

Universidad Central “Marta Abreu” de Las Villas

Facultad de Ingeniería Eléctrica

Departamento de Automática y Sistemas Computacionales



TRABAJO DE DIPLOMA

**Arquitectura de hardware de bajo costo para un
vehículo autónomo sumergible**

Autor: Alejandro Sánchez Sánchez

Tutor: Dr.C. Luis Hernández Santana

Ing. Hector Daniel Alvarez Pérez

Santa Clara

2016

"Año 58 de la Revolución"

Universidad Central “Marta Abreu” de Las Villas

Facultad de Ingeniería Eléctrica

Departamento de Automática y Sistemas Computacionales



TRABAJO DE DIPLOMA

Arquitectura de hardware de bajo costo para un vehículo autónomo sumergible

Autor: Alejandro Sánchez Sánchez

email: alejandross@uclv.cu

Tutor: Dr.C. Luis Hernández Santana

Dpto. de Automática, Facultad de Ing. Eléctrica, UCLV

email: luishs@uclv.edu.cu

Ing. Hector Daniel Alvarez Pérez.

Dpto. de Automática, Facultad de Ing. Eléctrica, UCLV

email: hectorap@uclv.edu.cu

Santa Clara

2016

"Año 58 de la Revolución"

PENSAMIENTO

*Cuando se innova, se corre el riesgo de cometer errores. Es mejor admitirlo
rápidamente y continuar con otra innovación.*

Steve Jobs.

DEDICATORIA

*A mis padres Adis y Raúl por brindarme su confianza,
apoyo y amor durante toda mi vida.*

*A mi hermana Rachel que espero ser su guía siempre aunque
no esté a mi lado.*

*Al resto de mi familia que de una manera u otra han
estado involucrados en el logro de este trabajo.*

AGRADECIMIENTOS

A mis padres que con su apoyo y paciencia he logrado realizar este trabajo y que han estado a mi lado a pesar de las distancias.

A Suanny que ha estado a mi lado durante toda la realización de este trabajo brindándome su amor y comprensión.

A los profesores del departamento y de la facultad que me han formado como profesional y que de una forma u otra me han sabido aconsejar.

A mi abuela mama que siempre se ha preocupado por mí y ha estado presente siempre que lo he necesitado.

A mi primo Pedro Julio por servirme de guía durante toda mi vida estudiantil.

A mis tutores por confiar en mí y ayudarme con la realización del trabajo.

A mis amigos Ariel, Vela, Navas, Giovanni, Damián, Tony, Yudaisy, demás compañeros de grupo y amistades cercanas por los momentos que vivimos en estos cinco años.

A mi hermano de universidad Ricardo por todos los consejos brindados.

A Dios por ser la luz que me ha guiado y protegido en mi vida.

RESUMEN

Los vehículos submarinos autónomos son objeto de investigación y utilización a nivel mundial debido a la gran variedad de aplicaciones que poseen. El CIDNAV, en nuestro país, ha mostrado gran interés en el tema, del cual se han realizado investigaciones previas como el HRC-AUV. En la actualidad la arquitectura del sistema es costosa, por lo que el objetivo de este trabajo es proponer una solución de piloto automático de bajo costo basado en Pixhawk para un vehículo submarino autónomo. En esta investigación se analizan diferentes arquitectura de hardware y software, reportadas en la literatura especializada, seguidamente se realiza un análisis de la factibilidad del uso de la plataforma Pixhawk, basados en esa información se presenta la propuesta de la nueva arquitectura de hardware para el sistema, la cual se basada en el aprovechamiento de las características y prestaciones del Pixhawk en combinación con un hardware de bajo nivel basado en microcontroladores DsPIC. También se define la estructura del software a implementar.

La arquitectura de hardware y software empleada en este trabajo fue validada mediante simulaciones en el software Proteus y la programación del microcontrolador se realizó en el software MPLAB.

