

**DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA DE GUÍA INERCIAL PARA EL  
CONTROL DE VUELO AUTÓNOMO EN UN UAV**

**ANDRÉS FELIPE LASSO ALDANA  
ALEXANDER MURCIA CASTRO**

**UNIVERSIDAD SURCOLOMBIANA  
FACULTAD DE INGENIERÍA  
PROGRAMA DE INGENIERÍA ELECTRÓNICA  
NEIVA-HUILA  
2012**

**DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA DE GUÍA INERCIAL PARA EL  
CONTROL DE VUELO AUTÓNOMO EN UN UAV**

**ANDRÉS FELIPE LASSO ALDANA  
ALEXANDER MURCIA CASTRO**

**Trabajo de grado presentado como requisito para optar al título de  
INGENIERO ELECTRÓNICO**

**Director  
Ing. MSc. JOSÉ DE JESÚS SALGADO PATRÓN**

**UNIVERSIDAD SURCOLOMBIANA  
FACULTAD DE INGENIERIA  
PROGRAMA DE INGENIERÍA ELECTRÓNICA  
NEIVA – HUILA  
2012**

Nota de aceptación:

---

---

---

---

---

---

---

Firma del presidente del jurado

---

Firma del jurado

---

Firma del jurado

Neiva, 7 de Diciembre de 2012

## **DEDICATORIA**

A mi familia  
A mis padres por ser el pilar fundamental en todo lo que soy, en toda mi educación,  
tanto académica, como de la vida, por su incondicional apoyo durante todo el  
transcurso de mi vida académica.

A mi mamá  
Por haberme apoyado en todo momento, por sus consejos, sus valores, por la  
motivación constante que me ha permitido ser una persona de bien, pero más que  
nada, por su amor.

**ANDRÉS FELIPE LASSO ALDANA**

A Dios,  
porque gracias a Él he logrado concluir esta etapa de mi vida.

A mis padres,  
por su apoyo y amor incondicional.

**ALEXANDER MURCIA CASTRO**

## **AGRADECIMIENTOS**

A todas las personas que participaron e hicieron posible este proyecto, muchas gracias por su apoyo y enseñanza.

Al ingeniero JOSE DE JESÚS SALGADO PATRÓN, quien nos guió como director en la elaboración de este proyecto.

Al ingeniero JAVIER HUMBERTO RUBIO PLAZAS, por todo el apoyo, colaboración y enseñanzas que nos ofreció desde el inicio hasta el final del proyecto, a él muchas gracias.

Al TecnoParque Colombia nodo Neiva, por apoyarnos y por todos los servicios prestados.

# CONTENIDO

	pág.
INTRODUCCIÓN.....	15
1. MARCO TEÓRICO.....	16
1.1 UNIDAD DE MEDIDA INERCIAL.....	16
1.2 SISTEMA DE NAVEGACIÓN INERCIAL.....	16
1.3 ÁNGULOS DE EULER.....	17
1.4 SISTEMA DE POSICIONAMIENTO GLOBAL.....	17
1.5 INSTRUMENTOS DE CONTROL DE UN UAV.....	17
1.6 MATRÍZ DE COSENO DIRECTORES.....	19
2. HARDWARE.....	21
2.1 UNIDAD DE MEDIDA INERCIAL.....	22
2.2 TARJETA DE ADQUISICIÓN DE DATOS.....	24
2.3 MÓDULO RECEPTOR GPS.....	26
2.4 MÓDULOS DE TELEMETRÍA.....	27
2.5 DISEÑO DE LA TARJETA.....	27
3. SOFTWARE.....	31
3.1 PROGRAMACIÓN ARDUINO.....	31
3.1.1 Protocolo de comunicación.....	31
3.1.2 Cálculo de la actitud.....	33
3.1.2.1 Cálculo de la actitud usando acelerómetro.....	33
3.1.2.2 Cálculo de la actitud usando giroscopio.....	34
3.1.2.3 Estimación de la orientación con el magnetómetro.....	35
3.1.2.4 Cálculo de la actitud usando el Algoritmo DCM.....	35
3.2 PROGRAMACIÓN DE LA INTERFAZ DE USUARIO.....	37
3.2.1 Eliminación de la componente de la gravedad del acelerómetro.....	40
3.2.2 Ecuaciones cinemáticas.....	41
4. RESULTADOS.....	43
4.1 RESULTADOS EN AMBIENTE CONTROLADO.....	43
4.1.1 Comprobación de los ángulos de Euler.....	43
4.1.2 Resultados del GPS.....	48
4.1.3 Resultados de posición con la unidad de medida inercial.....	48
4.2 RESULTADOS EN AMBIENTE REAL.....	51
4.2.1 Actitud.....	52
4.2.2 Distancia.....	52
4.2.3 GPS.....	53
5. CONCLUSIONES.....	54
6. RECOMENDACIONES.....	55

	pág.
BIBLIOGRAFÍA .....	56
ANEXOS .....	57

## LISTA DE FIGURAS

	pág.
Figura 1. Ejemplos de unidad de medida inercial .....	16
Figura 2. Representación de los giros en <i>roll</i> , <i>pitch</i> y <i>yaw</i> .....	17
Figura 3. Instrumentos de control de un UAV .....	19
Figura 4. Sistema ArduPilot Mega .....	21
Figura 5. Tarjeta MinIMU-9.....	22
Figura 6. Diagrama esquemático de la unidad de medida inercial minIMU-9.....	23
Figura 7. Partes importantes de la tarjeta Arduino Uno .....	24
Figura 8. Diagrama esquemático de la tarjeta Arduino Uno .....	25
Figura 9. GPS MediaTek MT3329 .....	26
Figura 10. Módulos XBee-PRO 900 utilizados en el proyecto .....	27
Figura 11. Diagrama esquemático de la tarjeta diseñada y su conexión con la tarjeta Arduino Uno.....	28
Figura 12. Cara inferior de la tarjeta diseñada.....	28
Figura 13. Cara superior o de componentes de la tarjeta diseñada .....	29
Figura 14. Cara inferior de la tarjeta terminada .....	29
Figura 15. Cara superior de la tarjeta terminada .....	29
Figura 16. Tarjeta terminada conectada a la tarjeta Arduino Uno .....	30
Figura 17. Representación del sistema completo.....	30
Figura 18. Diagrama de flujo del microcontrolador Atmega328.....	32
Figura 19. Identificador de trama de los ángulos de Euler .....	33
Figura 20. Ángulo pitch en sentido horario .....	33
Figura 21. Diagrama de bloques del proceso .....	37
Figura 22. Diagrama de flujo del algoritmo DCM en el microcontrolador .....	37
Figura 23. Diagrama de flujo del programa en Visual C# .....	38
Figura 24. Lectura de los datos en el puerto serial.....	39
Figura 25. Interfaz de usuario del sistema de guía inercial.....	40
Figura 26. División de la señal de aceleración en subáreas .....	42
Figura 27. Maqueta propuesta para medir los ángulos <i>roll</i> y <i>pitch</i> .....	43
Figura 28. Maqueta para medir el ángulo <i>yaw</i> .....	43
Figura 29. Medida de los ángulos positivos ( <i>roll</i> ).....	44
Figura 30. Medida de los ángulos negativos ( <i>roll</i> ) .....	45
Figura 31. Medida de los ángulos positivos ( <i>pitch</i> ).....	46
Figura 32. Medida de los ángulos negativos ( <i>pitch</i> ) .....	46
Figura 33. Medida de los ángulos ( <i>yaw</i> ).....	47
Figura 34. Posición marcada por el GPS MT3329 .....	48
Figura 35. Gráfica de aceleración .....	50
Figura 36. Gráfica de velocidad .....	50
Figura 37. Gráfica de posición .....	51
Figura 38. UAV utilizado para probar el sistema de guía inercial .....	51
Figura 39. Sistema de guía inercial acondicionado en el UAV .....	52
Figura 40. Recorrido del UAV entregado por el GPS .....	53
Figura 41. Componentes ZigBee .....	57



Figura 42. Ventana de configuración del programa X-CTU .....	57
Figura 43. Ventana que indica conexión correcta del módulo XBee .....	58
Figura 44. Ventana que indica la lectura de la programación del XBee .....	58
Figura 45. Ventana de configuración del programa X-CTU .....	59
Figura 46. Mapeo entre los pines del Arduino y los puertos del ATmega328.....	60
Figura 47. Diagrama de bloques del microcontrolador Atmega328.....	60

## LISTA DE CUADROS

pág.

Cuadro 1. Conexión entre la tarjeta MinIMU-9 y la tarjeta Arduino Uno .....	23
Cuadro 2. Conexión entre el GPS MT3329 y la tarjeta Arduino Uno .....	26
Cuadro 3. Porcentaje de error para el ángulo <i>roll</i> .....	44
Cuadro 4. Porcentaje de error para el ángulo <i>pitch</i> .....	45
Cuadro 5. Porcentaje de error para el ángulo <i>yaw</i> .....	47
Cuadro 6. Porcentaje de error para la distancia recorrida en el eje x.....	49
Cuadro 7. Porcentaje de error para la distancia recorrida en el eje y.....	49
Cuadro 8. Porcentaje de error para la distancia recorrida en el eje z.....	49

## LISTA DE ANEXOS

	pág.
ANEXO A. Configuración de los módulos ZigBee .....	53
ANEXO B. Microcontrolador Atmega328.....	56
ANEXO C. Ecuaciones para el cálculo de la actitud.....	61

## GLOSARIO

**ACELERÓMETRO:** sensor electromecánico que mide la aceleración lineal, y por medio del cual se puede calcular la actitud y posición de un móvil.

**ACTITUD:** es la orientación o referencia angular del avión con respecto al horizonte. Su representación clásica son los ángulos de Euler (*roll*, *pitch* y *yaw*).

**GIROSCOPIO:** sensor electromecánico que mide la velocidad angular de un cuerpo y que tiene mucha utilidad en la navegación inercial para el cálculo de la actitud.

**INERCIA:** es la tendencia de los cuerpos a mantener su velocidad de translación y de rotación constantes, a no ser que se vean afectados por fuerzas o por momentos, respectivamente.

**NAVEGACIÓN:** es el conjunto de técnicas empleadas para realizar un desplazamiento desde un punto a otro, ambos conocidos, siguiendo una trayectoria que también se conoce.

**NAVEGACIÓN INERCIAL:** es un tipo de navegación que utiliza sensores como giroscopios y acelerómetros para determinar la aceleración, velocidad, posición y actitud del vehículo en el que se encuentren dichos sensores.

**SISTEMA DE NAVEGACIÓN INERCIAL (INS):** es el sistema conformado por una unidad de medida inercial, un microcontrolador, un GPS y módulos de telemetría, para realizar la determinación de la actitud y la posición del vehículo en el que se encuentre ubicado.

**UAV:** son las siglas de *Unmanned Aerial Vehicle*, que traducido al español significa vehículo aéreo no tripulado.

**UNIDAD DE MEDIDA INERCIAL (IMU):** es un dispositivo electrónico conformado por un acelerómetro, un giroscopio y un magnetómetro, que realizan sus respectivas medidas en los ejes x, y, z.

## RESUMEN

El proyecto consiste en diseñar e implementar un sistema de guía inercial para ser utilizado en el cálculo de la posición y de la actitud de un UAV, para ello se hace uso de una unidad de medida inercial (IMU), una tarjeta de adquisición de datos, un receptor GPS y módulos de telemetría. La unidad de medida inercial está conformada por un acelerómetro, un giroscopio y un magnetómetro, de tres ejes cada uno, necesarios para el cálculo de la posición y actitud del UAV.

Para la realización de este proyecto se ha tenido en cuenta el uso de herramientas matemáticas como la matriz de cosenos de dirección para el cálculo de los ángulos de Euler (*roll*, *pitch* y *yaw*) e integración numérica utilizando la regla del trapecio, en combinación con ecuaciones cinemáticas, para el cálculo de la velocidad y de la posición del UAV, a partir de los datos que entrega el acelerómetro de la unidad de medida inercial.

Además, se cuenta con el sistema de posicionamiento global (GPS) para graficar el recorrido y la posición del UAV, complementando a la instrumentación con datos satelitales en tiempo real para mejorar la ubicación del UAV y poder observar su posición en un mapa.

## **ABSTRACT**

The project consists in designing and implement an inertial guidance system to be used in calculating the position and attitude of a UAV, so it makes use of an inertial measurement unit (IMU), a data acquisition card, a GPS and telemetry modules. The inertial measurement unit is formed by an accelerometer, magnetometer and a gyroscope, each one of three axes necessary for calculating the position and attitude of the UAV.

For the realization of this project it has taken into account the use of mathematical tools such as direction cosine matrix to calculate the Euler angles (roll, pitch and yaw) and numerical integration using the trapezoidal rule, combined with kinematic equations to calculate the speed and position of the UAV from the data delivered by the accelerometer inertial measurement unit.

In addition, it has the global positioning system (GPS) to plot the path and position of the UAV, complementing the instrumentation with real-time satellite data to improve the location of the UAV and observe its position on a map.

## INTRODUCCIÓN

Una unidad de medida inercial (IMU) es un dispositivo electrónico conformado por un acelerómetro, un giroscopio y un magnetómetro, que forma parte de un sistema de navegación inercial, el cual es el encargado de determinar la posición y la actitud del vehículo donde éste se encuentre instalado, normalmente la IMU se encuentra en aviones, aunque el primer vehículo en el que se utilizó fue en un barco; actualmente casi todos los barcos y satélites tienen una IMU, pero para ser más exactos, casi cualquier vehículo que deba utilizar la electrónica para conocer su aceleración, velocidad, posición y actitud, tiene una IMU.

En el presente proyecto de grado se realiza el diseño e implementación de un sistema de guía inercial para ser utilizado en un vehículo aéreo no tripulado (UAV), como la primera etapa que se debe realizar para implementar el sistema de control de despegue del UAV, ya que es necesario conocer en primera instancia la posición en la que éste se encuentre sin importar el ambiente externo que lo rodee. Luego de que se tiene conocimiento de la ubicación del UAV por medio del sistema de guía inercial, se puede tener en cuenta la perturbación producida por el viento y demás factores que conlleven a la implementación del sistema de control de despegue para el UAV en futuros proyectos de grado.

El presente libro se divide en cuatro capítulos, el primero de los cuales presenta el marco teórico necesario para la comprensión del proyecto.

El capítulo dos aborda el tema del hardware utilizado en el desarrollo del proyecto, en el cual se habla del sistema ArduPilot Mega, utilizado en la parte inicial, así como de la información de la unidad de medida inercial, la tarjeta de adquisición de datos, el GPS y los módulos de telemetría.

El capítulo tres presenta el software empleado, describiendo el código que se utiliza para obtener, procesar y visualizar la información que proviene de la unidad de medida inercial y del GPS.

El capítulo cuatro da a conocer los resultados obtenidos en las pruebas realizadas, tanto en un ambiente controlado como en un ambiente real utilizando el vehículo aéreo no tripulado.

## MARCO TEÓRICO

En este capítulo se presentan los conceptos más relevantes y necesarios, para la comprensión del presente proyecto de grado.

### 1.1 UNIDAD DE MEDIDA INERCIAL

Una unidad de medida inercial (IMU) es un dispositivo electrónico que está conformado por un acelerómetro, un giroscopio y un magnetómetro, de tres ejes cada uno; la unidad de medida inercial se utiliza en barcos, aviones, UAVs, satélites y en cualquier otro sistema al cual se le desea conocer su aceleración, velocidad, posición y orientación.

La unidad de medida inercial es solo un componente de un sistema de navegación inercial, sistemas como el GPS, compensan las limitaciones de la IMU, se debe tener en cuenta que la mayoría de los otros sistemas de navegación, tienen sus propios defectos los cuales son compensados entre ellos<sup>1</sup>. En la Figura 1 se observan algunos ejemplos de unidad de medida inercial.

Figura 1. Ejemplos de unidad de medida inercial



Fuente: <http://www.pololu.com/picture/view/0J3679>, <http://www.bricogeeek.com/shop/61-medicion-inercial-imu>

### 1.2 SISTEMA DE NAVEGACIÓN INERCIAL

Un Sistema de Navegación Inercial (INS) está conformado por una unidad de medida inercial y un microcontrolador, que se encuentran ubicados en una plataforma común, además, puede utilizar módulos de telemetría y apoyarse en un GPS u otro sistema complementario, para realizar la determinación de la actitud y posición del vehículo en el que se encuentre ubicado.

Hay diferentes tipos de INS y se clasifican en sistema *gimbaled* y sistema *strapdown*. El sistema *gimbaled* pretende aislar la plataforma con los sensores inerciales de los movimientos de rotación externos, mientras que en el sistema *strapdown*, los ejes de los sensores están alineados con los ejes del móvil. Este último sistema es el que se va a utilizar en este proyecto.

---

<sup>1</sup> RIVAS GIL, Miguel. Unidad de medida inercial: algoritmo de estimación e implementación software. Trabajo de grado. Universidad de Sevilla. 2006. p. 16.

