C:\Archivos de programa\Microchip \MPASM Suite**. Cabe mencionar que en los anteriores trabajos de grado el proceso de selección tanto en **Compiler** como en **Assembler** se realizaba en **find automatically**, estas acciones generaban compilaciones erróneas ya que en algunas ocasiones no se detectaba la carpeta actualizada **PBP** del disco **C**, sino alguna copia no actualizada en cualquier otra parte del PC, por otra lado en el menú **Assembler** se detectaba una carpeta errónea que correspondía al programa **Isis PROTEUS**<sup>4</sup> y no al **Microchip MPLAB ID**.

Es de gran importancia mencionar que la plantilla para **Visual Basic** que genera el **EasyHid USB Wizard**<sup>5</sup>, ubicado en el programa **MicroCode** y base de la aplicación realizada en el proyecto, presenta una pequeña e importante deficiencia en la sección de declaración, en las variables del **buffer USB** de entrada y salida como se muestra en las siguientes líneas.

Variables declaradas por el EasyHid USB Wizard:

```
Dim BufferIn(0 To BufferInSize)
Dim BufferOut (0 To BufferOutSize)
```

Al realizar una declaración de variable sin definir el tipo al final de la expresión como lo hace el **EasyHid**, el **Visual Basic** asume el tipo por defecto **Variant**, el cual proporciona a cada elemento de los arreglos **bufferin** y **bufferout** un tamaño de 16 bytes. Esta definición por defecto es incorrecta ya que el tamaño verdadero de los elementos de los arreglos de estos buffers es de 8 bytes, corrección que se logro después de un arduo estudio de la compleja y profunda plantilla generada por el **EasyHid** y su explicación se sale de los objetivos de este proyecto, mostrando a continuación solo el resultado.

Forma correcta de la declaración:

```
Dim BufferIn (0 To BufferInSize) as byte
Dim BufferOut (0 To BufferOutSize) as byte
```

---

<sup>4</sup> Software de simulación que utiliza librerías con la misma extensión y nombre del Mplab ID

<sup>5</sup>GARCIA LOPEZ, Juan Gilberto y MONTEALEGRE VANEGAS, John Wilder. Diseño del sistema de control del desplazamiento y software de monitoreo de la estación terrena de la ciudad de Neiva del macroproyecto cubesat. Neiva. 2007. p. 60. Trabajo de grado (Ingeniero Electrónico). Universidad Surcolombiana. Facultad de Ingeniería.

## 2.2 SOFTWARE DE PREDICCIÓN

Se hace referencia a aquellas aplicaciones que actúan como servidores, proporcionando la posición y visibilidad de satélites artificiales, por medio de cálculos matemáticos y el conocimiento de algunos parámetros de estos.

**2.2.1 Winorbit.** Un software de este tipo cuyo uso fue justificado en el trabajo de grado del diseño del control de desplazamiento es el Winorbit versión 3.6<sup>6</sup>, a pesar de ser uno de los más utilizados por los radioaficionados a nivel mundial, la versión que se encuentra disponible en Internet es la misma con la que se trabajó en los anteriores trabajos, por lo cual no hay veracidad en los elementos keplerianos<sup>7</sup> que trae incorporados, los cuales son los datos que utiliza el software para determinar la órbita y por consiguiente, calcular la posición del satélite en el espacio en un momento dado.

Una forma muy eficiente para la actualización de estos elementos se logró accediendo manualmente a la base de datos del Winorbit que se encuentra en la misma dirección de la aplicación » (**path winorbit \database\sat**), en un archivo llamado **database**. Dicha actualización se llevó a cabo reemplazando los elementos keplerianos por una actualización adecuada que se adquirió en la página de la **NORAD** » ([www.celestrak.com/NORAD/elements/](http://www.celestrak.com/NORAD/elements/)), la cual mostró resultados muy satisfactorios al comparar los datos de predicción que genera el programa Winorbit con datos de predictores en línea ubicados en páginas como la **AMSAT** » ([www.amsat.org/amsat-new/tools/predict/](http://www.amsat.org/amsat-new/tools/predict/)) y la **NASA** » (<http://science.nasa.gov/realtime/>).

Otro inconveniente que se presentó en la configuración del Winorbit es la diferencia de tiempo entre meridiano cero o meridiano de Greenwich y la hora local, correspondiente al punto donde está situada la estación terrena. Esta operación se corrigió en el menú principal del Winorbit, en **Setup>Time Zone** mostrando una ventana similar a la figura 2, donde se introduce la diferencia exacta que para la ciudad de Neiva es de 05 horas 01 minuto 12.19 segundos.

**2.2.2 Wxtrack.** Se utiliza por primera vez en este trabajo de grado. Su diseño tiene como objetivo predecir los rastros de los satélites, es decir los pasos por encima de la Tierra, y mostrar las imágenes producidas cuando estos "barren" la faz de la Tierra.

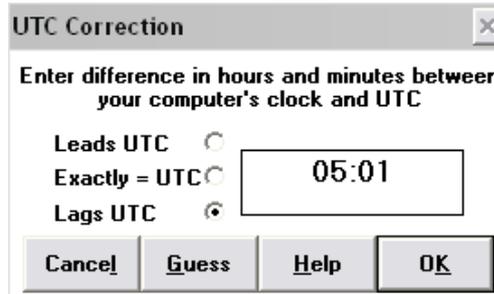
Puede leer imágenes de satélites procesadas de una o más carpetas de su elección, y presentar estas imágenes, una al lado de la otra, con la predicción del "barrido" en el terreno. Para hacer más fácil la correlación visual, son producidas marcas por minuto sobre estas imágenes. Pueden producirse máscaras de imágenes (overlays) apropiadas para en el futuro combinarlas con las imágenes reales de los satélites.

---

<sup>6</sup> GARCIA LOPEZ, Juan Gilberto y MONTEALEGRE VANEGAS, John Wilder. Diseño del sistema de control del desplazamiento y software de monitoreo de la estación terrena de la ciudad de Neiva del macroproyecto cubesat. Neiva. 2007. p. 58-59, 78-80. Trabajo de grado (Ingeniero Electrónico). Universidad Surcolombiana. Facultad de Ingeniería.

<sup>7</sup> Ibid., p. 39-40

**Figura 2. Diferencia horaria**



Este software de predicción es gratuito, aunque para soporte técnico y adquirir ciertas facilidades extras, se debe registrar el programa con algún costo. Sin embargo el programa que se ofrece de forma gratuita, funciona muy bien y tiene excelentes opiniones en Internet, de radio-aficionados, a cerca de su funcionamiento y precisión. La versión que se está manejando es la V3.6.8.853, desarrollada el 22 de febrero del 2008.

Este software no va a ser usado en este proyecto como el software predictivo principal, sino que será usado como un apoyo al sistema de control de la estación terrena para predecir la próxima pasada del satélite, ya que a este programa no se le pueden extraer los datos necesarios de azimut y la elevación en tiempo real, proceso que se puede hacer con cierta facilidad en el Winorbit.

Para ello el Wxtrack genera una base de datos en un documento de texto llamado **PassPrediction**, en donde se puede encontrar la latitud, la longitud, el corrimiento en doppler, la elevación, el azimut y la distancia, obteniendo estos datos por minuto en cada pasada. Como se quiere generar este archivo, hay que hacer ciertos ajustes al programa para que esté acorde a la realidad.

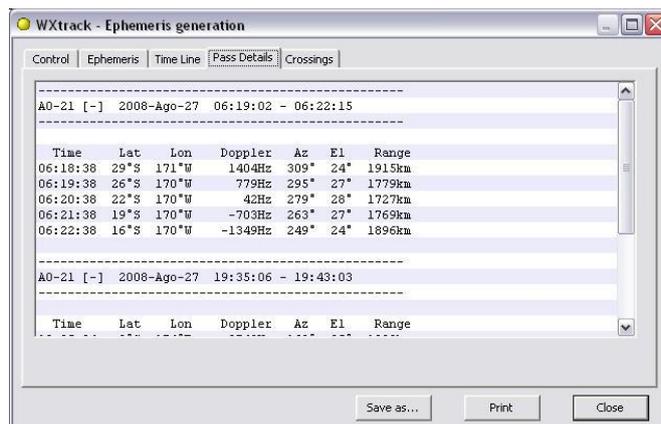
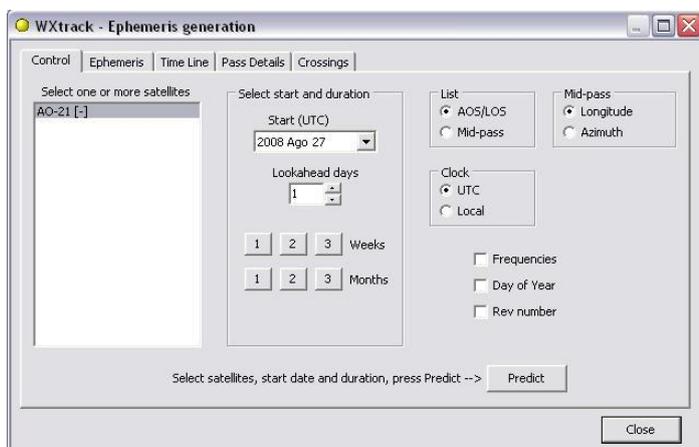
Primero se debe actualizar la base de datos propia del software en donde contiene los elementos keplerianos, los cuales son los que emplea para realizar sus cálculos de la posición y trayectoria de los satélites. Para ello se da clic en **Options** y seguidamente en **Update keplers**. Realizado esto, aparece una nueva ventana llamada **WxTrack-Update keplers from the Internet**, en la cual se seleccionan todas las casillas de los archivos a descargar y se revisa que en la casilla **From** se encuentre ingresada correctamente la siguiente dirección web: <http://www.celestrak.com/NORAD/elements/>, por último se le da clic en el botón **Update**. Cuando se están descargando los archivos aparecen unos mensajes de advertencia a los cuales se les debe simplemente dar clic en **Aceptar**.

Posteriormente se selecciona la pestaña **Setup** de la pantalla principal del **WxTrack**, y clic en el botón **Edit** del submenú **My locations details**. Aparece una nueva ventana en la cual se debe colocar las coordenadas geográficas de la ubicación de la estación terrena, en este caso 2°56'45.24" N de latitud y 75°18'14.79" W de longitud, el nombre de la estación, su peso y el mínimo de elevación en el cual el satélite se puede considerar visible. Por último

en la pantalla de la opción **Setup** se selecciona un satélite de la lista que está en **Available**, dando doble clic sobre el nombre de dicho satélite.

Una vez hecho esto, ya se puede visualizar la estación terrena y el satélite siguiendo su trayectoria correspondiente. Como lo que se desea es obtener la lista de las siguientes pasadas del satélite por la estación terrena que se estableció en el proceso anterior, se selecciona en **View** la opción **Ephemeris...** o en la pantalla **Setup** se da clic en el botón **Print Ephemeris...**, en ambos casos aparecerá la ventana **WxTrack- Ephemeris Generation**, como se muestra en la figura 3, en donde se puede escoger la fecha de inicio de la predicción y a partir de ahí cuantos días en adelante. Para obtener el archivo se da clic en el botón **Predict** y luego se selecciona la pestaña **Pass Details**, mostrada en la figura 3 en donde aparecerá la lista deseada para dicho satélite y para la estación terrena escogida. Esta lista se puede grabar en un archivo de texto seleccionando la opción **Save as...** o imprimirla en **Print**. Para este proyecto se debe guardar obligatoriamente dicha lista.

**Figura 3. Generación del archivo de texto**



### 3. SOFTWARE DESARROLLADO PARA LA INTERFAZ ICU G5500

Es importante en esta parte del trabajo, mencionar y reconocer el desarrollo que se hizo en los trabajos anteriores, a los que se ha referenciado de forma significativa ...en el capítulo 2..., como el trabajo de grado del diseño del control de desplazamiento y el trabajo de grado del diseño del control físico. Su trabajo realizado en el área de control, se basó en el desarrollo de dos programas que se encuentran explicados detalladamente en sus trabajos de grado y además son la base del funcionamiento de la interfaz ICU G5500. Uno de estos es utilizado en el microcontrolador PIC18F4550 (Unidad de proceso de la interfaz), y el otro realizado en Visual Basic 6.0, se sitúa en el PC permitiendo la comunicación USB<sup>8</sup> con el microcontrolador. Como de la validación y optimización de estos programas dependía en gran parte el éxito de este proyecto, se han desarrollado tres programas de los cuales solo dos buscan la mayor eficiencia de esta interfaz.

Estos dos, son los primeros que se explicarán en este capítulo, son versiones superiores y mejoradas de los realizados anteriormente. Por lo contrario el tercero no tiene ninguna relación con el dispositivo ICU G5500, ya que utiliza un hardware diferente mencionado por primera vez en este trabajo de grado y que por lo tanto, dicho programa junto con el hardware que utiliza, serán explicados de manera detallada ...en el capítulo 4...

Es fundamental mencionar que una vez estudiados los programas anteriores de forma detallada y bajo previa instalación de las herramientas de software de desarrollo ...de la sección 2.1..., se dio inicio a su optimización con el fin de desarrollar versiones superiores. Fue en este período que se presentó el primer inconveniente. Los dos programas no corrían en conjunto. Tras la aparición de este problema, se decidió optimizarlos de forma separada, parte por parte, siendo esta la metodología que se adoptó en el proyecto y por medio de la cual se explicarán las mejoras de cada programa con su debida explicación.

#### 3.1 SOFTWARE PARA EL PIC18F4550

La interfaz ICU G5500, siendo su cerebro principal el microcontrolador PIC18F4550, fue diseñada y elaborada cumpliendo con ciertas generalidades y características<sup>9</sup>. Su diseño fue pensado principalmente como medio de comunicación entre el PC y el Rotor Yaesu G-5500<sup>10</sup>. Teniendo en cuenta dichas características y buscando el mejor acople posible entre el funcionamiento del PC y la interfaz, se desarrolló un software para el microcontrolador, que presenta una mayor eficiencia con respecto al desarrollado anteriormente y alcanza de forma satisfactoria las metas de esta trabajo de grado.

---

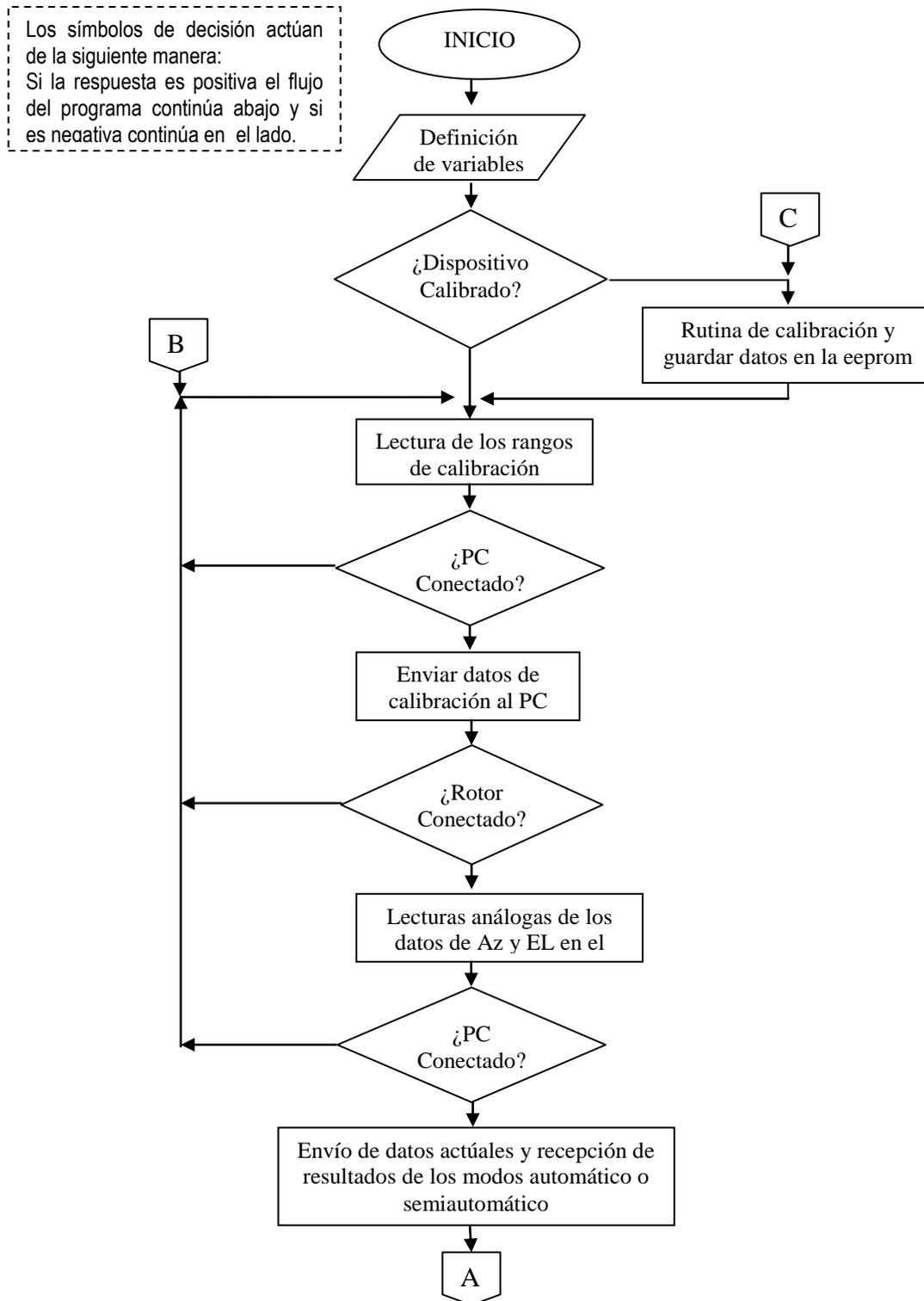
<sup>8</sup> ANDRADE VIDARTE, Diego Mauricio y NUÑEZ ROMERO, Leonardo Ernesto. Diseño del sistema de control físico para el posicionamiento de la antena de la estación terrena del proyecto cubesat en la universidad Surcolombiana. 2007. p. 41-46. Trabajo de grado (Ingeniero Electrónico). Universidad Surcolombiana. Facultad de Ingeniería.

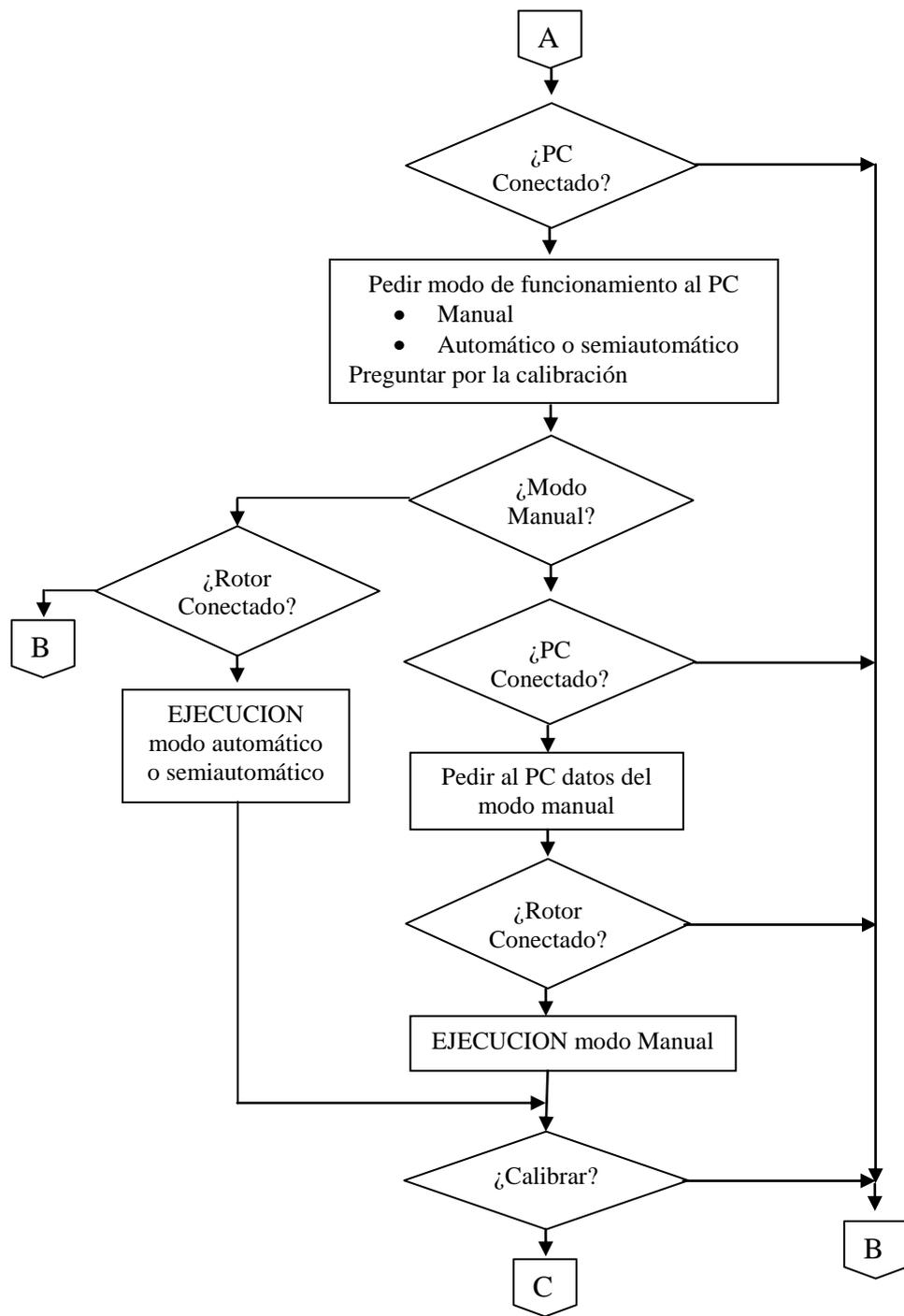
<sup>9</sup> Ibid., p. 33

<sup>10</sup> Ibid., p. 21-25

En la figura 4, se muestra el diagrama de flujo que se obtuvo, con el cual se pasara a explicar parte por parte el programa desarrollado para el microcontrolador.

**Figura 4. Diagrama de flujo interfaz ICU G5500**





**3.1.1 Definición de variables, calibración y lectura de rangos de calibración.** La primera sección del programa que representa el algoritmo de la figura 4 y parte inicial en casi todos los programas, es la sección de declaración de variables. Esta sección se debe encontrar libre de errores ya que una mala definición allí, puede afectar en gran parte el funcionamiento del programa.