TABLA DE CONTENIDOS

PENSAMIENTO	i
DEDICATORIA	ii
AGRADECIMIENTOS	iii
RESUMEN	iv
INTRODUCCIÓN	1
Organización del informe	4
ESTUDIO DEL DESARROLLO DE LOS AUTOPILOTOS COMERCIALES PARA VEHÍCULOS AUTÓNOMOS.	6
1.1 Desarrollo Actual de los AUV.	6
1.1.1 Aplicaciones.....	7
1.2 Arquitectura general de un AUV.	7
1.2.1 Estructura sensorial.....	8
1.3 Requerimientos.	8
1.4 Arquitectura general para el HRC-AUV.	8
1.5 Pilotos automáticos comerciales.	10
1.5.1 Lockheed Martin.....	10
1.5.2 Micropilot.	11
1.5.3 3D robotics.....	12
1.5.4 Cloud Cap Technology.	13
1.6 Definición de la arquitectura de hardware.	13
1.6.1 Pixhawk.	13
ARQUITECTURA DE HARDWARE DEL SISTEMA A BORDO.	16
2.1 Requerimientos del sistema.	16
2.2 Procesador de la interfaz de bajo nivel.	17

2.2.1 Características de los DsPIC33FJxxMCx04.....	18
2.2.2 Estructura general del hardware.	19
2.3 Interfaces de comunicación.	20
2.3.1 Interfaz SPI.	20
2.3.2 Interfaz UART.	22
2.4 Señal entrada-salida.	23
2.4.1 Acondicionamiento de entradas analógicas.....	23
2.4.2 Salidas analógicas.	24
2.5 Sensores.	25
2.5.1 Presión barométrica.	25
2.5.2 Nivel de baterías.	26
2.5.3 Temperatura.	26
2.5.4 Presencia de agua.....	27
2.5.5 Medición de las RPM del motor de propulsión.	27
2.6 Fuente de alimentación.	28
2.7 Consideraciones finales del capítulo.....	30
SOFTWARE DEL SISTEMA A BORDO.	31
3.1 Software de Bajo nivel.....	31
3.2 Planificación de tareas.....	32
3.3 Inicialización.	33
3.4 Comunicaciones.	34
3.4.1 Comunicación UART.	34
3.4.2 Comunicación SPI.	36
3.5 Adquisición.	38

3.6	Análisis económico.....	39
3.7	Consideraciones finales del capítulo.....	40
CONCLUSIONES Y RECOMENDACIONES		41
	Conclusiones.....	41
	Recomendaciones	42
REFERENCIAS BIBLIOGRÁFICAS		43
ANEXOS		46
	Anexo I Convertidor DAC.....	46
	Anexo II Transmisión SPI.	46
	Anexo III Transmisión de datos mediante el módulo UART.....	47

INTRODUCCIÓN

Desde todos los tiempos el océano ha significado un gran desafío para el hombre, el mismo posee incontables misterios en lugares a los que el ser humano no sería capaz de acceder sin poner en riesgo su vida; a su vez también es un medio rico en recursos bióticos y minerales. La necesidad de explotarlo se ha convertido en un reto y es por ello que la robótica submarina ha ido alcanzando grandes avances tecnológicos a lo largo de la historia de su desarrollo.

Los Vehículos Autónomos Sumergibles (AUV) son máquinas extraordinarias que han revolucionado el proceso de la obtención de datos del mar. Sus mayores progresos fueron resultado del desarrollo exitoso de tecnologías complementarias para vencer los desafíos asociados con las operaciones autónomas en ambientes hostiles.

Un vehículo autónomo submarino (AUV), es un submarino que porta su propia fuente de energía y medios de cómputo, ejecutando software y soluciones de control que le permiten la ejecución de una misión sin intervención humana. Esta misión puede consistir en una secuencia de instrucciones pre-programadas potencialmente modificables en línea, dependiendo de los datos obtenidos por los sensores a bordo. Los mismos se basan en dos tipos de vehículo: vehículos tipo crucero, y vehículos tipo hover. Los primeros son usados en misiones de “sobre vuelo” durante las cuales la meta fundamental consiste en obtener información con los sensores a bordo (Guerra, 2010).

