

Sistema de monitoreo para una plataforma aérea usando sistemas embebidos

Raúl Fernando Beltrán

Escuela Politécnica Nacional. Ingeniería Eléctrica y Electrónica
rfbsepn1991@outlook.com

Iván Alexander Llumiquinga

Escuela Politécnica Nacional. Ingeniería Eléctrica y Electrónica
ivan_pm92@hotmail.com

Eduardo Ávalos

Escuela Politécnica Nacional. Ingeniería Eléctrica y Electrónica
eduardo.avalos@epn.edu.ec

Ana Rodas Benalcázar

Escuela Politécnica Nacional. Ingeniería Eléctrica y Electrónica
ana.rodas@epn.edu.ec

Recibido: 01 de septiembre / *Aprobado:* 30 de noviembre 2017

Resumen

El sector aeronáutico está atravesando por una etapa donde la construcción de vehículos aéreos no tripulados, llamados UAV's de las siglas en inglés "Unmanned Aerial Vehicles", es un área de gran desarrollo. Este proyecto se encuentra enfocado al desarrollo de un sistema de monitoreo para ser embarcado en la plataforma aérea el cual está constituido por dos partes esenciales: el módulo de adquisición de datos que es incorporado al fuselaje del vehículo de prueba y de una estación en tierra en la cual se presentan todos los datos adquiridos, y que cuenta adicionalmente con un simulador de un hexacóptero implementado en el entorno de desarrollo Unity.

Palabras clave: aeronáutica, UAV's, sistema de monitoreo, sistemas embebidos.

Abstract

The aeronautical sector is going through a stage where the construction of unmanned aerial vehicles, called UAV's of the acronym in English "Unmanned Aerial Vehicles", is an area of great development. This project is focused on the development of a monitoring system to be shipped on the aerial platform which is constituted by two essential parts: the data acquisition module that is incorporated into the fuselage of the test vehicle and also of a ground station in which all the acquired data, is presented and which additionally has a simulator of a hexacopter implemented in the Unity development environment.

Keywords: aeronautics, UAV's, monitoring system, embedded systems.

Introducción

Un sistema de monitoreo utilizado en vehículos aéreos no tripulados consta de elementos necesarios para obtener información de diferentes variables, tales como: temperatura ambiental, temperatura de la fuente de alimentación, temperatura de la placa del sistema, humedad, posicionamiento satelital, presión barométrica, velocidad de aire, altitud sobre el nivel del mar, voltaje y corriente de la fuente de alimentación y datos del sistema de medida inercial. Para lograr este objetivo se utilizan diversos mecanismos tanto de adquisición como de procesamiento de señales. En este proyecto se recolectan los datos proporcionados por los sensores y se los envía a una tarjeta de desarrollo utilizando un hardware diseñado e implementado mediante un Arduino y una tarjeta UDOO. [1]

Los datos enviados al microcontrolador del Arduino se lo hace mediante comunicación serial UART de alta velocidad al procesador de alto nivel de la UDOO que mediante el módulo wifi envía los datos hacia una estación en tierra por medio de una red de área local mediante sockets con protocolo UDP.

La estación en tierra, que consta de un ordenador portátil, contiene una interfaz hombre máquina utilizada para el monitoreo del sistema, la cual es desarrollada mediante el lenguaje de programación Java en la que se aprecia de manera concreta las variables antes mencionadas y cuenta con alarmas para advertir al operador de condiciones riesgosas en el vuelo tales como vientos demasiado fuertes, carga baja de la batería, sobrecalentamiento de la tarjeta implementada, excesiva humedad que afecte el funcionamiento del sistema, etc.

Adicionalmente se implementó como parte de la HMI un simulador de un hexacóptero que tiene la opción de comunicarse con los instrumentos virtuales de la interfaz gráfica para que el operador tenga una experiencia previa con la estación de tierra.

I. Hardware

El desarrollo del hardware parte de la necesidad de tener un sistema dedicado especialmente para poder ser montado sobre una plataforma aérea lo que implica que debe ser lo más liviano y pequeño posible y tener además un mínimo consumo de corriente.

El esquema general del sistema de monitoreo implementado consta de varios sensores cuyas señales se acondicionan y procesan por un circuito de recolección de datos con el microcontrolador Atmel SAM3X8E ARM Cortex-M3 que está integrado en la tarjeta de desarrollo y un procesador Freescale i.MX 6 ARM Cortex-A9 Quad Core que en conjunto con el módulo wifi de la tarjeta de desarrollo envían los datos procesados hacia una HMI que conforma la estación en tierra para la visualización de los datos adquiridos por los sensores, como muestra la figura 1.

