

Desarrollo de un sistema de Navegación Autónoma para UAV basado en FPGA

Amparo Meythaler N. MSc
Departamento de Eléctrica y Electrónica
Universidad de las Fuerzas Armadas ESPE-EL
Latacunga, Ecuador
ammeythaler@espe.edu.ec

David Rivas L. MSc
Departamento de Eléctrica y Electrónica
Universidad de las Fuerzas Armadas ESPE-EL
Latacunga, Ecuador
drrivas@espe.edu.ec

Fernando A. Chicaiza C.
Departamento de Eléctrica y Electrónica
Universidad de las Fuerzas Armadas ESPE- EL
Latacunga, Ecuador
fernandochicaiza137@gmail.com

Cristian P. Chuchico A.
Departamento de Eléctrica y Electrónica
Universidad de las Fuerzas Armadas ESPE- EL
Latacunga, Ecuador
cristian.pca@gmail.com

Abstract- A system of autonomous flight navigation for UAV based on FPGA technology is development. Control logic FPGA was analyzed and the frame format for the acquisition and signal processing for controlling a stabilizing flight card was established. Subsequently programming input communication interfaces, algorithms routes execution and control signals is carried out.

Keywords- FPGA, UAV, VHDL, PID

Resumen - Se desarrolla un sistema de navegación autónoma de vuelo para vehículos aéreos no tripulados basado en tecnología FPGA. Se analizó la lógica de control del FPGA y se estableció el formato de trama para la adquisición y procesamiento de señales para el control de una tarjeta estabilizadora de vuelo. Posteriormente se realizó la programación de interfaces de comunicación de entrada, algoritmos de ejecución de rutas y señales de control de salida.

I. INTRODUCCIÓN

La aplicación de Vehículos Aéreos No Tripulados ha tenido un crecimiento exponencial en los últimos años. Los UAV son la más reciente innovación para la recopilación de información visual sobrevolada de manera remota.

Los UAV son empleados no sólo en actividades militares, sino también en labores civiles como estudios topográficos, control del tráfico aéreo, inspección de líneas de alta tensión, labores de monitoreo y vigilancia en zonas afectadas por catástrofes, acceso a lugares peligrosos o contaminados, agricultura inteligente, entre otros. [1] [2]

Los UAV pueden operar a control remoto o con instrucciones pre-programadas (de manera autónoma). Para misiones de largo alcance, la comunicación desde tierra con el UAV requiere de una infraestructura costosa y compleja; adicionalmente, al aumentar la separación física entre el operador en tierra y el vehículo aéreo, disminuye la confiabilidad del sistema de teleguiado y el enlace de comunicación aumenta su vulnerabilidad a las interferencias. Por ésta y otras razones adicionales se han desarrollado sistemas para que el UAV pueda operar de manera autónoma, independientemente del control terrestre. Entre otras ventajas, la navegación autónoma permite incrementar el rango de vuelo, mantener la integridad del UAV debido a su capacidad de reacción autónoma, contar con planes de contingencia que

permitan al vehículo implementar soluciones automatizadas en caso de fallas y disponer de mayor confiabilidad y versatilidad en la planificación y cumplimiento de nuevas misiones.

En virtud de las ventajas mencionadas, se desarrolló un sistema de navegación autónoma basado en FPGA, el cual se caracteriza por un algoritmo flexible que permite incrementar bloques de control, así como sensores y actuadores, mediante el protocolo de comunicación RS-232.

La actitud y orientación del UAV son comandadas por el FPGA, cuyas señales de salida son interpretadas por una tarjeta estabilizadora de vuelo, esta interacción es posible gracias a la incorporación de una etapa de acoplamiento debido a la diferencia de voltajes lógicos de operación.

El diagrama general de conexión y componentes de este sistema se muestra en la Figura 1.