Durante las últimas décadas, los AUVs han sufrido desarrollos notables. Al final de la década de los 80 y principio de los 90, el primer prototipo requirió un tremendo esfuerzo y complejas soluciones de ingeniería, para compensar las limitaciones tecnológicas en términos de potencia de cómputo, baterías y sensores de navegación. Desplegar estos costosos vehículos de navegación autónoma en un ambiente implacable y esperar que regresaran sin peligro

alguno se convirtió en un acto de fe en la ingeniería, una versión adaptada de los esfuerzos tempranos en la tecnología del espacio.

El desarrollo inicial continuó gradualmente y, al final del siglo pasado, los AUVs se fueron moviendo gradualmente del ambiente académico controlado a escenarios de desafío operacional, capa científica, comercial y aplicaciones militares. A medida que fue madurando la tecnología, fueron eficazmente logrados un gran número de diferentes soluciones, en varios tamaños y configuraciones, y algunos envueltos en aplicaciones comerciales.

La robótica submarina es un campo peculiar del conocimiento, trayendo consigo conocimientos complementarios en ingeniería mecánica y eléctrica, así como en ciencias computacionales. En la última década, con las mejoras grandiosas en cuanto al poder computacional, tecnología de baterías y la miniaturización de los sistemas electrónicos, los AUVs se han convertido en menos voluminosos y más receptivos para ser usados como bases de prueba para nuevas técnicas de procesamiento de datos. A medida que los equipos más pequeños, ligeros y menos costosos fueron disponibles, el acceso a los vehículos operacionales fue facilitado y muchos más prototipos fueron accesibles para las pruebas de nuevos algoritmos y soluciones. Esto ha resultado un incremento exponencial en el desarrollo y despliegue de los AUVs, los cuales se han convertido en una herramienta común para todas las comunidades envueltas en el muestreo oceánico (Cruz, 2011).

La mayoría de los avances en las capacidades de los AUV apuntan a alcanzar nuevas aplicaciones y el decremento del costo de la colección de datos del océano, reduciendo el tiempo de embarcación y automatizando el proceso de obtención de datos con geolocalización exacta. Aunque esto produjo mejoras importantes en la eficiencia, los nuevos enfoques también fueron previstos para una utilización más productiva de esta herramienta. Con las presentes capacidades, nuevos paradigmas están siendo empleados para explotar la inteligencia a bordo, tomando decisiones en línea basadas en la interpretación en tiempo real de los datos de los sensores (Ruiz, 2010).

El diseño de Vehículos Autónomos Sumergibles es gobernado por una relación compleja entre los requerimientos críticos de las misiones planificadas y las principales restricciones en la fabricación, ensamblaje y logística operacional. Contrario a la tendencia temprana de desarrollar vehículos de propósito general, la persecución en curso de la eficiencia ha

desplazado este concepto por el de vehículos específicos para tareas específicas, siendo aprovechados frecuentemente diseños modulares para acelerar el tiempo de ensamblaje.

En nuestro país investigadores de la Universidad Central de las Villas (UCLV) y el Centro de Investigación y Desarrollo Naval (CIDNAV) han unido fuerzas en el desarrollo del proyecto HRC-AUV. Un proyecto para la implementación de un AUV capaz de operar en la plataforma marina cubana en tareas de exploración y supervisión (Guerra, 2010). La arquitectura diseñada en este proyecto posee un alto valor económico debido a los componentes utilizados en la misma y además el software implementado es de código propietario, por esta razón se tiene como objetivo el desarrollo de un autopiloto de bajo costo basado en Pixhawk para el sistema.

Pixhawk es un sistema de piloto automático avanzado, diseñado por el proyecto PX4-hardware abierto y fabricado por 3D Robotic. Cuenta con procesador ARM de 32 bit y tecnología de sensores ST Microelectronics, y un sistema operativo de tiempo real NuttX, que ofrece un rendimiento increíble, flexibilidad y fiabilidad, por lo que este autopiloto ofrece la posibilidad de controlar cualquier vehículo autónomo (Aerorecord, 2014). Además posee una amplia comunidad de desarrolladores elevando cada vez más su funcionalidad. Existen una gran variedad de aplicaciones que utilizan Pixhawk, entre estas: aviones, helicópteros y rovers; sin embargo este autopiloto no cuenta con un Firmware para controlar un vehículo sumergible. Por lo que se presenta el siguiente problema científico:

La plataforma Pixhawk posee incompatibilidad en cuanto a las entradas y salidas necesarias para lograr el control autónomo del HRC-AUV desarrollado por el CIDNAV.