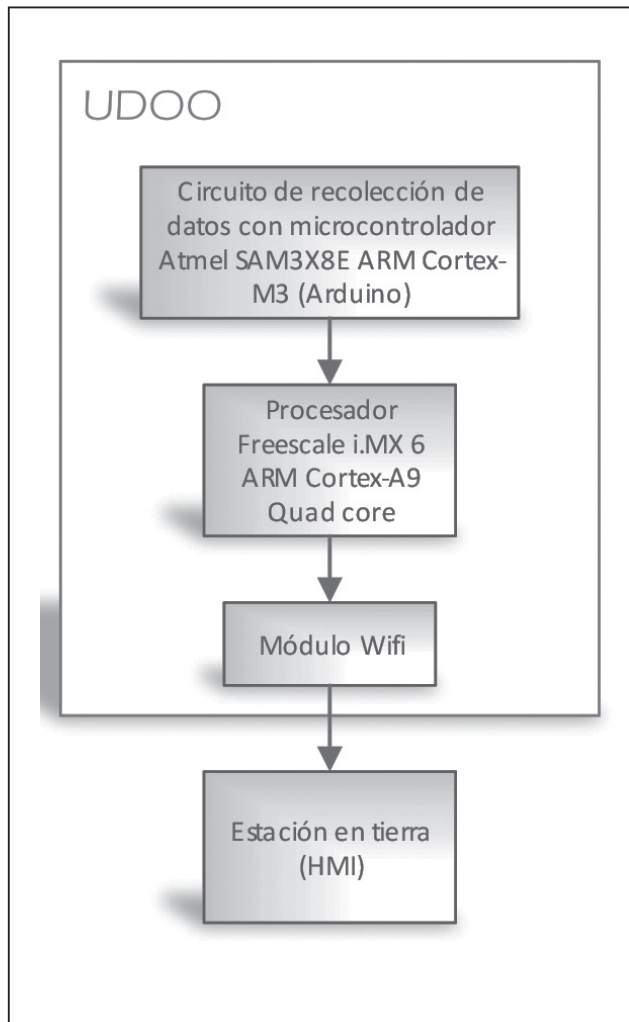


Figura 1. Diagrama de bloques general del sistema de monitoreo.

A. UDOO

Para recolectar los datos de los sensores y enviarlos hacia la estación en tierra se utiliza una tarjeta de desarrollo denominada “UDOO” la cual es una mini PC que puede ser usada con Android o Linux OS, con una tarjeta embebida Arduino. Esta tarjeta es un prototipo muy poderoso para el desarrollo y diseño de software. UDOO [4] mezcla dos mundos diferentes de computación, cada uno con sus fortalezas y debilidades, pero los dos son muy útiles hoy en día para propósitos educativos y la realización de prototipos. Es un sistema “open hardware”, de bajo costo equipado con un procesador ARM. I.MX6 Freescale y una sección compatible con Arduino basado en un procesador ATMEL SAM3X8E.

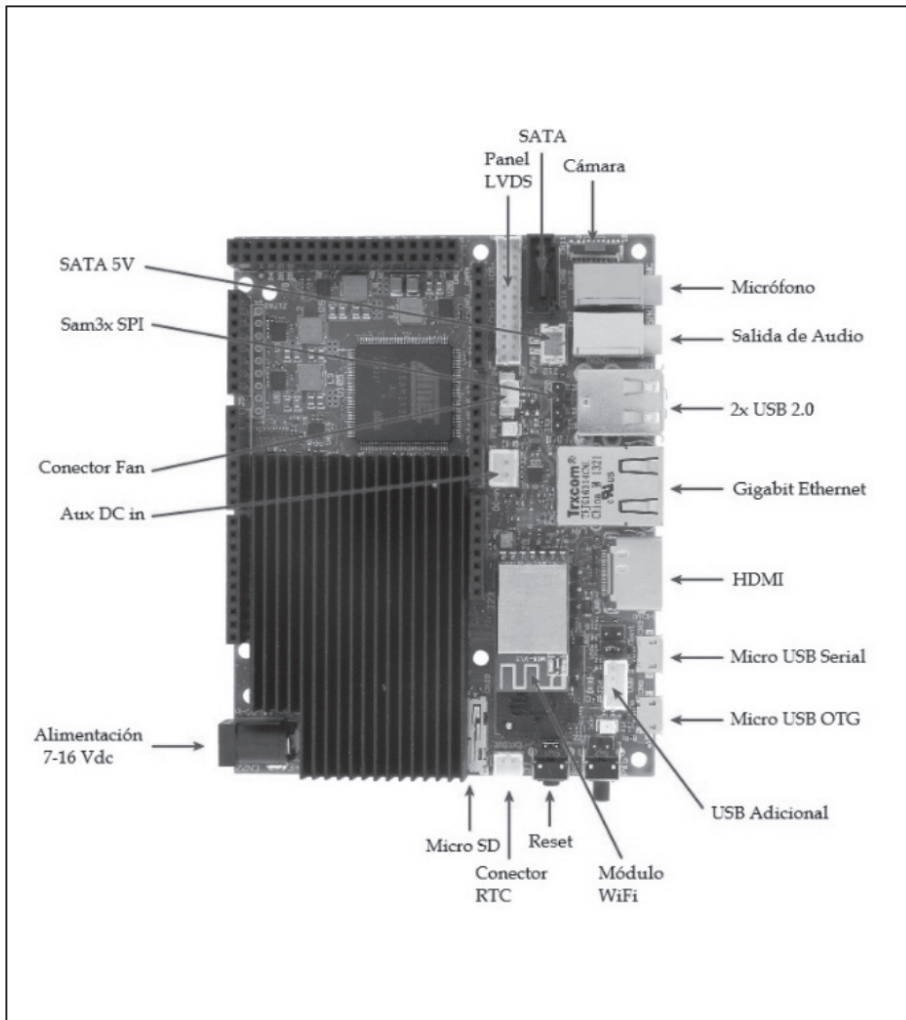


Figura 2. Tarjeta de desarrollo UDOO.

B. Circuito de recolección de datos

La tarjeta de desarrollo UDOO se convierte en la parte central del sistema al recibir y enviar los datos de todos los sensores empleados en el sistema de monitoreo. Los sensores se comunican con la tarjeta de desarrollo aprovechando los puertos serial, puertos I2C y entradas analógicas disponibles en el microcontrolador de bajo nivel que a su vez, vía serial, envía todos los datos de los sensores hacía el microcontrolador de alto nivel que tiene conexión con el módulo wifi para la transmisión de datos hacia tierra. Un esquema de la recolección de datos de los sensores se observa en la figura 3.