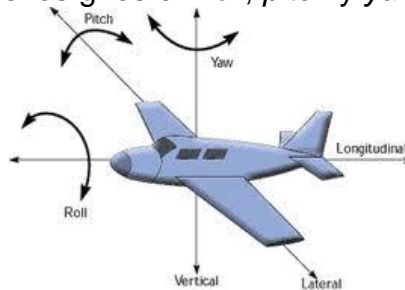


### 1.3 ÁNGULOS DE EULER

Los ángulos de Euler se corresponden con los ángulos de *roll*, *pitch* y *yaw*, que se utilizan para especificar la orientación o actitud de una aeronave, estos ángulos se pueden determinar a través de una herramienta matemática denominada la matriz de cosenos de dirección (DCM) y se miden de la siguiente manera:

- Ángulo *roll*: es positivo si el ala derecha del UAV está por debajo del plano horizontal, en caso contrario el ángulo *roll* es negativo.
- Ángulo *pitch*: es positivo si la nariz del UAV se inclina por encima del plano horizontal, en caso contrario el ángulo *Pitch* es negativo.
- Ángulo *yaw*: es positivo si la nariz del UAV se mueve en el sentido de las manecillas del reloj, respecto al norte. La Figura 2 muestra una representación de los giros en *roll*, *pitch* y *yaw*.

Figura 2. Representación de los giros en *roll*, *pitch* y *yaw*



Fuente: <http://quest.arc.nasa.gov/aero/virtual/demo/aeronautics/tutorial/motion.html>

### 1.4 SISTEMA DE POSICIONAMIENTO GLOBAL

El sistema de posicionamiento global (GPS) permite encontrar la posición de un determinado punto en la superficie terrestre en coordenadas de latitud, longitud y altitud, este sistema cuenta actualmente con un grupo de 28 satélites. Para realizar la determinación de la posición, se necesita contar con un receptor GPS y que por lo menos 4 satélites realicen el posicionamiento del módulo receptor GPS, para de esta forma obtener la posición esperada.

### 1.5 INSTRUMENTOS DE CONTROL DE UN UAV

La misión principal de los instrumentos a bordo es ejercer una ayuda instrumental al controlador de la aeronave en caso de no contar con referencia visual exterior, manteniéndose informado sobre el resultado de la actuación de aquellos mandos, y pudiendo así desarrollar con ellos un vuelo por instrumentos.

En el desarrollo de este proyecto se ha utilizado el código de software libre AvionicsInstrumentControlDemo, desarrollado en Visual C#, con el fin de visualizar los siguientes instrumentos y obtener la simulación de la cabina de un avión.

A continuación se encuentran las definiciones de estos instrumentos.

- **Horizonte artificial<sup>2</sup>**

Es el instrumento que le indica al piloto si vuela nivelado o si está inclinado longitudinal (nariz del UAV arriba o abajo) o transversalmente (con las alas horizontales). Consta de un giroscopio que gira alrededor de un eje vertical, sujeto a una cuna esférica que está pintada en azul y tierra para simular el horizonte. El avión se representa por una barra que se mantiene sujeta a la caja exterior del instrumento.

- **Anemómetro o visualizador de velocidad indicada**

Instrumento que refleja la velocidad indicada, es decir, la velocidad de vuelo del avión teniendo en cuenta que dicha velocidad es medida respecto al aire, no a la tierra. El anemómetro calcula la diferencia de presión que existe entre la presión dinámica obtenida a través del tubo de pitot y la procedente de la fuente estática.

- **Altímetro**

Es un instrumento que indica la altitud del vuelo. En este proyecto se tomará la medida que entregue el GPS como referencia para la altitud.

- **Indicador de viraje/coordinación**

Este aparato consta realmente de dos instrumentos independientes ensamblados en la misma caja: el indicador de viraje y el indicador de coordinación de viraje. Este fue uno de los primeros instrumentos usados por los pilotos para controlar un aeroplano sin referencias visuales al suelo o al horizonte.

- **Indicador de rumbo**

La brújula permite conocer el rumbo magnético de la aeronave. Desde los polos magnéticos de la Tierra, surgen líneas magnéticas o líneas de flujo, y los imanes o el magnetómetro incluido en la unidad de medida inercial, se orienta según las mismas.

En la Figura 3 se observan los instrumentos de control de un UAV que hacen parte de la interfaz de usuario diseñada para el sistema de guía inercial.

---

<sup>2</sup> SIMÓ ALONSO, María. Diseño de una cabina de vuelo virtual. Trabajo de grado. Universidad politécnica de Cataluña. 2006. p. 6.

Figura 3. Instrumentos de control de un UAV



Fuente: <http://www.codeproject.com/Articles/27411/C-Avionic-Instrument-Controls>

## 1.6 MATRIZ DE COSENOS DIRECTORES (DCM)<sup>4</sup>

Ciertos tipos de vectores, como direcciones, velocidades, aceleraciones y traslaciones, (movimientos) se pueden transformar entre marcos de referencia girados con una matriz de 3x3. Para este proyecto se necesita saber el cambio de marco de referencia del avión al marco de referencia de tierra. Es posible girar los vectores multiplicándolos por una matriz de cosenos de dirección:

$$Q = \begin{bmatrix} Q_X \\ Q_Y \\ Q_Z \end{bmatrix} = \text{Es un vector como aceleración, velocidad, dirección}$$

$$R = \begin{bmatrix} r_{xx} & r_{xy} & r_{xz} \\ r_{yx} & r_{yy} & r_{yz} \\ r_{zx} & r_{zy} & r_{zz} \end{bmatrix} = \text{Matriz de rotación} \quad \text{Ec. 1}$$

$$Q_G = RQ_P$$

Donde,

$Q_P$  = Vector  $Q$  medido en el marco de referencia del avión

$Q_G$  = Vector  $Q$  medido en el marco de referencia de la tierra

<sup>4</sup> PREMERLANI, William y BIZARD, Paul. Direction cosine matrix IMU: theory. 2009

La relación entre la matriz de cosenos de dirección y los ángulos de Euler es:

$$R = \begin{bmatrix} \cos \theta \cos \Psi & \sin \Phi \sin \theta \cos \Psi - \cos \Phi \sin \Psi & \cos \Phi \sin \theta \cos \Psi + \sin \Phi \sin \Psi \\ \cos \theta \sin \Psi & \sin \Phi \sin \theta \sin \Psi + \cos \Phi \cos \Psi & \cos \Phi \sin \theta \sin \Psi - \sin \Phi \cos \Psi \\ -\sin \theta & \sin \Phi \cos \theta & \cos \Phi \cos \theta \end{bmatrix} \text{ Ec. 2}$$

Donde,

$\Phi = \text{Ángulo roll}$

$\theta = \text{Ángulo pitch}$

$\Psi = \text{Ángulo yaw}$

La Ecuación 1 y la Ecuación 2 expresan cómo girar un vector medido en el marco de referencia del avión para el marco de referencia de la tierra. Ecuación 1 se expresa en términos de cosenos de dirección. Ecuación 2 es expresado en términos de ángulos de Euler.

## 2. HARDWARE

Para realizar el sistema de guía inercial, se necesita tener los datos proporcionados por sensores inerciales como acelerómetro, giroscopio y magnetómetro; además de disponer de una tarjeta de adquisición de datos para almacenar y procesar la información proporcionada por los anteriores sensores, un GPS para proporcionar información de posición y un módulo de telemetría con alcance suficiente para el envío de las medidas realizadas por los sensores inerciales y el GPS que se encuentran en el UAV, a la estación terrena.

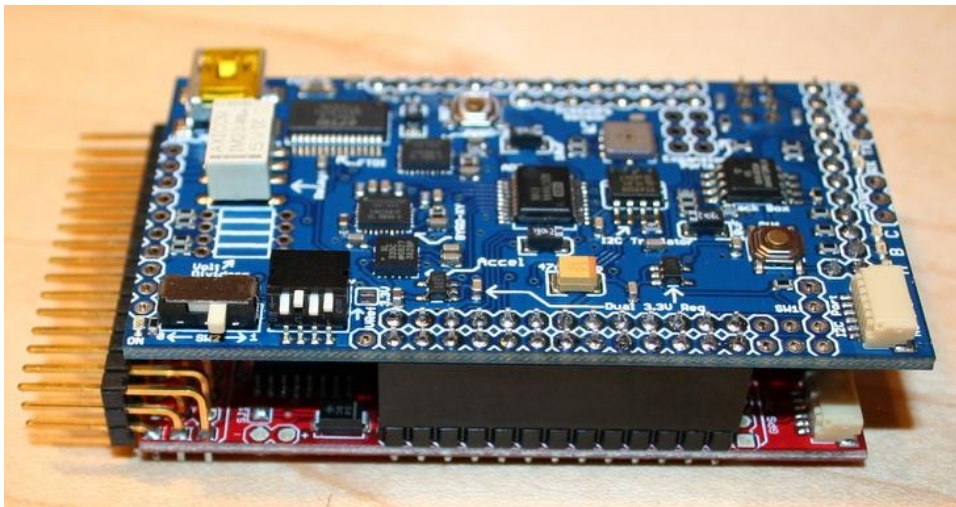
Para satisfacer estos requerimientos de hardware en la implementación del sistema de guía inercial, se hizo uso de los siguientes componentes:

- 1 unidad de medida inercial minIMU-9
- 1 tarjeta de adquisición de datos Arduino Uno, la cual cuenta con un microcontrolador Atmega328
- 1 GPS MT3329 de MediaTek
- Módulos de telemetría XBee-PRO 900

### Sistema ArduPilot Mega

En la etapa inicial de este proyecto se utilizó el sistema ArduPilot Mega para desarrollar el sistema de guía inercial, pero debido a una baja resolución de los datos del acelerómetro, se desistió de su utilización. Pese a esto, todo el trabajo realizado con este sistema, en cuanto al código e interfaz de usuario, sirvió como base para desarrollar el sistema de guía inercial definitivo. En la Figura 4 se puede observar el sistema ArduPilot Mega.

Figura 4. Sistema ArduPilot Mega



Fuente: <http://code.google.com/p/ardupilot-mega/wiki/Assembly>

## 2.1 UNIDAD DE MEDIDA INERCIAL

Los sensores inerciales utilizados en este proyecto vienen implementados en el pequeño chip minIMU-9, el cual facilita el montaje y adquisición de datos por el bus I2C; las lecturas de los sensores L3G4200D (giroscopio) y LSM303DLH (acelerómetro y magnetómetro), los cuales actúan como dispositivos esclavos en el bus I2C, pueden ser consultados a través de dicho bus. Los nueve datos independientes que proporciona la unidad de medida inercial, rotación, aceleración, y datos magnéticos, sumado al algoritmo apropiado, al microcontrolador y al computador, se utilizan para calcular la orientación de la tarjeta MinIMU-9 y consecuentemente la orientación del UAV. La Figura 5 muestra la cara superior de la tarjeta minIMU-9.

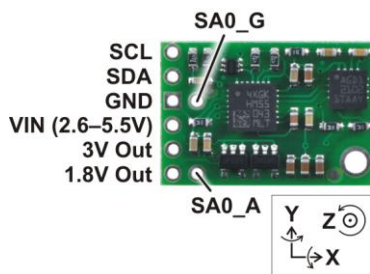
El chip L3G4200D es un sensor de velocidad angular, con un bajo consumo de energía y que proporciona medidas en los tres ejes. Este chip incluye al sensor y a una interfaz IC capaz de proporcionar la medida de velocidad angular al mundo exterior a través de la interfaz digital I2C/SPI. Las características más importantes del chip L3G4200D, se nombran a continuación.

- Tensión de alimentación: 2.4 V a 3.6 V
- Temperatura de funcionamiento: -40°C a +85°C
- Interfaz de salida digital I2C/SPI
- Filtro pasa bajo y pasa alto integrado
- FIFO embebido
- Rango de medida:  $\pm 250$ ,  $\pm 500$ ,  $\pm 2000$  dps

El chip LSM303DLH contiene un sensor de aceleración lineal y un sensor magnético que realizan mediciones en los tres ejes cada uno. Las características más importantes del chip LSM303DLH, son:

- Tensión de alimentación analógica: 2.5 V a 3.3 V
- Interfaz serial I2C
- Opera en un rango de temperatura de -30°C a +85°C
- Rango de medición de la aceleración lineal:  $\pm 8.0$  g
- Rango de medición magnética:  $\pm 8.1$  gauss

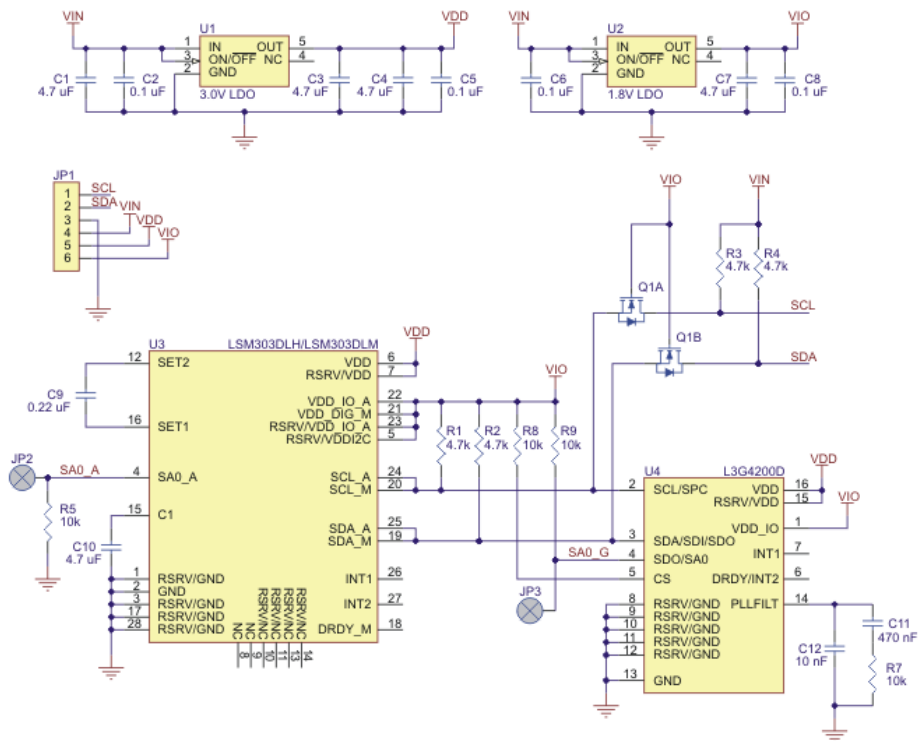
Figura 5. Tarjeta MinIMU-9



Fuente: <http://www.pololu.com/catalog/product/1264>

En la Figura 6 se puede observar el diagrama esquemático de la unidad de medida inercial minIMU-9.

Figura 6. Diagrama esquemático de la unidad de medida inercial minIMU-9



Fuente: <http://www.pololu.com/catalog/product/1264>

- **Conexión entre la tarjeta minIMU-9 y la tarjeta Arduino Uno.**

Para realizar la conexión de la tarjeta minIMU-9 con la tarjeta Arduino Uno, solo es necesario realizar cuatro conexiones, que son las de los pines VIN, GND, SCL, y SDA de la tarjeta minIMU-9 con sus correspondientes de la tarjeta Arduino Uno. El pin VIN debe ser conectado a una fuente de 2.6 a 5.5V, el pin GND a 0 voltios, y los pines SCL y SDA deben ser conectados al bus I2C del microcontrolador Atmega328 que se encuentra en la tarjeta Arduino Uno; más específicamente, la conexión de la tarjeta MinIMU-9 con la tarjeta Arduino Uno, se realiza como lo muestra el Cuadro 1.

Cuadro 1. Conexión entre la tarjeta MinIMU-9 y la tarjeta Arduino Uno

Arduino Uno	MinIMU-9
5 V	VIN
GND	GND
Pin analógico 4	SDA
Pin analógico 5	SCL

Fuente: <https://github.com/pololu/MinIMU-9-Arduino-AHRS>

Donde,

VIN: conexión de voltaje de entrada principal, 2.6V a 5.5V

GND: conexión a tierra, 0V

SCL: Línea de reloj

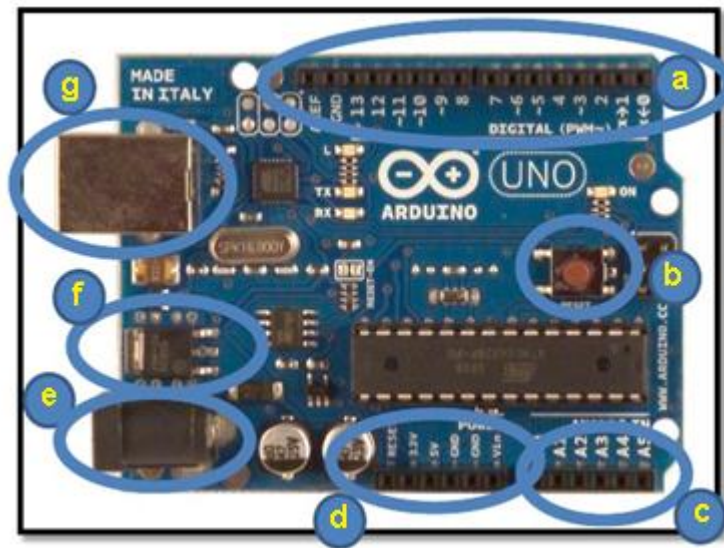
SDA: Línea de datos

## 2.2 TARJETA DE ADQUISICIÓN DE DATOS

El diseño del sistema de guía inercial se realizó con el apoyo de la tarjeta de adquisición de datos Arduino Uno, la cual contiene el microcontrolador Atmega328 necesario para realizar el almacenamiento y el procesamiento de los datos provenientes de la unidad de medida inercial y del GPS, además de que permite la conexión de la telemetría a través de los módulos ZigBee.