Para resolver este problema se plantea la siguiente hipótesis:

Con la incorporación de un hardware de bajo nivel que sea capaz de comunicar la plataforma de alto nivel Pixhawk con los sensores y actuadores del HRC-AUV se puede lograr el desarrollo de un autopiloto para el control del mismo.

Por consiguiente los objetivos propuestos para este trabajo son:

Objetivo general:

- Proponer una solución de piloto automático de bajo costo para vehículos subacuáticos

Objetivos específicos:

- Realizar un estudio bibliográfico referente al desarrollo de autopilotos comerciales y la arquitectura del HRC-AUV.
- Realizar estudio de factibilidad de la implementación de la plataforma PIXHAWK en vehículos subacuáticos.
- Diseñar una interfaz de bajo nivel basada en microcontroladores DsPIC que cumpla con las exigencias de E/S del hardware PIXHAWK y el HRC-AUV.
- Realizar la programación adecuada del microcontrolador para lograr un flujo de datos coherentes entre el PIXHAWK y el HRC-AUV

El resultado que se pretende alcanzar con este trabajo es contribuir al desarrollo de los vehículos subacuáticos y mediante la demostración de la compatibilidad de la plataforma PIXHAWK en los mismos, el diseño de la arquitectura de hardware correspondiente y la adecuada programación de los microcontroladores de la misma se pueden obtener soluciones económicamente factibles. Con la implementación de esta herramienta se propone continuar con el desarrollo de la misma en función de las aplicaciones deseadas.

Con la ejecución del proyecto, se dan soluciones a problemáticas como la adquisición del hardware existente para el HRC-AUV, el cual posee un alto valor económico, mientras que con la nueva arquitectura se alcanzaría un gran ahorro económico para el país, debido a que sus componentes (PIXHAWK, incluyendo sensores inerciales baratos y periféricos) están más asequibles que los actualmente instalados (PC-104, IMU, radio módems). También se establece un criterio para, aplicar o no, la arquitectura diseñada a los prototipos de AUV existentes, o proyectos en curso, para así obtener mayor rendimiento funcional y mayor factibilidad en cuanto a componentes y recursos económicos.

Organización del informe

La tesis, posterior a esta introducción, incluye tres capítulos. Los contenidos de los capítulos, en forma resumida, son los siguientes:

Capítulo 1

En el primer capítulo se realiza un estudio de la literatura que se ha consultado, así como la arquitectura actual del HRC-AUV. Se lleva cabo la investigación acerca de los autopilotos comerciales, así como sus principales características. Se plantea la panorámica general existente en torno al problema que se aborda y se define una propuesta para la arquitectura a diseñar.

Capítulo 2

En el capítulo 2 se analiza lo referente a la arquitectura de hardware de bajo nivel seleccionada para el AUV. Primeramente se hace un estudio de los principales requerimientos del sistema en cuanto a E/S y posteriormente se realiza el diseño del hardware teniendo en cuenta criterios bibliográficos.

Capítulo 3

En el capítulo 3 se analiza el software de bajo nivel a implementar así como las principales características del mismo, además se brinda una descripción de sus principales funciones y criterios tomados en cuenta a la hora de la implementación. También se realiza un análisis económico para el presente trabajo.

Finalmente son presentadas las conclusiones y recomendaciones generales del trabajo, seguidas de los anexos.

Capítulo 1

ESTUDIO DEL DESARROLLO DE LOS AUTOPILOTOS COMERCIALES PARA VEHÍCULOS AUTÓNOMOS.