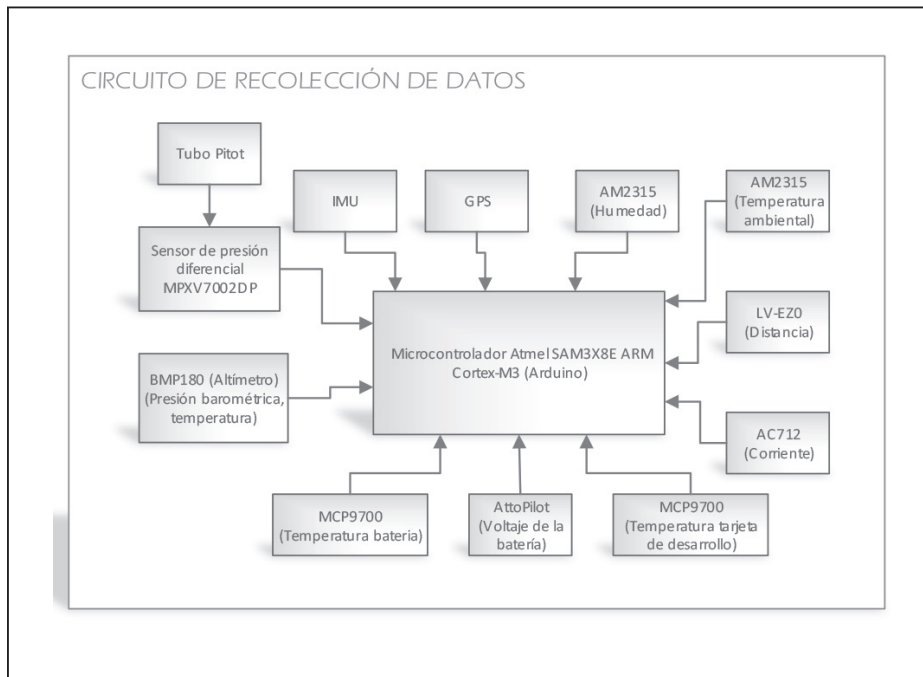


Figura 3. Diagrama del sistema de recolección de datos.

C. Sensores

El sistema de monitoreo emplea una serie de sensores clasificados en tres grupos [10]:
Sensores de detección y apoyo a la navegación autónoma:

- Unidad de medición inercial (IMU).
- GPS [8].
- Sensor de presión barométrica y temperatura para el cálculo de la altitud (altímetro) [11].
- Sensor ultrasónico para aterrizaje.
- Sensor de velocidad del aire.

Sensores de captación de parámetros del interior de la estructura:

- Sensor de temperatura de la tarjeta de desarrollo [12].
- Sensor de temperatura en la batería.
- Sensor de corriente del sistema de monitoreo.
- Sensor de voltaje de la batería.

Sensores de captación de parámetros del exterior de la estructura [9] :

- Sensor de humedad ambiental.

- Sensor de temperatura ambiental.

Todos los sensores están acoplados físicamente con la tarjeta de desarrollo por medio de una placa diseñada especialmente con las medidas de un Shield para Arduino Due como se la aprecia en la figura 4.

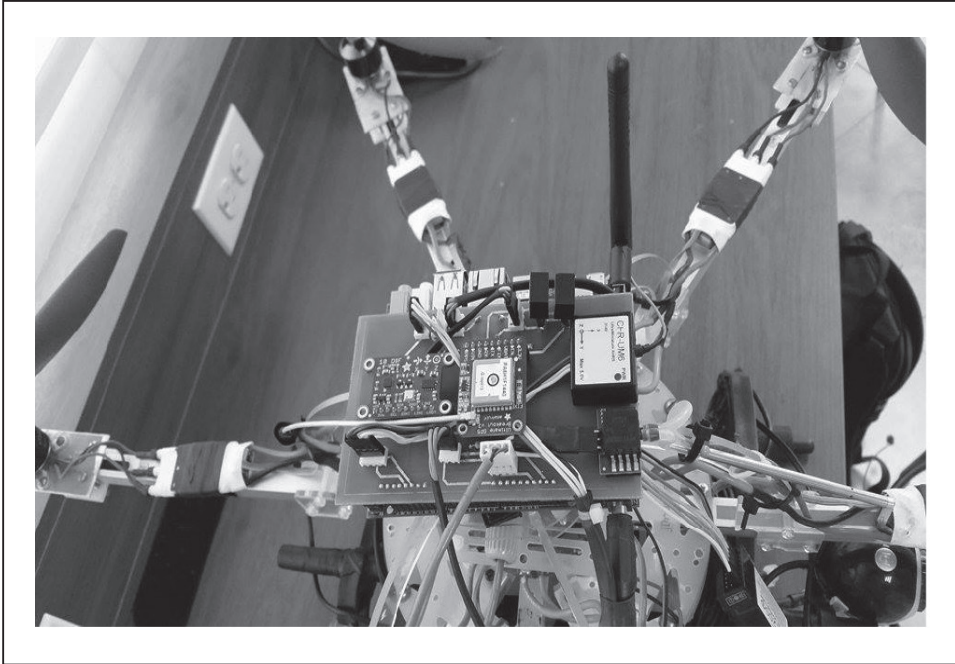


Figura 4. Tarjetas de desarrollo y de sensores instaladas en un hexacóptero.

II. Software

El software de recolección de datos desarrollado se lo hizo en el entorno Arduino IDE [7] para el microcontrolador Atmel SAM3X8E que se encuentra en la sección de la tarjeta de UDOO compatible con Arduino Due; el software HMI que se lo realizó en Netbeans y el software de simulación programado en Unity 3D que se ejecutan en una laptop.