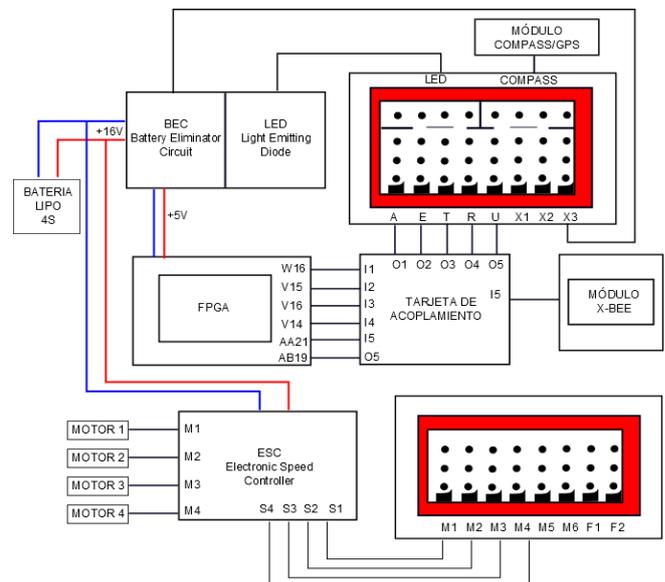


Figura 1. Diagrama de conexión y componentes

II. COMPONENTES FÍSICOS DEL CONTROLADOR

La implementación del sistema de vuelo autónomo se desarrolla sobre la plataforma modular de vuelo ARF-450 [3]. De esta manera se centró el interés en el diseño de las interfaces de comunicación y los algoritmos de control.

Los elementos primarios y actuadores se describen en los siguientes párrafos.

Debido a las ventajas expuestas en la introducción, el algoritmo de control del sistema de navegación autónoma será desarrollado en una plataforma FPGA, específicamente en la tarjeta Spartan XC3S700AN de XILINX [4]. Esta familia de tarjetas es la primera del mundo con característica de memoria no volátil, soportando dos o más archivos de configuración en un mismo dispositivo, además de admitir actualizaciones de campo, modos de prueba y configuraciones de múltiples sistemas.

El prototipo de vuelo sobre el cual se implementa el sistema de navegación autónoma se presenta en la figura 2 y sus características se muestran en la tabla 1.



Figura 2: Prototipo de Vuelo.

Tabla 1: características Físicas del prototipo de vuelo.

PARÁMETRO	DIMENSIÓN
Dimensión	72.5 cm x 72.5 cm.
Distancia entre motores	46 cm.
Peso Total	1,702 Kg.
Capacidad de Carga Útil	200 g.
Tiempo máximo de vuelo	23 min.
Altura máxima de vuelo	75 m.
Tiempo de carga de batería	60 min.
Alcance de mando remoto	150 m. (En línea de vista)
Variación de la altura de vuelo	±1m.

Los actuadores utilizados son motores DC sin escobillas (brushless), los cuales presentan diversas ventajas en relación a los motores con escobillas, tales como: eficiencia, par de fuerza superior, físicamente más livianos y no producen chispas cuando son accionados. En base a lo expuesto, un motor DC sin escobillas es más robusto, eficiente y confiable en este tipo de aplicaciones [5].

El hecho de no tener escobillas para generar la variación del campo magnético, hace indispensable la aplicación de un Controlador Electrónico de Velocidad (Electronic Speed Controller - ESC) para poder operar [6].

III. COMPONENTES LÓGICOS DEL CONTROLADOR

a) LÓGICA FPGA

La adopción de tarjetas FPGA en campos de la investigación ha sido impulsada por el hecho de que éstas combinan lo mejor de los Circuitos Integrados de Aplicación Específica (Application-Specific Integrated Circuit - ASIC) y de los sistemas basados en procesadores. Las FPGA ofrecen velocidades temporizadas por hardware sin requerir altos volúmenes de recursos. El silicio reprogramable tiene la misma capacidad de ajustarse que un software que se ejecuta en un sistema basado en procesadores, pero no está limitado por el número de núcleos disponibles [7]. A diferencia de los procesadores, las tarjetas FPGA llevan a cabo diferentes operaciones de manera paralela, por lo que éstas no necesitan competir por los mismos recursos. Cada tarea de procesos independientes se asigna a una sección dedicada del chip, y puede ejecutarse de manera autónoma sin ser afectada por otros bloques de lógica. Como resultado, el rendimiento de una parte de la aplicación no se ve afectado cuando se agregan otros procesos.