1.1 Desarrollo Actual de los AUV.

Al pasar de los años el desarrollo de los vehículos subacuáticos ha alcanzado un gran auge a nivel mundial. Con el tiempo se ha vuelto imprescindible para el hombre la exploración de un medio rico en recursos el cual no conoce lo suficiente y a su vez es de difícil acceso. Por este y otros motivos, el desarrollo de la tecnología de navegación en estas aplicaciones ha avanzado considerablemente, muchos problemas han surgido, algunos ya se han resuelto, otros están en desarrollo de soluciones y se supone que otros problemas saldrán (Ruiz, 2010).

Existen un gran número de universidades y centros de investigación que desarrollan la temática de vehículos sumergibles autónomos y que cada año perfeccionan y enriquecen el proceso de automatización para estos vehículos, como son “Universidad Nacional del Sur” (UNS) en Argentina, “Universidad BEIHANG”, China, “Universidad de Zagreb” en Croacia, así como el “Instituto de Problemas Tecnológicos Marinos de la Academia Rusa de Ciencias del Lejano Oriente” (IMTP FEB RAS) (Martinez, 2012). Entre la gran cantidad de proyectos desarrollados podemos citar como ejemplos el AUV NRDE desarrollado por Norwegian Defence Research Establishment (NDRE) (Ruiz, 2010), el AUV Hugin, sucesor del AUV NRDE, es desarrollado por Kongsberg Maritime y por el Forsvarets Forsknings Institute (FFI) de Noruega, el AUV Ocean Voyager II, el Solar AUV, el URASHIMA" AUV completado en 2000 por “Japan Agency for Marine-Earth Science and Technology” (JAMSTEC) (Wakita et al., 2010), entre otros.

AUV URASHIMA L 10.7 m × W1.3 m × H1.5 m



AUV Remus 100



Figura 1.1 Ejemplos de AUVs.

1.1.1 Aplicaciones

Aplicaciones Industriales:

- Supervisión de tuberías.
- Mantenimiento de plataformas petrolíferas.
- Inmersión en aguas poco profundas.
- Comunicaciones submarinas, instalación e inspección de cables.

Aplicaciones científicas:

- Monitoreo de volcanes marinos.
- Seguimiento de peces y de grandes mamíferos.
- Recolección de datos sobre corrientes oceánicas.

Aplicaciones militares:

- Exploración.
- Detección y mapeo de campos minados.

1.2 Arquitectura general de un AUV.

Para un AUV es de gran necesidad poseer una arquitectura computacional que sea capaz de cumplir las demandas de operación de la aplicación, las cuales son: medios de computo con funciones de navegación, control, adquisición y comunicación; cámaras, sensores, entre otros. A medida que transcurren los años, los sistemas empotrados que portan estas características se han fabricado de manera más compacta y a su vez se han utilizado sistemas operativos más óptimos.

1.2.1 Estructura sensorial.

La estructura sensorial de un AUV es la que permite conocer el estado del vehículo así como el de su entorno. Entre los sensores más utilizados en este tipo de aplicación se encuentran: acelerómetros, giróscopos, magnetómetros, altímetros, modem acústico, sonar, sensor de presión, profundidad, temperatura, velocidad de traslación y rotación, GPS (Global Positional System) entre otros.

En la actualidad se emplean sensores de muchas prestaciones como los llamados IMU (Inertial Measurement Unit), los cuales por lo general presentan internamente variados tipos de sensores, entre los que se encuentran acelerómetros, giróscopos, magnetómetros, temperatura, presión y GPS (Guerra, 2010).

1.3 Requerimientos.

Tras analizar la estructura de un AUV se puede llegar a la conclusión de que estos sistemas poseen requerimientos para su funcionamiento, como son:

- Sistema empotrado.
- Operación en tiempo real.
- Interacción multisensorial.
- Capacidad de cómputo media o elevada.

1.4 Arquitectura general para el HRC-AUV.

El colectivo del Grupo de Automatización, Robótica y Percepción (GARP), adoptó una estructura general para el proceso de creación del autopiloto para el HRC-AUV. En la figura 1.2 se muestra la estructura general del sistema.