Fue en esta sección donde se presentó un error en el momento en que se corrieron en conjunto los programas desarrollados en los trabajos de grado anteriores. Según las especificaciones del trabajo de grado del diseño del control físico, la calibración lleva de extremo a extremo<sup>11</sup> los dos actuadores del rotor YAESU G5500, acción que se efectuó de manera correcta con el azimut pero de manera incorrecta con la elevación.

Después de analizar el problema, se detectó que éste se encontraba en la sección de definición de variables, ya que los pines del microcontrolador que maneja la salida de elevación, se encontraban invertidos, a lo cual se dio solución reinvirtiéndolos, logrando superar el inconveniente y consiguiendo la corrección del programa como se muestra a continuación.

Forma incorrecta:

ELEVACION_UP	VAR PORTD.6	'Elevación hacia abajo.
ELEVACION_DOWN	VAR PORTD.7	'Elevación hacia arriba.

Forma correcta:

ELEVACION_UP	VAR PORTD.7	'Elevación hacia abajo.
ELEVACION_DOWN	VAR PORTD.6	Elevación hacia arriba.

Una vez definidas todas las variables del programa de manera correcta, la sección o rutina que continúa es la de calibración.

Se puede decir que como la calibración tanto de azimut como de elevación siguen un proceso similar, se hará referencia solo a una de ellas (elevación), la cual es la primera que realiza el programa en el microcontrolador.

La primera acción que ejecuta el microcontrolador antes de la calibración y seguido de la definición de variables, es preguntar si la interfaz ya ha sido calibrada, esto se realiza para evitar que todas las veces que se reinicie, ingrese de forma inevitable a esta sección. Cabe mencionar que el dispositivo entra a calibrarse solo cuando es arrancado por primera vez o por decisión del usuario que se encuentra manipulando el programa del PC realizado en Visual Basic, lo cual se explicará ... en la sección 3.2... con más detalle.

---

<sup>11</sup> ANDRADE VIDARTE, Diego Mauricio y NUÑEZ ROMERO, Leonardo Ernesto. Diseño del sistema de control físico para el posicionamiento de la antena de la estación terrena del proyecto cubesat en la universidad Surcolombiana. 2007. p. 39. Trabajo de grado (Ingeniero Electrónico). Universidad Surcolombiana. Facultad de Ingeniería.

Una vez el microcontrolador entra en la sección de calibración, publica un letrero en la pantalla LCD de la interfaz ICU G5500 y envía un mensaje al Programa del PC, donde indica que el proceso que se encuentra realizando es el de calibración. Ejecutada esta acción, se procede a mover el actuador correspondiente a la elevación a un valor igual o cercano a cero grados y así, tomar el valor entregado por el conversor análogo digital del microcontrolador como el primer valor de calibración llamado “valor mínimo de elevación”. Se recuerda que el PIC18F4550 utilizado en la interfaz ICU G5500 posee un conversor análogo digital de 10 bits, por lo cual el valor que se lee de éste, es una medida proporcional al valor real y no una igual, teniendo que hacer su respectiva transformación para obtener el valor verdadero.

Terminado este proceso, el microcontrolador hace girar el actuador de elevación hasta que llegue a su valor máximo, esto se realiza esperando un tiempo determinado ya que el tiempo que toma esta acción no varía en más de un segundo en repetidas ocasiones, tiempo que aproximadamente es de 67seg. y proporciona el valor máximo de elevación. La calibración del azimut se realiza de la misma forma que la elevación y al terminar ambas, el microcontrolador escribe los cuatro datos obtenidos en la calibración (mínimo de elevación, máximo de elevación, mínimo de azimut y máximo de azimut) en su memoria **EEPROM**.

A diferencia del programa para el microcontrolador, desarrollado en los trabajos de grado anteriores donde el estado del rotor solo se revisaba al comienzo de la rutina de calibración<sup>12</sup>, la rutina desarrollada en este proyecto, revisa en tiempo real el estado del rotor, dando aviso por medio de la pantalla LCD en la interfaz y al PC por medio de mensaje en Visual Basic, cuando éste se desconecta, parando el flujo del programa, hasta que el rotor se conecte y así iniciar el proceso de nuevo. Esta mejora es de suma importancia, ya que en la calibración se utilizan tiempos exactos, y la desconexión del rotor sin ser detectada en el transcurso de la rutina generaría medidas erróneas de los valores máximos y mínimos tomados en la calibración.

Los mensajes en el Visual Basic que indican la desconexión del rotor se han agregado en el programa actual y no hacían parte del anterior.

Por último se sigue con la lectura de rangos de calibración, aunque no es muy extensa, tiene la función de almacenar en variables, los valores tomados en la calibración, que se encuentran en la **EEPROM** del microcontrolador y así asegurar que al reiniciarse el microcontrolador estos datos se carguen y puedan ser usados en el programa.

**3.1.2 Detección de PC conectado y envío datos de calibración al PC.** La forma en que se comparten los datos entre el microcontrolador y el programa del PC se puede resumir de la siguiente manera.

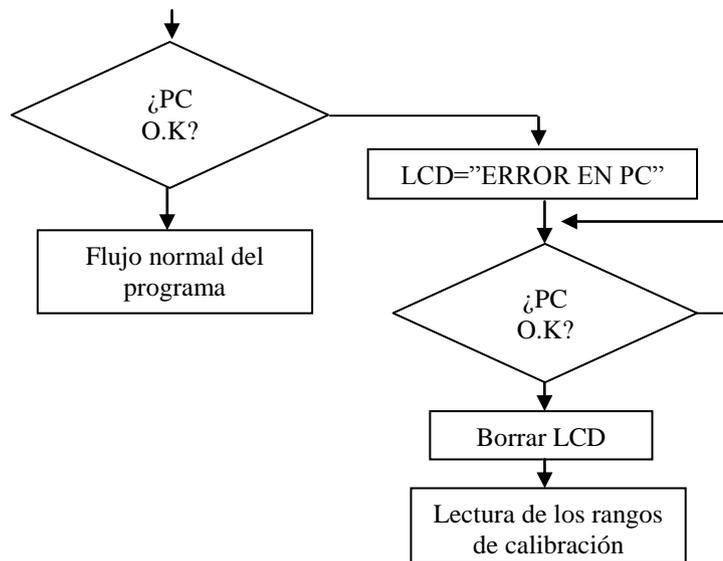
---

<sup>12</sup>Código de programa en: ANDRADE VIDARTE, Diego Mauricio y NUÑEZ ROMERO, Leonardo Ernesto. Diseño del sistema de control físico para el posicionamiento de la antena de la estación terrena del proyecto cubesat en la universidad Surcolombiana. 2007. p. 62-63. Trabajo de grado (Ingeniero Electrónico). Universidad Surcolombiana. Facultad de Ingeniería.

En el envío, el microcontrolador manda datos cuando desea, siendo recibidos en aquel momento por el programa del Visual Basic en el PC. Sin embargo en la recepción, el microcontrolador debe realizar una petición al PC por medio de un envío de datos, a los cuales este responde proporcionándole los datos que necesite. En el PC la situación es mucho más sencilla, este solo envía datos cuando el microcontrolador se los pide.

Con este mecanismo de comunicación se podría asegurar que en ningún momento existirá conflicto en la transmisión, ya que una de las partes solo transmite cuando la otra lo solicita. Sin embargo si se desea conocer en el microcontrolador el momento en que se abre o cierra el programa del PC, es necesario que este programa en ciertas ocasiones, envíe algún dato de forma independiente, que dé a conocer estas acciones. Este hecho es una de las causas por la cual, antes de cada acceso al bus **USB** por parte del microcontrolador, ya sea para enviar o para recibir, se coloca una rutina que detecta la conexión con el PC, la cual se observa con más detalles en la figura 5.

**Figura 5. Detección de PC conectado**



Otra razón por la cual se ubicó la rutina **Detección de PC conectado** de esta forma, es por el inconveniente que presentaba el microcontrolador en el programa de los trabajos anteriores, que cuando realizaba una petición de datos y el programa del PC no se encontraba disponible (el programa se había cerrado), el microcontrolador se quedaba en un bucle infinito, ya que no hay dato disponible al ingresar en la rutina de recepción de datos, como se ve y explicara a continuación.

Código para recepción de datos en el microcontrolador

```

DoUSBIn:
USBBufferCount = USBBufferSizeRX          ' RX buffer size
USBService      ' keep connection alive
  
```

```

USBIn 1, USBBuffer, USBBufferCount, DoUSBIn      ' read data, if available
return

```

Refiriéndose a la explicación que proporciona la ayuda del **MicroCode Studio--PICBASIC PRO** mencionado ...en la sección 2.1..., la palabra **DoUSBIn** que se encuentra en la cuarta línea del código, es el nombre de la etiqueta o salto al cual se dirigirá la secuencia del programa cuando no haya algún dato disponible, salto que se encuentra en la primera línea de código y convierte la rutina de recepción en un bucle infinito.

Siguiendo con el envío de los datos de calibración al PC, es importante decir que los cuatro datos de los que se hablaba ...en la sección 3.1.1..., son usados para hallar el equivalente en milivoltios de cada grado del rotor, tanto en azimut como elevación. Esto se hace restando el valor máximo obtenido menos el valor mínimo, dividido por 450 o 180, si es para azimut o elevación respectivamente, los valores obtenidos son los divisores de los datos entregados por el conversor análogo digital y de esta manera, se pasan al valor correspondiente en grados.

La rutina **envío datos de calibración al PC**, toma estos cuatro valores y se los envía al programa del PC para que realice esta operación. Si la operación se realiza en el mismo microcontrolador, como se hacía en el trabajo de grado del diseño del control físico y como se muestra a continuación:

```

WRITE 10, (MAX_AZ-MIN_AZ)/450
WRITE 11, (MAX_EL-MIN_EL)/180

```

Se pierden decimales importantes, ya que la programación en **PicBasic de Microcode** solo trabaja con enteros, y al realizar esta división en el microcontrolador, solo se obtendría el modulo del resultado.

Para entender más fácilmente la magnitud del problema, se calibró varias veces la interfaz, de donde se obtuvieron los valores de calibración que se muestran en el cuadro 1.

### Cuadro 1. Valores de calibración

Nombre del dato	Valor
Máximo _ azimut	1022
Mínimo _ azimut	2
Máximo_elevación	1023
Mínimo _ elevación	3

Si se realizan las operaciones en el microcontrolador y teniendo en cuenta que el **PicBasic** solo trabaja con enteros, obtendríamos el siguiente resultado:

Para azimut =  $(1022-2)/450 = 2$   
 Para elevación =  $(1023-3)/180 = 5$

Siendo los valores correctos de la operación, los que se ven a continuación y los cuales se obtienen con el programa del PC:

$$\text{Para azimut} = (1022-2)/450 = 2.26666$$

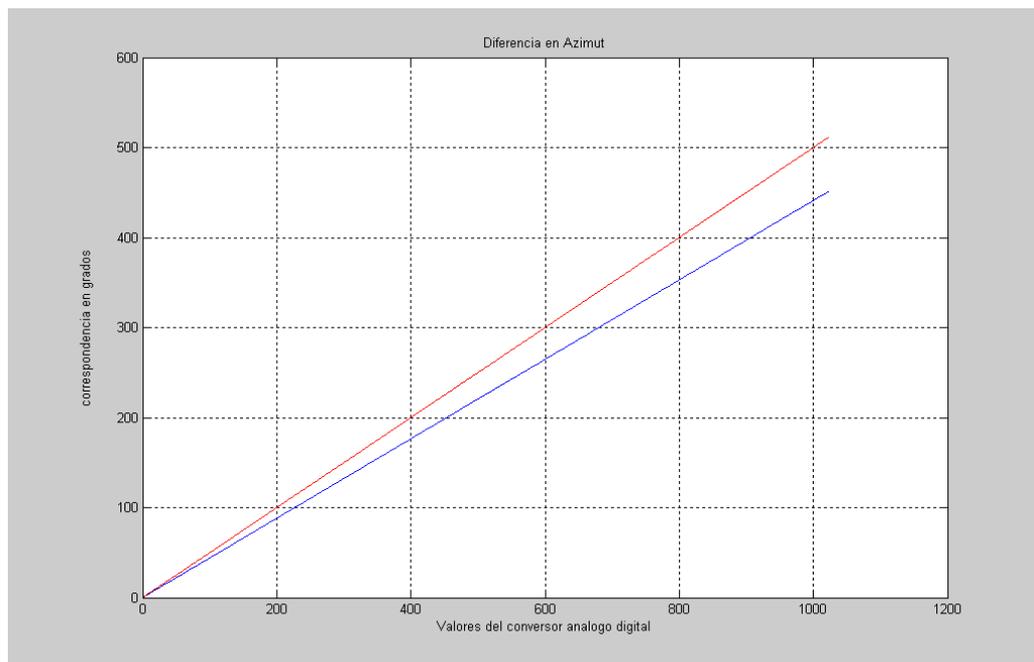
$$\text{Para elevación} = (1023-3)/180 = 5.66666$$

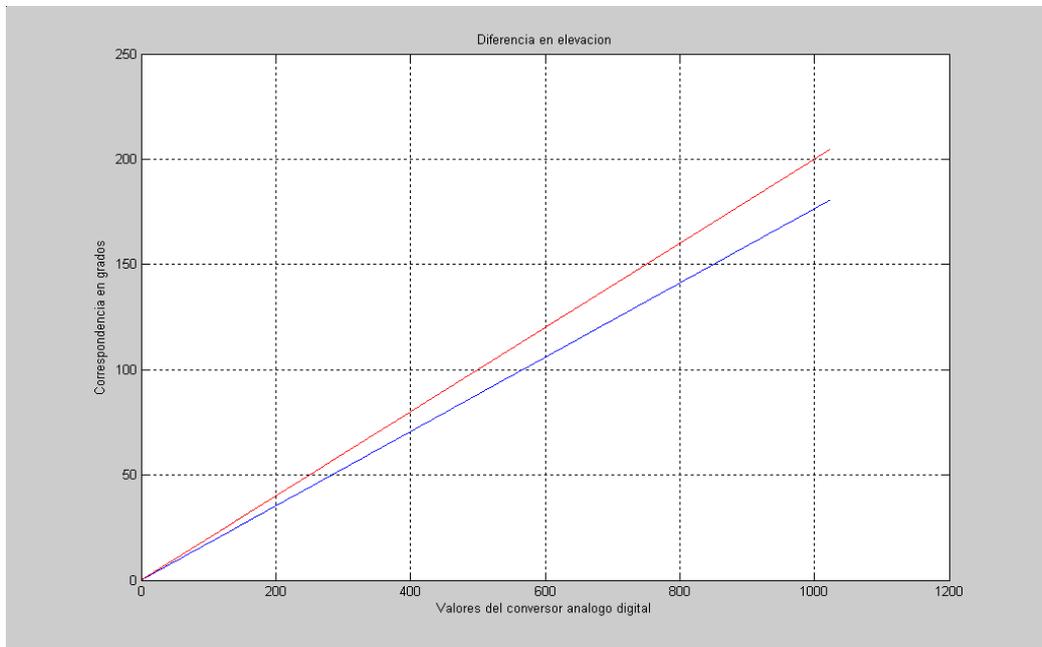
Como se ve hay una gran diferencia, ya que se pierden 0.26666 para azimut y 0.6666 para elevación.

Ahora si se divide los 1024 valores que entrega el conversor, tanto en azimut como en elevación, por los resultados de las operaciones anteriores (correctos realizados en el programa del PC y los incorrectos del trabajo de grado del diseño del control físico) y se grafican los resultados, se obtiene la figura 6.

En la gráfica se observa que la diferencia entre el valor real, correspondiente al color azul y el valor erróneo, correspondiente al color rojo es demasiada, razón por la cual dicha operación se le asigna al programa del PC realizado en Visual Basic y no al microcontrolador.

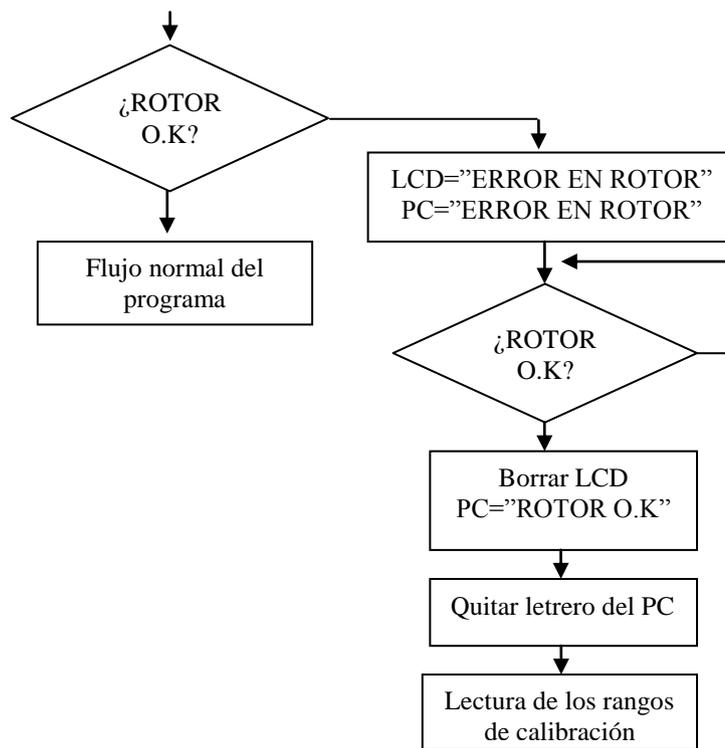
**Figura 6. Gráficas de las dispersiones**





**3.1.3 Detección de rotor conectado, lecturas análogas, envío de datos actuales y recepción de resultados.** La conexión con el rotor se encuentra presente a lo largo del programa, su comportamiento se puede observar en el diagrama de flujo de la figura 7.

**Figura 7. Detección de rotor conectado**



A consideración de los autores la ubicación de esta rutina antes de acceder al rotor, presenta mayor eficiencia. Esta decisión fue tomada después de realizar muchas pruebas, siendo la que mejores resultados proporcionó. A diferencia de la rutina anterior, la que se utiliza en este proyecto le indica al programa del PC realizado en Visual Basic cuando sucede una interrupción en la operación el rotor.

Otra rutina importante es **envío de datos actuales y recepción de resultados**, en la cual se leen los valores actuales que entrega el conversor análogo digital tanto en azimut como elevación y a diferencia del programa anterior, no realiza la respectiva conversión a grados, por razones obvias mencionadas ...en la sección 3.1.2....

Terminada estas lecturas, el microcontrolador envía los valores obtenidos al programa del PC, el cual realiza la conversión a grados y la comparación con los valores deseados, devolviéndole al microcontrolador los datos necesarios para ubicar el rotor en la posición correcta. Estas operaciones se explicarán con más detalle ...en la sección 3.2.2...

Los datos obtenidos en esta rutina, se utilizan para mover el rotor en los modos automático y semiautomático que se verán a continuación.

**Modo automático-semiautomático, manual y calibración.** Como se había mencionado, los datos para la ubicación del rotor en los modos automático o semiautomático, se obtienen en la rutina **envío de datos actuales y recepción de resultados**. Tal operación se realiza antes de seleccionar cualquier modo, ya que el modo automático es el más utilizado por el usuario.

Al observar el diagrama de flujo de la figura 4, se observa que los modos automático y semiautomático utilizan la misma rutina de ejecución, esto se debe a que los datos enviados por el PC al microcontrolador para ubicar el rotor, en ambos modos dan a conocer de forma similar el estado en que debe ser colocado el rotor, siéndole al microcontrolador indiferente el modo del que provengan.

Cuando los datos llegan a la rutina de ejecución, sin importar cual sea (automático o semiautomático), proporcionan al microcontrolador la información para que mueva el rotor de forma correcta, siendo allí donde se controla el movimiento del rotor tanto en azimut como en elevación, de acuerdo a tres estados para cada movimiento como se muestra en el cuadro 2.

**Cuadro 2. Estados de movimiento**

<b>Movimiento</b>	<b>Estado 1</b>	<b>Estado 2</b>	<b>Estado 3</b>
Elevación	Pare	Arriba	Abajo.
Azimut	Pare	Izquierda	Derecha

El otro modo de operación utilizado en el programa es manual, diferenciándose de los anteriores por la intervención del usuario en el programa del PC, quien es el encargado de mover el rotor.

Una vez que el microcontrolador entra al modo manual, se comunica con el PC, pidiéndole el estado al que debe mover el rotor, de acuerdo a la interacción que el usuario se encuentre realizando con la interfaz de usuario, ubicada en el programa del PC explicada mas adelante. Cuando dicho estado es entregado al microcontrolador, el rotor es desplazado por la rutina **Ejecución modo manual** que de forma parecida a los modos semiautomático o automático mueve el rotor, con la diferencia que en esta rutina el estado de parada es el mismo para los dos movimientos, ya que el usuario solo puede realizar un movimiento a la vez.

Como se ve en la figura 4, en el mismo momento en que se define el modo de funcionamiento, se solicita el estado de la calibración, siendo este valor el que se utiliza para tomar la decisión en la última acción que realiza el programa del microcontrolador, la cual pregunta si se desea calibrar de nuevo el sistema. Allí, puede tomar esta decisión o en caso contrario se devuelve a leer los datos de calibración iniciando de nuevo el flujo del programa.

### 3.2 SOFTWARE DESARROLLADO PARA EL PC

El siguiente software desarrollado, ha sido mencionado ... en la sección 3.1... como el **programa del PC realizado en Visual Basic**. Este programa actúa como interfaz de usuario en el PC y además, realiza las operaciones mencionadas ..en la sección 3.1.2..., en busca del correcto funcionamiento del sistema. Como su programación fue realizada en Visual Basic, el cual es un lenguaje de programación orientado a objetos, y basado en eventos, su representación de casos de uso se muestra en el anexo B y su explicación se basará en los tres eventos más importantes que contiene y serán explicados en las siguientes secciones.

En la figura 8 se muestra la interfaz de usuario que presenta dicho programa y a continuación se explicará de manera general sus partes y funcionamiento. Se puede observar el sistema de brújulas que posee el programa, las cuales permiten visualizar gráficamente los valores de la ubicación actual del rotor tanto en azimut como en elevación, presentando también los valores numéricos encima de cada una.