La tarjeta Arduino Uno se utiliza debido a que es una tarjeta de hardware libre, que cuenta con las características necesarias para el desarrollo del presente proyecto, entre estas características se puede destacar: 14 pines de entrada/salida digitales, 6 entradas analógicas, conexión USB para la alimentación y transmisión de datos desde la tarjeta al computador, posee un regulador de voltaje con un rango de 6 V a 20 V, y un botón de reinicio. En la Figura 7 se puede observar resaltadas las partes importantes de esta tarjeta.

Figura 7. Partes importantes de la tarjeta Arduino Uno



Fuente: <http://es.scribd.com/doc/62962340/Manual-Arduino-Uno>

**a.** Línea de comunicaciones, donde se encuentran los pines de entradas/salidas digitales, del 2 al 13.

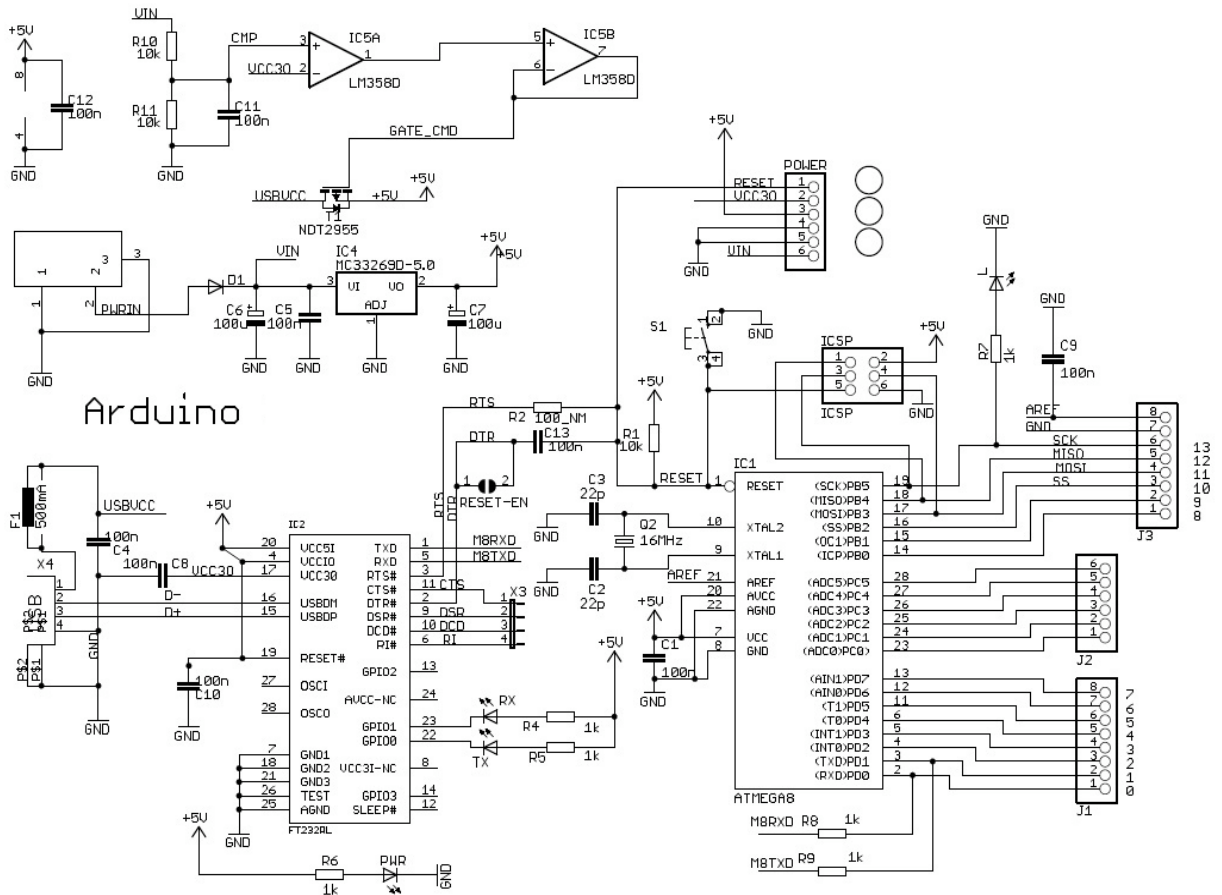
**b.** Botón de reinicio de la tarjeta.



- c. Línea de entradas analógicas, de la A0 a la A5.
- d. Línea de alimentación. En estos pines se encuentran Vin, GND, 5V, 3.3V y reinicio.
- e. Conector de alimentación de la tarjeta, para voltajes entre 7 a 12 voltios máximo.
- f. Regulador de voltaje.
- g. Conector USB.

Esta tarjeta tiene un tamaño pequeño, el cual la hace ideal para ser utilizada en el UAV porque es necesario que el peso y el tamaño del sistema implementado no interfieran en el normal funcionamiento del vehículo aéreo no tripulado. Además, la tarjeta Arduino Uno permite que la tarjeta diseñada (con la unidad de medida inercial, el GPS y el módulo ZigBee), sea dispuesta en la parte superior de la tarjeta Arduino Uno, logrando un diseño del sistema de guía inercial compacto y atractivo visualmente con un estilo similar al sistema ArduPilot Mega. La Figura 8 muestra el diagrama esquemático de la tarjeta Arduino Uno.

Figura 8. Diagrama esquemático de la tarjeta Arduino Uno



Fuente: <http://www.projectpageone.com/informes/gettingstartedarduino.html>

## 2.3 MÓDULO RECEPTOR GPS

El módulo receptor GPS utilizado en el presente proyecto es el GPS MT3329 de MediaTek, el cual es un receptor GPS que trabaja a una frecuencia de actualización por defecto de 1Hz configurable hasta 10Hz, velocidad de transmisión por defecto de 9600 bps, y una precisión de la posición sin ayuda de 3 m. En la Figura 9 se puede observar el módulo receptor GPS MT3329.

Figura 9. GPS MT3329



Fuente: <https://store.diydrones.com/ProductDetails.asp?ProductCode=MT3329-02>

El protocolo de comunicación del módulo receptor GPS MT3329 es el protocolo NMEA, el cual es definido y controlado por la organización estadounidense NMEA<sup>5</sup> (asociación electrónica marina nacional). La trama de salida utilizada por este GPS es la RMC la cual entrega la información de navegación mínima recomendada. Para utilizar el módulo receptor GPS MT3329, es necesario incluir las librerías NewSoftSerial y TinyGPS en el directorio de Arduino.

- **Conexión entre el GPS MT3329 y la tarjeta Arduino Uno.**

Para realizar la conexión del receptor GPS a la tarjeta Arduino Uno, se necesitan realizar cuatro conexiones: un pin para la alimentación, uno para GND, un pin para la transmisión de datos y otro para recepción, la interfaz de salida de datos utilizada para la comunicación del GPS con la tarjeta Arduino Uno, es la interfaz UART (transmisor-receptor asíncrono universal) que utiliza niveles TTL (lógica transistor a transistor); mas específicamente, la conexión del GPS MT3329 a la placa Arduino, consiste en conectar la salida de transmisión (TXD) del GPS, al pin digital 2 de la placa Arduino Uno, como se puede observar en el Cuadro 2.

Cuadro 2. Conexión entre el GPS MT3329 y la tarjeta Arduino Uno

Arduino Uno	GPS MT3329
5 V	VIN
GND	GND
Pin digital 2	TXD
Pin digital 3	RXD

---

<sup>5</sup> Para más información: <http://www.gpsinformation.org/dale/nmea.htm>

## 2.4 MÓDULOS DE TELEMETRÍA

Para realizar este proyecto se necesitó utilizar módulos de telemetría para el envío y recepción de la información, desde los sensores inerciales y el GPS hacia el computador, en el cual se encuentra el programa e interfaz de usuario encargados de identificar, convertir y visualizar toda la información correspondiente al sistema de guía inercial.

Los módulos de telemetría ZigBee utilizados son los módulos XBee-PRO 900, los cuales tienen una frecuencia de operación de 900 MHz y ofrecen características como: un alcance de 10 km utilizando antenas de alta ganancia y una velocidad de datos rápida de 156 Kbps, convenientes para su utilización en el sistema de guía inercial, debido a que es ideal contar con unos módulos de telemetría de alto alcance que permitan obtener los datos provistos por la unidad de medida inercial y el GPS en el momento de realizar los vuelos en el UAV.

Para realizar la conexión entre el módulo XBee y la tarjeta Arduino Uno, solamente hay que conectar los pines DODI Y DIDO de la tarjeta adaptadora *XtreamBee*, a los pines digitales 0 (Rx) y 1 (Tx) de la tarjeta Arduino Uno, respectivamente. El módulo XBee que se encuentra conectado al computador, tiene protocolo de comunicación serial a USB, mientras que el módulo XBee que está conectado a la tarjeta Arduino Uno, utiliza un protocolo de comunicación serial RS-232. En la Figura 10 se observan los módulos de telemetría utilizados en el proyecto.

Figura 10. Módulos XBee-PRO 900 utilizados en el proyecto.

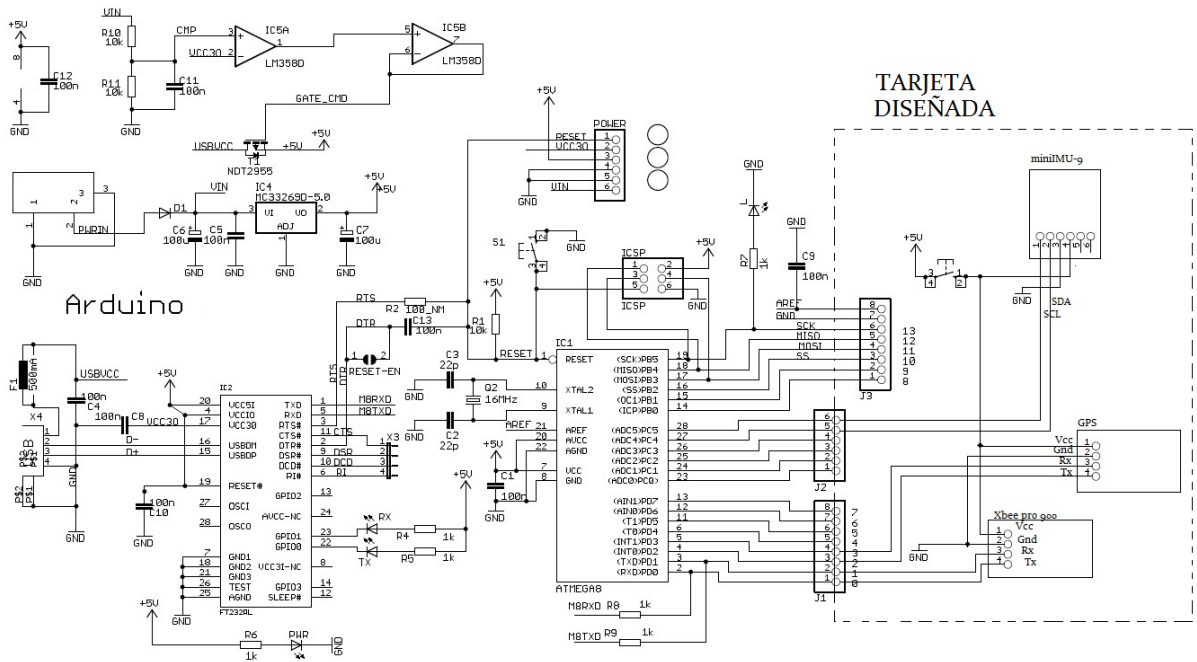


Fuente: <https://code.google.com/p/ardupilot-mega/wiki/Wireless>

## 2.5 DISEÑO DE LA TARJETA

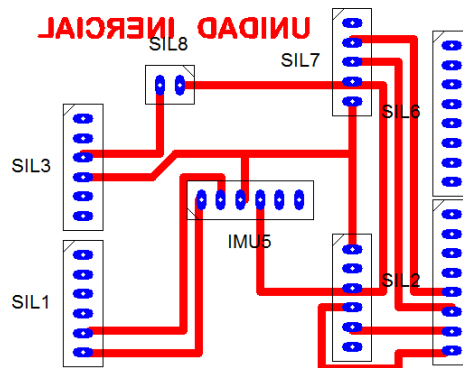
El diseño de la tarjeta se realizó en el programa PCB Wizard, teniendo en cuenta la polaridad y colocación de cada componente. En la Figura 11 se observa el diagrama esquemático de la tarjeta diseñada y su conexión con la tarjeta Arduino Uno.

Figura 11. Diagrama esquemático de la tarjeta diseñada y su conexión con la tarjeta Arduino Uno.



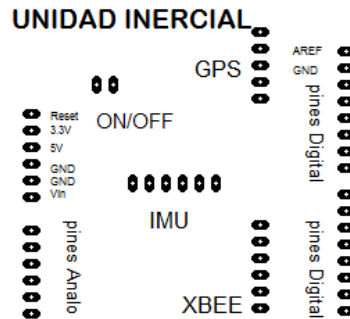
La Figura 12 muestra la cara inferior de la tarjeta diseñada, en la cual se observan los caminos o buses de transmisión de datos de los sensores a los pines de entrada de la tarjeta Arduino uno.

Figura 12. Cara inferior de la tarjeta diseñada



En la Figura 13 se aprecia la cara superior o de componentes de la tarjeta diseñada, la cual muestra las etiquetas donde van cada uno de los componentes, en este caso, la IMU, el GPS y el módulo de telemetría XBee, también se encuentran las referencias de los voltajes de entrada, tierra y algunas entradas analógicas y digitales disponibles para agregar más sensores.

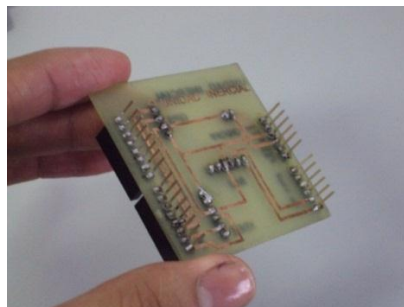
Figura 13. Cara superior o de componentes de la tarjeta diseñada



- **TARJETA TERMINADA**

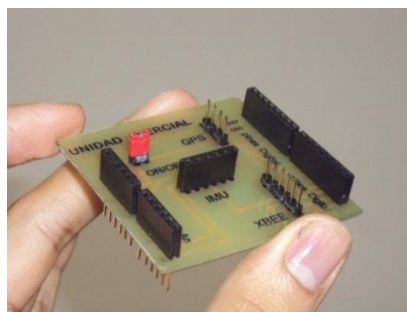
El diseño se imprimió en una impresora láser en papel press-n-peel<sup>5</sup> para luego introducirse en ácido cloruro férrico y así terminar el proceso. La Figura 14 muestra la cara inferior de la tarjeta terminada, en donde se observa la placa con sus componentes soldados.

Figura 14. Cara inferior de la tarjeta terminada



En la Figura 15 se observa la tarjeta terminada con sus hoyos perforados y sus componentes soldados.

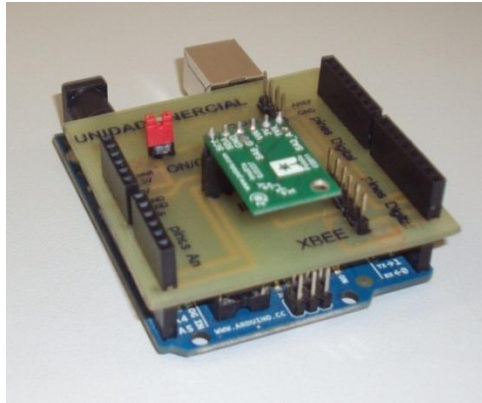
Figura 15. Cara superior de la tarjeta terminada



<sup>5</sup> Tipo de papel utilizado en la realización de circuitos impresos y que funciona solo con impresoras láser.

La Figura 16 muestra la tarjeta diseñada conectada en la parte superior de la tarjeta Arduino Uno.