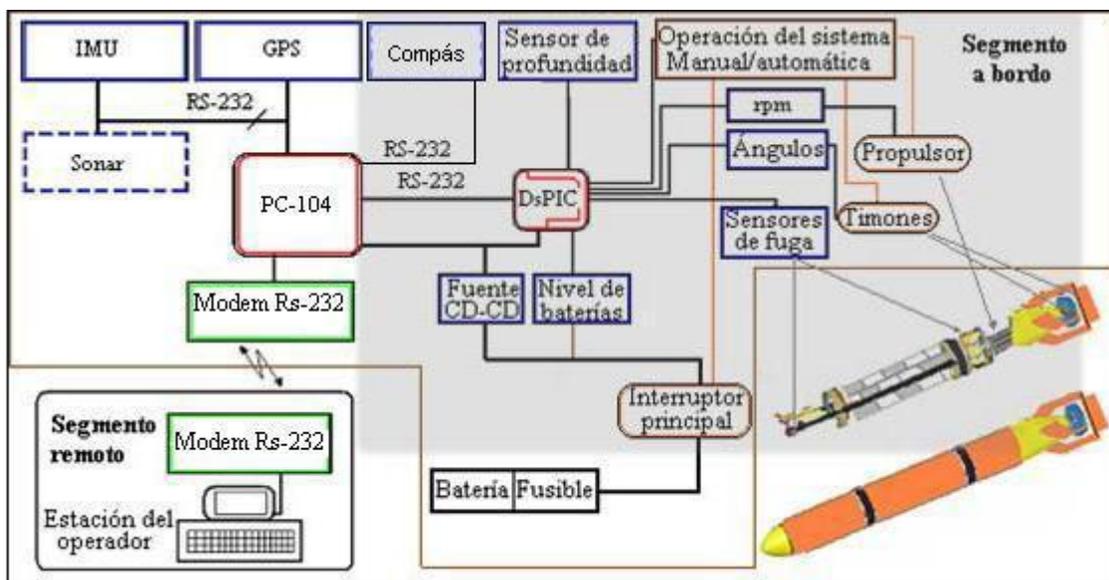


Figura 1.2 Estructura actual del sistema.

En trabajos previos se implementó la arquitectura de hardware del segmento a bordo, la cual está compuesta por dos unidades de cómputo y una unidad de potencia. Las unidades de cómputo son: una computadora industrial PC-104 y un sistema empotrado basado en los Microcontroladores DsPIC 30F4013 y DsPIC 30F3013 de Microchip (Díaz, 2012).

La unidad DsPIC es una de las unidades de cómputo del sistema. Su tarea es la relacionada con todas las mediciones tanto analógicas como digitales adquiridas desde los sensores. Esta unidad también es la encargada del control de rumbo y profundidad del AUV. Por otra parte la PC-104 es la unidad en donde se realizan todas las mediciones digitales relacionadas con: estimación de posición y actitud a partir de la relación INS/GPS. Esta unidad también ejecuta el software de navegación y comunicaciones (Guerra, 2010). La unidad de potencia está compuesta por un bloque DC-DC, el cual suministra todos los voltajes requeridos por los sensores y componentes del AUV y un driver que maneja los ángulos en los timones.

Con la arquitectura de hardware implementada se cumplen perfectamente todos los requerimientos del HRC-AUV, no obstante, todos sus componentes presentan un alto valor económico, es una arquitectura cuyo software es de código propietario y además posee una potencia de cómputo que excede la necesaria. Es una arquitectura en la que no es económicamente factible su diseño y montaje en el país, debido al escaso desarrollo de la industria electrónica. Además en el mundo, en la actualidad, existen numerosos

desarrolladores de hardware propietario y libre que brindan soluciones de alto desempeño y calidad, las cuales pueden ser utilizadas en numerosas aplicaciones con el mínimo de adaptaciones a los requerimientos propios. Por estas razones es necesario realizar un estudio de opciones de autopilotos que sean más factibles económicamente y que a su vez su implementación cumpla con todos los requerimientos desde el punto de vista de hardware y software para el correcto funcionamiento del AUV, el cual es el objetivo del presente trabajo y se describirá en el transcurso del mismo.