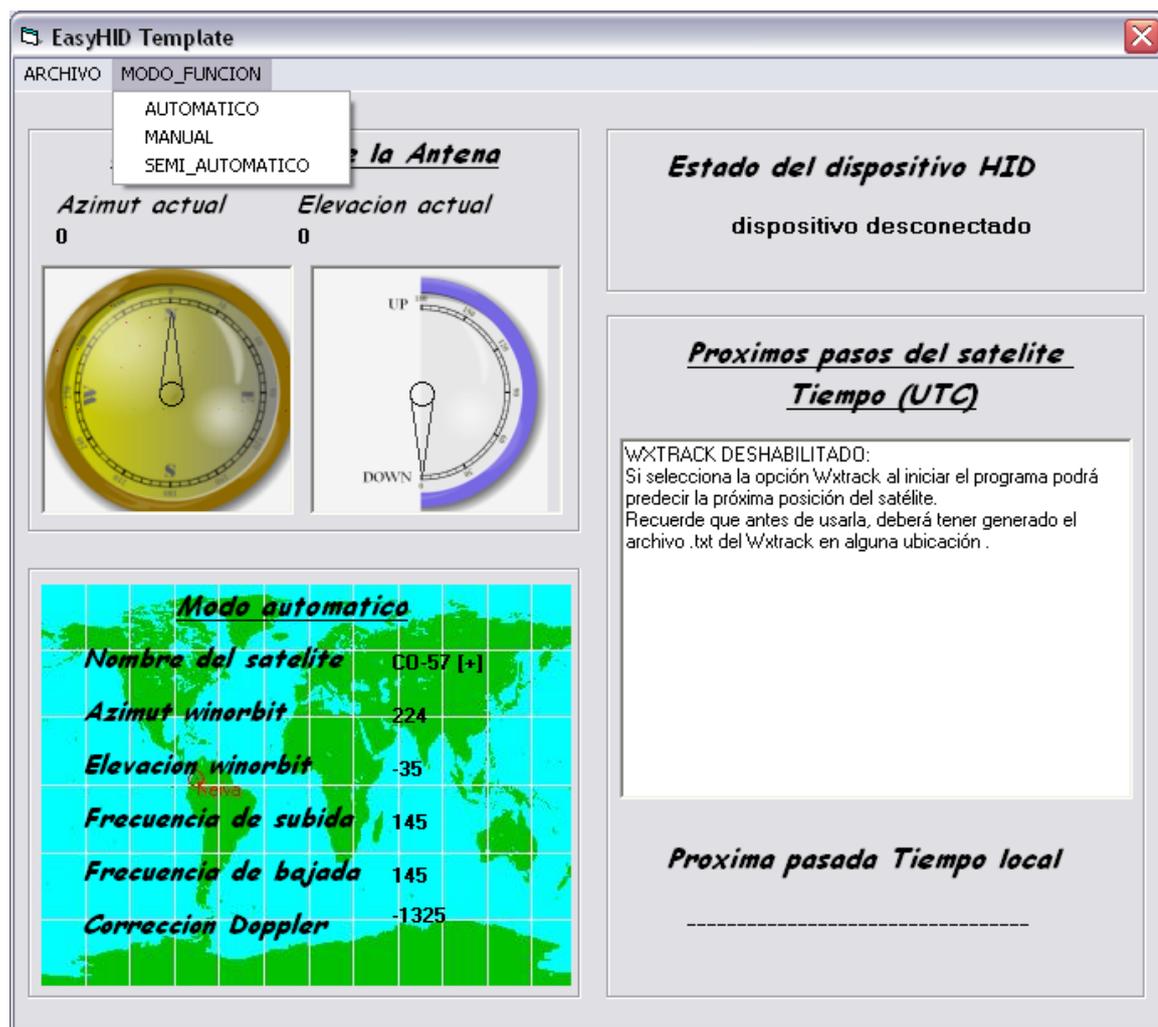
Otro elemento importante que además contiene la interfaz de usuario, es la barra de menú, en la cual se puede escoger si se desea calibrar de nuevo la interfaz ICU G5500 o el modo de funcionamiento, que a diferencia del programa que existía anteriormente, no contaba con el modo semiautomático. También se puede apreciar, que el modo de funcionamiento por defecto que posee la interfaz es el automático, el cual toma los valores de azimut y elevación del software de predicción (Winorbit) por medio del protocolo DDE<sup>13</sup> para procesarlos mas adelante y enviar el resultado al microcontrolador, siendo este modo el más importante y

---

<sup>13</sup>GARCIA LOPEZ, Juan Gilberto y MONTEALEGRE VANEGAS, John Wilder. Diseño del sistema de control del desplazamiento y software de monitoreo de la estación terrena de la ciudad de Neiva del macroproyecto cubesat. Neiva. 2007. p. 78. Trabajo de grado (Ingeniero Electrónico). Universidad Surcolombiana. Facultad de Ingeniería.

más utilizado por el usuario. Sin embargo, al seleccionar cualquiera de los otros modos en la barra de menú, aparecerá una ventana sobre la correspondiente al modo automático, la cual tomara el lugar de esta, pasando a ser la utilizada por el programa. Esta misma acción se repetirá con cualquiera que sea el caso como se observa en el diagrama de secuencia del anexo D.

**Figura 8. Interfaz de usuario**



Si se elige el modo semiautomático en al barra de menú, aparecerá una ventana como la que se muestra en la figura 9.

En este modo, se puede ingresar la posición en la cual se desea ubicar los dos actuadores del rotor. Una vez se hayan ingresado dichos valores, se da clic en el botón **INGRESAR VALOR**, e inmediatamente el rotor se ubicara en la posición de azimut y elevación ingresada.

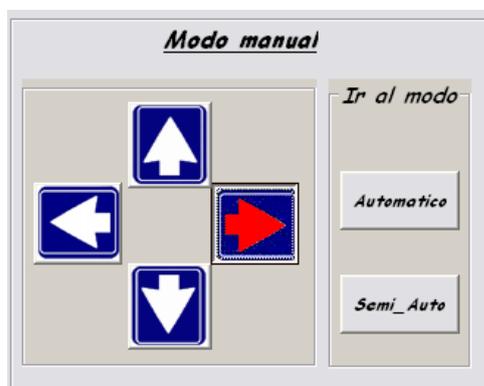
Seleccionando el modo manual aparecerá una ventana como la que se muestra en la figura 10. Es allí donde se puede mover manualmente el rotor, seleccionando con el Ratón cualquier estado de los cuatro posibles (arriba, abajo, izquierda y derecha). Después de haber realizado esta selección, el rotor se moverá hacia el estado seleccionado, que en el caso de la figura 10 es el derecho.

**Figura 9. Modo semiautomático**



En la figura 8, se puede apreciar que hay un cuadro que indica el estado del dispositivo HID (PIC18F4550), el cual siempre estará visible permitiéndole al usuario conocer en cualquier momento el estado de la interfaz ICU G5550.

**Figura 10. Modo manual**

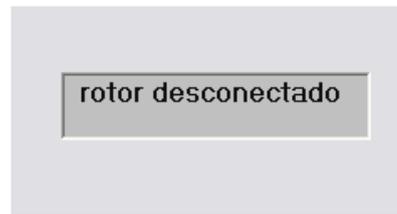


Como se menciona ...en la sección 2.2..., el software de predicción Winorbit entrega datos confiables acerca de la posición de azimuth y elevación del satélite en el espacio. Sin embargo un problema que se presentó, fue el hecho de que Winorbit no deja obtener la próxima pasada del satélite, situación que presentaba inconveniente en el momento en que el rotor, se encontraba esperando la próxima pasada del satélite en una posición lejana a la del paso, tardando segundos en ubicarse y perdiendo tiempo valioso para seguirlo. Este inconveniente se puede afrontar cargando al comienzo del software desarrollado, lo cual se explicará ...en la sección 3.2.1..., el archivo **PassPrediction.txt** que entrega el Wxtrack, y utilizando sus datos de predicción se ubica el rotor en la posición del siguiente paso del

satélite, evitando la pérdida de tiempo y logrando un mejor seguimiento. Cuando se carga el archivo **PassPrediction.txt**, se puede observar en una ventana del software desarrollado, sin embargo hay que tener en cuenta que las fechas que aparecen allí, están referenciadas al tiempo UTC (tiempo del meridiano cero) y es por esta razón que debajo de dicha ventana, se muestra la fecha de la siguiente pasada, esta vez en tiempo local (tiempo de la ciudad de Neiva).

Otros beneficios que posee el programa del PC, son una serie de letreros, que indican el estado de conexión del rotor en la interfaz ICU G5500, como el que se ve en la figura 11, que aparecen cuando el rotor se desconecta y desaparecen cuando se conecta.

**Figura 11. Letrero de conexión del rotor**



**3.2.1 Evento form load.** Cada vez que se carga un formulario en una aplicación de Visual Basic, éste genera una serie de eventos dentro de los cuales se encuentra el evento **form load**. Aunque este evento no es el primero que se genera, es el más usado en las aplicaciones porque es el más apropiado para introducirle los valores y parámetros iniciales en un programa.

En la aplicación desarrollada en el presente proyecto, el evento **form Load** es el primero en ejecutarse, ya que en dicha aplicación no se utilizaron los eventos que el Visual Basic genera antes de él, por lo cual su explicación se realiza primero que los demás eventos. En el anexo C se representa el diagrama de secuencia que se relaciona con dicho evento.

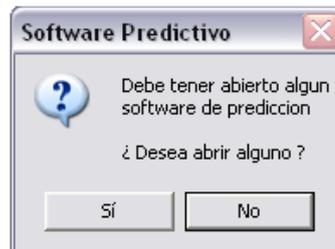
Lo primero que realiza este evento es iniciar algunas variables que se utilizan a lo largo del programa y ocultar los letreros que hacen referencia al rotor como el que se vio en la figura 11. Seguido de este proceso, el programa realizado en el trabajo de grado del diseño del control físico presentaba el siguiente inconveniente: si se iba a utilizar el programa del PC, se debía tener abierto con anterioridad el Winorbit<sup>14</sup>. Este problema se resolvió, mostrando una caja de texto como se muestra en la figura 12, que le recuerda al usuario el uso del software de predicción, y le pregunta si desea abrir alguno, en el caso de no tenerlo abierto.

La acción de abrir otras aplicaciones, se realiza desde el mismo programa del PC

---

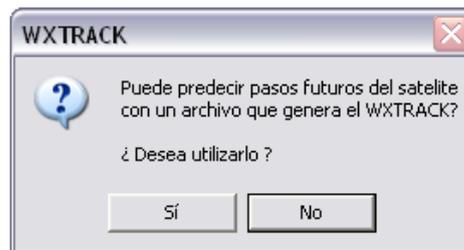
<sup>14</sup> Para mayor información: ANDRADE VIDARTE, Diego Mauricio y NUÑEZ ROMERO, Leonardo Ernesto. Diseño del sistema de control físico para el posicionamiento de la antena de la estación terrena del proyecto cubesat en la universidad Surcolombiana. 2007. p. 36. Trabajo de grado (Ingeniero Electrónico). Universidad Surcolombiana. Facultad de Ingeniería.

**Figura 12. Primer mensaje de advertencia**



Una vez abierto el Software predictivo, se utiliza un mensaje igual que en la figura 13, para informar sobre la posibilidad de predecir los pasos futuros del satélite, por medio de un archivo del Wxtrack e invitar a usarlo.

**Figura 13. Segundo mensaje de advertencia**



A diferencia del Winorbit, el Wxtrack proporciona beneficios para el funcionamiento del programa del PC y no es esencial para su funcionamiento, por lo cual el usuario puede rechazar la utilización, y trabajar sin él. Al presentarse este caso (el usuario rechaza el uso del Wxtrack), el evento **form load** seguirá con la siguiente acción, ignorando todo lo referente al Wxtrack. En caso de aceptarlo, el usuario debe cargar por medio del programa del PC realizado en Visual Basic, el archivo **PassPrediction** que genera el Wxtrack, en el cual se encuentran los próximos pasos del satélite y será generado con anterioridad por el usuario. Tras esto, el archivo **PassPrediction** es utilizado por el PC, para generar otro archivo **.txt** en el cual almacenara todas las pasadas del satélite y con el cual trabajara en el resto del Programa.

Antes de finalizar el evento **form load**, el programa del PC establece la conexión con el host controlador USB, siguiendo con el envío de un paquete de datos por el bus, el cual le indica al dispositivo HID (PIC18F4550) que se ha conectado.

Terminada estas acciones, el programa inicializa otras variables utilizadas en el programa, e inicializa el intervalo del evento **Timer** que se verá ...en la sección 3.2.3....

**3.2.2 Evento on read.** Con respecto a la comunicación entre el microcontrolador y el Programa del PC, se puede asegurar que hay dos formas de comunicación entre ellos. La unidireccional, del microcontrolador al programa del PC y la bidireccional, del microcontrolador al programa del PC y viceversa. Con un sistema de comunicación como este, debe existir un mecanismo de control, que permita saber que hacer con los diferentes tipos de datos que se reciben, esto requerimiento se logró mediante la utilización de un byte de control, ubicado en el byte cero del buffer USB del microcontrolador, y en el byte uno del buffer USB del programa del PC<sup>15</sup>. Mediante este byte de control utilizado en ambas partes, el microcontrolador le indica al programa del PC la función que debe realizar con los datos que le envía. Todas estas funciones que debe realizar el programa del PC, se llevan a cabo en el evento **on read**, el cual es un evento que genera la plantilla del **EasyHID**, que se ejecuta cada vez que haya un envío de datos por parte del microcontrolador y donde se manipulan los datos de control mencionado anteriormente asociándole la tarea respectiva. Esta manipulación se realiza por medio de un seleccionador (**select case**) dentro del evento, al que se le asigna un dato de control por cada **case**, y dentro del cual se ubica la tarea correspondiente.

Las tareas que debe realizar el programa del PC ante un envío de datos por parte del microcontrolador, se pueden clasificar en dos y son: las que ejecutan alguna acción en el Programa del PC pero no devuelven resultado al microcontrolador y las que ejecutan alguna acción en el programa del PC y devuelven resultado al microcontrolador.

Para mayor detalle, se muestran las tareas del evento **on read** con su respectivo número de control en el cuadro 3 y en el anexo E. se muestra su diagrama.

**Cuadro 3. Tareas realizadas**

<b>Dato de control enviado por el microcontrolador en el Buffer(0)</b>	<b>Acción realizada en el programa del PC en el evento on read</b>	<b>Respuesta que da el programa del PC al microcontrolador</b>
Buffer(0)=1	Tomar el modo de funcionamiento y estado de la calibración, seleccionado por el usuario en el PC	Enviar modo de funcionamiento y el estado de calibración actual al microcontrolador
Buffer(0)=3	Tomar datos de azimuth y elevación actual enviados por el microcontrolador y compararlos con los valores deseados del PC.	Enviar resultado de comparación: estados en los que se debe ubicar azimuth y elevación en el rotor.
Buffer(0)=7	Tomar el estado en el que se encuentra el control manual del PC	Enviar el estado del control manual al microcontrolador
Buffer(0)=8	Publicar en letrero del PC "Calibrando"	Sin respuesta
Buffer(0)=9	Publicar en letrero del PC "Calibración terminada"	Sin respuesta

<sup>15</sup> Tanto el microcontrolador como el programa del PC, poseen un arreglo de bytes para la transmisión y recepción de datos, el cual es llamado buffer USB

Buffer(0)=10	Publicar en letrero del PC "rotor desconectado "	Sin respuesta
Buffer(0)=11	Tomar los datos de calibración enviados por el microcontrolador y hallar el equivalente en milivoltios de cada grado del rotor para azimut y elevación	Sin respuesta
Buffer(0)=12	Quitar letrero del PC	Sin respuesta
Buffer(0)=13	Publicar en letrero del PC "rotor OK "	Sin respuesta

**Tareas que no devuelven respuesta.** No hay duda que las tareas que no devuelven respuesta al microcontrolador, son en su mayoría aquellas que tienen que publicar algún aviso en el programa del PC. Estas tareas son de gran importancia ya que le indican al usuario cuando se ha presentado alguna desconexión en el rotor, permitiéndole hacer la revisión correspondiente. Algunos de estos letreros son como el que se ve en la figura 11, que han sido agregados en el programa del PC desarrollado por este proyecto, ya que no eran usados en el anterior.

Pero no todas las tareas sin respuestas corresponden a letreros, una muy importante y cuyas razones de uso son explicadas ...en la sección 3.1.2..., es la correspondiente al dato de control 12, en la cual se halla el equivalente en milivoltios de cada grado del rotor. Esta es una de las primeras tareas que pide realizar el microcontrolador, ya que el resultado que produce es utilizado por otras tareas en este evento.

**Tareas que devuelven respuesta.** Estas tareas son solo tres en el evento **on read**. Las que tienen los datos de control 7 y 3 responden a las peticiones del modo manual y el modo automático o semiautomático en el microcontrolador, respectivamente.

La tarea correspondiente al modo manual, tiene una operación muy sencilla, que consiste en entregarle al microcontrolador el estado en el que se encuentre este modo. Como se observa en la figura 10 se encuentra presionado el botón correspondiente a la derecha, por lo cual la tarea debe de informarle al microcontrolador que éste es el estado del dicho modo.

Sin embargo en el modo automático o semiautomático es diferente. La tarea obtiene los valores que vienen del microcontrolador, correspondientes a la ubicación actual del rotor y los transforma a grados utilizando el resultado obtenido en la tarea cuyo dato de control es 12. Una vez realizada la conversión, se procede a comparar los valores actuales del rotor con los valores deseados, ya sean tomados del Winorbit en el modo automático o de las cajas de texto en el modo semiautomático. La comparación entrega como resultado los estados vistos ...en la sección 3.1.3..., en los que el microcontrolador debe colocar el rotor. De manera similar a las razones escogidas ...en la sección 3.1.2..., para calcular el equivalente en milivoltios de cada grado del rotor, la sección de comparación se realiza en el PC, ya que al realizarse en el microcontrolador como se realizaba anteriormente, se pierden cifras significativas lo cual lleva a un error en más de dos grados en la precisión para valores grandes de azimut y elevación, presentando errores indeseados en la ubicación del rotor.

Por último, la tarea cuyo dato de control es uno, se encarga de proporcionar al microcontrolador el estado que indica si hay deseo de calibración por parte del usuario y el modo en el que se encuentra funcionando el programa del PC.

Las tareas, a pesar de estar ubicadas en un evento, se ejecutan en el orden en que los solicita el microcontrolador, lo cual se observa de forma clara en el diagrama de flujo de la figura 4.

**3.2.3 Evento Timer.** Al evento **Timer** se le puede comparar con un temporizador ya que se ejecuta cada tiempo determinado. Este tiempo es fijado en su intervalo, en la propiedad **Timer.interval**, que para el presente proyecto se colocó de 100 ms. y cuyo funcionamiento en el programa del PC es muy sencillo, encontrándose dividido en tres partes como se muestra en el diagrama del anexo F.

La primera de ellas, es la sección en la que se obtienen los datos del Winorbit por medio del protocolo DDE. Este software predictivo, actúa como un servidor proporcionando los datos de azimut y elevación del satélite. El programa del PC puede acceder a estos datos cuando desee, sin embargo en este caso y debido al evento **Timer** se obtienen cada 100 ms. Siendo este tiempo mas que suficiente y con el cual se obtuvieron muy buenos resultados.

Una vez conseguidos los datos del Winorbit, se procede con la detección de la siguiente pasada del satélite. Recordando que esta parte se realiza solo si el Wxtrack ha sido seleccionado al comienzo del programa. Si se ha seleccionado, esta sección se ejecuta en el momento en que la elevación pasa de un valor negativo a un valor negativo mayor, este instante se da cuando el movimiento del satélite toma dirección hacia la estación terrena. Una vez la elevación se comporte de esta manera, el programa del PC obtiene la fecha actual del computador y realiza la comparación con el archivo **.txt** mencionado ...en la sección 3.2.1... y que el programa genera en el evento **form load**. Dicha comparación, obtiene la fecha correspondiente a la siguiente pasada del satélite y tomando los respectivos valores de azimut y elevación, se ubica el rotor en esta posición.

Por último se pasará a explicar el funcionamiento de las brújulas, animaciones que se habían tomado en los trabajos de grado anteriores para mostrar la posición actual del rotor. El inconveniente que se encontró es que se necesitaba un programa externo para poderlas operar, ya que dichas animaciones estaban desarrolladas en el programa Macromedia Flash, es decir que como mínimo el PC debería tener instalado los **plug-in** que esta animación requería, y para los programadores, que en cierto caso quisieran desarrollar algunos cambios, deberían instalar el programa, aprenderlo a manejar, y como es un proyecto universitario, se debe contar con la licencia para el uso de dicho software. Viendo este inconveniente y ya que la universidad cuenta con la licencia del Visual Studio 2005, se optó por realizar la misma animación a través del Visual Basic 6, con el fin de que el resultado no perdiera el hecho de ser amigable y de buena apariencia para el usuario, pero que a su vez solo dependiera de un solo programa toda la aplicación, para evitar futuras complicaciones por software.

Es así como en un control del Visual llamado **PictureBox**<sup>16</sup> se puede introducir la imagen de las brújulas, establecer un entorno circular y se hace girar unas líneas, que indican los grados en los que se ubica el rotor. Para ello se deben usar cálculos geométricos para determinar la posición exacta en las que las líneas deben tener sus extremos. Los valores de azimut y elevación obtenidos del rotor se deben relacionar mediante estos cálculos con las posiciones finales de las líneas, para que la representación gráfica indique fielmente los valores del rotor.

---

<sup>16</sup> Es un control del Visual Basic 6, que permite presentar todo tipo de ficheros gráficos (.BMP, WMF, .ICO, .CUR) y puede ser destino de los datos de un enlace DDE.

#### 4. INTERFAZ GS-232

Una opción que se puede tomar para controlar de forma automática desde un PC al rotor YAESU G-5500, es con el dispositivo denominado GS-232. Cabe anotar que este dispositivo no ha sido manejado en los trabajos de grado anteriores, debido a que el módulo no se encontraba en la Universidad cuando éstas se realizaron. Es importante también mencionar que el GS-232 cuenta con sus propios comandos de control, y está programado para realizar ciertas funciones predeterminadas, es decir que para desarrollar el software que permita controlar este dispositivo de forma automática desde el PC, se debe limitar al uso de dichos códigos y a las funciones que estos realizan, que se encuentran en el manual del GS-232 anexo en el CD del presente proyecto. Esta interfaz no presenta la flexibilidad, en cuanto a programación se refiere, que sí otorga la ICU-G5500, debido a que sus funciones ya están diseñadas y no se pueden modificar.

Ya que este es un producto de la misma empresa que fabrica los rotores G-5500, es decir YAESU, está especialmente diseñado para servir de interfaz digital, proporcionando el control desde el puerto serie de cualquier PC a la mayoría de los rotores para antenas marca YAESU. El GS-232, mostrado en la figura 14, contiene su propio microprocesador (PIC18C452) con un convertidor análogo digital de 10 bits, una EEPROM, y puede ser configurado para trabajar a velocidad de comunicación de datos en serie de 1200 a 9600 baudios.