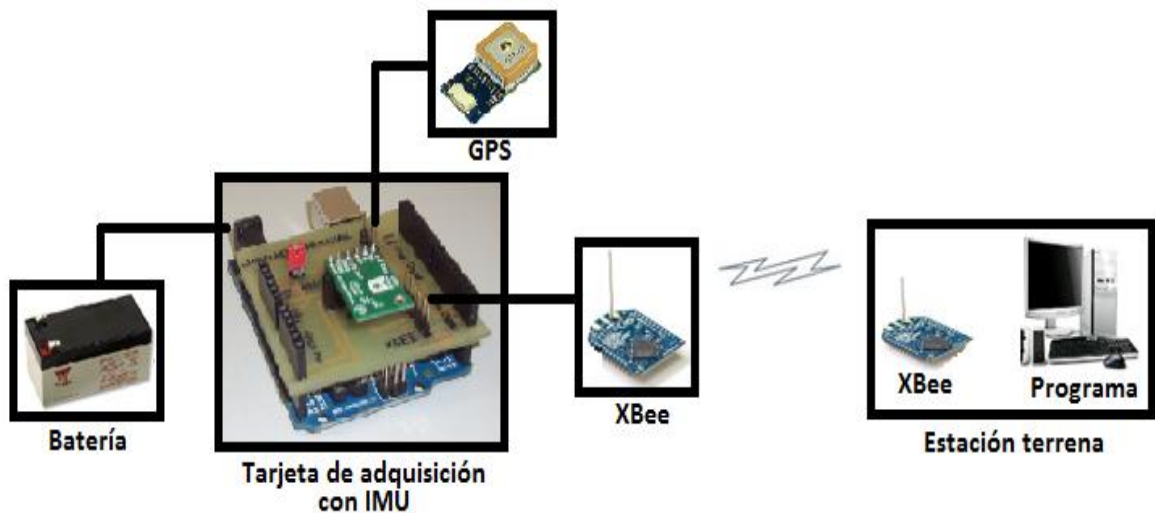
Figura 16. Tarjeta terminada conectada a la tarjeta Arduino Uno



- **Sistema completo:**

El sistema de guía inercial diseñado para ser utilizado en la determinación de la posición del UAV, está conformado por la tarjeta de adquisición de datos Arduino Uno, en donde se encuentran conectados la unidad de medida inercial minIMU-9, el GPS MT3329 y el módulo XBee, quien es el encargado del envío de los datos a la estación terrena, la cual a su vez está compuesta por un computador en donde se recibe y procesa toda la información para el cálculo de la posición deseada. La representación del sistema de guía inercial completo, se observa en la Figura 17.

Figura 17. Representación del sistema completo



### 3. SOFTWARE

Para realizar la obtención, decodificación, procesamiento y visualización de toda la información que proporciona el sistema de guía inercial, se ha utilizado el lenguaje de programación Visual C# 2010 de Microsoft, debido a que es un lenguaje de programación orientado a objetos y que permite la fácil creación de interfaz de usuario, brindando al diseñador las herramientas suficientes para la realización de cualquier tipo de proyecto en lo concerniente al software, además tiene la ventaja de dar a los estudiantes una versión gratuita.

#### 3.1 PROGRAMACIÓN ARDUINO

La plataforma Arduino se programa mediante el uso de un lenguaje propio *Open Source*, el cual está basado en el lenguaje de programación Java; es por esta razón, que el lenguaje Arduino se utiliza para realizar la programación del microcontrolador Atmega328 de la familia Atmel, contenido en la tarjeta Arduino Uno.

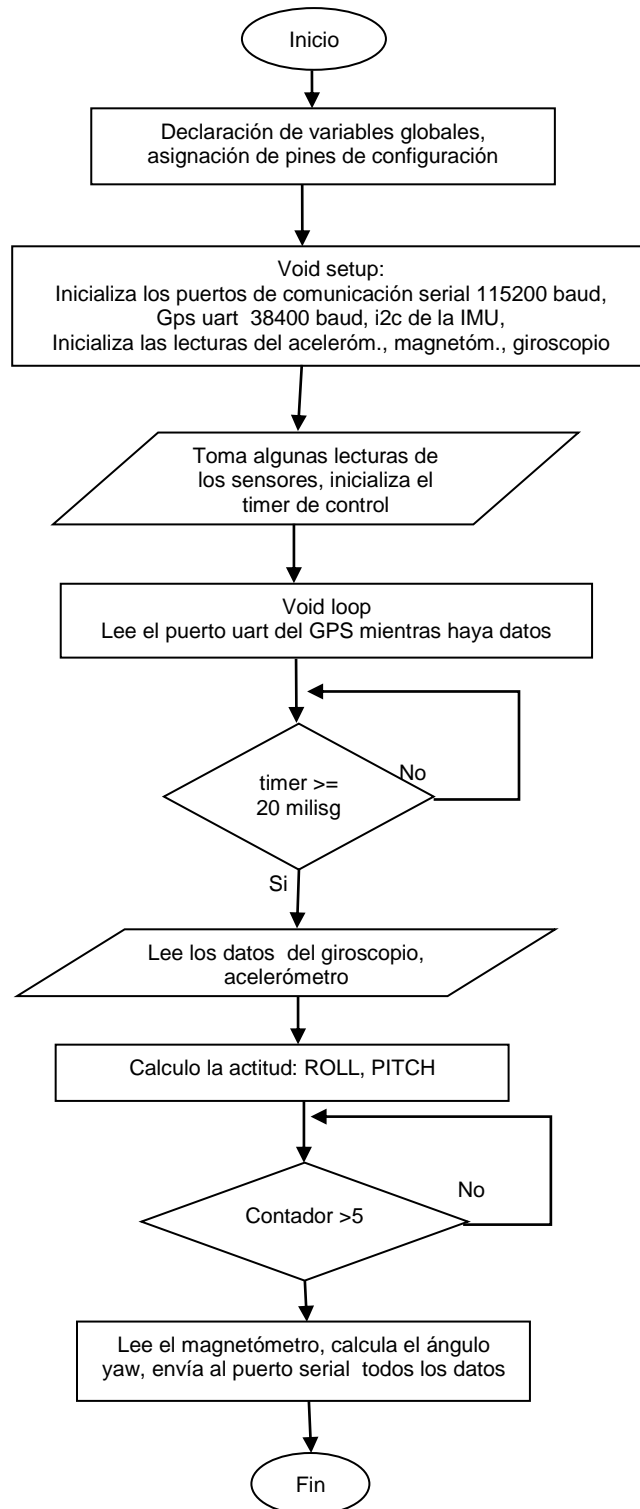
**3.1.1 Protocolo de comunicación.** La obtención de la información se realiza mediante la tarjeta Arduino uno, la cual es la encargada de recibir y procesar todos los datos de los sensores, una vez estos llegan a la tarjeta; debido a que la información enviada es recibida en el computador como una trama de datos en formato texto y la cual no especifica a que corresponde cada dato, se procede a utilizar el identificador de trama implementado en el microcontrolador, el cual consiste en agregarle una letra mayúscula a cada dato que se envía al computador, que corresponde a la información de los sensores inerciales y el GPS.

Las letras identificadoras que se anteponen a cada uno de los datos son:

- R: *roll*
- P: *pitch*
- Y: *yaw*
- A: aceleración en x
- C: aceleración en y
- L: aceleración en z
- T: latitud
- D: longitud
- H: altitud
- G: curso
- S: velocidad

En la Figura 18 se muestra el diagrama de flujo del microcontrolador Atmega328 realizado en el lenguaje de programación Arduino.

Figura 18. Diagrama de flujo del microcontrolador Atmega328



En la Figura 19 se puede apreciar cómo se antepone un identificador a los ángulos de Euler, en este caso, a los ángulos *roll*, *pitch* y *yaw* se le anteponen las letras R, P, Y, respectivamente.



Figura 19. Identificador de trama de los ángulos de Euler

```
File Edit Sketch Tools Help
MinIMU9AHRS Compass DCM I2C Output $ Vector gps mat

//codigo grabado
void printdata(void)
{

    #if PRINT_EULER == 1

    Serial.print("R");
    Serial.print(ToDeg(roll));
    Serial.print("\t"); // \t: tabulador

    Serial.print("P");
    Serial.print(ToDeg(pitch));
    Serial.print("\t");

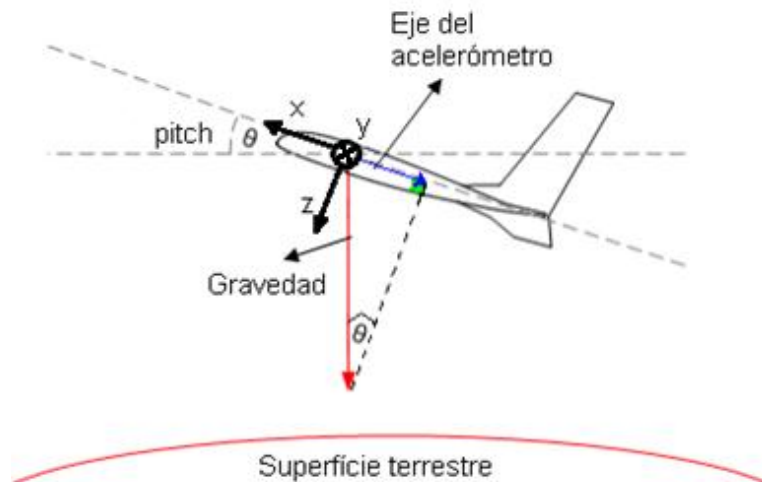
    Serial.print("Y");
    Serial.print(ToDeg(yaw));
    Serial.print("\t");

    }
```

**3.1.2 Cálculo de la actitud.** El objetivo en esta parte es obtener la actitud del UAV a partir de las medidas de los sensores inerciales. Existen varios métodos para calcular la actitud, a continuación se explican tres de ellos.

**3.1.2.1 Cálculo de la actitud usando acelerómetro.** El acelerómetro da una medida de la aceleración en cada uno de sus ejes, una de sus más importantes aplicaciones, es como sensor de inclinación o inclinómetro, a continuación se describe cómo medir la actitud con los datos entregados por el acelerómetro:

Figura 20. Ángulo *pitch* en sentido horario



Fuente: POZO ESPÍN, David Fernando. Diseño y construcción de una plataforma didáctica para medir ángulos de inclinación usando sensores inerciales como acelerómetro y giroscopio. 2010. p. 19

Cuando el UAV se inclina como en la Figura 20, el ángulo *pitch* entrega una señal debido a la componente de la gravedad presente. En la gráfica la flecha azul representa lo que el acelerómetro está midiendo a causa de la gravedad y la flecha roja representa la aceleración de la gravedad.

Para obtener el valor de la actitud o ángulo de inclinación se utilizan las siguientes fórmulas para cada eje:

$$\sin \theta = \frac{Ax}{1g} \rightarrow pitch = \text{asin} \left( \frac{Ax}{1g} \right)$$

$$\sin roll = \frac{Ay}{1g} \rightarrow roll = \text{asin} \left( \frac{Ay}{1g} \right)$$

Dónde,

$$\theta = Pitch$$

$Ax$  = Componente de la gravedad medida con el acelerómetro en el eje x

$Ay$  = Componente de la gravedad medida con el acelerómetro en el eje y

**3.1.2.2 Cálculo de la actitud usando giroscopio.** El giroscopio entrega la velocidad con la que está rotando (velocidad angular). Para conseguir la actitud es necesario integrar la velocidad angular, el cual consiste en multiplicar este dato por un valor de tiempo determinado "dt".

$$\int \text{velocidad angular} = \text{velocidad angular} \cdot dt = \text{ángulo}$$

Una pequeña variación en el ángulo es posible calcularla a través del producto de la velocidad angular por el tiempo en el cual se está realizando el movimiento.

Debido a que el giroscopio solo entrega datos cuando detecta un cambio de giro o en otras palabras cuando detecta movimiento, si se aplica la fórmula anterior solo se encuentra un ángulo en un instante determinado, para encontrar la actitud, es necesario tomar en cuenta que el proceso antes descrito se realiza en intervalos de tiempo establecidos "dt", con lo cual la actitud sería el resultado de un algoritmo acumulativo, en donde el sistema depende de los valores iniciales para que los cálculos sucesivos sean consistentes, obteniendo así el siguiente modelo:

$$\text{ángulo}_k = \text{ángulo}_{k-1} + U_k * dt$$

Donde,

$U_k$  = Señal acondicionada del giroscopio

$dt$  = Tiempo de muestreo

$\text{ángulo}_k$  = Estado presente

$\text{ángulo}_{k-1}$  = Estado anterior

**3.1.2.3 Estimación de la orientación con el magnetómetro.** Luego de hallar el *roll* y *pitch* con la ayuda del giroscopio y acelerómetro, se procede a encontrar el ángulo *yaw* usando el compás magnético o magnetómetro, tomando los datos que este sensor entrega en sus tres ejes y aplicándolo a la siguiente fórmula:

$$X_h = X \cos(\theta) + Y \sin(\emptyset) + Z \sin(\emptyset) \cos(\emptyset)$$

$$Y_h = Y \cos(\emptyset) - Z \sin(\emptyset)$$

Donde,

$\theta = Pitch$

$\emptyset = Roll$

$X = Medida del magnetómetro en el eje x$

$Y = Medida del magnetómetro en el eje y$

$Z = Medida del magnetómetro en el eje z$

Por último, se utiliza la siguiente ecuación para encontrar el ángulo *yaw*

$$yaw = atan\left(\frac{-Y_h}{X_h}\right)$$

Se ha tenido en cuenta el método descrito en el numeral 3.1.2.2 para calcular la actitud con el giroscopio, el método descrito en el numeral 3.1.2.3 para calcular el ángulo *yaw* y posteriormente hacerle una corrección a este ángulo, y el acelerómetro se incluye para corregir la deriva del giroscopio a los ángulos *roll* y *pitch*, los procedimientos descritos hacen parte del algoritmo principal de este trabajo, el cual es la matriz de cosenos directores o DCM.

**3.1.2.4 Cálculo de la actitud usando el Algoritmo DCM.** Para el cálculo de la actitud se necesitan los datos que entrega la unidad de medida inercial, es decir, los datos del acelerómetro, giroscopio y magnetómetro, posteriormente se utiliza un algoritmo matemático denominado matriz de cosenos directores (DCM) que realiza los cálculos correspondientes y halla los datos de actitud o ángulos de Euler (*roll*, *pitch*, *yaw*).

El procedimiento para el cálculo de la actitud es el siguiente,

a. Los giroscopios se utilizan como la principal fuente de información para encontrar la orientación, se integra la ecuación diferencial no lineal que relaciona la tasa de cambio en la orientación de la aeronave a la orientación correcta (marco de referencia de tierra) este proceso se realiza a un ritmo de 50 Hz<sup>6</sup>.

---

<sup>6</sup> PREMERLANI, William y BIZARD, Paul. Direction Cosine Matrix IMU: Theory. 2009. p. 2

La ecuación diferencial no lineal es,

$$r(t) = r(0) + \int_0^t d\theta(t) \times r(t)$$

Donde,

$$d\theta(t) = w(t)dt$$

$r(0)$  = Valor inicial del vector

$$\int_0^t d\theta(t) \times r(t) = \text{Cambio en el vector}$$

Debido a que los sensores están en el fuselaje del avión, estos entregan medidas en un marco de referencia diferente al de tierra, para obtener las medidas en el marco de referencia de la tierra, se aplica:

$$R(t + dt) = R(t) \begin{bmatrix} 1 & -d\theta_z & d\theta_y \\ d\theta_z & 1 & -d\theta_x \\ -d\theta_y & d\theta_x & 1 \end{bmatrix}$$

Donde,

$$d\theta_x = w_x dt$$

$$d\theta_y = w_y dt$$

$$d\theta_z = w_z dt$$

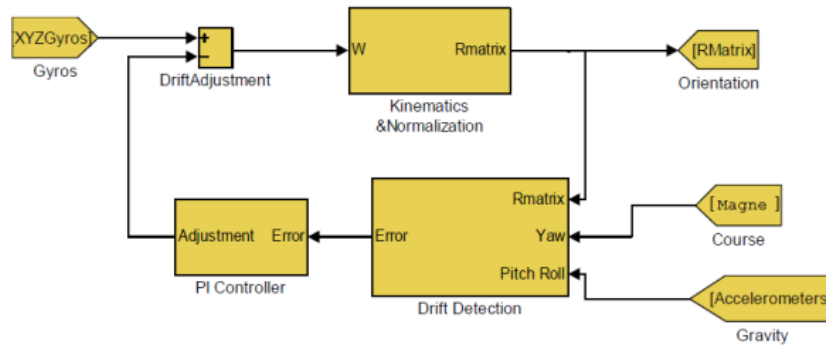
$w_x, w_y, w_z$  = Vector velocidad angular

**b.** Teniendo en cuenta que los errores numéricos en la integración, gradualmente violan las restricciones de ortogonalidad del algoritmo DCM, se hacen ajustes regulares a los elementos de la matriz para satisfacer estas restricciones.

**c.** Reconociendo que los errores numéricos, la deriva del giroscopio y offset poco a poco acumulan errores en los elementos de DCM, se utilizan vectores de referencia para detectar los errores, y un controlador proporcional integral (PI) de retroalimentación negativa entre los errores detectados y las entradas del giroscopio utilizados en el paso (a), para disipar los errores más rápido. El magnetómetro se utiliza para detectar el error de yaw, los acelerómetros se utilizan para detectar *pitch* y *roll*.