Software de adquisición de datos

Para integrar todas las tareas que el microcontrolador de bajo nivel de la tarjeta desarrollo debe ejecutar sin saturarse se manejan “timers” y eventos seriales para la correcta distribución de los tiempos de ejecución para cada uno de los sensores utilizados de la siguiente manera, como se muestra en la figura 5.

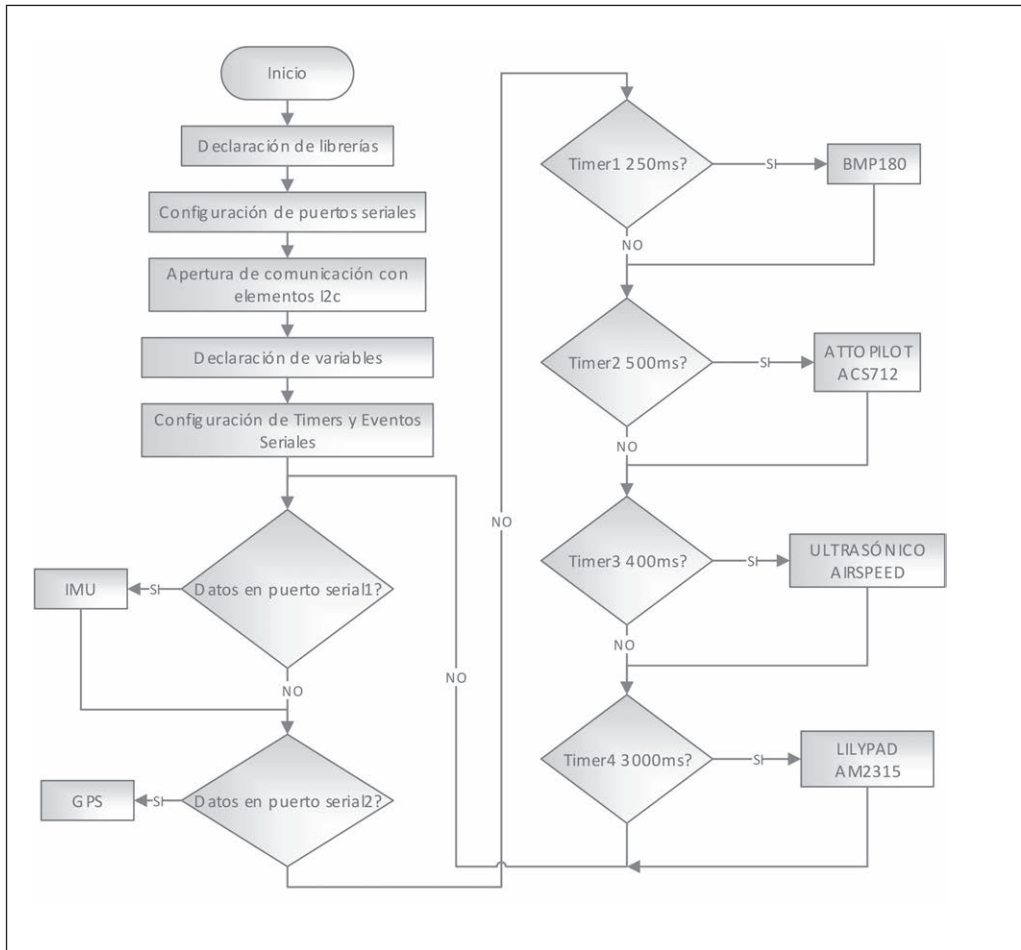


Figura 5. Diagrama del programa de lectura de sensores.

B. Software de comunicación hacia la estación en tierra

El programa principal que se ejecuta en la computadora a bordo tiene dos fines, el primero: la recepción de datos desde el microcontrolador de bajo nivel y la transmisión de datos hacia la estación de tierra, y el segundo fin brindar información de los sensores para cualquier otro programa que se desarrolle en este medio como puede ser el de control del hexacóptero. En la figura 6 se describe la estructura y secuencia de ejecución de las tareas programadas en la clase de JAVA [6] que se ejecuta cíclicamente en la tarjeta de desarrollo.

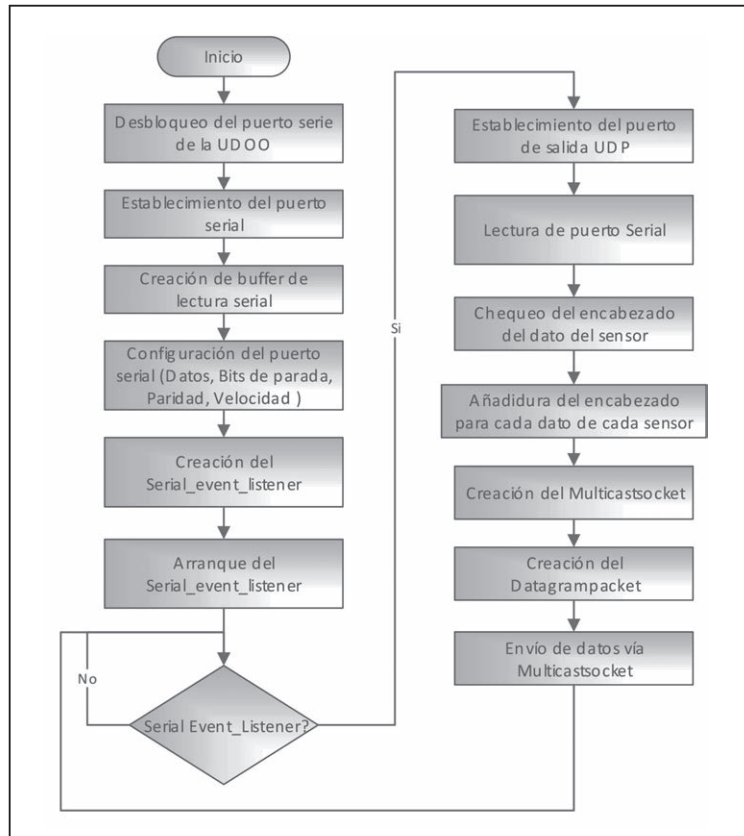


Figura 6. Diagrama del programa de enlace UDOO-HMI.