La FPGA puede ser programada mediante distintos tipos de lenguajes, siendo el más utilizado el Lenguaje de Descripción de Hardware para circuitos integrados de muy alta velocidad (Very high speed integrated circuit Hardware Description Language - VHDL). El VHDL es un lenguaje de descripción de hardware, el cual determina el comportamiento de un circuito electrónico [8]. Un proyecto VHDL puede contener varios archivos, haciéndolo apropiado para aplicaciones donde la escalabilidad sea una consideración de diseño.

b) DISEÑO DEL CONTROLADOR LÓGICO

La lógica de programación del componente principal identifica inicialmente a qué tipo de control está sujeto el comportamiento del cuadricóptero. Al recibir una trama que contenga el encabezado del carácter "H", seguido de un carácter "1", el control está comandado por la ruta de navegación cargada. Por el contrario, al recibir un carácter de "0" (valor por defecto) el cuadricóptero responderá a los comandos enviados por el radio control. Ver figura 3.

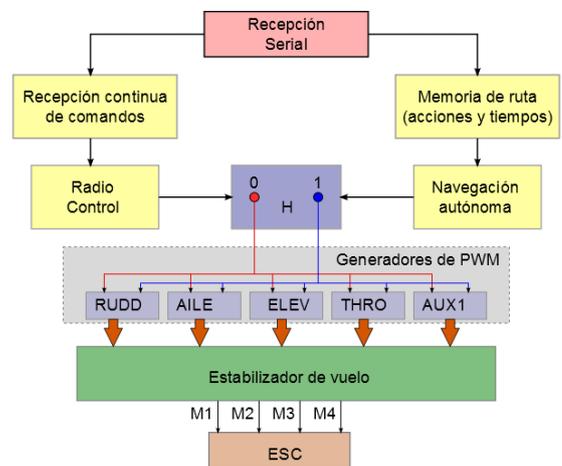


Figura 3: Comportamiento del FPGA al recibir el carácter "H".

El conjunto componentes implementadas en VHDL, es utilizado para ejecutar los diferentes procesos implicados en el sistema de navegación autónoma en el FPGA.

La programación de la librería de recepción serial en VHDL se estructura en dos partes. La primera parte genera relojes de captura de datos, así como también la recepción bit a bit de la línea de comunicación. La segunda parte convierte un vector de bits receptados en valores decimales operables, los cuales son almacenados en variables globales. La figura 4 presenta la interacción entre los dos subcomponentes para adquirir un valor de la trama enviada mediante la comunicación inalámbrica. Para la lectura de un dato serial se genera las señales que se muestran en la figura 5 dentro del FPGA.

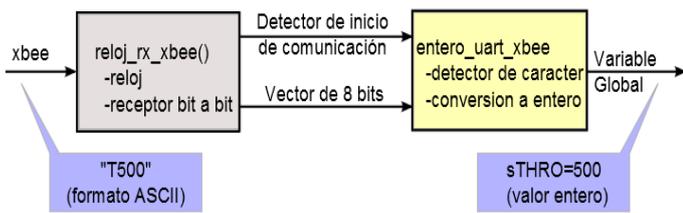


Figura 4: Interacción entre subcomponentes de recepción serial.

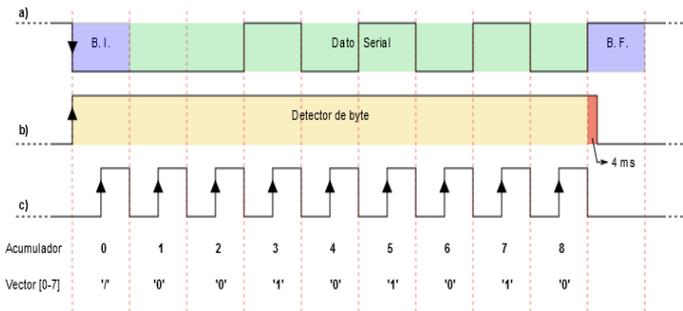


Figura 5: Señales generadas en el FPGA para la captura de datos seriales.