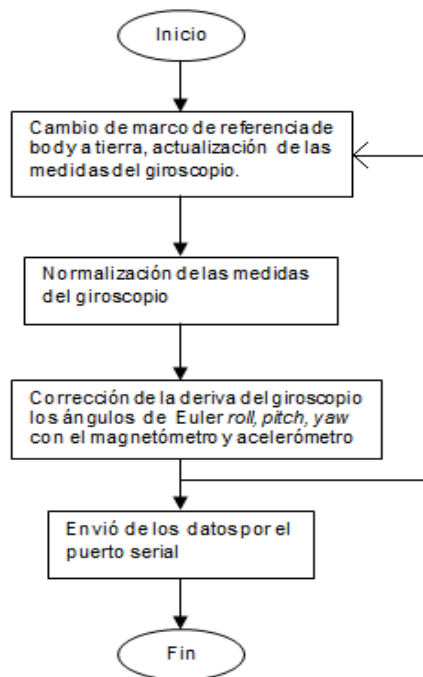
En la Figura 21 se observa el diagrama de bloques para el cálculo de la actitud por medio de la matriz DCM y en la Figura 22 se muestra el diagrama de flujo del algoritmo DCM implementado en el microcontrolador, siendo éste una forma alternativa del ver el diagrama de bloques de la Figura 21 pero en forma de diagrama de flujo.

Figura 21. Diagrama de bloques del proceso



Fuente: PREMERLANI, William y BIZARD, Paul. Direction cosine matrix IMU: theory. 2009. p. 3

Figura 22. Diagrama de flujo del algoritmo DCM en el microcontrolador



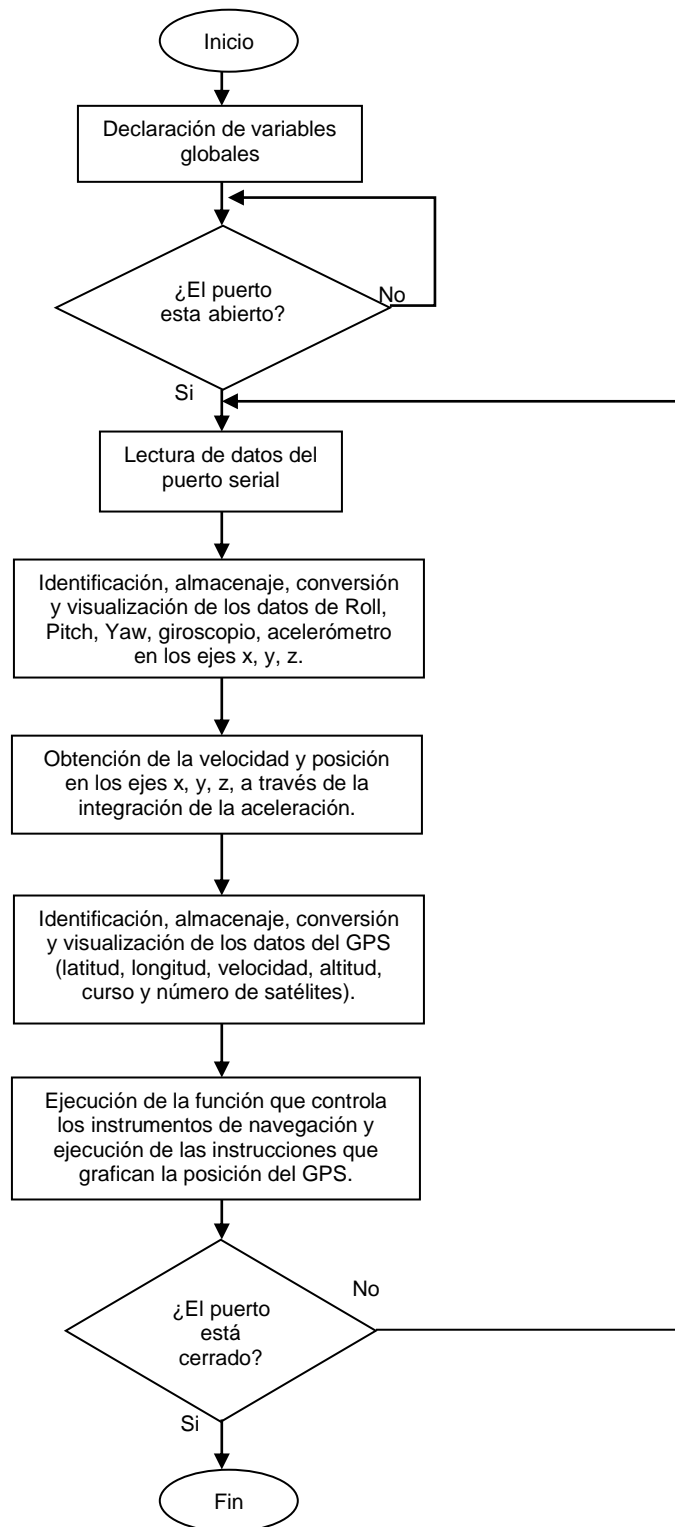
Las ecuaciones para el cálculo de la actitud se presentan en el ANEXO C.

### 3.2 PROGRAMACIÓN DE LA INTERFAZ DE USUARIO

Al empezar a utilizar el sistema conformado por la unidad de medida inercial y la tarjeta de adquisición de datos, lo primero que se necesitó resolver fue leer y decodificar todos los datos encontrados en el puerto serial, para ello se utilizó el componente *serialport* y su evento *DataReceived*, dentro de este evento se invoca una función que realiza todo el proceso de leer la información contenida en el puerto serial; para realizar esta parte del código y su correspondiente interfaz de usuario, se utilizó como referencia el código de software libre llamado *SerialPortTerminal*, en el cual se convierte la información disponible en el puerto serial a una cadena de

hexadecimales, esto con el objetivo de realizar la visualización y el adecuado tratamiento de los datos en cálculos posteriores. La Figura 23 muestra el diagrama de flujo del programa diseñado en Visual C#.

Figura 23. Diagrama de flujo del programa en Visual C#



Mediante el código, se crea un arreglo de bytes con el tamaño del número de bytes que se encuentran en el puerto serial esperando por ser leídos, mediante la propiedad *BytesToRead*; todos estos bytes se leen y se almacenan en el búfer creado a través del método *SerialPort.Read*, luego se hace la conversión de arreglo de bytes a cadena de hexadecimales como se mencionó anteriormente. En la Figura 24 se observa como se realiza la lectura de los datos del puerto serial.

Figura 24. Lectura de los datos en el puerto serial

```
//Inicio de la lectura de los datos en el puerto serial
private void serialPort1_DataReceived(object sender, SerialDataReceivedEventArgs e)
{
    this.Invoke(new EventHandler(Leer));
}

private void Leer(object sender, EventArgs e)
{
    //Selección de la velocidad de transmisión en baudios
    serialPort1.BaudRate = int.Parse(baudrate.SelectedItem.ToString());

    //Obtiene el número de bytes en espera en el búfer del puerto
    int bytes = serialPort1.BytesToRead;

    //Crea un arreglo de bytes para guardar los datos que llegan
    byte[] buffer = new byte[bytes];

    //Lee los datos desde el puerto y los almacena en el búfer
    serialPort1.Read(buffer, 0, bytes);

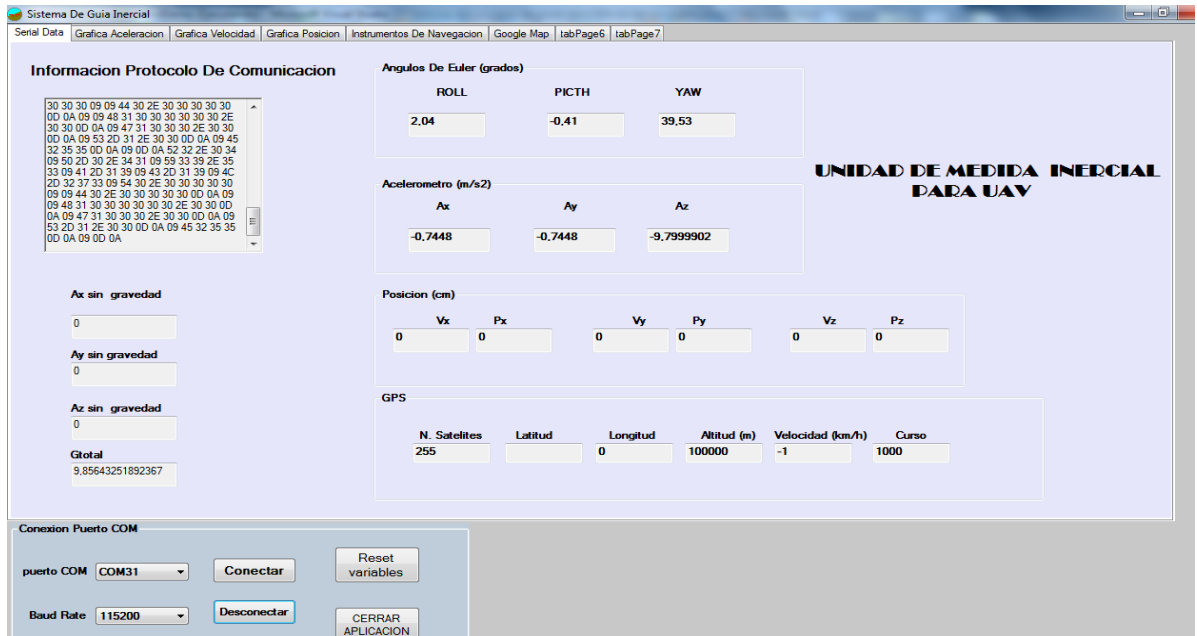
    //Muestra los datos que llegan en formato hexadecimal
    ArregloDeBytesA_CadenaDeHexadecimales(buffer);
}
```

El programa diseñado para la visualización de las variables se realizó en el lenguaje de programación Visual C# de la plataforma .net, el cual realiza la lectura del puerto serial y obtiene la información correspondiente a cada variable utilizando un ciclo condicional (if ( )) igual a la letra identificadora de cada dato en formato hexadecimal que llegaba a la variable buffer, el condicional (if) compara la letra R que en hexadecimal es 0x52, con la trama de datos que llega al buffer[i], si la condición es verdadera se procede a obtener la información.

Una vez identificado cada dato, se procede a transformar la información entrante de formato texto a formato número, en este caso *double* (), porque se trabaja con número decimales, esta tarea la realiza el comando *tryparse* de C#, para que este comando realice esta tarea se declara una matriz tipo *byte[]* que almacena los caracteres perteneciente a cada número, luego con el comando *TrimEnd* se eliminan los caracteres inválidos como el retorno de carro “ / ” y algunas letras, posteriormente se realiza la conversión a formato número.

Para finalizar, los datos obtenidos se utilizan en las ecuaciones o se muestran en los instrumentos y *textbox* correspondientes. En la Figura 25 se observa la interfaz de usuario diseñada para el sistema de guía inercial.

Figura 25. Interfaz de usuario del sistema de guía inercial



**3.2.1 Eliminación de la componente de la gravedad del acelerómetro.** Al tener los datos del acelerómetro en los tres ejes, se hace necesaria la eliminación de la componente de la gravedad de los mismos, porque de esta forma se evita que al inclinar la unidad de medida inercial, el acelerómetro mida la componente de gravedad en esa dirección y calcule erróneamente la velocidad y la posición a partir de ella. Una vez eliminada, el acelerómetro solo entrega la medida de la aceleración producida por una fuerza externa. Para implementar este proceso, se necesitan los valores de *roll* ( $\Phi$ ) y *pitch* ( $\theta$ ). Las ecuaciones para eliminar la componente de la gravedad de los valores de aceleración, son:

$$\begin{aligned} a_{Ix} &= a_x - g_{total} * \sin \theta \\ a_{Iy} &= a_y + g_{total} * \cos \theta * \sin \varphi \\ a_{Iz} &= a_z + g_{total} * \cos \theta * \cos \varphi \end{aligned}$$

Donde

$$g_{total} = \sqrt{a_x^2 + a_y^2 + a_z^2}$$

$$\theta = pitch$$

$$\varphi = roll$$

El resultado de utilizar estas ecuaciones es que los datos de aceleración en los tres ejes siempre van a ser cero, sin importar el grado de inclinación en el que se encuentre la unidad de medida inercial, resultado que es indispensable para el correcto funcionamiento del sistema de guía inercial, específicamente en la parte que tiene que ver con los cálculos de la velocidad y la posición, debido a que estos



cálculos necesitan que los datos de la aceleración sean cero cuando la unidad de medida inercial no se encuentra en movimiento.

**3.2.2 Ecuaciones cinemáticas.** Una de las partes fundamentales de un sistema de guía o navegación inercial, es el hecho de conocer el recorrido y la posición del vehículo en el cual se encuentre instalado dicho sistema; es por esta razón que se realiza el cálculo de la posición a partir de las medidas de aceleración entregadas por un sensor como el acelerómetro. Debido a que la posición se obtiene con una doble integración de la aceleración y que realizar una integral es hallar el área bajo la curva, se procede a realizar esta integración mediante un tipo de integración numérica denominada regla del trapecio.

De acuerdo con las leyes del movimiento que estudia la cinemática, la ecuación para hallar la velocidad a partir de la aceleración, es la siguiente

$$v = v_o + \int a . dt$$

Donde,

$v =$  *velocidad*

$v_o =$  *velocidad inicial*

$a =$  *aceleración*

Los datos de aceleración que entrega la unidad de medida inercial minIMU-9 forman una curva, se necesita integrar la aceleración para encontrar la velocidad, es decir, se necesita hallar el área bajo la curva de la aceleración, esta área se va a obtener utilizando la regla del trapecio y para ello se divide esta área en áreas con forma de trapecio de menor tamaño. El área total se encuentra realizando una sumatoria de las áreas más pequeñas y que tienen forma de trapecio.

Para obtener el área de un trapecio se divide éste en dos figuras geométricas, un rectángulo y un triángulo rectángulo, se calcula el área de cada una de ellas y se realiza la suma de estas dos áreas, dando como resultado el área del trapecio; esto está representado por la siguiente ecuación,

$$Area_{trapecio} = Area_{rectangulo} + Area_{trianguloRect.}$$

Donde,

$$Area_{rectangulo} = a_o * \Delta t$$

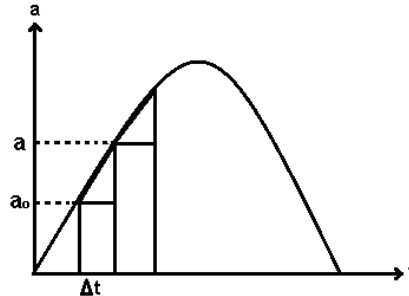
$$Area_{trianguloRect.} = \frac{1}{2} (a - a_o) * \Delta t$$

Por lo tanto,

$$Area_{trapecio} = a_o * \Delta t + \frac{1}{2} (a - a_o) * \Delta t$$

En la Figura 26 se observa una representación de los rectángulos y triángulos rectángulos que conforman cada trapecio, al realizar la suma de las áreas de todos los trapecios, se obtiene el área bajo la curva total aproximada, que corresponde a la integral de la aceleración y consecuentemente a la velocidad.

Figura 26. División de la señal de aceleración en subáreas



Para obtener la posición a partir de la velocidad se procede de una forma similar a la mencionada anteriormente, es decir, se integra la velocidad por medio de una integración numérica utilizando la regla del trapecio. La ecuación general para obtener la posición a partir de la velocidad es la siguiente,

$$x = x_o + \int v . dt$$

Donde,

$x$  = posición

$x_o$  = posición inicial

$v$  = velocidad

Las ecuaciones utilizadas en el programa para el cálculo de la posición son:

$$velocidad = velocidadInicial + aceleraciónInicial * t + \frac{1}{2}(x1 - aceleraciónInicial) * t$$

$$posición = posiciónInicial + velocidadInicial * t + \frac{1}{2}(velocidad - velocidadInicial) * t$$

Donde,

$x1$  = aceleración actual

$t$  = tiempo

Luego se iguala la aceleración, velocidad y posición actual, para que el algoritmo sea acumulativo, así:

$aceleraciónInicial = x1$

$velocidadInicial = velocidad$

$posiciónInicial = posición$

## 4. RESULTADOS

Luego de implementar el sistema se procede a realizar las pruebas para comprobar su correcto funcionamiento. Inicialmente se hicieron en un ambiente controlado y por último se realizaron en un UAV.

Para las pruebas realizadas en un ambiente controlado se tuvo en cuenta que las variables fueran manipuladas por el usuario, se verificaron los ángulos de Euler tomando como referencia una maqueta, luego el GPS y la posición determinada por los sensores inerciales.