**Figura 14. GS-232**



Este dispositivo posee un terminal macho RS-232 que se conecta con el PC mediante un cable DB-9 en el puerto COM RS-232 o serie de la computadora. También contiene un terminal eléctrico donde se le deben suministrar 12 VDC a 110 mA con los que trabaja. Igualmente posee un cable para la conexión con el controlador G-5500 consistente en un conector dual 5-pin y en su otra terminal un conector DIN<sup>17</sup>, la conexión y los cables pueden apreciarse en la figura 15.

El firmware del GS-232 soporta dos tipos de control, uno es directamente a través del teclado del PC y el otro es mediante los comandos dados desde programas escritos específicamente para este dispositivo.

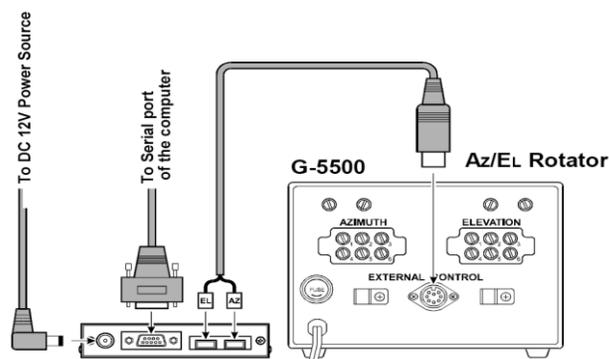
Ya que YAESU no proporciona junto con el dispositivo el software controlador, se debe considerar alguna de las dos opciones para manipularlo, pero como la idea es realizar un sistema autónomo como el de la ICU G5500, se considera entonces fundamental desarrollar

---

<sup>17</sup> Para mayor información referirse al manual del GS-232, anexo en el CD del presente proyecto

un software que permita mover el rotor YAESU G5500 con los comandos propios de la interfaz GS-232 y mezclando características propias del software controlador de la interfaz ICU-G5500, tales como el manejo de Visual Basic 6 para su desarrollo, el empleo del protocolo DDE para extraer datos de elevación y azimut del software predictivo denominado Winorbit, el uso del Wxtrack para determinar la posición a la que se debe poner la antena para la próxima pasada del satélite, entre muchas otras. Estas similitudes se dan porque se quiere que ambos programas sean lo más parecidos posibles en cuanto a funcionalidad y a características propias de la programación, con el fin de poder establecer una comparación en los resultados de algunas pruebas finales.

**Figura 15. Conexión del GS-232**



*YAESU GS-232B Computer control interface for antenna rotators*

#### 4.1 CALIBRACION

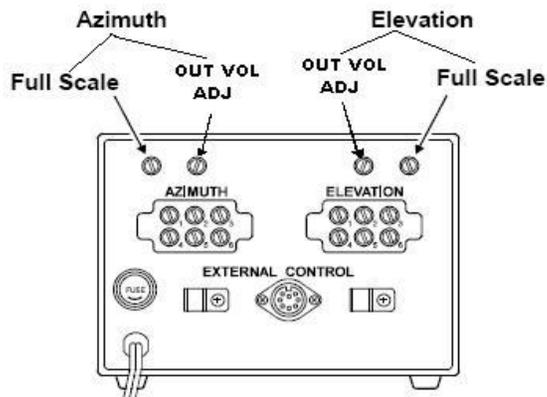
Este es el primer proceso que se debe llevar a cabo para obtener un óptimo funcionamiento del GS-232. El proceso de la calibración de este dispositivo es prácticamente manual, debido a que luego de la correcta conexión de los dispositivos, se debe establecer comunicación por medio del **Hyperterminal** de Windows y mover un potenciómetro denominado **out vol adj** que se encuentra en la parte posterior del controlador dual azimut-elevación G5500 como se muestra en la figura 16, también hay que observar que el dispositivo pueda llegar a su máximo rango de  $450^{\circ}$  para azimut y  $180^{\circ}$  para elevación, esto se realiza con el potenciómetro **full scale adj**.

Al mover el potenciómetro de **out vol adj**, se está variando el voltaje que entrega a la salida este módulo, ya sea para azimut o para elevación, y variando el potenciómetro **full scale**, se mueve la aguja para establecer cuál es el rango máximo que se quiere. Este último se debe dejar fijo en  $450^{\circ}$  para azimut y en  $180^{\circ}$  para elevación. Pero para ello se debe primero configurar el **Hyperterminal** de Windows para poder enviar comandos al GS-232 como se explicará a continuación.

En la **barra de tareas** de Windows se da clic en **Inicio**, luego en **todos los programas** se selecciona **accesorios**, allí se señala **comunicaciones** y clic en **Hyperterminal**. A continuación se despliega una ventana llamada **Descripción de la Conexión** en la cual le

damos un nombre a la conexión, que en este caso se denomina GS-232 y se identifica escogiendo un icono.

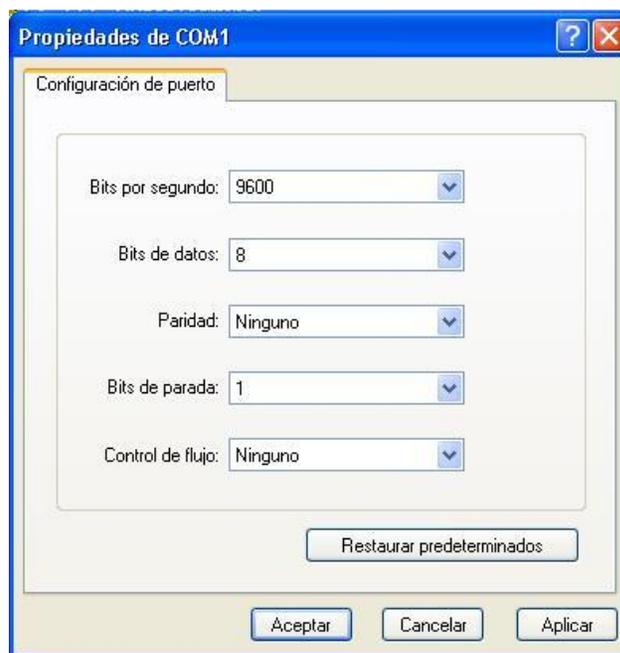
**Figura 16. Controlador Dual G5500**



*YAESU GS-232B Computer control interface for antenna rotators*

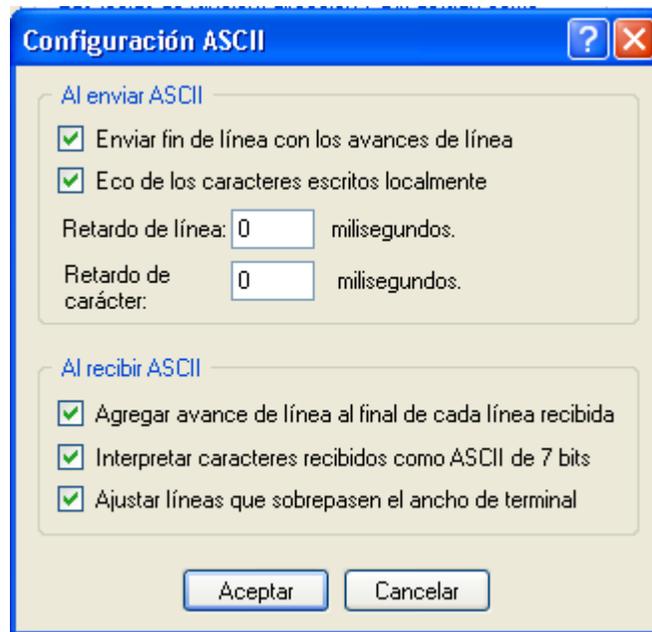
Al dar clic en **aceptar**, aparece la ventana **Conectar a**, en la cual se debe seleccionar en la casilla **conectar usando** el puerto serie que vamos a emplear para la comunicación con el GS-232, generalmente los PC vienen con un puerto serie llamado **COM1**, que preferiblemente debe usarse, si posee otro debe averiguar cual está libre para usarse. Posteriormente en la ventana de **Propiedades de COM1** se debe introducir la configuración que se ve en la figura 17.

**Figura 17. Propiedades de COM1**



Una vez configurado el puerto, aparece la interfaz de la conexión que se ha creado, luego se da clic en **archivo**, y en **propiedades**, para configurar la pantalla de la conexión. Luego en la pantalla de **Propiedades de GS-232** que aparece, se selecciona la pestaña que dice **Configuración** y se selecciona el botón llamado **Configuración ASCII**. Por último se activan todas las casillas presentes en la ventana **Configuración ASCII** como se muestra la figura 18 y se le da **aceptar**, esto se hace con el fin de visualizar lo que se escribe y se recibe además de que interprete el paso de líneas. De esta forma ya queda listo el **Hyperterminal** para la conexión con el GS-232.

**Figura 18. Configuración ASCII**



Una vez estén encendidos los dispositivos y activo el **Hyperterminal**, se debe visualizar en el diálogo éste un mensaje que dice **Connect OK**, el cual indica que ya se puede introducir códigos del GS-232, debido a que ya se presenta un flujo de información entre el PC y dicho dispositivo.

Para poder realizar la calibración del control G5500 con la interfaz GS-232, este último dispositivo posee un comando que permite recibir constantemente los datos tanto de azimut como de elevación, simultáneamente, en los que se encuentra el rotor. Presionando **F2** (F, 2) seguido por la tecla **ENTER**, el computador muestra el mensaje **AZ=aaa EL=eee**, donde **eee** son tres dígitos numéricos que indican la posición de la elevación en grados, y **aaa** son los tres dígitos numéricos que representan los grados de la posición del rotor en azimut.

Cabe anotar que el GS-232 también posee un código similar al anterior, con la diferencia que este solo sirve para visualizar el valor en grados de la posición del rotor en azimut.

Introduciendo desde el teclado la letra **F**, obteniendo en pantalla el mensaje **AZ=aaa**, pero para mayor eficiencia se puede emplear el primer código que es más general y contiene ambos datos. La diferencia entre estos dos códigos, que son similares se presenta debido a que los procesos de azimut se pueden dar de forma independiente, en cambio todo lo relacionado con la elevación, no puede estar solo y tiene una dependencia del azimut, por ello todos los códigos que emplean la elevación utilizan también el azimut.

Una vez introducidos los caracteres **F2** (F, 2) en el **Hyperterminal**, el proceso de visualización de los valores de azimut y elevación actuales del rotor se hace de forma constante y automática, lo cual permite mover el potenciómetro de **out vol adj** e ir modificando los valores de los voltajes de salida, hasta que el valor visualizado coincida con los valores mostrados en el control G-5500. Por último para comprobar que la calibración ha sido correcta, se debe poner el rotor en algún ángulo crítico para el proyecto, cuya posición sea conocida e indicada en el rotor, tal como  $360^0$ , y pedir de nuevo los valores actuales a la interfaz. Este mismo proceso de comprobación se realiza para ángulos indicados de la elevación, como  $180^0$ . Si los datos no coinciden se continuará variando el potenciómetro hasta que estos valores críticos sean exactamente los mismos. Para detener el proceso de visualización, que se inició para calibrar cuando se introdujo el comando **F2**, se presiona la tecla **ENTER**, que es la forma de salir de la rutina de calibración del GS-232.

Como ya se mencionó anteriormente se tomó la opción de crear un software controlador de la interfaz GS-232, que fuera capaz de enviar los comandos, de recibir las respuestas del dispositivo y visualizarlas, todo a través del puerto serie, para que de esta forma el rotor se pueda mover de una manera autónoma, sin la necesidad de que una persona esté allí al frente de la pantalla introduciendo los comandos directamente desde el teclado. Como se desea obtener parámetros uniformes de comparación entre ambas interfaces empleadas en el presente proyecto, el software se desarrolla en Visual Basic 6, para otorgar las mismas propiedades que contiene la interfaz de usuario de la ICU G5500 y porque también este programa permite manejar el puerto serie.

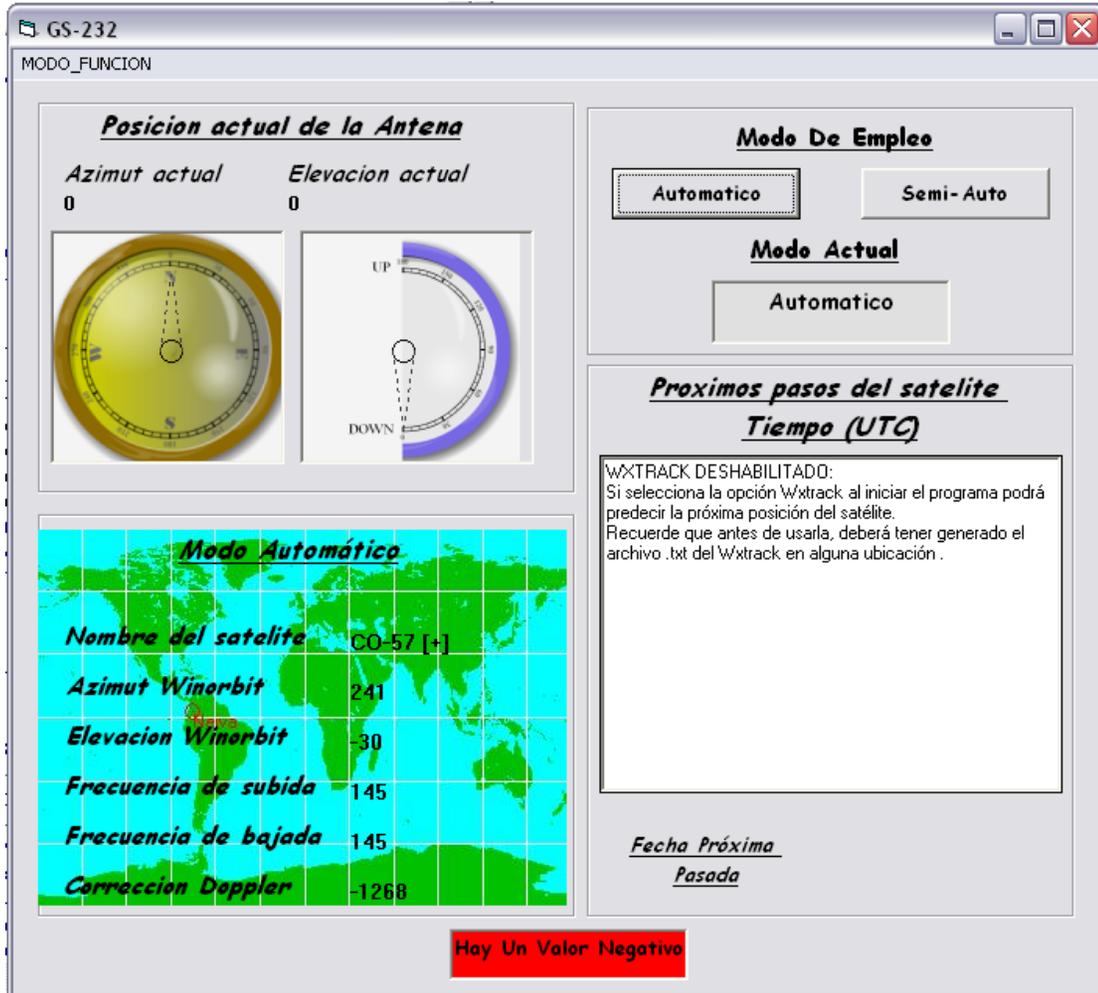
#### **4.2 SOFTWARE CONTROLADOR DE INTERFAZ GS-232**

El software que se desarrolló tiene dos formas de empleo, la primera corresponde al tipo automático, en la cual todo funciona de forma autónoma, es decir que apenas se activa dicho programa con sus respectivos requerimientos, este ubica al rotor en la posición deseada y continúa el seguimiento del satélite sin necesitar a una persona que esté ubicándolo, sino que emplea al Winorbit para obtener los datos de azimut y elevación, empleados en el posicionamiento del rotor. La segunda forma corresponde al tipo semi-automático, en la cual a pesar que la ubicación del rotor se hace por medio de la interfaz GS-232, necesita una persona que esté introduciendo los valores, y se le llamo semi-automático porque la persona solo digita el valor en grados de donde quiere posicionar al rotor y el software lo ubica sin importar en qué posición se encuentre.

Este software permite abrir un programa de predicción, como el Winorbit, y a su vez da la opción de predecir los pasos futuros que realiza el satélite por la estación terrena con un archivo que genera el Wxtrack, que se ha explicado ampliamente ...en la sección

3.2.1....También posee la capacidad de visualizar los valores actuales de la posición del rotor de forma dinámica y real, a través del empleo de una brújulas como lo muestra la figura 19, el cual es la interfaz de usuario del software controlador GS-232.

**Figura 19. Interfaz de usuario GS-232**



Como el Visual Basic 6 se programa por eventos, es complicado describir el programa de forma secuencial, así que la forma más indicada para explicar lo que se realizó es a través de los eventos.

**4.2.1 Evento form load.** En el evento **form load** se realizan tres cosas principalmente, que se mencionarán a continuación:

- **Se inicializan variables.** En esta parte del programa se establece el valor inicial de algunas variables o registros que se emplean y que posteriormente pueden cambiar de valor tal como el registro de salida del puerto serie, y también de algunas que permanecen

con el valor inicial sin cambiar nunca, como lo son los valores que equivalen al **ENTER** o a un **espacio en ASCII**. De igual forma se inicializa el **Timer** y se introduce el valor del intervalo de tiempo en el que se desea que se ejecute el evento **Timer**, esto se hace inicializando el registro **Timer.interval** con un valor numérico que estará en unidades de milisegundos (ms).

Cabe anotar que una particularidad que tiene el GS-232 son sus comandos, ya que estos tienen un formato especial, los cuales deben ser enviados correctamente para que funcionen. Por lo tanto para asegurar esto, se establecen algunas variables que guardan partes de los comandos que se emplean constantemente, algunos de ellos son las letras **W** y **S**, el valor numérico **0** y el comando **C2**, que deben ser introducidos como un **string**, también se encuentran algunos caracteres especiales que no están representados en el teclado, por lo cual debe introducirse su valor hexadecimal y luego pasarlo a un **string**, algunos ejemplos son la tecla **ENTER** que se compone de los valores hexadecimales **0D** y **0A** ( retorno de carro y salto de línea respectivamente) y el **espacio** cuyo valor en el sistema hexadecimal es **20**. Más adelante se explicará para que se emplean estos valores, y en el manual del GS-232 anexado en el CD del presente proyecto está la lista de los comandos.

- **Se establece el intercambio de datos.** Como el intercambio dinámico de datos (DDE) ya se había explicado en los trabajos de grado anteriores, solo se va a mencionar una pequeña variación realizada exclusivamente para este programa debido a un inconveniente que se presentó.

En el software de control de la Interfaz ICU-5500 se establece el **LinkMode=2**, esto quiere decir que los datos se traspasarán cuando lo pida la aplicación cliente o destino que en este caso es el software de control de la interfaz, mediante la orden **LinkRequest**. Esta orden se ejecutaba constantemente y con cierta periodicidad. Al ejecutar de la misma forma el intercambio de datos, pero esta vez con la interfaz GS-232, se presentaba una inestabilidad en el rotor al enviar varias veces el mismo dato sin necesidad, debido a que los valores de azimut y/o elevación no habían cambiado, pero si se aumentaba el intervalo de tiempo en el cual se aplica el **LinkRequest** los datos podían cambiar y el programa no los tendría en cuenta en tiempo real. Para ello entonces se decide establecer el **LinkMode=3**, o sea que para este caso cuando existe un cambio en los datos de la aplicación origen, ésta notifica a la aplicación destino que el dato ha cambiado, generando el evento **LinkNotify**, pero el dato nuevo aún no se envía. En el procedimiento de dicho evento se debe escribir la orden **LinkRequest**, para que permita el paso de los datos nuevos. De esta forma se asegura que el dato se reciba solamente cuando cambie.

- **Mensajes y WxTrack.** Esta parte realiza el mismo proceso que el explicado ...en la sección 3.2.1..., en la cual se envían dos mensajes al usuario, el primero pide abrir un software predictivo y el segundo da la opción de usar el Wxtrack para la predicción del próximo paso del satélite por el sitio donde se encuentra la estación terrena.

**4.2.2 Evento Timer.** Este evento se genera cada vez que se cumpla el intervalo de tiempo que se ha establecido para la propiedad **Timer.interval** en el evento **form load**, que para el caso del proyecto se tomó de 100 milisegundos (ms).

Este evento se emplea fundamentalmente para el funcionamiento de las brújulas, que son las que permiten visualizar la posición tanto en azimut como en elevación actuales del rotor. La única parte diferente a la explicada ...en la sección 3.2.3... es la obtención de los valores actuales de azimut y elevación del rotor que se realiza de la siguiente forma: el puerto serie se habilita desde la carga del **form**, en el evento **form load**, colocando en **enabled** el registro **PortOpen** del control **MSComm**. Este último es el control propio del visual para la comunicación serie. Luego se envía al registro de salida (**Output**) el comando del GS-232 que devuelve la posición actual del rotor, este comando es **C2** y como todo comando de esta interfaz debe ir acompañada del **ENTER**, entonces se envía en un solo **string** de datos en conjunto con el comando. El comando **C2** devuelve el azimut y la elevación de la siguiente forma **AZ=aaa EL=eee** mencionada ...en la sección 4.1..., que es el mismo formato entregado por el código **F2**, con la diferencia que este último hace que la Interfaz se quede ocupada devolviendo constantemente el valor actual de la posición del rotor, hasta que se desee, en cambio con **C2** la interfaz devuelve esta información una sola vez.

La interfaz responde de inmediato a este comando y se graba en una variable lo que contiene el registro de entrada del puerto serie (**Input**). Una vez se tiene la información almacenada, se divide la cadena y se separan los datos que interesan, por medio de la instrucción **Mid** del Visual Basic 6.

**4.2.3 Evento linknotify.** Este evento se genera cuando uno de los dos valores o ambos (azimut y/o elevación) de la posición del satélite cambia en el Winorbit. Debido a que los datos se transfieren por separado, es decir se visualizan en controles diferentes, por lo cual existen dos eventos **LinkNotify**, uno para el azimut y otro para elevación. Estos dos eventos realizan básicamente lo mismo, primero envían el **LinkRequest**, que es la orden necesaria para traer los datos desde el Winorbit al programa del PC, toman la información, la almacenan en registros y se procesa para saber si los valores son positivos. Posteriormente se comprueba si el valor grabado es menor de 10 o está entre 10 y 99 o es superior o igual a 100, esto con el objetivo de enviar la información completa que requiere la interfaz para trabajar, pues todos los valores que se envían a la interfaz tienen que poseer tres dígitos, ni uno más ni uno menos, para lo cual se hace dicha diferenciación, pues si es menor de diez se le agrega a la información obtenida del Winorbit dos ceros antes del dato, si está entre 10 y 99 se le agrega un cero, y para 100 y mayores, no se altera. Este proceso se hace tanto para el modo semiautomático, como para los **LinkNotify** que corresponden al modo automático, y también en los valores adquiridos del WxTrack.