El componente generador de PWM presenta un porcentaje de variación el cual depende del valor entero de entrada en un rango de 0 a 999 (1 a 2 ms respectivamente), según lo indicado en (1).

$$\text{Tiempo en 1 lógico} = \text{lim_bajo} + vTHRO * 50 \quad [1]$$

Dónde:

lim_bajo es el valor calculado que genera un 1 ms como ancho de pulso mínimo. El valor se calcula tomando en cuenta que el oscilador de entrada es de 50 MHz; por tanto, el valor de espera se toma del cálculo dado por (2).

$$\text{lim_bajo} = 1\text{ms} * 50\text{Mhz} \quad [2]$$

$$\text{lim_bajo} = \frac{1\text{ms}}{2 * 10^{-8}\text{s}}$$

$$\text{lim_bajo} = 50000$$

El almacenamiento de rutas en el FPGA se ha limitado a 14 maniobras diferentes, las cuales definen el comportamiento de vuelo del vehículo con un tiempo limitado de 15 minutos aproximadamente (tiempo calculado de la duración de carga de la batería LIPO 4S basado en la experimentación práctica). Las acciones de maniobra en la navegación autónoma son: despegue, estabilización, giro a la derecha, giro a la izquierda, avance frontal y aterrizaje.

La ruta puede ser enviada a través de cualquier interfaz de comunicación serial para esto es suficiente analizar la trama presentada en la imagen 6 y su respectiva descripción.

S0N1N0P1F050D010F100I010F060P0

Figura 6: Trama de ingreso de una ruta.

El conjunto de caracteres “S0” indica al FPGA la programación de una nueva ruta, terminando los procesos de navegación autónoma de rutas anteriores. “N1” formatea las acciones y tiempos generados para la ejecución de la ruta anterior y “N0” habilita el uso de localidades de memoria para el ingreso de la nueva ruta. El inicio de la nueva ruta es detectado mediante el conjunto de caracteres “P1”, y su finalización se determina con “P0”. Dentro de este conjunto de caracteres previamente expuestos, se incluyen las acciones de maniobra y sus tiempos de ejecución.

Una vez recibida la trama correspondiente se procede a la ejecución del algoritmo el cual se describe de forma general en la figura 7.

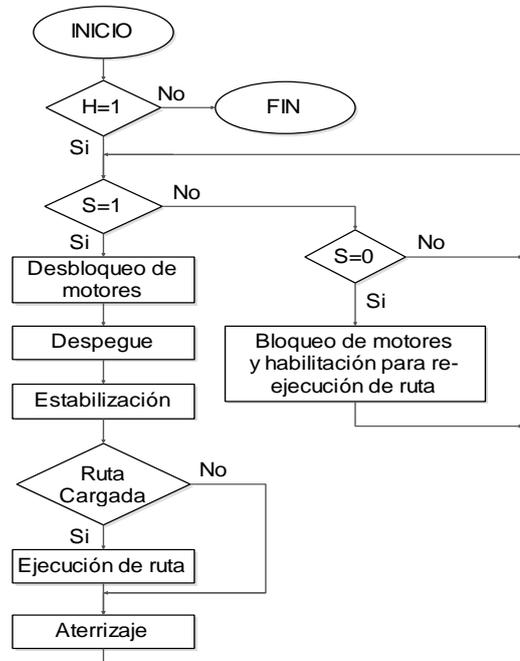


Figura 7: Esquema general de programación de la navegación autónoma.

c) INTERACCIÓN CON EL SISTEMA DE NAVEGACIÓN AUTÓNOMA

GUI de Matlab es un entorno que permite desarrollar interfaces gráficas que hacen amigable la interacción entre el usuario y el programa en ejecución, la figura 8 muestra la ventana desarrollada para el ingreso de una ruta, mientras que la figura 9 muestra la interface de control de mando remoto.