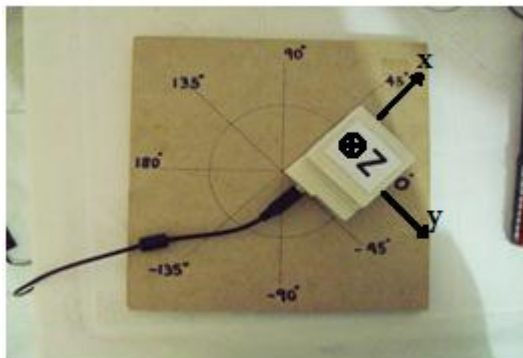
### 4.1 RESULTADOS EN UN AMBIENTE CONTROLADO

**4.1.1 Comprobación de los ángulos de Euler.** Se realizaron diferentes medidas angulares para comprobar la actitud de la IMU (unidad de medida inercial) utilizando una maqueta como muestra la Figura 27. Siendo *roll* una rotación en el eje  $x+$ , *pitch* una rotación en el eje  $y+$ , *yaw* una rotación en el eje  $z+$ . El sistema se dispuso como se muestra en las Figuras 27 y 28, mostrando sus valores en la interfaz de usuario diseñada en Visual C# y los resultados obtenidos se describen a continuación.

Figura 27. Maqueta propuesta para medir los ángulos *roll* y *pitch*



Figura 28. Maqueta para medir el ángulo yaw



## Resultados para los ángulos *roll*, *pitch* y *yaw*

Se realizaron varias medidas para los ángulos *roll*, *pitch* y *yaw*; el valor del porcentaje de error para todas las pruebas realizadas en este capítulo, se calcula por medio de la siguiente ecuación,

$$error(\%) = \frac{|valor_{medido} - valor_{real}|}{|valor_{medido}|} * 100\%$$

### Porcentaje de error para el ángulo *roll*

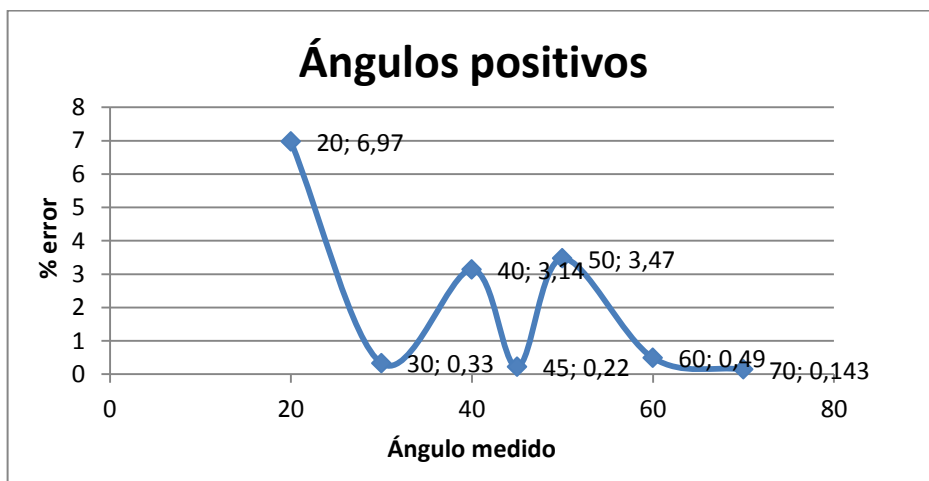
El Cuadro 3 muestra las medidas realizadas para diferentes ángulos *roll* y sus correspondientes porcentajes de error, en el cual se observa que los porcentajes de error obtenidos son bajos.

Cuadro 3. Porcentaje de error para el ángulo *roll*

Ángulo real (grados)	Ángulo medido (grados)	Porcentaje de error (%)
20	21,5	6,97
30	30,1	0,33
40	41,3	3,14
45	44,9	0,22
50	51,8	3,47
60	60,3	0,49
70	69,9	0,14

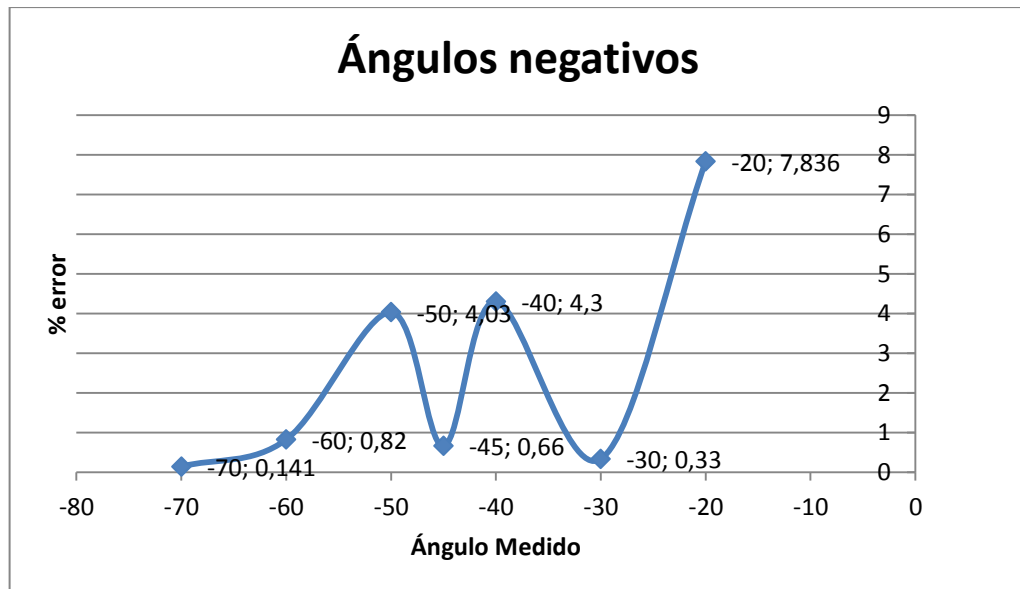
La Figura 29 es una gráfica de los ángulos positivos (*roll*) con respecto al porcentaje de error.

Figura 29. Medida de los ángulos positivos (*roll*)



En la Figura 30 se observa la gráfica de los ángulos negativos (*roll*) con respecto al porcentaje de error.

Figura 30. Medida de los ángulos negativos (*roll*)



### Porcentaje de error para el ángulo *pitch*

El Cuadro 4 muestra las medidas realizadas para diferentes ángulos *pitch* y sus correspondientes porcentajes de error, se aprecia que los valores para el porcentaje de error obtenidos son bajos.

Cuadro 4. Porcentaje de error para el ángulo *pitch*

Ángulo real (grados)	Ángulo medido (grados)	Porcentaje de error (%)
20	20,3	1,47
30	29,1	3,09
40	40,2	0,49
45	44,1	2,04
50	50,4	0,79
60	59,8	0,33
70	69,7	0,43

En las Figuras 31 y 32 se aprecia las gráficas de los ángulos positivos y negativos (*pitch*), respectivamente, con respecto al valor del porcentaje de error.

Figura 31. Medida de los ángulos positivos (*pitch*)

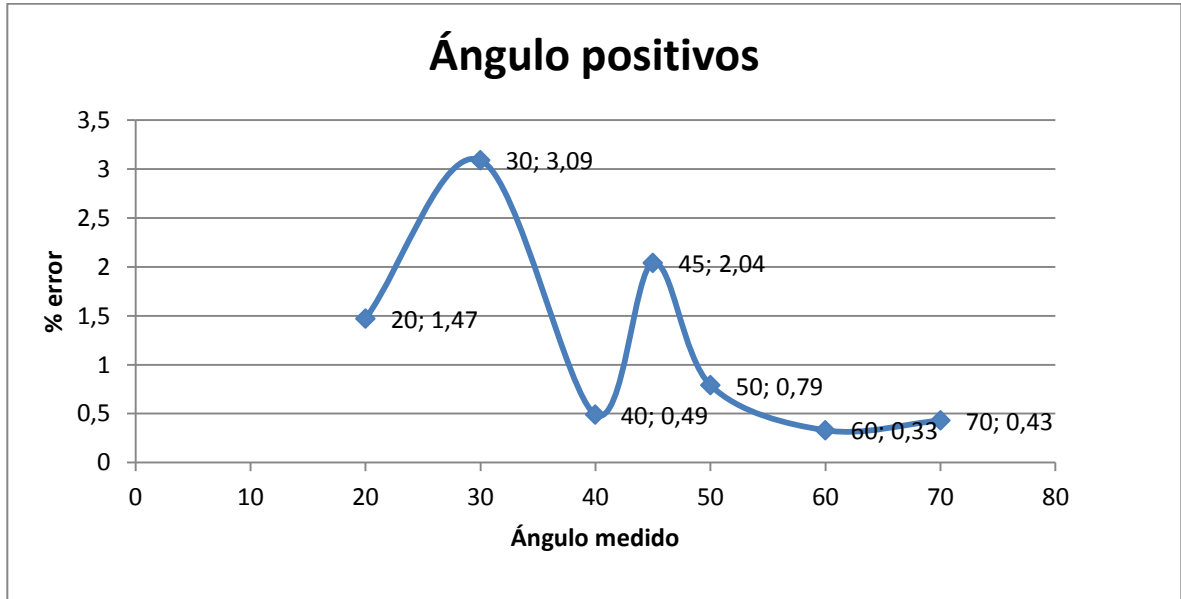
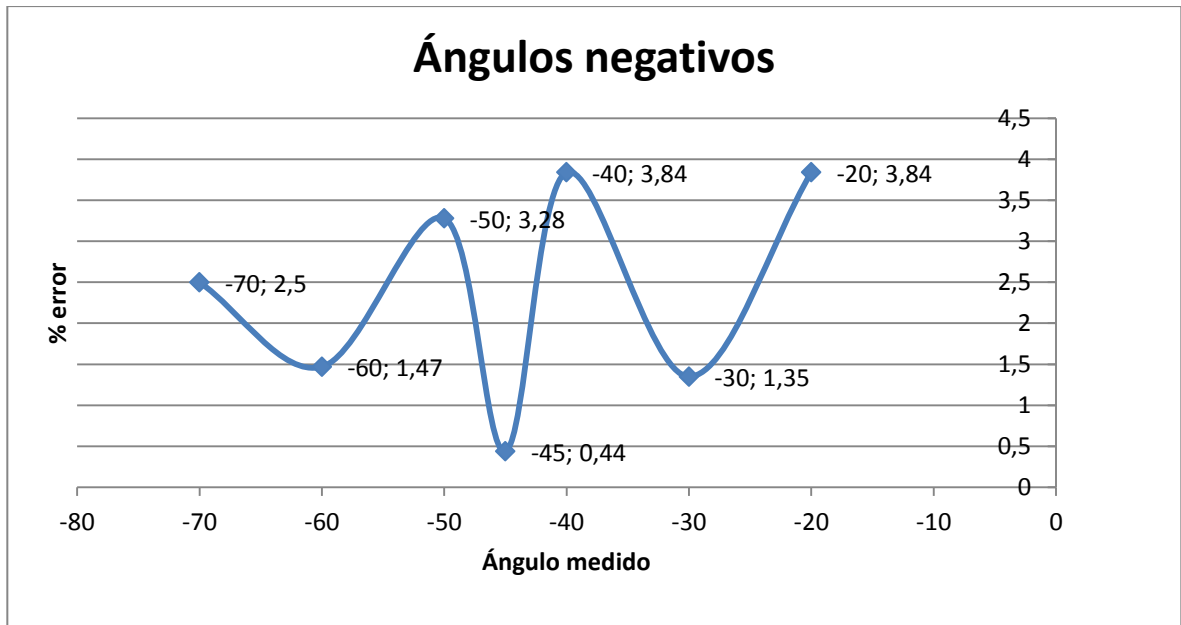


Figura 32. Medida de los ángulos negativos (*pitch*)



### Porcentaje de error para el ángulo yaw

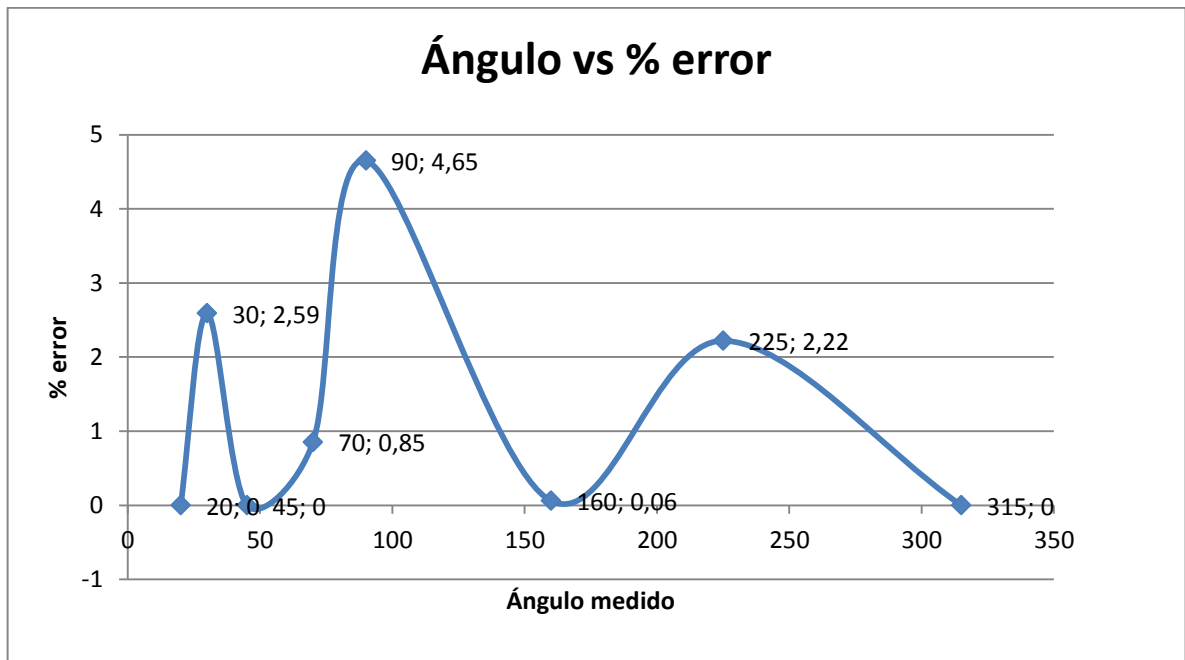
El Cuadro 5 muestra las diferentes medidas realizadas para el ángulo yaw y sus correspondientes porcentajes de error, observándose que los valores obtenidos para los porcentajes de error son bajos.

Cuadro 5. Porcentaje de error para el ángulo yaw

Ángulo real (grados)	Ángulo medido (grados)	Porcentaje de error (%)
20	20	0.00
30	30.8	2.59
45	45	0.00
70	69	0.85
90	86	4.65
160	159.9	0.06
225	220	2.22
315	315	0.00

La Figura 33 es una gráfica de los ángulos yaw con respecto al valor del porcentaje de error.

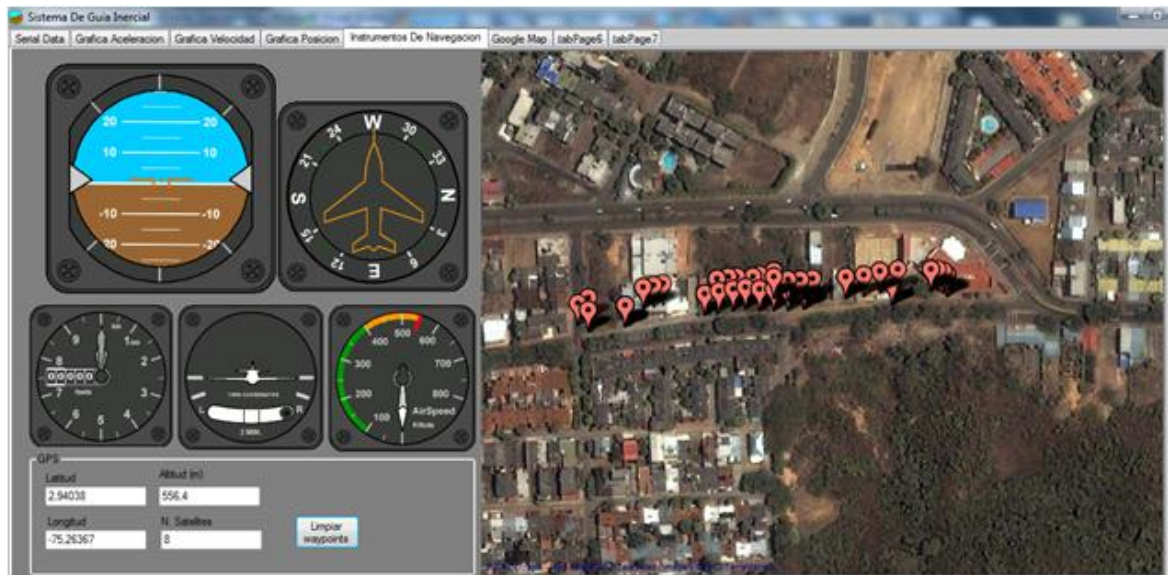
Figura 33. Medida de los ángulos (yaw)



Luego de realizar la comprobación y ajuste por programación a cada uno de los ángulos, se puede afirmar que el sistema de guía inercial entrega resultados muy favorables para las mediciones angulares en cada uno de los ejes, con un mínimo porcentaje de error. Concluyendo esta parte del proyecto como satisfactoria, para la implementación de la instrumentación del UAV.