C. HMI

La interfaz de usuario desarrollada proporciona una serie de instrumentos, gráficos y diversas utilidades para el monitoreo constante de los parámetros de vuelo del hexacóptero. Este software se encuentra dividido en nueve pestañas con el fin de tener una buena organización, funcionalidad y facilidad en la manipulación del programa. Las pantallas en las que se encuentra dividida la interfaz son:

- Pantalla Principal.
- Pantalla de Actitud.
- Pantalla de la Unidad de Medición Inercial (IMU).
- Pantalla Velocidad.
- Pantalla Posición.
- Pantalla Otros Parámetros.
- Pantalla World Wind.
- Pantalla de Configuración.
- Pantalla de Recursos.

- 1) *Pantalla Principal*: En esta pantalla se encuentran las principales utilidades para monitorear los parámetros de vuelo del hexacóptero. Los instrumentos de la parte izquierda permiten la visualización de variables como altitud, ángulos de inclinación y velocidades, mientras que en la parte derecha se localiza la utilidad de mapas para registrar el posicionamiento del hexacóptero.



Figura 7. Pantalla Principal.

- 2) *Pantalla de Actitud*: La pantalla presenta gráficos en tiempo real de los ángulos de Euler junto a la altura para el monitoreo y registro adecuado de estas variables durante el transcurso del vuelo. [12]

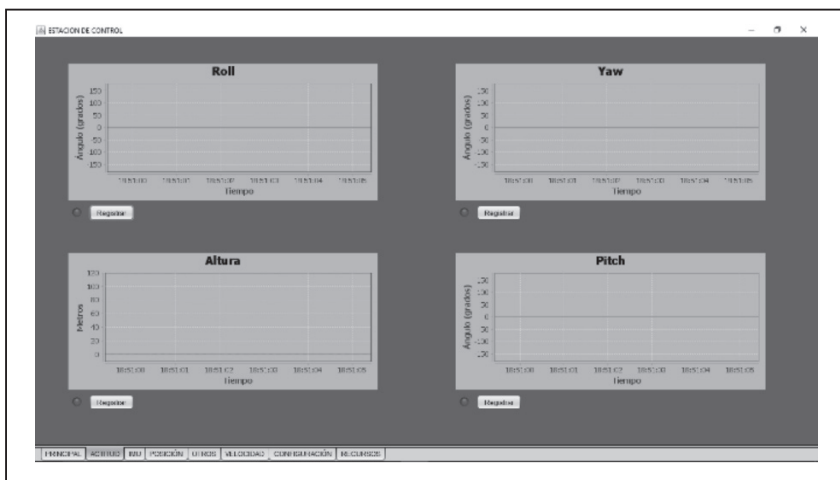


Figura 8. Pantalla de Actitud.

3) *Pantalla de la Unidad de Medición Inercial (IMU)*: Esta pestaña se compone de un mundo tridimensional OpenGL [2], en el cual, un hexacóptero animado simula los movimientos durante el vuelo y tres indicadores tipo barra indican la magnitud de los tres ángulos en tiempo real.

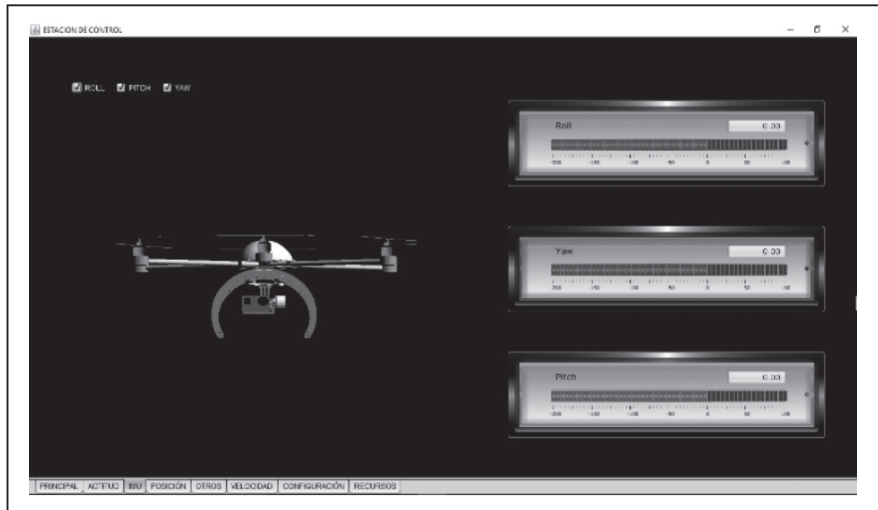


Figura 9. Pantalla de la Unidad de Medición Inercial (IMU).

4) *Pantalla Velocidad*: En esta pantalla se encuentran tres gráficos en tiempo real para cada una de las velocidades de navegación junto con un radar e indicadores de distancias para obtener la posición estimada del hexacóptero como respaldo del mapa de la Pantalla Principal.