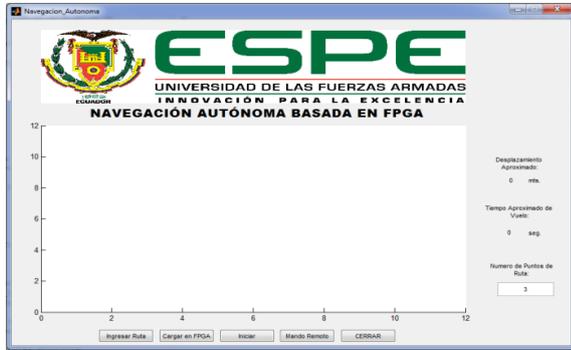


Figura 8: Interfaz de Navegación Autónoma



Figura 9: Interfaz de Mando Remoto

d) PRUEBAS EN LABORATORIO

Para la verificación final antes de las pruebas prácticas se procede a cargar una ruta y a observar las señales de control de PWM generadas por el FPGA. En la figura 10 se observa una trama transmitida mientras que en la figura 11 se muestra la variación del ancho de pulso de una de las señales de salida del FPGA.



Figura 10: Trama Correspondiente a una Ruta.

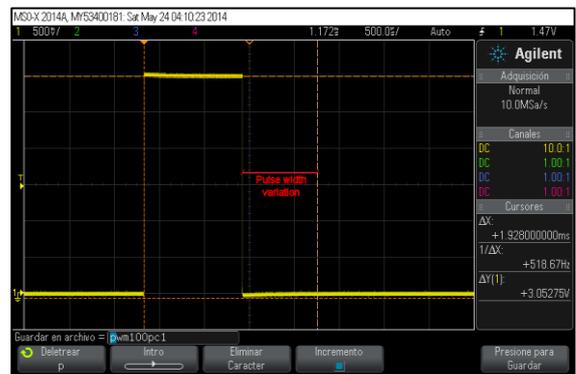


Figura 11. Variación de PWM para el control de velocidad de los motores.

e) PRUEBAS PRÁCTICAS

Una vez que se ha verificado el correcto funcionamiento de todos los componentes se puede ejecutar el sistema de navegación autónoma, para este caso la prueba de vuelo consiste en realizar una figura que describe un cuadrado de 8m de lado, se ha trazado una ruta equivalente en tierra, ver figura 12.

El análisis contempla un sobrevuelo con baja y alta perturbación el contraste ante estas dos situaciones se la puede observar en la figura 13. En la figura 14 se observa el comportamiento del UAV ante la orden de permanecer fijo en un punto determinado.



Figura 12: Ruta de navegación capturada en sobrevuelo.

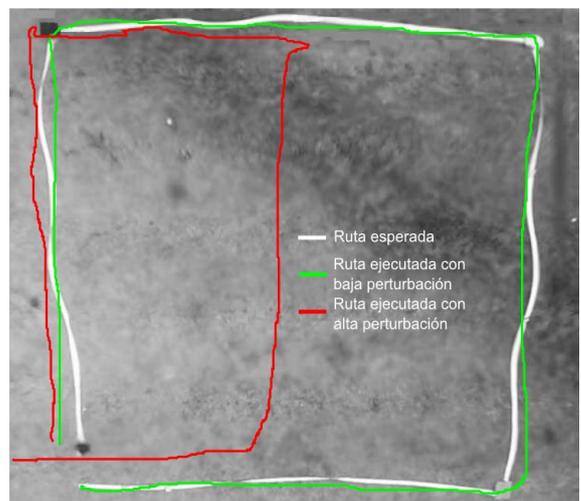


Figura 13: Contraste de ejecuciones con baja y alta perturbación.