Luego de haber realizado el debido proceso para transformar los valores del Winorbit en información que pueda entender la interfaz, se procede a escribir el registro de salida (**Output**) del puerto serie. Este debe contener un comando de la interfaz GS-232, que para este caso corresponde al siguiente comando: **Waaa eee**, que como se sabe **aaa** corresponde

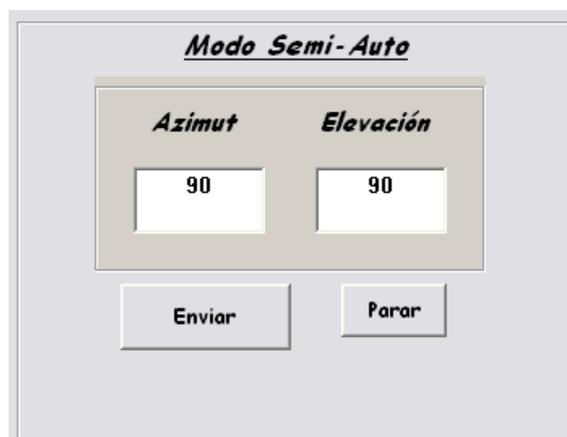
al valor de azimut en el que se desea poner el rotor y **eee** es el ángulo de elevación en grados en el que se debe establecer el rotor. Este comando debe ir acompañado de la tecla **ENTER** (cuya equivalente en el sistema hexadecimal es **0D** y **0A**). Cuando la elevación es negativa, ambos eventos **LinkNotify** esperan que dicho dato negativo comience a disminuir en su valor absoluto, para que la información de la posición de la próxima pasada del satélite, otorgada por el WxTrack si se ha activado, sea enviada a la interfaz y ejecutada en el rotor.

**4.2.4 Modo semiautomático.** Este modo se puede activar desde un botón el cual establece el **LinkMode=2** del intercambio dinámico de datos, lo cual quiere decir que los datos del Winorbit solo se transferirán si el programa desarrollado del PC lo solicita a través de la instrucción **LinkRequest**. Pero como para este caso no se desea trabajar con los valores del software predictivo, sino con los introducidos por teclado en las casillas correspondientes, entonces no se pone dicha instrucción.

Para que este modo funcione y los datos se envíen una sola vez, se activa un botón o **CommandButton**, que al recibir un clic tomará los datos introducidos por el usuario, en las cajas de texto destinadas para azimut y elevación, procesará estos valores para verificar que sean positivos, que sean valores aceptados por el rotor y la interfaz y para convertirlos en números de tres cifras; por último envía los datos al registro de salida del puerto serie (**OutPut**) mediante el comando **Waaa eee** de la interfaz GS-232. Si el valor es negativo envía un mensaje y no ejecuta nada.

Para el modo semiautomático, aparece una nueva ventana en la interfaz de usuario como la que se muestra en la figura 20, reemplazando o sobreponiéndose a la que corresponde al modo automático.

**Figura 20. Modo semiautomático**



<i>Modo Semi-Auto</i>	
<i>Azimut</i>	<i>Elevación</i>
90	90
Enviar	Parar

## 5. PRUEBAS REALIZADAS

En esta sección, se mostrarán los resultados obtenidos de pruebas realizadas tanto al sistema ICU G5500 como al GS232. Dichas pruebas se realizaron por medio de un círculo dibujado de 98 cm. de diámetro, el cual fue realizado en Autocad<sup>18</sup> y que contiene representados los grados de 1 a 360 en variaciones exactas de 1 grado. Las mediciones que se realizaron para los dos sistemas, se hicieron de forma similar. Para el azimut, se ubicó el rotor Yaesu G5500 en el centro del círculo ubicado sobre el suelo, al cual se le adecuó un **Boom** del tamaño del radio con una plomada en el extremo, que indicaba en el perímetro de la circunferencia el valor del grado en el que se encontraba el rotor. En el caso de la elevación, se utilizó la mitad del círculo mencionado ubicado sobre la pared, en esta ocasión en el extremo del **Boom** se le adecuó una extensión que estuviera de forma vertical formando un ángulo de 90 grados con este y un apuntador al extremo que señalara la ubicación del rotor sobre el papel de la circunferencia. Una vez instalado en el rotor apropiadamente el **Boom**, para tomar las medidas tanto de azimut como elevación, se ingresaron valores con variaciones de 5 grados utilizando los modos semiautomáticos realizados para ambas interfaces. Ingresados estos valores, el rotor se ubicaba en la posición seleccionada de la cual se toma el valor ingresado, el valor registrado en la circunferencia y el dato que entrega la interfaz que en el caso de la GS-232 es el mismo que se ingresa y en el caso de la ICU G5500 es el dato numérico obtenido después de transformar del valor entregado por el conversor análogo digital a grados.

El GS-232 entrega este valor debido a que no tiene en cuenta las cifras decimales y solo muestra el valor entero de la medida tomada. Se asume que el microcontrolador que contiene está programado de forma tal que solo permite visualizar un valor entero para determinado nivel de voltaje. En cambio la ICU G5500 está dispuesta de tal forma que permite visualizar el valor con decimales de la posición en que debería estar el rotor, ya que la posición real es la que se tomó de las mediciones del círculo y varían muy poco a las mostradas en la interfaz de usuario de la ICU G5500.

Ambos sistemas presentan una limitación en cero grados tanto en azimut como en elevación, ya que ninguno de los dos llega a dicho valor exactamente. En las pruebas realizadas se observó que el GS-232 llega hasta 1 grado como valor mínimo, y en el ICU G5500 también se obtiene el mismo valor ya que en cero grados el programa del microcontrolador no funcionaba correctamente, y la solución tomada fue asignarle a esta posición el mismo valor que el grado siguiente.

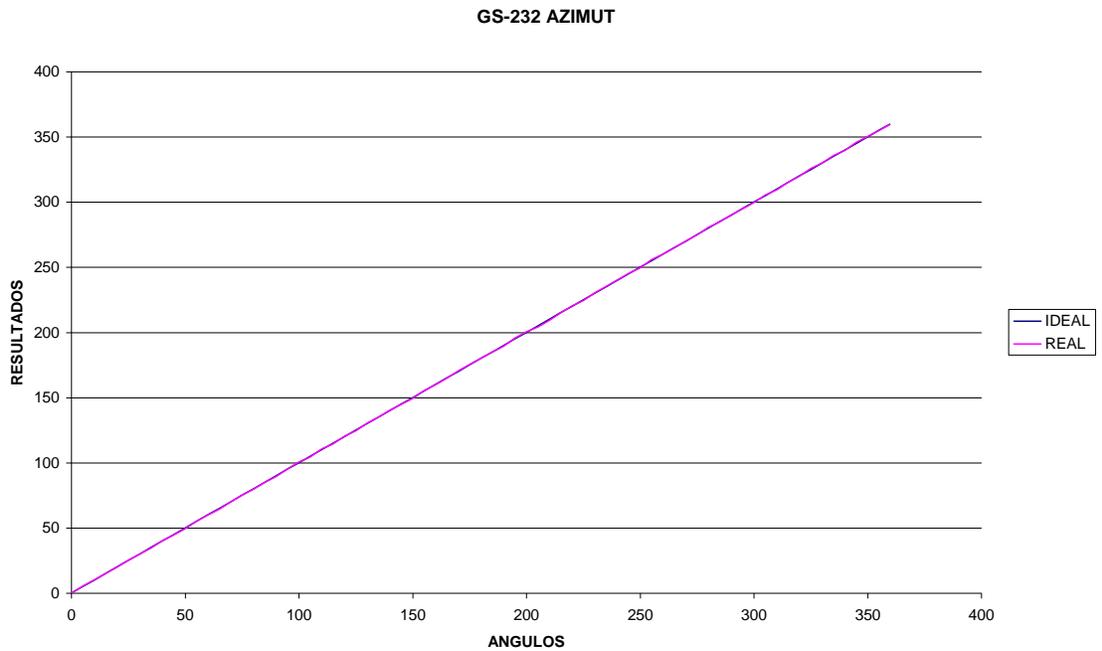
Por último se debe mencionar que para el GS-232 la tolerancia entre el valor real y el valor deseado que se obtuvo en ambas mediciones, tanto para azimut como para elevación fue de + o - 1 grado, en cambio la ICU G5500 presentaba este mismo parámetro solo en azimut, ya que en elevación presentaba una tolerancia de + o - 0.6 grados, siendo estos valores los mismos que los escogidos en el programa desarrollado, que fueron los mínimos en que se estableció el límite de funcionamiento esta interfaz.

---

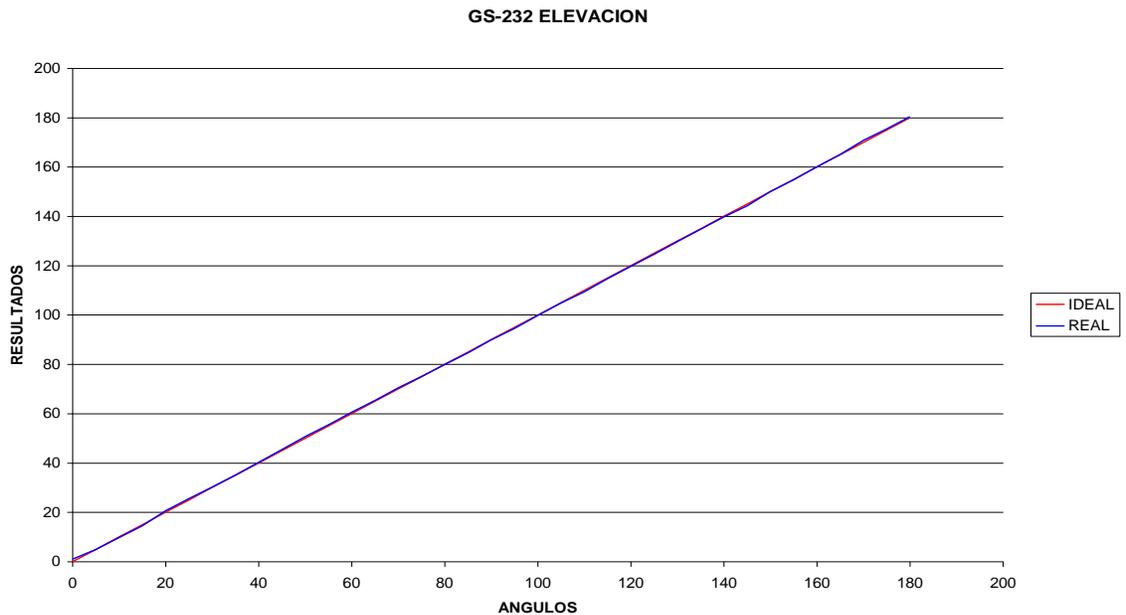
<sup>18</sup> Software de diseño grafico

A continuación se muestran las gráficas de las mediciones obtenidas de las pruebas realizadas, para ambas interfaces. Para la interfaz GS-232 solo se graficó el valor ideal en el que el rotor debería estar ubicado y el valor real o el tomado de la medición en la circunferencia. La gráfica de azimut se puede apreciar en la figura 21, y la de elevación en la figura 22.

**Figura 21. GS-232 Azimut**

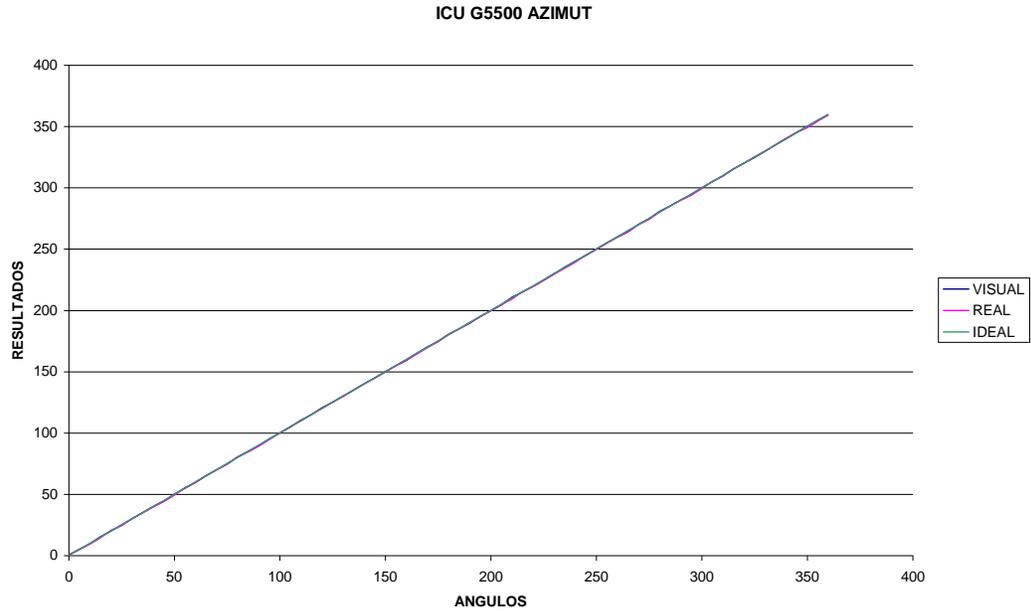


**Figura 22. GS-232 Elevación**

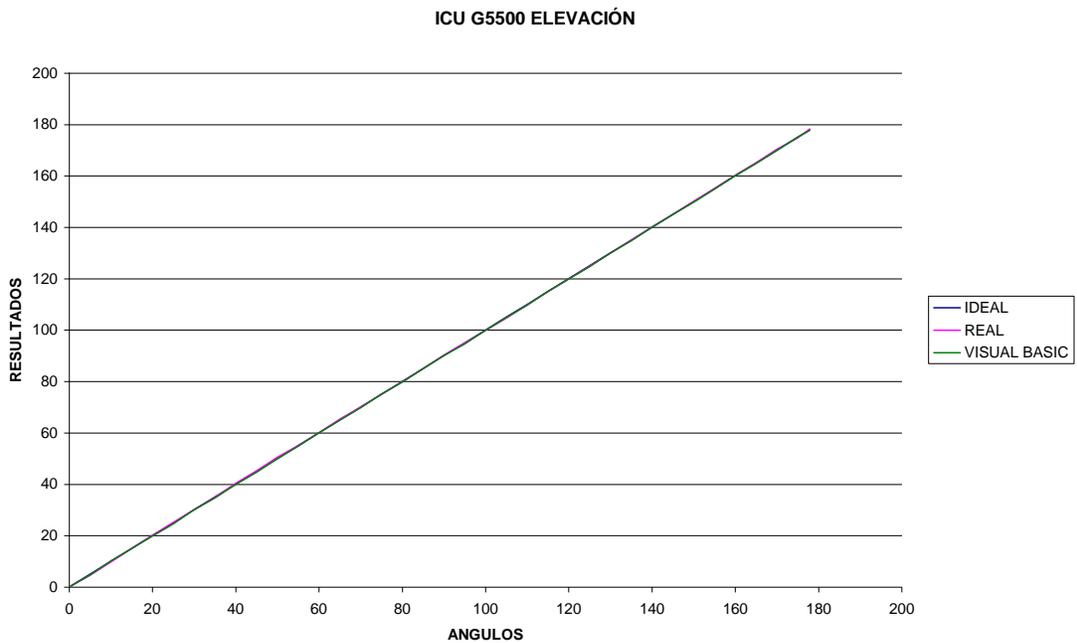


Para el ICU G5500 se realizaron las mismas gráficas que en las empleadas para el GS-232 y además la correspondiente a los valores que entrega la interfaz de usuario de dicho sistema. La gráfica de azimut se puede apreciar en la figura 23, y la de elevación en la figura 24.

**Figura 23. ICU G5500 Azimut**



**Figura 24. ICU G5500 Elevación**



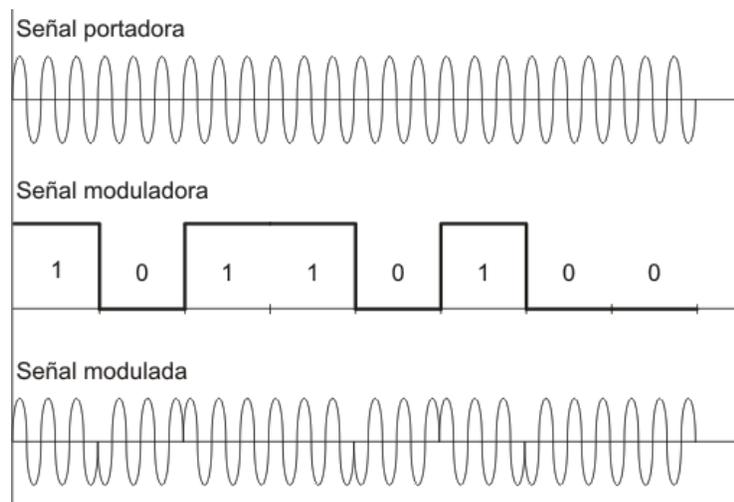
Como se observa en las figuras anteriores, las diferencia entre los valores graficados son muy pequeñas, de máximo 1 grado, por lo cual las líneas de los gráficas se aprecian casi encima una de la otra. Para observar los cuadros de los valores obtenidos en las mediciones, se pueden encontrar en el ANEXO A.

## 6. VERIFICACION DE MÉTODOS DE DETECCIÓN DE BEACON

Para comprender mejor lo que se va a analizar en esta sección, hay que saber que es una señal **Beacon**.

Pues bien, una señal **Beacon** es un tono en la banda de frecuencias audible codificada en Morse o **CW** mediante modulación **PSK**, que se muestra en la figura 25. Es una señal radio que se transmite de forma continua o periódica con una cantidad de información limitada, (por ejemplo su localización e identificación), en una frecuencia de radio establecida o fija, su propósito es doble, uno es el de enviar información vital propia del satélite (telemetría), y la otra para ubicar al satélite (determinar el azimut y la elevación) en el cielo, debido a que con ella se detecta la presencia del satélite.

**Figura 25. Modulación PSK**



<http://www.textoscientificos.com/imagenes/redes/modulacion-PSK.gif>

Se estudiaron dos posibilidades para detectar la señal **Beacon** de un satélite radioaficionado de órbita baja.

La primera posibilidad consistía en el uso de un dispositivo digital o análogo denominado **satfinder** o **satellite finder** (encontrador de satélites), que es un dispositivo de medición usado para determinar la dirección óptima en la cual se debe apuntar la antena para recibir lo más fuerte posible la señal del satélite que está en órbita.

Se descartó esta opción porque no se encontró en el mercado un dispositivo con las características deseadas para este proyecto, por ejemplo que sea capaz de detectar señales en las frecuencias de transmisión de los pico-satélites de baja potencia como es el caso del Cubesat, que tiene un rango asignado de 435 a 440 Mhz de enlace descendente, y estos valores no está comprendido dentro del rango que maneja el **satfinder** que es de 0.95 a 2,3 Ghz, debido a que estos dispositivos están especialmente diseñados para fines comerciales.

La segunda opción que se considera es adquirir la señal **Beacon** directamente a través de los radios con los que cuenta este proyecto, que son el **Kenwood TM-D700** y el **Yaesu FT-7800**<sup>19</sup>, y a partir de ellos escuchar la señal del satélite.

Este método es sencillo, eficiente y cabe mencionar que solo basta tener dos datos actualizados, que son fundamentales del satélite al que se le va a detectar la señal **Beacon** para desarrollar este proceso, que son la frecuencia del enlace descendente y el corrimiento en **doppler**.

Para las pruebas se debe contar con una lista o base de datos de los satélites actualizadas, es decir, una lista en donde figuren los satélites que cumplan ciertos requisitos como: esté en funcionamiento, que tengan señal **Beacon** y esté en el rango de potencia que acepte la estación terrena y que la frecuencia del enlace de bajada de la misma se encuentre dentro del rango de frecuencia que acepta la antena empleada en la estación. Estos datos se pueden obtener de la página oficial de la **AMSAT**<sup>20</sup>, la cual está en constante actualización.

El corrimiento en **doppler** se obtiene de los softwares predictivos, pero estos también tienen una base de datos que debe estar en constante actualización.

Para la prueba que fue realizada, en cuanto a la detección de **Beacon** se refiere, se empleó el radio **Kenwood TM-D700**, la base de datos y la corrección automática del corrimiento en **doppler**, que ofrece el programa denominado **PROET**<sup>21</sup> que permite controlar el radio de forma automática; al igual se utilizó todo el montaje realizado para el control del rotor, realizando las pruebas tanto con el GS-232 como con la ICU G5500 para el posicionamiento de la antena, y sus respectivos software controladores.

Dentro de los satélites encontrados hábiles y aptos para la prueba, se escogió el **LO-19** o **LUSAT-OSCAR 19**, desarrollado por **AMSAT** Argentina, que describe una órbita LEO es un micro satélite cuya **Beacon** está en modulación **CW** y envía 8 canales de telemetría y un canal de estado en la frecuencia de 437.1250 MHz, y en total entrega 12 palabras por minuto. Para esta prueba, los resultados fueron muy satisfactorios, ya que se siguió la trayectoria del satélite y en todo su recorrido se escuchó su señal **Beacon**, además que se puso a prueba lo diseñado en el software de control, como por ejemplo la predicción de la próxima pasada del satélite por la estación terrena que la otorga el Wxtrack.

Las pruebas se realizaron el 4 de agosto con la interfaz ICU G5500 y el 5 de agosto con la interfaz GS-232. Se obtuvieron muy buenos resultados, ya que con cada una de las

---

<sup>19</sup> LOZANO CLAROS, Diego y VELASQUEZ CHALA, Oscar Fernando. Implementación y puesta en funcionamiento del sistema de comunicaciones de la estación terrena cubesat-usco. Proyecto cubesat-ud-usco telemetría y telemetría. 2008. p. 35-38. Trabajo de grado (Ingeniero Electrónico). Universidad Surcolombiana. Facultad de Ingeniería.

<sup>20</sup> <http://www.amsat.org/amsat-new/satellites/status.php>

<sup>21</sup> LOZANO CLAROS, Diego y VELASQUEZ CHALA, Oscar Fernando. Implementación y puesta en funcionamiento del sistema de comunicaciones de la estación terrena cubesat-usco. Proyecto cubesat-ud-usco telemetría y telemetría. 2008. P. 53-65. Trabajo de grado (Ingeniero Electrónico). Universidad Surcolombiana. Facultad de Ingeniería.

interfaces se le realizó el seguimiento dos veces, con el fin de probar la sección del programa que corresponde a la próxima pasada dada por el Wxtrack. Se obtuvo un seguimiento de entre 14 y 15 minutos en cada uno de los casos, y se observó que el método empleado para acercar al rotor a la posición en donde se producirá la salida de la próxima pasada del satélite funciona muy bien, con eficiencia y efectividad.