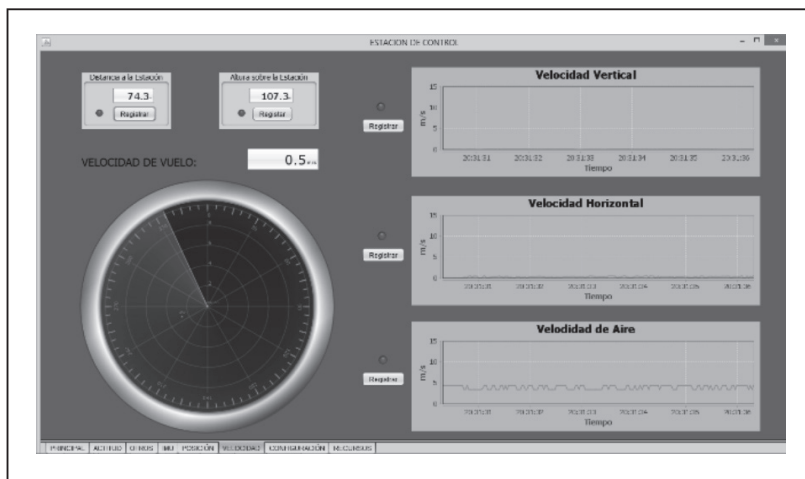


Figura 10. Pantalla Velocidad.

5) *Pantalla Posición:* La pantalla es una alternativa para la visualización dentro de un plano coordenado tridimensional de los movimientos en tiempo real del hexacóptero. Adicionalmente se encuentran herramientas para el registro y valores de velocidad, y posición en cada uno de los ejes.

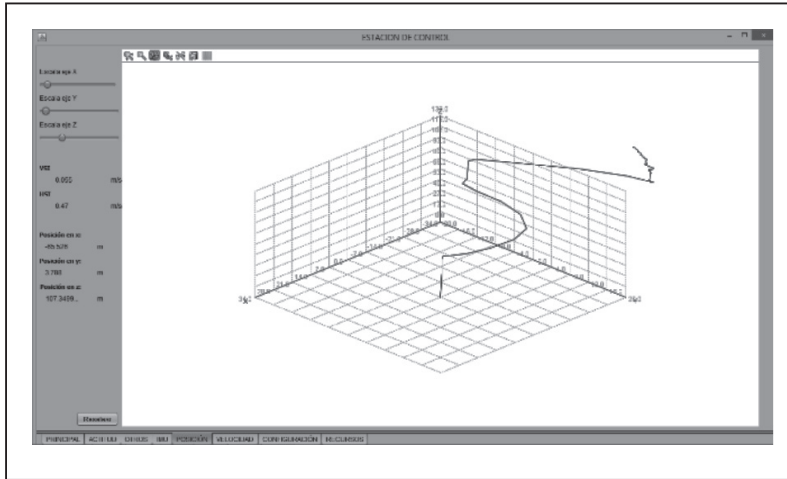


Figura 11. Pantalla Posición.

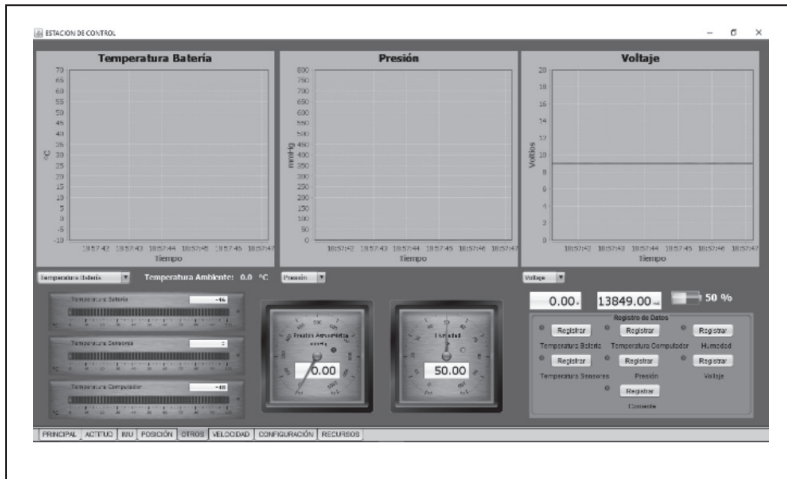


Figura 12. Pantalla otros parámetros.

6) *Pantalla Otros Parámetros:* La ventana de Otros Parámetros está destinada al registro y la visualización de parámetros ambientales y de funcionamiento como son: las temperaturas de los sistemas, voltaje, corriente, presión y la humedad presente en el ambiente. Todas estas variables se representan en gráficos en tiempo real e indicadores animados con la posibilidad de ser registrados de forma independiente.

7) *Pantalla World Wind*: Esta pantalla es un pestaña especial que integra el sistema de monitoreo con el servidor World Wind de la Nasa para aplicaciones Java. Esta extensión permite la visualización de un mapa especial, en el cual, es posible observar las características en tres dimensiones del terreno.

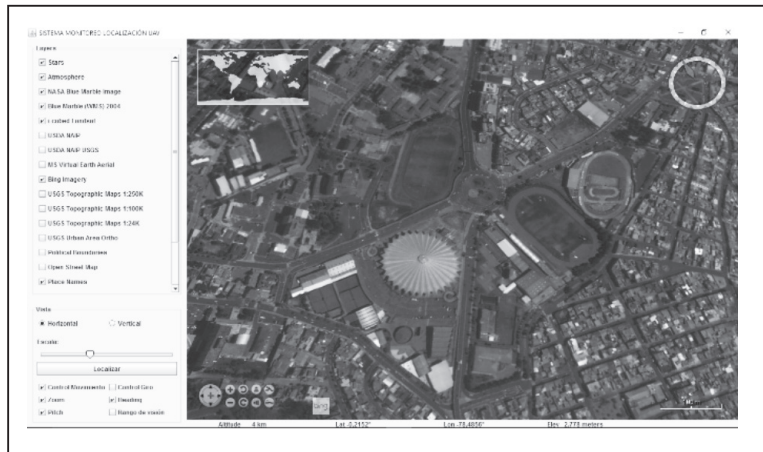


Figura 13. Pantalla World Wind.