## 4.1.2 Resultados del GPS

Figura 34. Posición marcada por el GPS MT3329



En la Figura 34 se aprecia el recorrido marcado por el sistema de guía inercial a través del GPS MT3329 a bordo de un automóvil, el recorrido forma una trayectoria cerrada correspondiente a una distancia de 550 metros aproximadamente.

La precisión del GPS también depende de diversos factores como son tener un cielo despejado, el número de satélites al que esté conectado, la velocidad del vehículo teniendo en cuenta que si ésta es muy baja el cálculo del rumbo no será válido, todos estos factores influyen en tener una buena fuente de información satelital.

**4.1.3 Resultados de posición con la unidad de medida inercial.** Antes de dar a conocer los resultados de posición, es de aclarar que la dinámica de los sistemas para el cálculo del desplazamiento varía mucho debido a que todos los sistemas producen diferentes tipos de vibración, debido a esto es muy distinto calcular el desplazamiento con el sistema acoplado a una persona moviéndose, que a un vehículo como un UAV o un automóvil, por tal razón hay que ajustar el sistema de guía inercial dependiendo del móvil al cual se acople, realizar diferentes ensayos hasta conseguir que el sistema trabaje de la mejor manera.

Se realizaron pruebas al sistema de guía inercial para la determinación de la distancia recorrida en un ambiente controlado, estas pruebas se realizaron caminando con el sistema de guía inercial sujeto al cuerpo de una persona y en un lugar cerrado. En los Cuadros 6, 7 y 8 se observa las diferentes distancias recorridas en los ejes x, y, z, respectivamente, y sus correspondientes porcentajes de error.



- Mediciones en el eje x

Cuadro 6. Porcentaje de error para la distancia recorrida en el eje x

Distancia real (metros)	Distancia medida (metros)	Porcentaje de error (%)
8	8.0	0.00
8	8.2	2.44
7	6.5	7.69
6	5.8	3.45
5	4.5	11.11
4	4.5	11.11
42	47.4	11.39

- Mediciones en el eje y

Cuadro 7. Porcentaje de error para la distancia recorrida en el eje y

Distancia real (metros)	Distancia medida (metros)	Porcentaje de error (%)
8	9.8	18.37
8	8.5	5.88
6	6.3	4.76
6	5.1	17.65
5	5.5	9.09

- Mediciones en el eje z

Cuadro 8. Porcentaje de error para la distancia recorrida en el eje z

Distancia real (metros)	Distancia medida (metros)	Porcentaje de error (%)
7	8.2	14.63
7	7.0	0.00
6	5.5	9.09
4	3.9	2.56
4	3.4	17.65

Las gráficas de la aceleración, velocidad y posición que se observan en la interfaz de usuario y que corresponden a una distancia recorrida de 8 metros, se pueden apreciar en las Figuras 35, 36 y 37, respectivamente.

Figura 35. Gráfica de aceleración

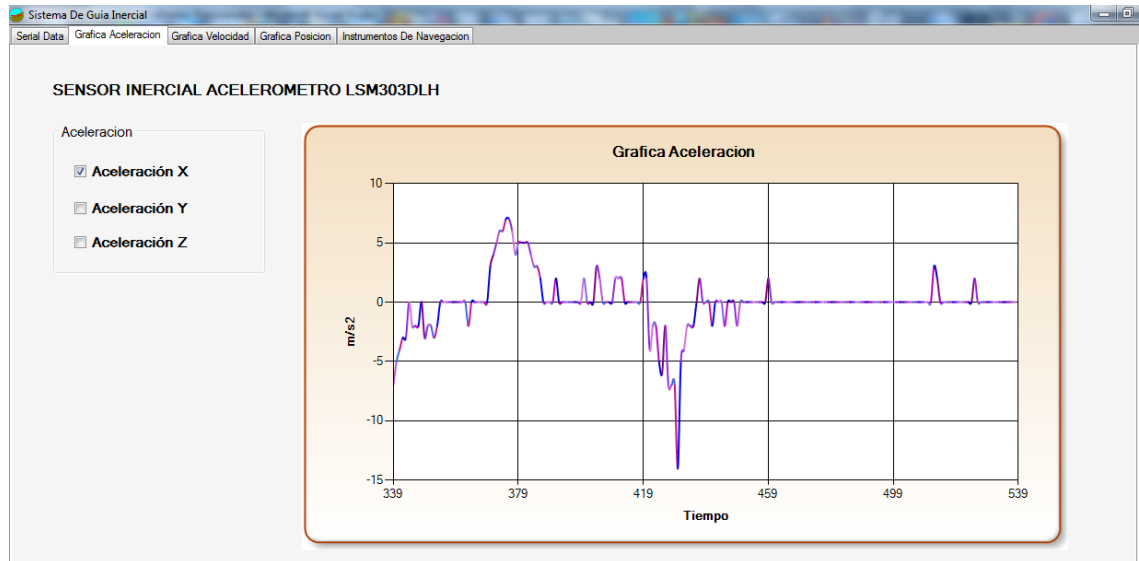


Figura 36. Gráfica de velocidad

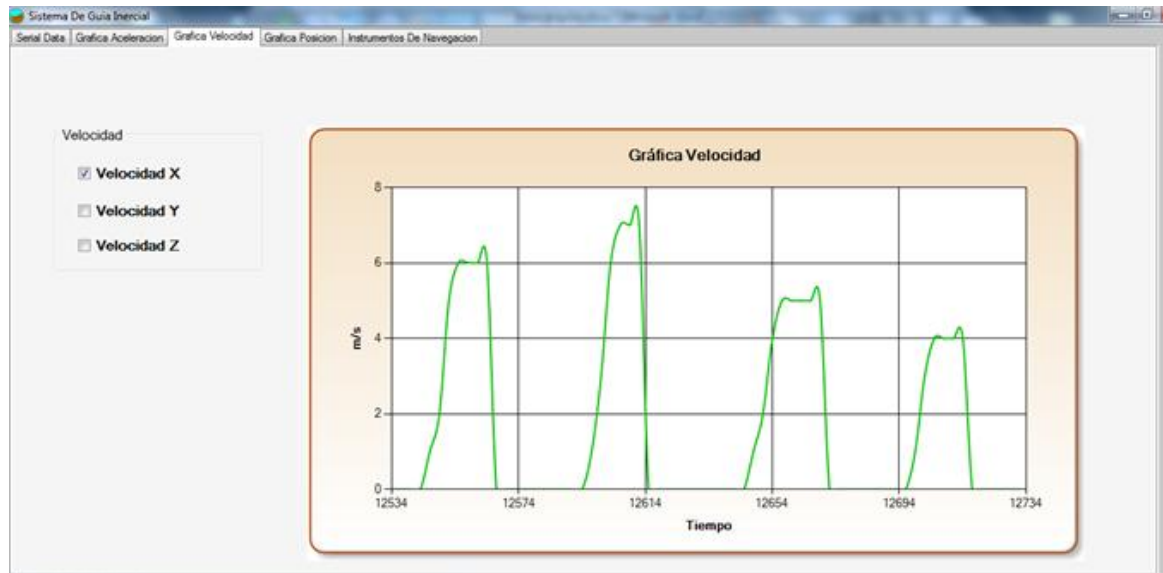
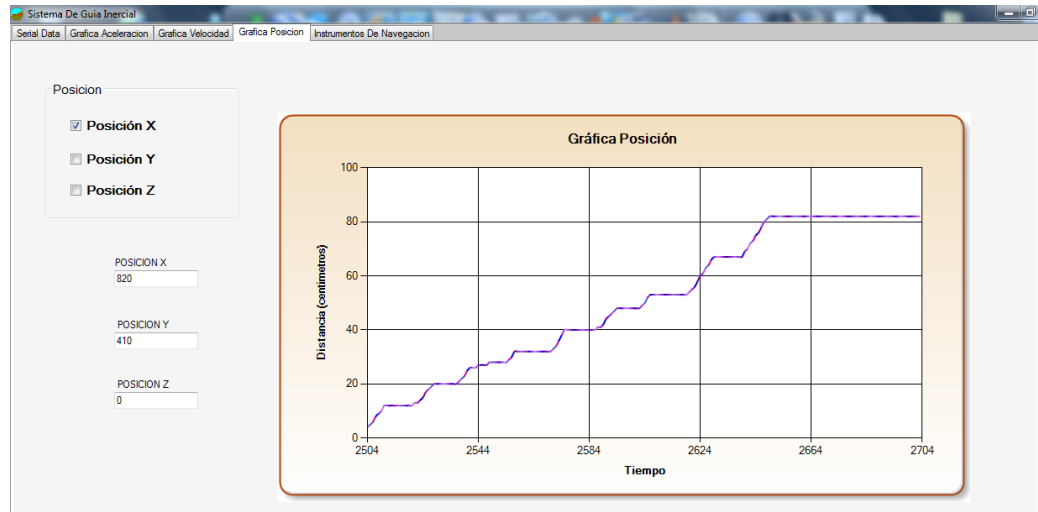


Figura 37. Gráfica de posición



## 4.2 RESULTADOS EN AMBIENTE REAL

Para las pruebas realizadas en ambiente real se ejecutaron varios recorridos con el UAV de manera que se lograra observar el comportamiento del sistema y determinar su correcto funcionamiento.

Los ensayos fueron realizados en un avión entrenador NexSTAR de cuatro canales marca Hobbico como muestra la Figura 38, con las siguientes características: envergadura del ala 1.70 m, longitud del fuselaje 1.40 m, motor glow<sup>6</sup> de 7.5 cc marca Mágnum, las pruebas fueron realizadas en campo en el lugar denominado Huila extrema.

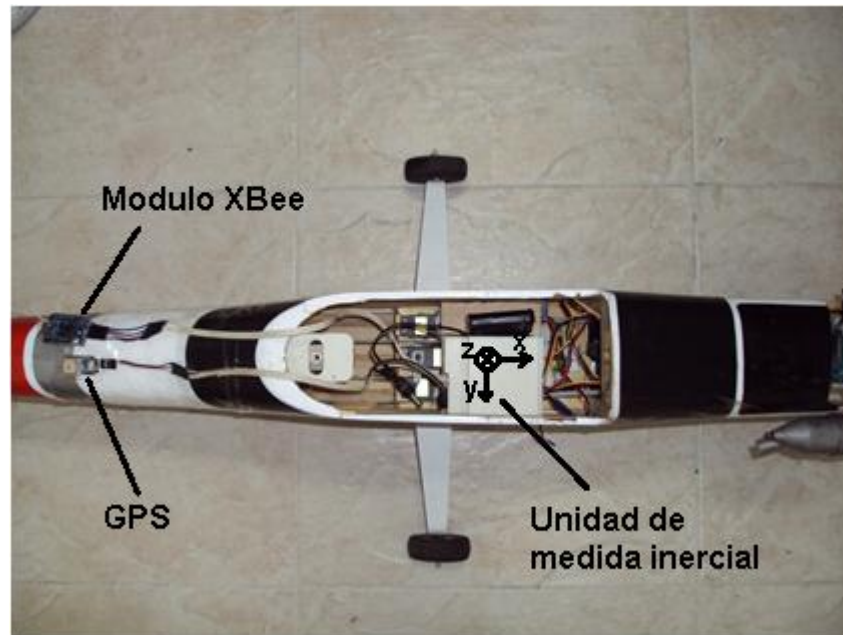
Figura 38. UAV utilizado para probar el sistema de guía inercial



<sup>6</sup> Para más información: [http://es.wikipedia.org/wiki/Motor\\_glow](http://es.wikipedia.org/wiki/Motor_glow)

El sistema fue ajustado al UAV como muestra la Figura 39 de manera que este no interrumpiera el funcionamiento normal del avión.

Figura 39. Sistema de guía inercial acondicionado en el UAV



Debido a que no se cuenta con referencias visuales para medir las diferentes variables en el avión, se realizaron varios recorridos teniendo en cuenta la actitud, desplazamiento y posición con el GPS.

**4.2.1 Actitud.** Se dispuso el UAV en movimientos de ascensos, descensos y giros laterales para comprobar los ángulos *roll*, *pitch* y *yaw*. En el momento del despegue del UAV, se pudo observar el horizonte artificial en la interfaz de usuario, que el ángulo *pitch* aumentaba positivamente hasta alcanzar un valor aproximado de  $45^\circ$  para posteriormente equilibrarse el avión en el aire a un ángulo *pitch* aproximado de  $0^\circ$ ; el ángulo *roll* respondía de acuerdo al movimiento del UAV en vuelo, es decir, cuando éste realizaba un movimiento de su ala derecha por debajo de la línea del horizonte, se marcaba un ángulo *roll* positivo que podía ser observado en el horizonte artificial, así mismo sucede para el ángulo *roll* negativo; para el instrumento indicador de rumbo, se visualizó el ángulo *yaw* respondiendo de acuerdo al movimiento del UAV cuando éste se movía.

**4.2.2 Distancia.** Las pruebas de distancia se realizaron tanto en tierra como en aire para comprobar el correcto funcionamiento de los sensores inerciales, para los ensayos en tierra se dispuso el UAV en un terreno plano para luego poner en marcha el motor trasladando el UAV hasta un punto determinado y posteriormente devolverlo al punto inicial, para la pruebas en el aire se hicieron varios recorridos circulares para comprobar su funcionamiento, para ambos ensayos se ajustó el sistema de manera que éste arrojará los mejores resultados.

**4.2.3 GPS.** Para observar la respuesta entregada por el módulo GPS MT3329, fue necesario contar con un modem de internet, ya que se estaba trabajando con Google maps<sup>7</sup> y éste necesita tener acceso a la red de internet para mostrar sus mapas. En la Figura 40, se observa uno de los recorridos realizados por el UAV, el número de satélites conectados al módulo receptor fue de 10.

Figura 40. Recorrido del UAV entregado por el GPS



---

<sup>7</sup> Servidor de aplicaciones de mapas en la Web, de Google.

## 5. CONCLUSIONES

La realización del presente proyecto ha sido una muy importante fuente de aprendizaje y un repaso de los conocimientos adquiridos durante toda la carrera, en el cual se puso mucho esfuerzo y dedicación, ya que este es un tema muy poco investigado en la región surcolombiana y centro de estudios.

Se diseñó e implementó un sistema de guía inercial para ser utilizado en un vehículo aéreo no tripulado, teniendo en cuenta un tamaño reducido y un bajo peso, debido a que estos factores son muy importantes en su implementación; por otro lado, se espera seguir con la siguiente fase que es implementar el piloto automático del UAV, cabe resaltar que este sistema de guía inercial puede ser utilizado en cualquier tipo de vehículo que requiera conocer su desplazamiento.

En la parte inicial de este proyecto se utilizó el sistema ArduPilot Mega para desarrollar el sistema de guía inercial, pero debido a que los resultados no eran los esperados, se decidió utilizar la unidad de medida inercial minIMU-9 en conjunto con la tarjeta Arduino Uno, obteniendo mejores resultados para los datos entregados por el acelerómetro y consecuentemente para el cálculo de la posición esperada.

La interfaz de usuario diseñada para el presente proyecto de grado, tiene un funcionamiento eficiente que permite ver toda la información requerida de un sistema de navegación inercial, tanto de forma numérica como de forma gráfica, esto por medio de los instrumentos de control del UAV, el código de libre distribución para implementar google maps que permite observar la información del GPS y las gráficas para visualizar los datos de aceleración, velocidad y posición.

Las pruebas realizadas al sistema de guía inercial, arrojaron buenos resultados tanto en aquellas realizadas en un ambiente controlado como las realizadas en un ambiente real, utilizando el vehículo aéreo no tripulado; lo cual indica que el tratamiento realizado a los datos de los sensores inerciales, para obtener los resultados esperados, fueron realizados de una forma correcta y eficiente.

Con el desarrollo del presente proyecto se pretende crear tecnología propia a bajo costo y de calidad, evitando la dependencia extranjera y que sirva para la región en cualquier campo que lo necesite.

## 6. RECOMENDACIONES

El desarrollo del presente proyecto es una plataforma para el inicio de diferentes equipos o instrumentos de navegación como por ejemplo, el desarrollo de un estabilizador de vuelo ya sea para una avión o helicóptero, el cual es muy utilizado para realizar un vuelo estable sin importar las condiciones atmosféricas presentes o la pericia del piloto, también es muy utilizado en un nuevo sistema de vuelo llamado FPV o vuelo en primera persona en donde el UAV realiza vuelos de largo alcance ente 1km - 7km de distancia, y el control de este, solo se basa en la instrumentación presente y la transmisión de video en vivo desde el UAV, otra aplicación utilizada seria en una plataforma de estabilización de video cámara para fotografía o video aéreo, ya que el sistema permite estabilizar y controlar la cámara al objetivo deseado.