Los seguimientos efectuados el 4 de agosto fueron desde las 4:11:58 p.m. hasta las 4:26:57 p.m., para la primera pasada cuya azimut de salida fue  $158^\circ$  y la elevación de  $2^\circ$ , y el segundo paso del satélite por la estación se produjo desde 5:53:13 p.m. hasta las 6:07:01 p.m. cuya azimut de salida fue  $214^\circ$  y la elevación de  $1^\circ$ . La frecuencia de la señal de bajada inicial fue de 437.135 MHz debido al ajuste del corrimiento en **Doppler** para la primera pasada y de 437.130 para la segunda.

De igual manera, los seguimientos efectuados el 5 de agosto fueron desde las 3:41:41 p.m. hasta las 3:55:21 p.m., para la primera pasada cuya azimut de salida fue  $139^\circ$  y la elevación de  $2^\circ$ , y el segundo paso del satélite por la estación se produjo desde 5:20:48 p.m. hasta las 5:34:33 p.m. cuya azimut de salida fue  $196^\circ$  y la elevación de  $3^\circ$ . En ambos casos la frecuencia de la señal de bajada inicial fue de 437.130 MHz debido al ajuste del corrimiento en **Doppler**.

## 7. CONCLUSIONES

Mediante el empleo de la misma plataforma de programación sobre la que se desarrolló el software de control para la interfaz ICU G5500, se logró realizar el programa controlador propio de la interfaz GS-232, manejando las mismas características en la interfaz de usuario y las mismas propiedades de funcionamiento de la ICU G5500, obteniendo buenos resultados en funcionamiento y con un margen tolerable de error en las mediciones desarrolladas, pero con la limitación de que esta interfaz ya contiene sus funciones establecidas y solo se puede acceder a ellas mediante comandos, a los cuales se restringe el desarrollo del software controlador. Además su costo en el mercado es de \$ 806.67 dólares, que comparado con la interfaz ICUG5500 que no supera los \$ 600.000 pesos colombianos, se puede considerar demasiado caro.

En el programa de la interfaz ICU G5500, se solucionó el problema que presentaba con las cifras significativas el programa del microcontrolador desarrollado en los anteriores trabajos de grado y que generaba diferencias de más de 20 grados tanto en azimut como elevación, al utilizar el programa del PC desarrollado en visual Basic para realizar todas las operaciones que manejaran cifras significativas y así no tener problemas con la pérdidas de éstas. También se desarrolló una verificación del estado de conexión del rotor en tiempo real tanto en el microcontrolador como en el PC mejorando el sistema de calibración de esta, evitando el inconveniente de que se obtuvieran valores erróneos en los datos que se obtienen de la calibración.

Ambos sistemas (interfaz ICU G5500 y GS-232) presentan una limitación en cero grados tanto en azimut como en elevación, ya que ninguno de los dos llega a dicho valor exactamente. Se observó que el GS-232 llega hasta 1 grado como valor mínimo, y el ICU G5500 también obtiene el mismo valor ya que en cero grados el programa del microcontrolador no funciona correctamente, y la solución tomada fue asignarle a esta posición el valor del grado siguiente (1 grado), por esto se concluye que esta limitación es propia del rotor y su control dual G5500, ya que también se presenta en la interfaz GS 232, la cual está diseñada para manejar los rotores de la marca YAESU.

Como el programa Winorbit no proporciona las futuras pasadas del satélite ya que predice solo en tiempo real, el rotor espera la salida del satélite sobre el horizonte en una posición errónea de azimut y elevación, esto lleva a una pérdida de tiempo en el seguimiento ya que una vez el satélite es visible el rotor pasa a reubicarse en los valores correctos dados por el Winorbit. Éste inconveniente se solucionó utilizando los datos de azimut y elevación que proporciona el programa Wxtrack sobre las próximas pasadas del satélite, para ubicar el rotor en dichos valores antes de que el satélite salga en el horizonte y así evitar la pérdida de tiempo.

Se integró todo el software empleado para el control del rotor bajo una sola plataforma de programación que es Visual Basic 6, que admite el traspaso de información entre aplicaciones de forma sencilla mediante protocolo DDE, permitiendo también acceder

directamente desde el software desarrollado a otras aplicaciones como el Winorbit y el WxTrack.

Se demostró, mediante pruebas con satélites, que la confiabilidad de los programas predictivos puede mejorarse mucho si se les actualiza constantemente la base de datos propia de cada uno, ya que los elementos keplerianos que estos contienen pueden variar con el paso del tiempo y de esta forma deteriorar los resultados de los cálculos de posición y trayectoria de los satélites. Se puede considerar que la opción de emplear como software predictivo al Winorbit puede extenderse en el tiempo mientras se logra desarrollar e implementar uno propio para el macro-proyecto Cubesat Usco-UD.

Anteriormente cuando el microcontrolador de la interfaz ICU G5500 realizaba una petición de datos al programa del PC y éste no se encontraba disponible (el programa se había cerrado), en algunas ocasiones se quedaba en un bucle infinito, éste inconveniente se solucionó colocando una rutina que detecta la conexión del PC antes de cada acceso al bus USB por parte del microcontrolador, eliminando la posibilidad de que éste entre al bucle infinito. Por otra parte en el programa del PC se agregó el modo semiautomático muy útil para la calibración del sistema y se utilizó el Visual Basic para reemplazar el sistema de brújulas creado anteriormente en FLASH y que se utilizaba para visualizar la posición actual del rotor, evitando el uso de programas diferentes al Visual Basic, unificando el programa desarrollado en un solo lenguaje de programación.

A través de pruebas realizadas el 4 y 5 de Agosto con satélites de órbita baja, se comprobó que los programas desarrollados en Visual Basic para el PC para ambas interfaces (ICU G5500 y GS-232) funcionan muy bien, ya que se consiguió el seguimiento completo a los satélites desde que aparecían en el horizonte hasta cuando salían de él. En dichas pruebas se pudo escuchar la señal Beacon y probar el sistema empleado para la predicción del próximo paso del satélite por medio del Wxtrack, al igual que se verificó la perfecta integración del software de control con el de comunicaciones. Las señales Beacon se escucharon entre 14 y 15 minutos cada una, tiempo que demoraba pasando el satélite por la estación terrena de la USCO. También se comprobó la precisión de la posición del rotor con ambas interfaces, a través de pruebas en las que se tomaba como referencia un transportador dibujado por computadora, el cual indicaba en que grados estaba ubicado el rotor y se comparaba con la posición deseada que era introducida por el usuario, obteniendo un error máximo de  $+ o - 1$  grados en ambos casos para azimut y elevación en el GS-232, en cambio en la ICU G5500 se presentó un error máximo en la elevación de  $+ o - 0.6$  grados.

## 8. RECOMENDACIONES

- Así como se realizó una interfaz para comunicar el PC con el control G-5500, también es posible desarrollar en el macro proyecto un dispositivo que cumpla las funciones que en estos momentos son desarrolladas por el control dual azimut –elevación G-5500, y de esta forma obtener ciertas ventajas como la solución del inconveniente que presentaba el rotor en cero grados mencionado en el presente trabajo de grado, la posibilidad de incrementar funciones necesarias exclusivamente para el macroproyecto, suavizar los movimientos del rotor y mejorar las aplicaciones que posea el proyecto. Esto puede conducir a la creación e implementación no solo de la interfaz sino de un dispositivo de propósito específico para este macroproyecto que contenga tanto las funciones de la interfaz como las del control dual de azimut y elevación en uno solo.
- Se puede realizar un software predictivo, con base a las necesidades propias del proyecto, que permita varios protocolos de transferencia de información, que tenga facilidades de actualización, que se abra de forma automática en el PC del control de la estación terrena , entre otras funciones, esto con el fin de ir desarrollando un perfil propio a la estación terrena de la Usco, con todos los módulos desarrollados por la propia Universidad, ya que permite mayor versatilidad, variar la programación, es más económico y puede suplir necesidades futuras del proyecto. Una vez hecho esto se puede llegar a pensar en cambiar el software controlador para el PC, por otro que este desarrollado en un lenguaje de última generación como los son c++, c# y Visual Basic .Net, que permita mayores facilidades como la creación de formularios Web tan fácilmente como formularios de Windows, soporte mejorado orientado a objetos y la creación de servicios Web rápidamente entre otros.
- Se pretende que para un futuro el control de la estación terrena se pueda desarrollar por otros sistemas de programación, como la lógica difusa, o las redes neuronales, realizando pruebas pilotos de desarrollos en estas ramas para mirar eficiencia en su aplicación y posibles mejoras, con el fin de ir encaminándose a un seguimiento realimentado, y no tener que depender 100% del software predictivo.

## **BIBLIOGRAFÍA**

### **LIBROS**

ÁVILA, Miguel. Mecánica orbital, curso. Bogotá: Universidad Distrital “Francisco José de Caldas”, 2006. 18p.

AXELSON, Jan. USB complete. Lakeview Research, 1ª edición, 1999.

### **TESIS**

ANDRADE VIDARTE, Diego Mauricio y NUÑEZ ROMERO, Leonardo Ernesto. Diseño del sistema de control físico para el posicionamiento de la antena de la estación terrena del proyecto cubesat en la universidad Surcolombiana. 2007. 68 p. Trabajo de grado (Ingeniero Electrónico). Universidad Surcolombiana. Facultad de Ingeniería.

GARCIA LOPEZ, Juan Gilberto y MONTEALEGRE VANEGAS, John Wilder. Diseño del sistema de control del desplazamiento y software de monitoreo de la estación terrena de la ciudad de Neiva del macroproyecto cubesat. Neiva. 2007. 90 p. Trabajo de grado (Ingeniero Electrónico). Universidad Surcolombiana. Facultad de Ingeniería

LOZANO CLAROS, Diego y VELASQUEZ CHALA, Oscar Fernando. Implementación y puesta en funcionamiento del sistema de comunicaciones de la estación terrena cubesat-usco. Proyecto cubesat-ud-usco telemedicina y telemetría. 2008. 112 p. Trabajo de grado (Ingeniero Electrónico). Universidad Surcolombiana. Facultad de Ingeniería.

### **MANUALES**

Manual de instrucciones G-5500, vertex standard co ltd.

Manual de instrucciones Yaesu GS-232A

### **ENLACES**

<http://www.astromia.com>

<http://www.mecanique.co.uk/>

[http://picmania.garcia-cuervo.com/USB\\_0\\_Desencadenado.htm](http://picmania.garcia-cuervo.com/USB_0_Desencadenado.htm)

<http://www.todopic.com.ar/foros/index.php?topic=13406.20>

<http://www.astromia.com/glosario/mecanicaceleste.htm>

[http://ar.geocities.com/seti\\_argentina/mecacel/mecesa.html](http://ar.geocities.com/seti_argentina/mecacel/mecesa.html)

<http://www.xelrcs.org.mx/programas/diversos.html>

<http://celestrak.com/NORAD/elements/master.asp>

<http://www.amsat.org/amsat-new/satellites/status.php>

## ANEXO A. CUADROS DE LAS PRUEBAS

<b>PRUEBA ELEVACIÓN DE LA ICU-G5500</b>		
<b>Ángulos</b>	<b>Real</b>	<b>Visual Basic</b>
0	0	0
5	4,5	4,7
10	9,7	10,2
15	15	15
20	20	19,8
25	25,1	24,6
30	30	30
35	34,9	34,6
40	40,3	40
45	45,1	44,6
50	50,5	49,9
55	55,1	54,8
60	60	60
65	65,3	64,9
70	70	69,7
75	75	75,1
80	79,9	79,8
85	85	84,9
90	90,2	90,1
95	95	94,6
100	99,8	99,9
105	104,6	104,8
110	109,8	109,6
115	114,9	115
120	119,8	119,8
125	124,9	124,7
130	129,9	130
135	134,9	134,7
140	140,1	140
145	145	144,9
150	150,1	149,8
155	155,1	154,8
160	160,2	160
165	165,2	164,8
170	170,3	170

175	174,7	175
178	178,4	177,8

<b>PRUEBA AZIMUT DE LA ICU-G5500</b>		
<b>Ángulos</b>	<b>Real</b>	<b>Visual Basic</b>
0	0	0,43
5	5,1	4,8
10	9,3	9,6
15	14,3	15,3
20	20,5	20,1
25	24,3	24,9
30	30	29,8
35	34,7	35
40	39,6	39,9
45	44,1	44,2
50	49,1	49,9
55	55,1	54,8
60	59,6	59,6
65	65,1	64,9
70	69,7	70,1
75	74,4	74,55
80	80,5	80,25
85	84,3	85,07
90	89,2	89,46
95	94,7	95,1
100	99,8	99,98
105	105,1	104,81
110	109,7	110,51
115	115	114,88
120	120,1	120,599
125	125	124,98
130	129,7	129,8
135	135	135,01
140	139,9	139,89
145	144,8	144,71
150	149,7	149,9
155	154,8	154,8
160	159,6	159,1
165	164,4	164,89

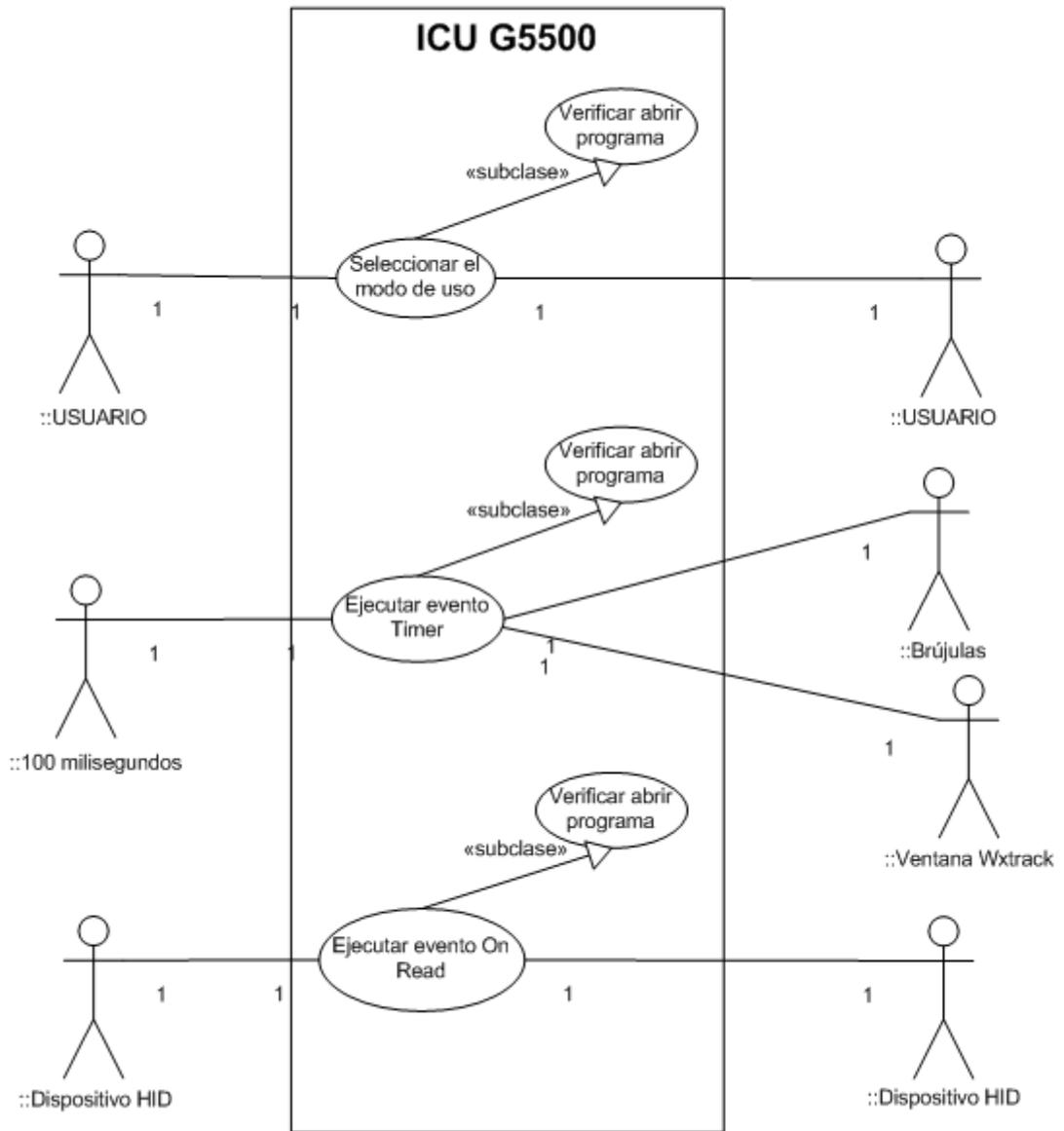
170	169,7	170,15
175	174,4	174,54
180	180,2	180,24
185	184,9	184,9
190	190,1	189,45
195	194,7	195,1
200	199,7	199,53
205	204,3	204,36
210	209,2	210,5
215	215	214,88
220	219,4	219,7
225	224,5	224,53
230	229,4	229,558
235	234,4	235,02
240	239	239,88
245	244,9	244,7
250	249,6	249,91
255	254,6	254,79
260	259,9	259,61
265	264,3	264
270	269,8	270,1435
275	274,2	274,53
280	280,1	280,668
285	285,1	285,054
290	289,6	289,87
295	294,1	294,26
300	299,3	299,8
305	305,1	304,78
310	309,4	309,61
315	315,2	315,31
320	320	320,13
325	325,2	324,52
330	329,6	329,78
335	335	335,048
340	340,3	339,87
345	345,3	345,13
350	348,8	349,95
355	354,3	354,78
360	359,3	359,606

<b>PRUEBA ELEVACION DEL GS-232</b>	
<b>Ángulos</b>	<b>Real</b>
0	1
5	5
10	9,7
15	14,6
20	20,6
25	25,5
30	30,2
35	35,1
40	40,2
45	45,5
50	50,7
55	55,4
60	60,5
65	65,3
70	70,4
75	75,1
80	79,9
85	84,7
90	89,9
95	94,6
100	99,8
105	104,8
110	109,4
115	114,7
120	119,7
125	124,6
130	129,7
135	134,8
140	139,8
145	144,3
150	150
155	154,9
160	160,1
165	165,2
170	170,8
175	175,4
180	180,4

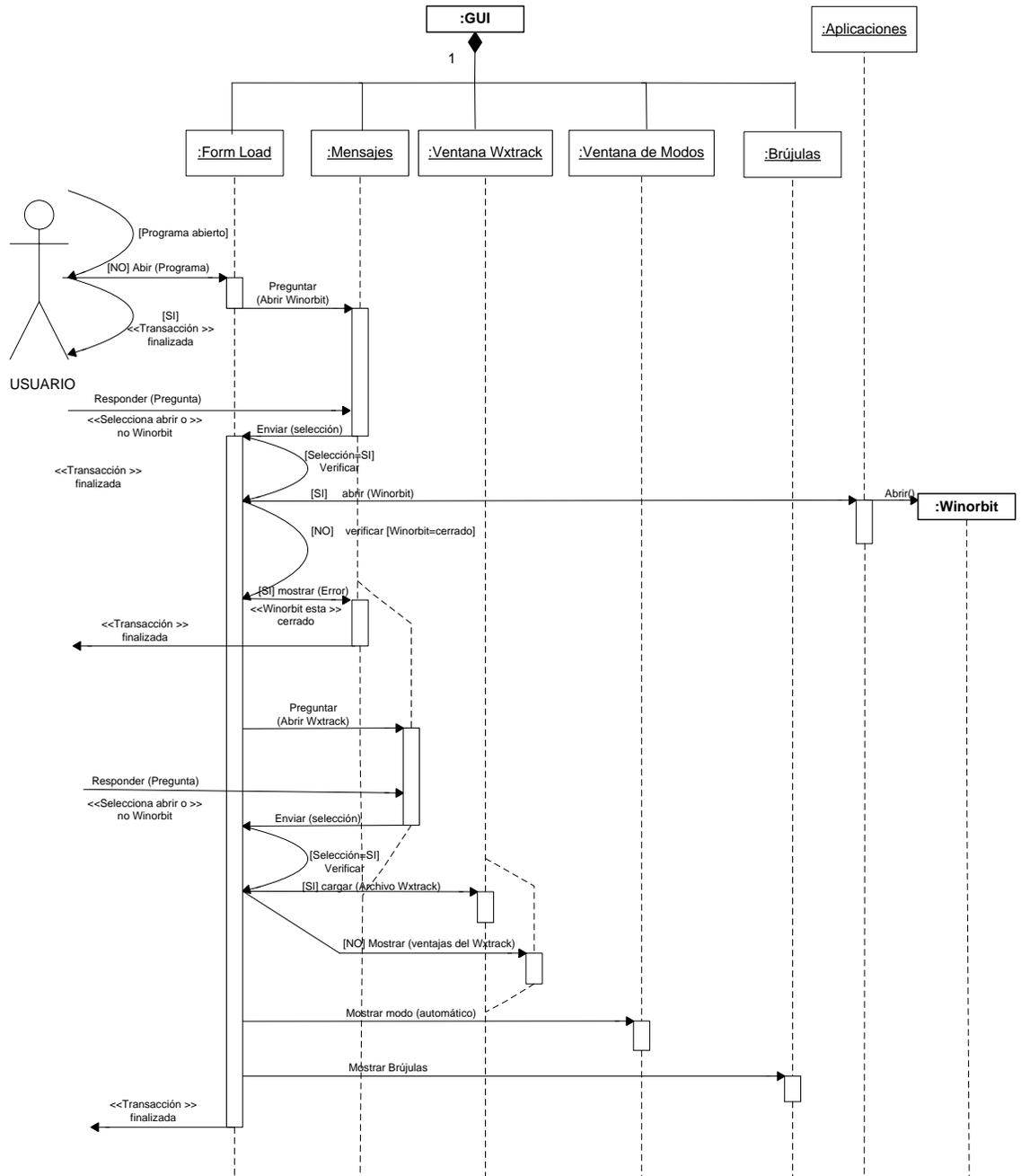
<b>PRUEBA AZIMUT DEL GS-232</b>	
<b>Ángulos</b>	<b>Real</b>
0	0
5	5,5
10	10,3
15	15
20	19,8
25	25,1
30	30
35	34,5
40	40,1
45	44,6
50	49,4
55	55,1
60	59,9
65	64,4
70	70
75	75
80	79,7
85	85
90	89,5
95	95
100	100,5
105	104,5
110	110,6
115	114,4
120	120,2
125	124,5
130	130,5
135	134,8
140	140,2
145	144,7
150	149,2
155	155,2
160	159,7
165	164,8
170	170,6
175	175,3
180	180,1