8) *Pantalla de Configuración*: Dentro de las configuraciones disponibles en esta pantalla se tiene la posibilidad de personalizar el nombre de la red de comunicación del hexacóptero, el número de datos por archivo de registros, los valores críticos y la habilitación de alarmas, el enlace con el simulador de prueba para el sistema de monitoreo y bloc de sucesos para el chequeo de alarmas y demás eventos.

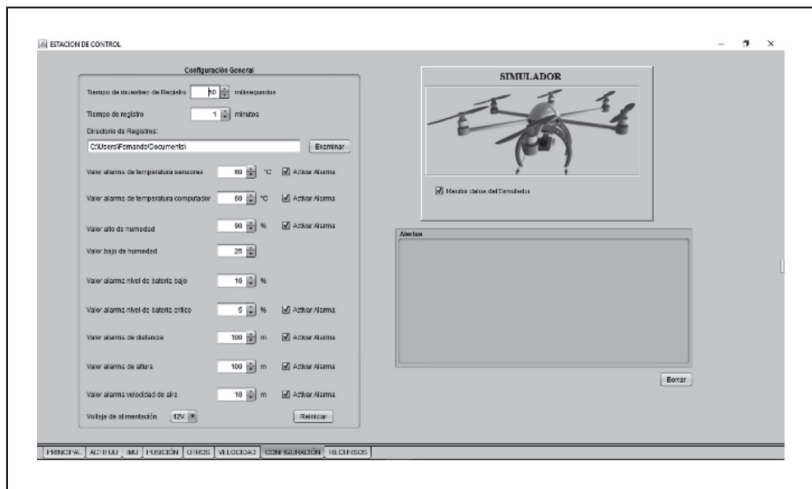


Figura 14. Pantalla de Configuración.

9) *Pantalla de Recursos*: La última pantalla es la de Recursos, en donde se puede encontrar información variada de todos los componentes que forman parte del sistema, así como también manuales para el usuario e información de los desarrolladores.

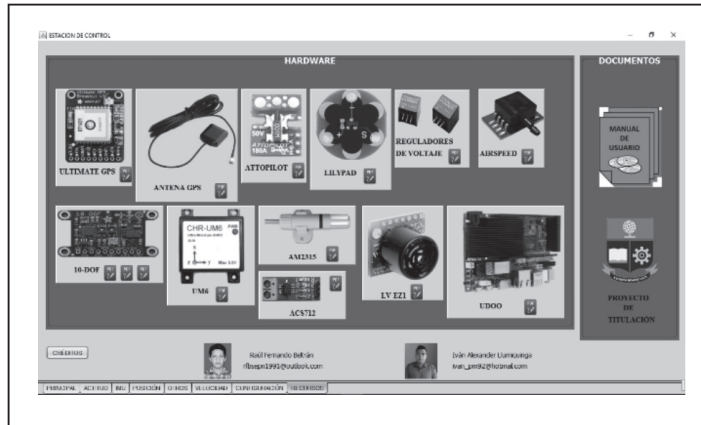


Figura 15. Pantalla de Recursos.

D. Simulador

Para el desarrollo del simulador se utilizó Unity3D [3], que es un motor gráfico 3D para la creación de juegos, animaciones y aplicaciones interactivas para múltiples plataformas como PC, MAC, Nintendo Android, iOS, XBOX, etc.

Este motor gráfico ofrece un completo editor visual para la creación de juegos. El juego se lo construye desde el editor y las acciones y controles se programa mediante la creación de scripts. Los scripts son implementados en tres lenguajes de programación disponibles que son: una versión de JavaScript [5], C# o Boo.

El modelo 3D del hexacóptero se lo dibujó y exportó desde el programa SketchUp. Para simular los movimientos se añadieron propiedades físicas a las hélices gracias al módulo de física de Unity, además se implementaron scripts de control para calcular las fuerzas y torques generadas por cada hélice al girar.

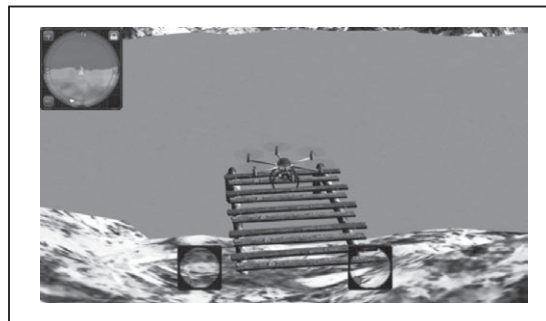


Figura 16. Simulador posición de partida.

El simulador cuenta con varias utilidades como una cámara abordo para recrear el vuelo por medio de vídeo a distancia, un indicador de la trayectoria y lo más importante la opción de simular datos que son enviados hacia el HMI del sistema de monitoreo todo esto con el fin de que el usuario se pueda familiarizar con todas las opciones de la estación de tierra y tenga una experiencia previa en el vuelo de un hexacóptero.



Figura 17. Cámara a bordo.