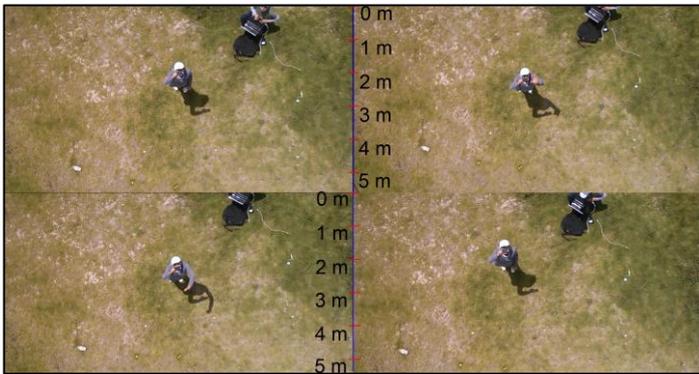


Figura 14: Capturas de la estabilidad de vuelo del cuadricóptero.

IV. CONCLUSIONES

- Se ha implementado un controlador de navegación autónoma para un vehículo aéreo no tripulado basado en FPGA, mediante el desarrollo de componentes de programación en dicha tecnología.
- El FPGA es una alternativa flexible y escalable para lograr la navegación autónoma del cuadricóptero. Aproximadamente el 50% de memoria PROM de la FPGA ha sido utilizada en la programación del proyecto, lo que permite la adición de nuevos componentes que optimicen la navegación o incluyan nuevos algoritmos de control más eficientes.
- La documentación de los componentes programados en lenguaje VHDL permiten reforzar los conocimientos adquiridos por los investigadores de este proyecto. Además, se recopilan características y virtudes exclusivas del FPGA.
- La interfaz desarrollada en el GUI de MATLAB es predecible y fácil de utilizar para cualquier usuario.
- La tarjeta de estabilización tiene un comportamiento satisfactorio, además de que posee características de seguridad embebidas.
- Debido a la diferencia en el manejo de tensiones entre la FPGA y el resto de componentes, se incluye una etapa de acoplamiento. En el apartado 3.1.2 se muestran todos los elementos electrónicos y su interconexión.
- El prototipo de cuadricóptero tiene la capacidad de levantar 200 gramos de carga útil, por lo cual, una cámara de video o un sistema de captura de imágenes puede ser incluido en el vehículo. Esto permite enfocarlo a fines de beneficio civil.

V. REFERENCIAS

[1] Instituto nacional de estadística y geografía, «El uso de Vehículos Aéreos No Tripulados,» Abril 2014. [En línea]. Available: <http://www.inegi.org.mx/inegi/contenidos/espanol/pr>

ensa/Boletines/Boletin/Comunicados/Especiales/2014/abril/comunica5.pdf. [Último acceso: 8 Mayo 2014].

- [2] L. Ariza, «Drones para todo,» Febrero 2014. [En línea]. Available: http://elpais.com/elpais/2014/02/11/eps/1392119032_902847.html. [Último acceso: 13 Febrero 2014].
- [3] DJI [en línea]. [Fecha de consulta: 1 Febrero 2014]. Flame Wheel ARF. Disponible en: <http://www.dji.com/product/flame-wheel-arf>.
- [4] XILINX, Spartan-3AN FPGA Family Data Sheet, Abril 2011, Disponible en: http://www.xilinx.com/support/documentation/data_sheets/ds557.pdf
- [5] BARRERA, Daniel, “Diseño e implementación de un cuadricóptero con sistema de control automático de estabilidad y comunicación inalámbrica de datos utilizando plataformas de hardware y software libre”, Universidad de Costa Rica, Julio 2012.
- [6] NEOTEO [en línea]. [Fecha de consulta: 25 Junio 2014]. Motores Brushless o BLDC (Métodos de accionamiento). Disponible en: <http://www.neoteo.com/motores-brushless-blcd/>.
- [7] National Instruments [en línea]. [Fecha de consulta: 30 Junio 2014]. Introducción a la tecnología FPGA: Los Cinco Beneficios Principales. Disponible en: <http://www.ni.com/white-paper/6984/es/>
- [8] Control Engineering [en línea]. [Fecha de consulta: 14 Febrero 2014]. Advantages of FPGAs. Disponible en: <http://www.controleng.com/single-article/advantages-of-fpgas/c5a8c5a1dffd102e83ae9d9d7c518746.html>