185	184,8
190	189,3
195	195,5
200	200,3
205	204,2
210	209,3
215	214,7
220	220,1
225	224,4
230	230,4
235	235,2
240	240,2
245	245,2
250	249,6
255	255,7
260	260
265	264,9
270	270,2
275	274,8
280	280,6
285	284,8
290	290,1
295	294,7
300	299,6
305	305,3
310	309,4
315	314,9
320	319,7
325	325,7
330	330
335	335,6
340	339,5
345	345,7
350	350,3
355	354,9
360	359,6

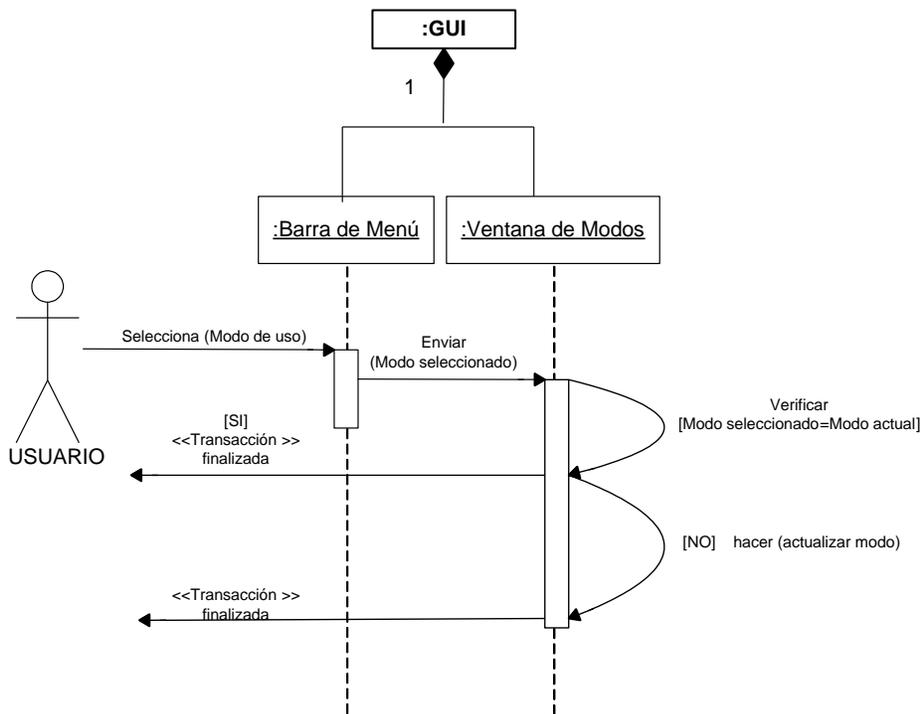
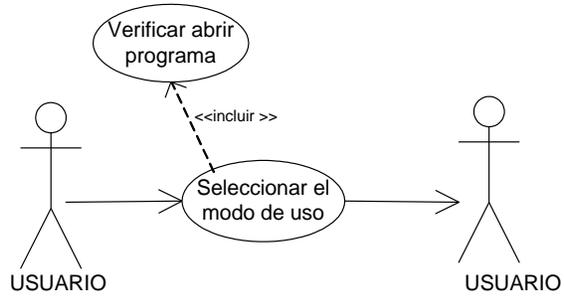
**ANEXO B. DIAGRAMA UML DE CASOS DE USO DEL PROGRAMA DEL PC**



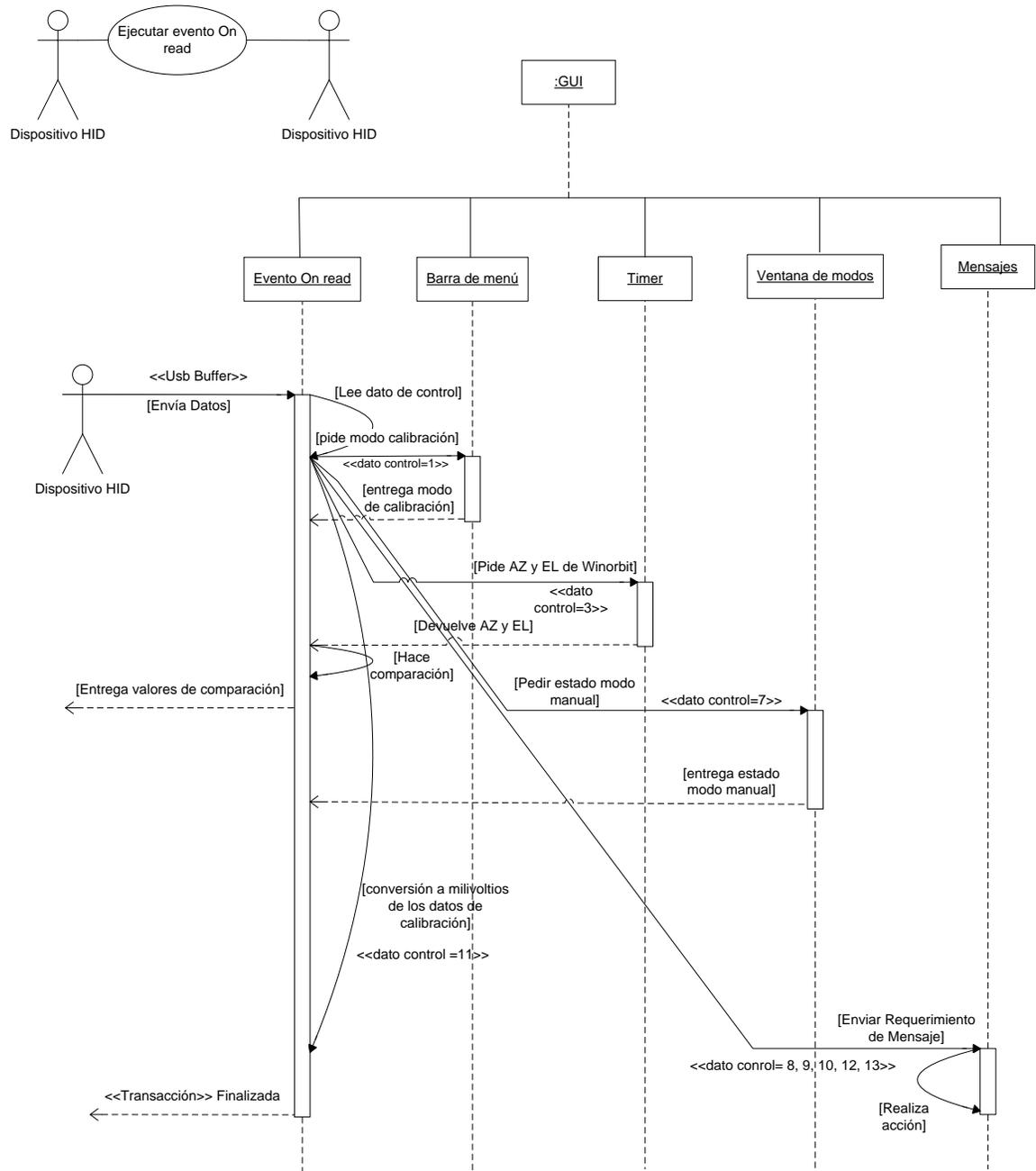
## ANEXO C. DIAGRAMA UML DE SECUENCIA VERIFICAR PROGRAMA ABIERTO



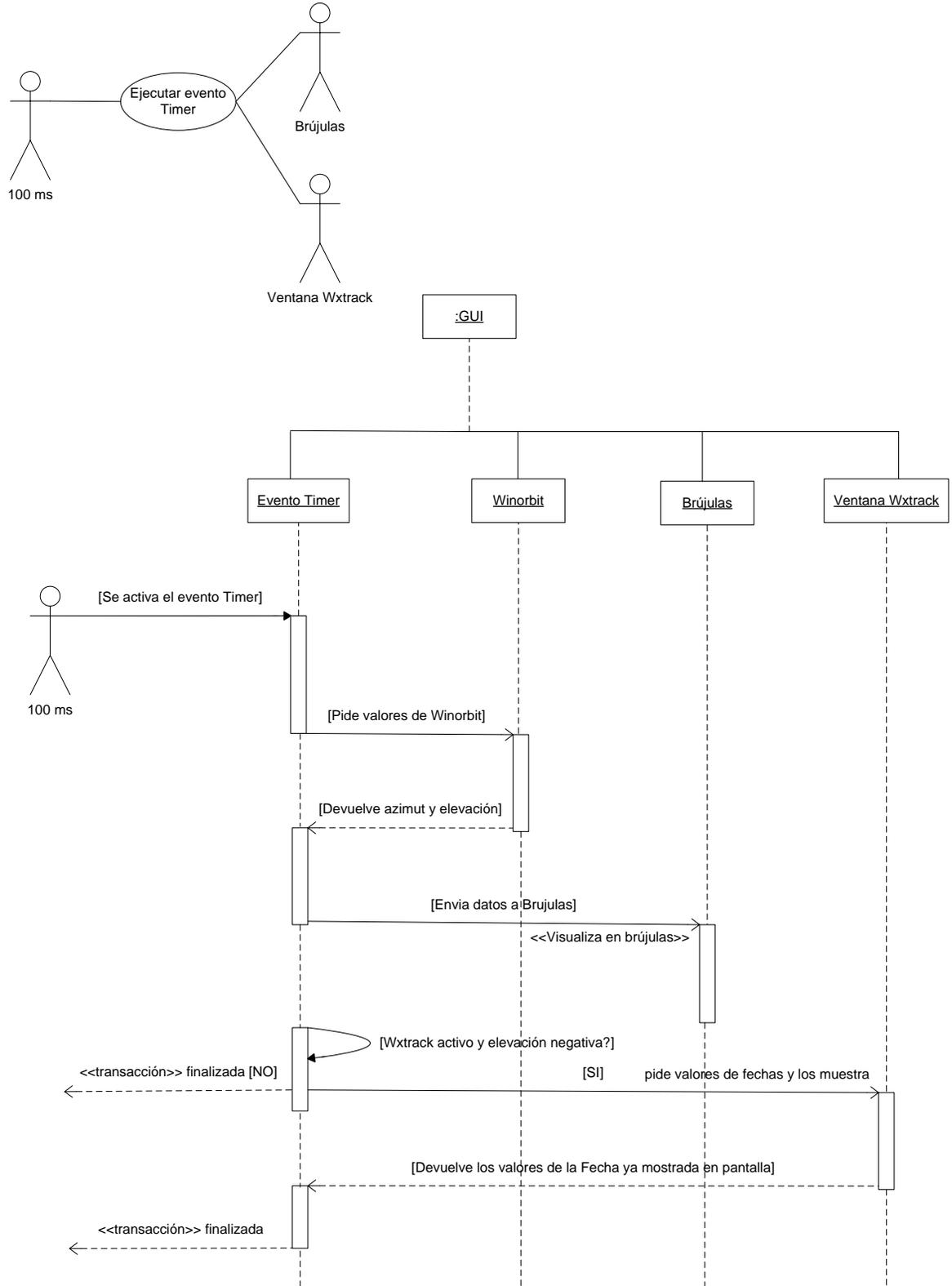
## ANEXO D. DIAGRAMA UML DE SECUENCIA SELECCIONAR MODO DE USO



## ANEXO E. DIAGRAMA UML DE SECUENCIA EJECUTAR EVENTO ON READ



## ANEXO F. DIAGRAMA UML DE SECUENCIA EJECUTAR EVENTO TIMER



 **YAESU**

**GS-232B**

*Computer Control Interface  
for Antenna Rotators*



**VERTEX STANDARD CO., LTD.**

4-8-8 Nakameguro, Meguro-Ku, Tokyo 153-8644, Japan

**VERTEX STANDARD**

US Headquarters

10900 Walker Street, Cypress, CA 90630, U.S.A.

**YAESU EUROPE B.V.**

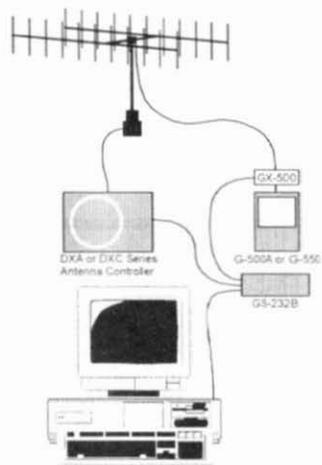
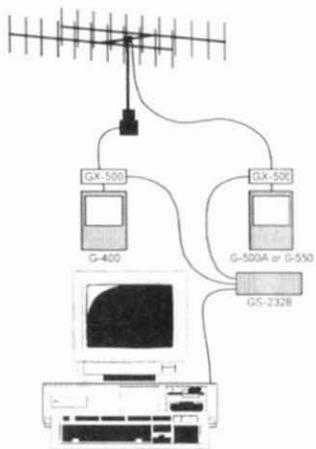
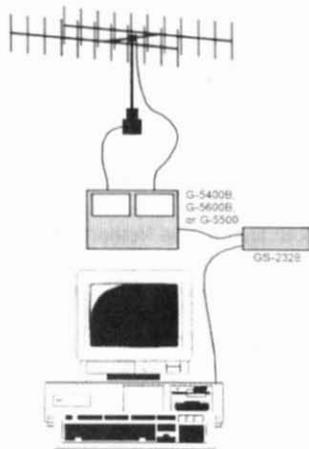
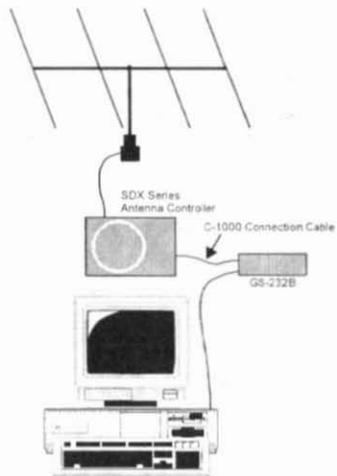
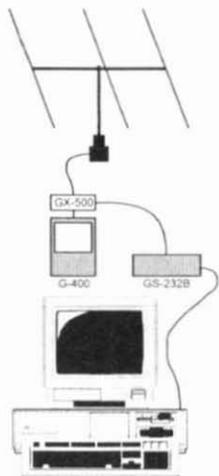
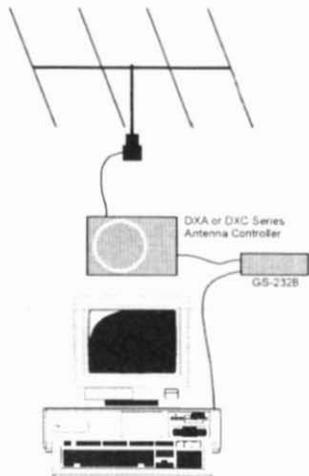
P.O. Box 75525, 1118 ZN Schiphol, The Netherlands

**YAESU UK LTD.**

Unit 12, Sun Valley Business Park, Winnall Close  
Winchester, Hampshire, SO23 0LB, U.K.

**VERTEX STANDARD HK LTD.**

Unit 5, 20/F., Seaview Centre, 139-141 Hoi Bun Road,  
Kwun Tong, Kowloon, Hong Kong



# GS-232B Computer Control Interface

## *for Yaesu Antenna Rotators*

The **GS-232B** provide digital control of most models of Yaesu antenna rotators\* from the serial port of an external personal computer.

The **GS-232B** contains its own microprocessor with a 10-bit analog-to-digital (A-D) converter and EEPROM. The async serial line can be configured for serial data rates from 1200 to 9600 baud. The **GS-232B** has a DB-9 "male" connector for connection to the (RS-232C) COM port of your computer. Purchase or construct a "straight" type serial cable, ensuring it has the correct gender and number of pins for connection to your system.

Firmware on the **GS-232B** supports either direct keyboard control, or commands from programs written specifically to support it (software is not supplied by Yaesu). In addition to reading and setting antenna angle and rotation speed, the firmware includes clocked positioning routines to auto-

matically step the antenna through up to 3800 angles at programmable intervals, such as for tracking band openings or satellites (with an elevation rotator).

Please read this manual carefully to install the **GS-232B**. If also installing a **G-400**, **G-500A** or **G-550** with the **GX-500** Automatic Control Adapter, follow the procedures in the **GX-500** manual before installing the **GS-232B**.

- ※ **G-800DXA/G-1000DXA/G-2800DXA** Azimuth Rotator,  
**G-800DXC/G-1000DXC/G-2800DXC** Azimuth Rotator,  
**G-400** Azimuth Rotator,  
**G-500A/G-550** Elevation Rotator,  
**G-5400B/G-5600B/G-5500** Az-El Rotator, and  
above Azimuth and Elevation rotator combination.
- G-400** Azimuth Rotator and **G-500A/G-550** Elevation Rotator requires one **GX-500** Automatic Control Adapter each.

---

**GENERAL**

---

- Power Requirements:** DC 12 V, 70 mA  
**Case Size:** 4.3" (W) x 0.8" (H) x 5.4" (D)  
(110 x 21 x 138 mm)  
**Weight (approx.):** 13.4 oz. (380 g)

**Semiconductors**

- Microprocessor:** PIC18C452  
(includes 10 bits A/D converter)  
**EEPROM:** 24LC256  
**Serial Comms:** RS-232C voltage levels,  
1200 to 9600 baud, 8 data bits,  
1 stop bit, no parity  
**Overflow Cont.:** Hardware Control (CTS port)

---

**CONNECTOR PINOUTS**

---

**Serial I/O:**

- 9-pin DB-9 connector (**RS-232C** connector)  
Pin 2 - Tx Data  
Pin 3 - Rx Data  
Pin 5 - Signal Ground  
Pin 7 - RTS  
Pin 8 - CTS

**Rotator Control:**

- 5-pin connector (**EL** connector)  
Pin 1 - UP switch (open collector)  
Pin 2 - DOWN switch (open collector)  
Pin 3 - analog output (0.5 - 4.5 V, four steps)  
Pin 4 - analog input (0-5V elevation)  
Pin 5 - analog ground  
5-pin connector (**AZ** connector)  
Pin 1 - RIGHT switch (open collector)  
Pin 2 - LEFT switch (open collector)  
Pin 3 - analog output (0.5 - 4.5 V, four steps)  
Pin 4 - analog input (0-5V azimuth)  
Pin 5 - analog ground

---

## SUPPLIED ACCESSORIES

---

- Control cable for the Azimuth Rotator<sup>×1</sup> ..... 1 pc  
(“5-pin” ↔ “Min-DIN” cable)
- Control cable for the Az/El Rotator<sup>×2</sup> ..... 1 pc  
(“Dual 5-pin” ↔ “DIN” cable)
- DC cable w/coaxial plug ..... 1 pc
- Hook & loop fasteners (for mounting) ..... 1 pc

×1: **G-800DXA, G-1000DXA, G-2800DXA,**  
**G-800DXC, G-1000DXC, and G-2800DXC**

×2: **G-5400B, G-5600B, and G-5500**

---

## AVAILABLE OPTIONS

---

- GX-500** Control Adapter  
(Check with your dealer)
  - C-1000** Connection Cable  
(for **SDX** series Azimuth Rotator)
  - NC-72B/C/F/U**<sup>×3</sup> AC Adapter
- ×3: “**B**” *suffix* is for use with 117 VAC,  
“**C**” *suffix* is for use with 220-240 VAC,  
“**F**” *suffix* is for use with 220 VAC, or  
“**U**” *suffix* is for use with 230 VAC

During installation, a personal computer with a serial port and terminal software is required to calibrate trimmers on the Controller and on the Control Interface. Any simple interactive terminal program can be used - it only has to transmit keystrokes as typed, and display characters received from the **GS-232B**.

## POWER & CONTROL CONNECTIONS

### DXA or DXC Series Azimuth Rotator

- ❑ Connect the supplied DC cable to a source of 12 VDC. The red lead connects to the Positive (+) DC terminal, and the black lead connects to the Negative (-) DC terminal. The **GS-232B** requires 70 mA. The supplied cable has a 500-mA fast-blow fuse. Use only the same type fuse for replacement.
- ❑ Plug the coaxial power connector into the **DC 12V** jack on the **GS-232B** rear panel.
- ❑ Connect the supplied Control cable (“5-pin” ↔ “Mini-DIN”) between the **EXT CONTROL** connector on the rotator’s controller and **AZ** connector on the rear panel of the **GS-232B** (Figure 1).

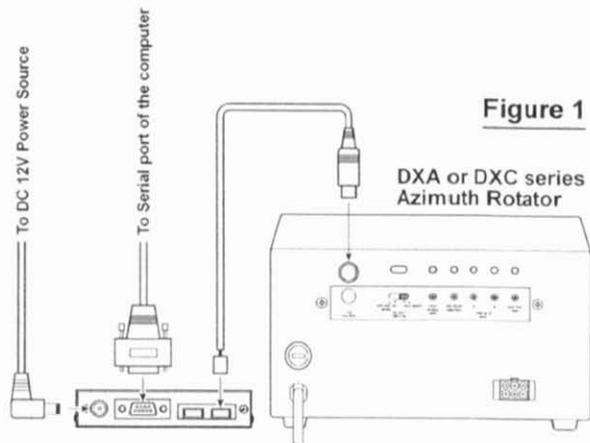
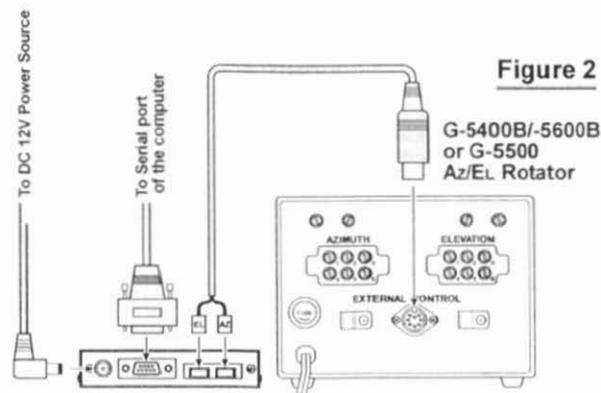


Figure 1

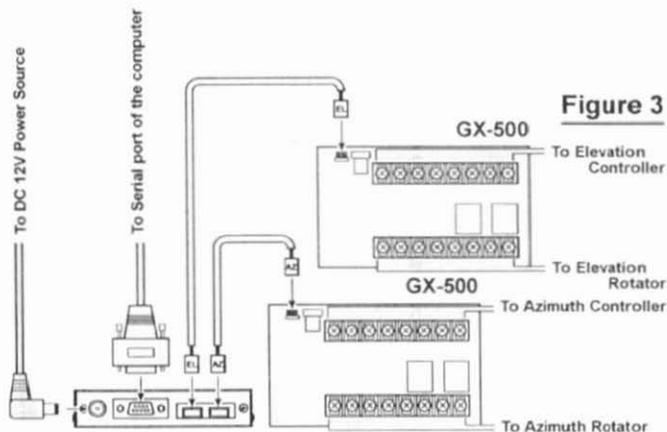
## POWER &amp; CONTROL CONNECTIONS

**G-5400B/-5600B Az-EL Rotator**

- ❑ Connect the supplied DC cable to a source of 12 VDC. The red lead connects to the Positive (+) DC terminal, and the black lead connects to the Negative (-) DC terminal. The **GS-232B** requires 70 mA. The supplied cable has a 500-mA fast-blow fuse. Use only the same type fuse for replacement.
- ❑ Plug the coaxial power connector into the **DC 12V** jack on the **GS-232B** rear panel.
- ❑ Connect the supplied Control cable ("Dual 5-pin" ↔ "DIN") between the rotator's controller and **GS-232B**. Be careful to match the "AZ" and "EL" labels on the cable with the same labels on the rear panel of the **GS-232B** (Figure 2).