Conclusiones

Para optimizar la comunicación de los datos se configuró el módulo GPS de tal manera que sólo envíe las sentencias GGA y RMC. Con la primera se tienen los datos de posición y número de satélites conectados y con la segunda se obtiene la velocidad de desplazamiento. De esta manera se evita que se transmitan sentencias innecesarias que podrían hacer lenta la comunicación y se dispone de todos los datos requeridos para utilizarlos en el sistema de monitoreo.

La velocidad de transmisión utilizada entre el microcontrolador y el procesador fue de 115200 baudios mediante comunicación UART a 3.3V TTL, esto ayuda a que la comunicación siempre sea fluida y exitosa.

Para evitar la congestión en la comunicación fue importante seleccionar un periodo de muestreo adecuado para la lectura de cada sensor. A las variables que sufren cambios rápidos tales como ángulos de Euler se determinó un periodo de muestreo pequeño, no menor al tiempo de respuesta de los sensores, con el fin de leer varios datos en un segundo; y, a las variables lentas como las temperaturas y humedad se determinó un tiempo de muestreo más grande, en el orden de segundos, tiempo de respuesta promedio de los sensores ambientales empleados. De esta manera se optimiza las comunicación en la tarjeta de desarrollo y hacia la estación de monitoreo.

La velocidad de transmisión de información es un factor importante en el caso de un sistema de monitoreo, por lo que se optó por usar el protocolo UDP para, de esta manera lograr un intercambio fluido de datos.

Los datos adquiridos del sistema de monitoreo se reciben con fluidez a una distancia de 100 metros, sin embargo se observa que al pasar los 150 metros la comunicación se torna más lenta. Para extender el alcance de comunicación se necesita utilizar un router de mayor potencia.

El uso de sensores que incluyen como parte de su estructura microcontroladores para procesamiento de señales tal como en el caso de la Unidad de Medición Inercial que acopla un filtro extendido de Kalman, ayuda a optimizar la programación al no ser necesario un tratamiento de datos adicional que el que ya lo realizan por su propia cuenta.

Se concluyó que, utilizando el protocolo de comunicación UDP se obtuvo una mayor velocidad de transmisión de datos que si se lo hacia mediante TCP pues UDP no pierde tiempo en establecer conexiones, ni se asegura de la recepción de los datos, lo que lo hace ideal para aplicaciones de monitoreo por su velocidad y no así en aplicaciones de control en donde se necesita una alta confiabilidad.

Se obtuvo un error máximo en la medición de altura de 0.4 m debido a los 0.25 m de error nativo del sensor, a la interferencia que produce el viento sobre el sensor de presión barométrica.

El sistema de monitoreo desarrollado utiliza software y lenguaje de programación libres como lo son Netbeans y Java, y demuestra que se pueden obtener mejores resultados que al utilizar herramientas privadas bajo licencia.

Utilizando una antena externa para el modulo GPS se logra una conexión de 8 a 10 satélites en exteriores, obteniendo una localización exacta ya que el sistema necesita por lo menos una conexión con 4 satélites.

Referencias

- [1] Microsoft. «Protocolo de datagramas de usuario (UDP)». Enero 2005. [En línea]. Available: [https://msdn.microsoft.com/es-es/library/cc785220\(v=ws.10\).aspx](https://msdn.microsoft.com/es-es/library/cc785220(v=ws.10).aspx). [Último acceso: 11 de febrero de 2016].
- [2] Open GL. [En línea]. Available: <https://opengl.softonic.com/> [Último acceso: 11 de febrero 2016].
- [3] Unity3D. [En línea]. Available: <https://unity3d.com/es> [Último acceso: 11 de febrero].
- [4] UDOO. «Starting Manual». Versión 0.4. [En línea]. Available: http://udoo.org/download/files/Documents/UDOO_Starting_Manual_beta0.4_11_28_2013.pdf. [Último acceso: 11 de enero de 2016].
- [5] Antonio García Gonzales. «Multiprocesos en Java: Como usar hilos (Threads)». 10 febrero 2013. [En línea]. Available: <http://panamahitek.com/multiprocesos-en-java-como-usar-hilosthreads/>. [Último acceso: 11 de enero de 2016].
- [6] Harmonic Code. «Java swing component library SteelSeries». 13 agosto 2010. [En línea]. Available: <http://harmoniccode.blogspot.com/2010/08/java-swing-component-library.html#uds-search-results>. [Último acceso: 28 diciembre de 2015].

- [7] José Manuel Ruiz. «Manual de Programación Arduino». Agosto 2007 [En línea]. Available: <http://arduinoobot.pbworks.com/f/Manual+Programacion+Arduino.pdf>. [Último acceso: 10 de febrero de 2016].
- [8] Eduardo Huerta, Aldo Mangiaterra, Gustavo Noguera. GPS Posicionamiento Satelital. UNR Editora, julio 2005.
- [9] Aosong Electronics. «Digital temperature and humidity sensor». [En línea]. Available: <http://www.adafruit.com/datasheets/AM2315.pdf>. [Último acceso: 2 de febrero de 2016].
- [10] Antonio Creus. *Instrumentación industrial*. Editorial Alfaomega-Marcombo, 2010.
- [11] Lady Ada. «Bosch BMP085 Breakout Board». Adafruit Industries. 7 noviembre 2014. [En línea]. Available: <https://learn.adafruit.com/downloads/pdf/bmp085.pdf>. [Último acceso: 2 de febrero de 2016].
- [12] Glenn Baddeley. «NMEA sentence information». Adafruit Industries. 24 mayo 2011. [En línea]. Available: <http://home.pacific.net.au/~gnb/gps/nmea.html>. [Último acceso: 2 de febrero de 2016].