Se espera que estos temas sean más investigados y tenidos en cuenta en éste centro de estudios, ya que se espera que los UAV sean próximamente la aviación del futuro, debido a que se logran las mismas funciones que en la aviación comercial a un menor costo y lo mejor, sin arriesgar vidas humanas.

Utilizar diferentes algoritmos para el tratamiento de los datos del acelerómetro y para realizar la fusión de sensores, que permitan obtener resultados más exactos en cuanto a la determinación de la posición del sistema de guía inercial, además de realizar una comparación de los resultados obtenidos mediante la aplicación de estos algoritmos.

Al sistema de guía inercial desarrollado en el presente proyecto de grado, se le puede realizar la adición de sensores como un altímetro o un sensor de velocidad del aire, para hacer que el sistema sea más robusto.

La próxima fase del proyecto será el diseño e implementación del piloto automático en base al sistema realizado en este proyecto.

## BIBLIOGRAFÍA

CERQUERA ROJAS, Yamil Armando. Integración numérica: regla trapezoidal. [en línea].

<<http://www.monografias.com/trabajos-pdf/integracion-numerica/integracion-numerica.pdf>> [citado en 14 de septiembre de 2012]

FERRER MÍNGUEZ, Gonzalo. Integración Kalman de sensores inerciales INS con GPS en un UAV. Trabajo de grado. UPC. 2009. 83 p.

GREWAL, Mohinder; WEILL, Lawrence y ANDREWS, Angus. Global positioning systems, inertial navigation, and integration. 2 ed. Hoboken: John Wiley. 2007. 525 p.

Manual oficial del ArduPilot Mega. [en línea].

<<http://code.google.com/p/ardupilot-mega/>> [citado en 14 de septiembre de 2012]

RIVAS GIL, Miguel. Unidad de medida inercial: algoritmo de estimación e implementación software. Trabajo de grado. Universidad de Sevilla. 2006. 120 p.

ROGERS, Robert M. Applied mathematics in integrated navigation systems. 2 ed. Gainesville: AIAA education series. 2003. 330 p.

SerialPort (RS-232 Serial COM Port) in C# .NET. [en línea].

<[http://msmvps.com/blogs/coad/archive/2005/03/23/SerialPort-\\_2800\\_RS\\_2D00\\_232-Serial-COM-Port\\_2900\\_-in-C\\_2300\\_-NET.aspx](http://msmvps.com/blogs/coad/archive/2005/03/23/SerialPort-_2800_RS_2D00_232-Serial-COM-Port_2900_-in-C_2300_-NET.aspx)> [citado en 14 de septiembre de 2012]

SIOURIS, George M. Aerospace avionics systems: a modern synthesis. San Diego: Academic Press. 1993. 466 p.



## ANEXO A. Configuración de los módulos ZigBee

Los módulos de telemetría vienen en un kit que incluye un convertidor serial USB, permitiendo la conexión al computador, para poder configurarlos, como muestra la Figura 41.

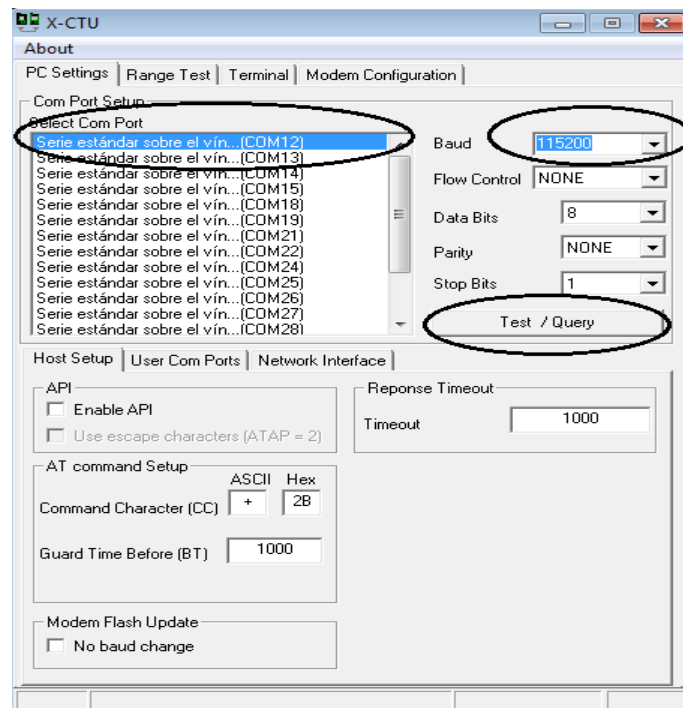
Figura 41. Componentes ZigBee.



Fuente: <http://code.google.com/p/ardupilot-mega/wiki/APM1Wireless>

Una vez conectado el módulo al computador se abre el programa X-CTU, se selecciona el puerto COM asignado al módulo (se puede verificar en el administrador de dispositivos del computador) y su velocidad de transmisión en baudios, originalmente vienen a una velocidad de transmisión de 9600 baudios, luego se da clic en el botón (*Test/Query*), como se indica en la Figura 42.

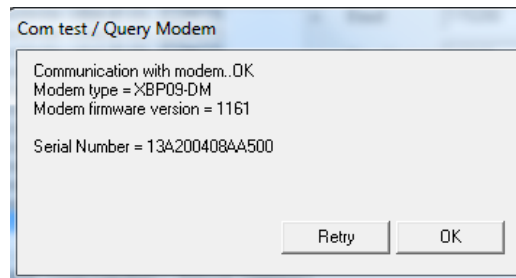
Figura 42. Ventana de configuración del programa X-CTU



Fuente: programa X-CTU programmer, Dig electronics

Si el modulo está correctamente conectado, debe aparecer una ventana como en la Figura 43.

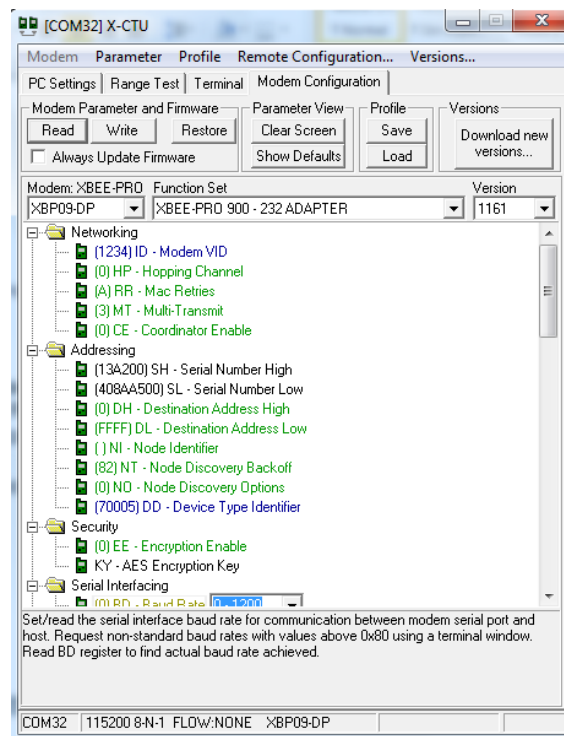
Figura 43. Ventana que indica conexión correcta del modulo XBee



Fuente: programa X-CTU programer, Dig electronics

Luego se procede a entrar a la pestaña de Modem Configuracion y se da clic a la opcion *Read*, de esta forma se hace una lectura de la programacion que el modulo posee, como lo muestra la Figura 44.

Figura 44. Ventana que indica la lectura de la programacion del X-Bee



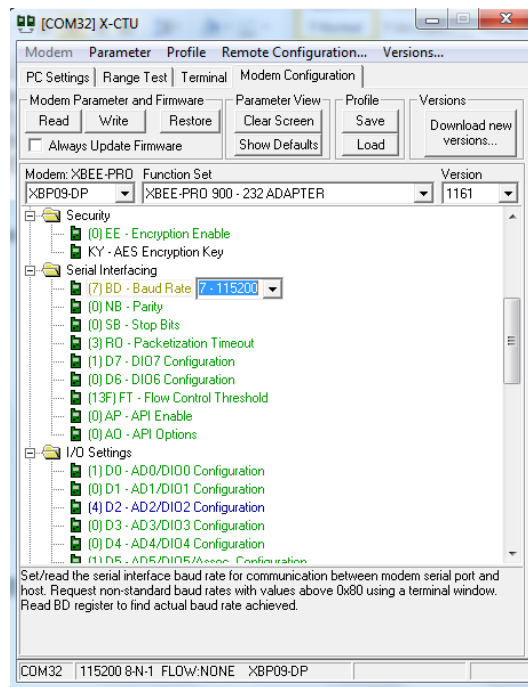
Fuente: programa X-CTU programer, Dig electronics

Luego se pasa a programar el modulo con los siguientes parámetros básicos:

- *Destination Address:* BBBB.
- *Source Address:* AAAA
- *Coordinator Enable:* ON
- *Interface Data Race:* 115200

La dirección de destino es a la cual se desean transmitir los datos, la dirección de fuente será la dirección de origen, la velocidad de transmisión en este caso será 115200 y se habilita la opción de coordinador debido a que este será el dispositivo de control el cual se encargara de recibir los datos enviados desde el módulo END DEVICE o final. El procedimiento se observa en la Figura 45.

Figura 45. Ventana de configuracion del programa X-CTU.



Fuente: programa X-CTU programmer, Dig electronics

Luego de estos pasos se da clic en el botón *Write* para escribir la configuración en el módulo, reprogramándolo con la nueva información. Posteriormente se repite el proceso con el otro modulo, la única diferencia será en la dirección fuente y la dirección destino y en la opción de coordinador se selecciona la opción *END DEVICE*.

## ANEXO B. Microcontrolador ATmega328

Figura 46. Mapeo entre los pines de Arduino y los puertos del ATmega328  
ATmega168/328-Arduino Pin Mapping

Note that this chart is for the DIP-package chip. The Arduino Mini is based upon a smaller physical IC package that includes two extra ADC pins, which are not available in the DIP-package Arduino implementations.

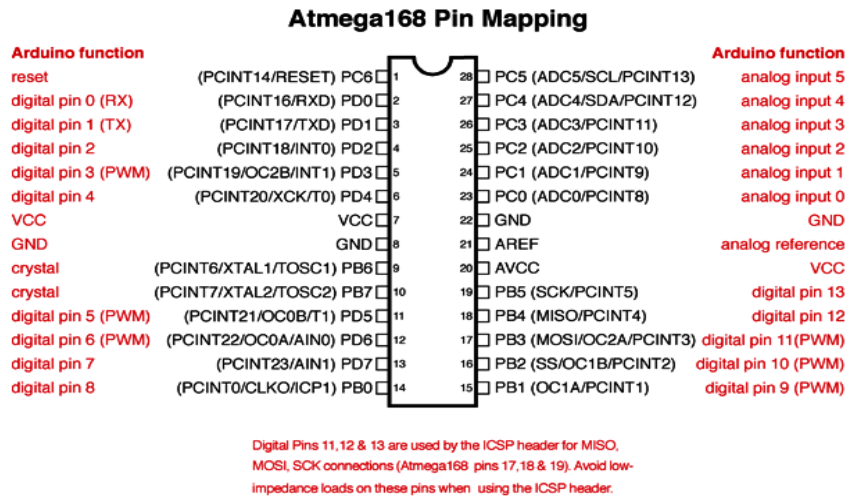
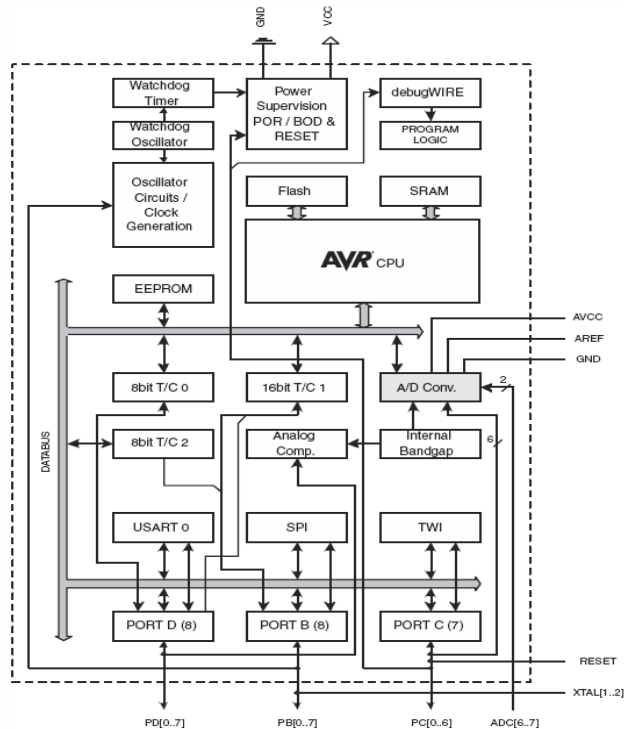


Figura 47. Diagrama de bloques del microcontrolador Atmega328  
Block Diagram



## ANEXO C. Ecuaciones para el cálculo de la actitud

Cambio de marco de referencia de body a tierra, actualización de las medidas del giroscopio:

$$R(t + dt) = R(t) \begin{bmatrix} 1 & -d\theta_z & d\theta_y \\ d\theta_z & 1 & -d\theta_x \\ -d\theta_y & d\theta_x & 1 \end{bmatrix}$$

Donde,

$$d\theta_x = w_x dt$$

$$d\theta_y = w_y dt$$

$$d\theta_z = w_z dt$$

$$w_x, w_y, w_z = \text{Vector velocidad angular}$$

- Normalización de las medidas del giroscopio:

Primero se calcula el producto escalar de las filas X e Y de la matriz de las medidas del giroscopio, que se supone que es cero por ser vectores ortogonales.

$$\mathbf{X} = \begin{bmatrix} r_{xx} \\ r_{xy} \\ r_{xz} \end{bmatrix} \quad \mathbf{Y} = \begin{bmatrix} r_{yx} \\ r_{yy} \\ r_{yz} \end{bmatrix}$$

$$\text{error} = \mathbf{X} \cdot \mathbf{Y} = \mathbf{X}^T \mathbf{Y} = \begin{bmatrix} r_{xx} & r_{xy} & r_{xz} \end{bmatrix} \begin{bmatrix} r_{yx} \\ r_{yy} \\ r_{yz} \end{bmatrix}$$

$$\begin{bmatrix} r_{xx} \\ r_{xy} \\ r_{xz} \end{bmatrix}_{\text{orthogonal}} = \mathbf{X}_{\text{orthogonal}} = \mathbf{X} - \frac{\text{error}}{2} \mathbf{Y}$$

$$\begin{bmatrix} r_{yx} \\ r_{yy} \\ r_{yz} \end{bmatrix}_{\text{orthogonal}} = \mathbf{Y}_{\text{orthogonal}} = \mathbf{Y} - \frac{\text{error}}{2} \mathbf{X}$$

El siguiente paso es ajustar la fila de la matriz Z que es ortogonal a la X e Y fila. La forma de hacerlo es simplemente hacer el producto cruz de la X y la Y filas:

$$\begin{bmatrix} r_{xx} \\ r_{xy} \\ r_{xz} \end{bmatrix}_{\text{orthogonal}} = \mathbf{Z}_{\text{orthogonal}} = \mathbf{X}_{\text{orthogonal}} \times \mathbf{Y}_{\text{orthogonal}}$$

El último paso en el proceso de renormalización es ampliar las filas de la matriz R Ecuación 2 para asegurar que cada uno tiene una magnitud igual a uno

$$\mathbf{X}_{normalized} = \frac{1}{2}(3 - \mathbf{X}_{orthogonal} \cdot \mathbf{X}_{orthogonal})\mathbf{X}_{orthogonal}$$

$$\mathbf{Y}_{normalized} = \frac{1}{2}(3 - \mathbf{Y}_{orthogonal} \cdot \mathbf{Y}_{orthogonal})\mathbf{Y}_{orthogonal}$$

$$\mathbf{Z}_{normalized} = \frac{1}{2}(3 - \mathbf{Z}_{orthogonal} \cdot \mathbf{Z}_{orthogonal})\mathbf{Z}_{orthogonal}$$

Corrección de la deriva del giroscopio para los ángulos de Euler *roll*, *pitch* y *yaw* con el magnetómetro

$$COGX = \cos(cog)$$

$$COGY = \sin(cog)$$

Dónde:

$$cog = \arctan\left(\frac{-Mag(y)}{Mag(x)}\right)$$

*Mag* = medida del magnetometro en cada eje

A continuación, se calcula la corrección del ángulo yaw:

$$YawCorrectionGround = r_{xx}COGY - r_{yx}COGX$$

Se obtiene la corrección de guiñada en el marco de referencia de tierra pero se necesita conocer la corrección del vector en la aeronave (cuerpo)

$$\mathbf{YawCorrectionPlane} = YawCorrectionGround \begin{bmatrix} r_{zx} \\ r_{zy} \\ r_{zz} \end{bmatrix}$$

Se puede combinar con la corrección roll-pitch que se utiliza para compensar la deriva.