**G-400/G-500 or G-400/G-550 & pair of GX-500**

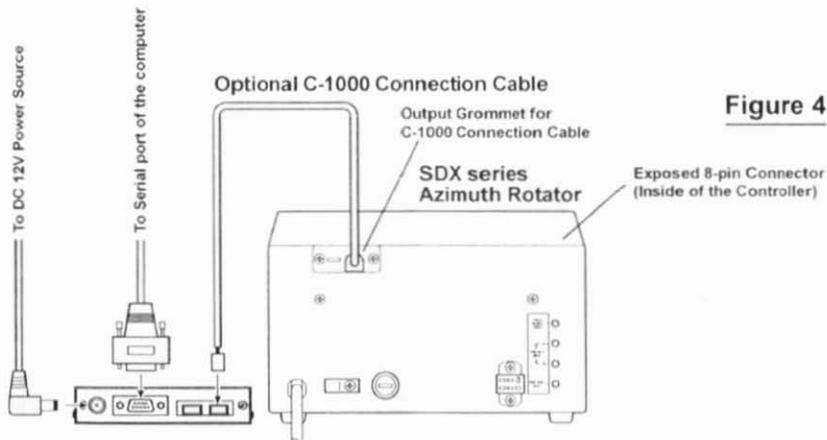
- ❑ Connect the supplied DC cable to a source of 12 VDC. The red lead connects to the Positive (+) DC terminal, and the black lead connects to the Negative (-) DC terminal. The **GS-232B** requires 70 mA. The supplied cable has a 500-mA fast-blow fuse. Use only the same type fuse for replacement.
- ❑ Plug the coaxial power connector into the **DC 12V** jack on the **GS-232B** rear panel.
- ❑ Connect the 5-pin to 5-pin cable (supplied with the **GX-500**; requires two sets) between the **GX-500(s)** and **GS-232B** (Figure 3).



## POWER &amp; CONTROL CONNECTIONS

**SDX Series Azimuth Rotator**

- ❑ Prepare the optional **C-1000** Connection Cable.
  - ❑ Remove the Top Cover from the controller.
  - ❑ Connect the 8-pin connector of the **C-1000** Connection cable to the exposed 8-pin connector located the rear left corner in the controller.
  - ❑ Route the 5-pin connector of the **C-1000** Connection cable through out the rubber grommet on the rear panel of the controller, and connect it to the **AZ** connector on the rear panel of the **GS-232B** (Figure 4).
  - ❑ Replace the Top Cover.
- ❑ Connect the supplied DC cable to a source of 12 VDC. The red lead connects to the Positive (+) DC terminal, and the black lead connects to the Negative (-) DC terminal. The **GS-232B** requires 70 mA. The supplied cable has a 500-mA fast-blow fuse. Use only the same type fuse for replacement.
  - ❑ Plug the coaxial power connector into the **DC 12V** jack on the **GS-232B** rear panel.

**Figure 4**

---

## COMPUTER CONNECTION

---

- ❑ With the computer switched off, connect the DSUB-9 pin Straight RS-232C cable to the serial port of your computer, then connect the other end of the RS-232C cable to the **RS-232C** connector on the rear panel of the **GS-232B**.
- ❑ Turn the computer's power switch on, then turn the **GS-232B**'s power switch on.
- ❑ Within 15 seconds of swithing the **GS-232B**'s power, press the [↵] (ENTER) key several times to adjust the **GS-232B**'s data baud rate to the Computer's data baud rate. If the computer is successful to setting the data baud rate, the computer's diaplay will show "Connect OK."
- ❑ If you are using a **G-400** Azimuth Rotator, or **G-5400B/G-5600B** Az-EL Rotator, press [P36] → [↵] (P, 3, 6, and ENTER) on the computer keyboard to disable the 450° rotation capability of the **GS-232B**. To restore 450° rotation capability, press [P45] → [↵] (P, 4, 5, and ENTER) on the computer keyboard.

---

**DXA/DXC/SDX SERIES AZIMUTH ROTATOR**

---

**Azimuth Offset Null**

- From the Controller panel, set the Rotator fully counter-clockwise (set to 0°).
- Press [O] → [↵] (the letter "oh," and "ENTER") on the computer keyboard to activate the azimuth calibration routine. The computer display should show "are you sure?"
- Press [Y] → [↵] ("Y" and "ENTER") on the computer keyboard. The computer display should show "Completed," and save the calibration data and exit the azimuth calibration routine.

**Azimuth A-D Calibration**

- From the Controller panel, set the Rotator fully clockwise (to the right).
- Press [F] → [↵] ("F" and "ENTER") on the computer keyboard to activate the Control Interface's azimuth A-D converter calibration routine. The computer's display should show "AZ=aaa," where "aaa" is a three-digit number which indicates the azimuth heading in degrees.
- Adjust the **OUT VOL ADJ** potentiometer on Controller rear panel so as to get a reading of "450" on the computer's display. This reading ("450 degrees") corresponds to the actual beam heading you established when you pointed the azimuth rotator to the East.
- Press [↵] ("ENTER") on the computer keyboard to exit the azimuth A-D converter calibration routine.

---

**Important Note !**

---

If your controller's indicator needle starts from a point other than 0° (North) (such as the default setting of the USA version), align the starting point of the controller's indicator needle to 0° (North) before calibration. When finish the calibration, set the controller's indicator needle to the desired point. Refer to the rotator's user manual for details regarding the indicator needle alignment.

---

## G-400 AZIMUTH ROTATOR

---

### Azimuth Offset Null

- ❑ From the Controller panel, set the Rotator fully counter-clockwise (set to 0°).
- ❑ Press [O] → [↵] (the letter “oh,” and “ENTER”) on the computer keyboard to activate the azimuth calibration routine. The computer display should show “are you sure?”
- ❑ Press [Y] → [↵] (“Y” and “ENTER”) on the computer keyboard. The computer display should show “Completed,” and save the calibration data and exit the azimuth calibration routine.

### Azimuth A-D Calibration

- ❑ From the Controller panel, set the Rotator fully clockwise (to the right).
- ❑ Press [F] → [↵] (“F” and “ENTER”) on the computer keyboard to activate the Control Interface’s azimuth A-D converter calibration routine. The computer’s display should show “AZ=aaa,” where “aaa” is a three-digit number which indicates the azimuth heading in degrees.
- ❑ Adjust the **FULL SCALE ADJ** on the **GX-500** so as to get a reading of “180” on the computer’s display. This reading (“180 degrees”) corresponds to the actual beam heading you established when you pointed the azimuth rotator to the South (the fully clockwise setting).
- ❑ Press [↵] (“ENTER”) on the computer keyboard to exit the azimuth A-D converter calibration routine.

---

## G-5400B/-5600B AZ-EL ROTATOR

---

### Azimuth Offset Null

- ❑ From the Controller panel, set the Rotator fully counter-clockwise (set to  $0^\circ$ ).
- ❑ Press [O] → [↵] (the letter “oh,” and “ENTER”) on the computer keyboard to activate the azimuth calibration routine. The computer display should show “are you sure?”
- ❑ Press [Y] → [↵] (“Y” and “ENTER”) on the computer keyboard. The computer display should show “Completed,” and save the calibration data and exit the azimuth calibration routine.

### Azimuth A-D Calibration

- ❑ From the Controller panel, set the Azimuth Rotator fully clockwise (to the right).
- ❑ Press [F] → [↵] (“F” and “ENTER”) on the computer keyboard to activate the Control Interface’s azimuth A-D converter calibration routine. The computer’s display should show “AZ=aaa,” where “aaa” is a three-digit number which indicates the azimuth heading in degrees.
- ❑ Adjust the **OUT VOL ADJ** potentiometer on the “**AZIMUTH**” (left) side of the Controller rear panel so as to get a reading of “180” on the computer’s display. This reading (“180 degrees”) corresponds to the actual beam heading you established when you pointed the azimuth rotator to the South.
- ❑ Press [↵] (“ENTER”) on the computer keyboard to exit the azimuth A-D converter calibration routine.

---

## G-5400B/-5600B Az-EL ROTATOR

---

### Elevation Offset Null

- ❑ From the Controller panel, set the Elevation Rotator to the “left” horizon (down, set to 0°).
- ❑ Press [O2] → [↵] (the letter “oh,” “2,” and “ENTER”) on the computer keyboard to activate the elevation calibration routine. The computer display should show “are you sure?”
- ❑ Press [Y] → [↵] (“Y” and “ENTER”) on the computer keyboard. The computer display should show “Completed,” and save the calibration data and exit the elevation calibration routine.

### Elevation A-D Calibration

- ❑ From the Controller panel, set the Elevation Rotator to full scale (180°: “right” horizon).
- ❑ Press [F2] → [↵] (F, 2, and ENTER) on the computer keyboard to activate the Control Interface’s elevation A-D converter calibration routine. The computer will display “AZ=aaa EL=eee,” where “eee” is a three-digit number which indicates the elevation heading in degrees. For the purposes of this alignment, you may ignore the (azimuth) “aaa” numbers.
- ❑ Adjust the **OUT VOL ADJ** potentiometer on the “**EL-EVATION**” (right) side of the Controller rear panel so as to get a reading of “180” on the computer’s display. This reading (“180 degrees”) corresponds to the actual beam heading you established when you pointed the elevation rotator to the 180° position.
- ❑ Press [↵] (“ENTER”) on the computer keyboard to exit the elevation A-D converter calibration routine.

---

## G-5500 Az-EL ROTATOR

---

### Azimuth Offset Null

- From the Controller panel, set the Rotator fully counter-clockwise (set to 0°).
- Press [O] → [↵] (the letter “oh,” and “ENTER”) on the computer keyboard to activate the azimuth calibration routine. The computer display should show “are you sure?”
- Press [Y] → [↵] (“Y” and “ENTER”) on the computer keyboard. The computer display should show “Completed,” and save the calibration data and exit the azimuth calibration routine.

### Azimuth A-D Calibration

- From the Controller panel, set the Azimuth Rotator fully clockwise (to the right).
- Press [F] → [↵] (F and ENTER) on the computer keyboard to activate the Control Interface’s azimuth A-D converter calibration routine. The computer’s display should show “AZ=aaa,” where “aaa” is a three-digit number which indicates the azimuth heading in degrees.
- Adjust the **OUT VOL ADJ** potentiometer on the “**AZIMUTH**” (left) side of the Controller rear panel so as to get a reading of “450” on the computer’s display. This reading (“450: 360 degrees + 90 degrees”) corresponds to the actual beam heading you established when you pointed the azimuth rotator fully clockwise.
- Press [↵] (“ENTER”) on the computer keyboard to exit the azimuth A-D converter calibration routine.

---

## G-5500 AZ-EL ROTATOR

---

### Elevation Offset Null

- ❑ Press [O2] → [↵] (the letter “oh,” “2,” and “ENTER”) on the computer keyboard to activate the elevation calibration routine. The computer display should show “are you sure?”
- ❑ Press [Y] → [↵] (“Y” and “ENTER”) on the computer keyboard. The computer display should show “Completed,” and save the calibration data and exit the elevation calibration routine.

### Elevation A-D Calibration

- ❑ From the Controller panel, set the Elevation Rotator to full scal (180°: “right” horizon).
- ❑ Press [F2] → [↵] (F, 2, and ENTER) on the computer keyboard to activate the Control Interface’s elevation A-D converter calibration routine. The computer will display “AZ=aaa EL=eee,” where “eee” is a three-digit number which indicates the elevation heading in degrees. For the purposes of this alignment, you may ignore the (azimuth) “aaa” numbers.
- ❑ Adjust the **OUT VOL ADJ** potentiometer on the “**EL-EVATION**” (right) side of the Controller rear panel so as to get a reading of “180” on the computer’s display. This reading (“180 degrees”) corresponds to the actual beam heading you established when you pointed the elevation rotator to the 180° position.
- ❑ Press [↵] (“ENTER”) on the computer keyboard to exit the elevation A-D converter calibration routine.

---

## G-500 ELEVATION ROTATOR

---

### Elevation Offset Null

- ❑ Press [02] → [↵] (the letter “oh,” “2,” and “ENTER”) on the computer keyboard to activate the elevation calibration routine. The computer display should show “are you sure?”
- ❑ Press [Y] → [↵] (“Y” and “ENTER”) on the computer keyboard. The computer display should show “Completed,” and save the calibration data and exit the elevation calibration routine.

### Elevation A-D Calibration

- ❑ From the Controller panel, set the Elevation Rotator to full scale (180°: “right” horizon).
- ❑ Press [F2] → [↵] (F, 2, and ENTER) on the computer keyboard to activate the Control Interface’s elevation A-D converter calibration routine. The computer will display “AZ=aaa EL=eee,” where “eee” is a three-digit number which indicates the elevation heading in degrees. For the purposes of this alignment, you may ignore the (azimuth) “aaa” numbers.
- ❑ Adjust the **OUT VOL ADJ** potentiometer on the **GX-500** so as to get a reading of “180” on the computer’s display. This reading (“180 degrees”) corresponds to the actual beam heading you established when you pointed the elevation rotator to the 180° position.
- ❑ Press [↵] (“ENTER”) on the computer keyboard to exit the elevation A-D converter calibration routine.

---

**GENERAL**

---

If you wish, you can mount the **GS-232B** on top of your Rotator Controller using the two supplied hook-and-loop fastener strips. Just remove the backing from one side of each strip, and press into place on the bottom of the **GS-232B**. Then remove the backing from the other side, and press the **GS-232B** into place on the Controller.

After installation and calibration, the Control Interface can accept commands entered directly from the keyboard, or from a program written specifically to support it (not supplied by Yaesu). For brief summaries of the commands recognized by the Control Interface, press [H] → [↵] for a list of azimuth commands, or [H2] → [↵] for elevations commands. Keep in mind that all commands require that the **ENTER** key be pressed after the command letter (or “ODh” be sent by a control program), although we will not repeat this when discussing the commands. Also note that any command letter may be sent in either upper or lower case. The info screens shown on the next page will be returned by the Control Interface.

Most commands have two versions: one for azimuth, and one for elevation. Commands are not echoed by the Control Interface, but a carriage return character (“ODh”) is returned after every command, and also a line feed character (“OAh”) if the command invoked returned data. Invalid commands cause “? >” to be returned and the input buffer cleared. Note that all angles are in degrees, beginning with zero at the most counterclockwise azimuth (or horizontal elevation). Angles sent to the Control Interface must be 3 digits long (left-zero-padded).

## COMMAND LIST

In the following command descriptions, the elevation version of each command, where there is one, is shown in parentheses (but don't type the parentheses). Remember that elevation commands require the **G-5400B**, **G-5600B** or **G-5500** Az/El Rotators, or the **GX-500** adapter and the **G-500** or **G-550** Elevation Rotator.

## O (O2)

**Offset calibration** for the Azimuth (Elevation): preset rotator manually fully counter-clockwise ("left" horizon for elevation), send command, and press [Y] → [↵] ("Y" and "ENTER") on the computer keyboard to store setting.

## H (H2, H3)

Returns list of commands (see page 19 and 20).

## F (F2)

**Full Scale Calibration:** preset rotator manually to full scale, send command, adjust **OUT VOL ADJ** trimmer on rear of controller (or **GX-500** elevation adapter) until the returned data is "+180 or +450" ("Onn+180" for elevation). Press [↵] ("ENTER") on the computer keyboard to exit the calibration routine.

## R (U)

Start turning the rotator to the right (up)

## L (D)

Start turning the rotator to the left (down).

## A (E)

Stop azimuth (elevation) rotation.

## S

Stop: cancel current command before completion.

## C (B)

Return current azimuth (elevation) angle in the form "AZ=aaa" ("EL=eee" for "B" command) degrees.

## C2

Return azimuth and elevation ("AZ=aaa EL=eee", where "aaa" = azimuth, "eee" = elevation).

## P36

Switch the azimuth angle to 360 degree mode.

## P45

Switch the azimuth angle to 450 degree mode.

## Z

Toggle the starting point of the azimuth indicator needle between "N (North)" and "S (South)."

This command is ignored when the rotator is set to "450 degree mode."

---

**COMMAND LIST**

---

**X $n$** 

Select azimuth rotator turning speed, where  $n = 1$  (slowest) to 4 (fastest). This command can be issued during rotation, and takes effect immediately. There is no equivalent for elevation.

**Maaa**

Turn to *aaa* degrees azimuth, where *aaa* is a three-digit number between "000" and "360" or "450" (depending on rotator type). Rotation starts upon execution of the command.

**Msss aaa bbb ccc**

This command, together with the [T] command, provides automatic, timed tracking of moving objects or propagation by the Control Interface itself. This command stores the time value *sss* seconds to wait between stepping from azimuth *aaa* to *bbb*, and then to *ccc*, etc. (from "2" to as many as "3800" angles may be stored with one command).

Note that this command is completely different than the [T] command with only one parameter: when multiple parameters are present, the first one is interpreted by the Control Interface as the rotation interval *sss*, not an angle. Valid ranges are "001" to "999" for *sss*, and "000" to "360" or "450" (depending on rotator type) for the angles. When this command is sent, the parameters are stored in the Control Interface's RAM,

and the rotator turns to angle *aaa* and waits for a subsequent [T] command to begin the actual stepping. All numbers must be 3 digits, space-separated. Stored values remain in effect until another [M] command is issued (this may have no parameters, in which case the "?>" error prompt is returned, but memories are still cleared), or until the controller is turned off or by toggling the **GS-232B** off and on.

**T**

See the [M] (above) and the [W] (below) command. Start automatic stepping routine (both azimuth and elevation): turn rotator to next sequentially memorized azimuth (or az-el pair, for the [W] command), wait *sss* seconds, and turn to next angle (or pair), etc. This command works only if a long-form [M] or [W] has been issued since power-up or the last reset.

## COMMAND LIST

## N

Return serial number of currently selected memorized point [*nnnn*], and total number of memorized points [*mmmm*], in the form =*nnnn*=*mmmm*. Must be preceded by either a long-form [M] or [W], and a T command. Used only during stepping (see [T] command).

The meaning of a “point” in this command following an [M] command is only an azimuth angle, so in this case *nnnn* and *mmmm* can range up to “3800” (the limit defined in the EEPROM<sup>×</sup> in the Control Interface). However, when elevation is involved, a “point” following a [W] command is represented by both an azimuth and an elevation angle, in which case *nnnn* and *mmmm* can range up to only “1900,” since each “point” is a pair of angles.

× The EEP ROM may be written with up to one million data points.

*Elevation Control Commands*

These commands are only for az-el operation. Note that an azimuth angle must always be supplied when changing elevation, and that a setting point consists of a pair of angles.

*Waaa eee*

Turn to *aaa* degrees azimuth and *eee* degrees elevation, where *aaa* is a three-digit number between “000” and “360” or “450” (depending on rotator type) and *eee* is a three-digit number between “000” and “180.” Rotation starts upon execution of the command.

*Wsss aaa eee aaa sss ...*

This command is similar to the [M] command: the first parameter is a time interval, and succeeding parameters are angles. With this command, however, angles are in azimuth-elevation pairs, each pair representing one antenna location. At most “1900” pairs can be sent and stored in the Control Interface. As with the other commands, the time interval range is limited to “001” to “999” (seconds), azimuth to “000” to “360” or “450” (depending on rotator type), and elevation to “000” to “180.”

When this command is sent, the rotators turn to the first *aaa* azimuth parameter and the first *eee* elevation parameter, and wait for a subsequent [T] command to begin the actual stepping (to the next azimuth-elevation pair). Stored values

---

## COMMAND LIST

---

remain in effect until another [W] command is issued (this may have no parameters, in which case the “?>” error prompt is returned, but memories are still cleared), or until the controller is turned off or by toggling the **GS-232B** off and on.

### Returned by [H] Command:

----- COMMAND LIST 1 -----

- |    |  |
|----|--|
| R  | Clockwise Rotation   |
| L  | Counter Clockwise Rotation   |
| A  | CW/CCW Rotation Stop   |
| C  | Antenna Direction Value  |
| M  | Antenna Direction Setting, MXXX  |
| M  | Time Interval Direction Setting,<br>MTTT XXX XXX XXX ---<br>[TTT = Step value]<br>[XXX = Horizontal Angle] |
| T  | Start Command in the time interval direction setting mode.   |
| N  | Total number of setting angles in "M" mode and traced number of all datas (setting angles)                 |
| S  | All Stop   |
| O  | Offset Calibration   |
| F  | Full Scale Calibration   |
| X1 | Rotation Speed 1 (Horizontal) Low  |
| X2 | Rotation Speed 2 (Horizontal) Middle 1   |
| X3 | Rotation Speed 3 (Horizontal) Middle 2   |
| X4 | Rotation Speed 4 (Horizontal) High   |

---

**COMMAND LIST**


---

**Returned by [H2] Command:**

----- HELP COMMAND 2 -----

- U UP Direction Rotation
- D DOWN Direction Rotation
- E UP/DOWN Direction Rotation Stop
- C2 Antenna Direction Value
- W Antenna Direction Setting.  
WXXX YYY
- W Time Interval Direction Setting.  
WTTT XXX YYY XXX YYY ---  
(TTT = Step value)  
(XXX = Horizontal Angle)  
(YYY = Elevation Angle)
- T Start Command in the time interval direction setting mode.
- N Total number of setting angle in "W" mode and traced number of all datas (setting angles)
- S All Stop
- O2 Offset Calibration
- F2 Full Scale Calibration
- B Elevation Antenna Direction Value

**Returned by [H3] Command:**

----- HELP COMMAND 3 -----

- P45 Set\_mode 450 Degree
- P36 Set\_mode 360 Degree
- Z Swich N Center/S Center

----- MODE -----

- mode 450 Degree (or mode 360 Degree)
- N Center (or S Center)





Copyright 2004  
VERTEX STANDARD CO., LTD.  
All rights reserved.

No portion of this manual  
may be reproduced  
without the permission of  
VERTEX STANDARD CO., LTD.

Printed in Japan

0401P-0E