1.5 Pilotos automáticos comerciales.

Como se expuso en el epígrafe anterior es necesario realizar una revisión de los pilotos automáticos así como un estudio de los productos comerciales que se puedan encontrar actualmente en el mercado, para esto se realiza una comparación entre las diferentes arquitecturas a nivel de requerimientos básicos, a nivel de sensores, capacidad de procesamiento, tamaño, peso, relación E/S y precios. Para esta investigación se identificaron 4 Empresas productoras de estos instrumentos, se trata de Lockheed Martin, Micropilot, 3D robotics y Cloud Cap Technology.

1.5.1 Lockheed Martin.

1- Kestrel autopilot V2.4

Este piloto automático de software propietario cuenta con un microprocesador Rabbit 3000 que funciona a 29 MHz con una capacidad de 512 K de memoria RAM y Flash. En cuanto la IMU, este piloto automático cuenta con un giróscopo cuyo rango de medición máximo es de $\pm 300^\circ/\text{seg}$ con una resolución $0.0318^\circ/\text{LSB}$. Los acelerómetros de este piloto automático pueden medir hasta $\pm 10g$ y cuentan con una resolución de $0.0015g/\text{LSB}$, ambos componentes funcionan con un voltaje de 5V. Este piloto automático cuenta con un sensor de velocidad del aire que mide hasta 466.7Kph, además, el sensor de altitud tiene un rango máximo de 7Km. El tamaño total del piloto automático es de 52.654mm x 34.925mm x 11.98mm y tiene un peso de 16.65 gramos, además cuenta con 4 puertos para conexión servos que puede ser expandible a 12 puertos. Finalmente el precio del Kestrel autopilot V2.4 es de 5000 USD. Esta información fue extraída de los manuales de especificación del producto (Campo, 2014).

2- Kestrel autopilot V3.0

Este piloto automático cuenta con un procesador que opera a 500MHz y tiene una memoria flash y RAM de 32Mb, cuenta con una tarjeta IMU con compensación de temperatura para evitar mediciones erradas dadas por temperaturas anómalas. Tiene 11 puertos para conexión de servomotores y puede ser expandible a 19 puertos según el propósito que se tenga. Este piloto automático es de código cerrado tiene un peso de 24 gramos, además tiene las siguientes dimensiones 57.404mm x 37.084mm x 17.018mm y un precio de 8500 USD (Campo, 2014).

1.5.2 Micropilot.

La información que se muestra se encuentra en las especificaciones de los productos de la página web de esta compañía. Los siguientes autopilotos son de código cerrado.

1- MP 2028g

Este piloto automático cuenta con un acelerómetro que permite medir hasta 2g, posee un giróscopo de 3 ejes cuya tasa angular máxima es 300°/seg, en cuanto al sensor de velocidad de aire, la medida máxima que proporciona el mismo es de 500Kph. El puerto de altitud de este piloto automático puede medir hasta 12km de altura, en cuanto al GPS, la actualización de datos que tiene este es de 1Hz. Por otro lado, esta arquitectura cuenta con 8 puertos para servos, expandible a 16 o a 24 según los requerimientos del usuario, la actualización de datos que tienen los servos es de 50Hz, por último, el precio de este producto es de 3500 USD (Campo, 2014).

2- MP 2128g

Este piloto automático posee con un acelerómetro que permite la medición de hasta 5g, cuenta con un giróscopo de 3 ejes cuya tasa angular máxima es 300°/seg. En cuanto al sensor de velocidad de aire, la medida máxima que proporciona el mismo es de 500Kph. El puerto de altitud de este piloto automático puede medir hasta 12km de altura, en cuanto al GPS, la actualización de datos que tiene este es de 4Hz. Por otro lado, esta arquitectura cuenta con 8 puertos para servos, expandible a 16 o a 24 según los requerimientos del usuario, la actualización de datos que tienen los servos es de 50Hz. Las dimensiones de este piloto automático son de 100mm x 40mm x 15mm y tiene un peso de 24 gramos, por último, el precio de este producto es de 6000 USD (Campo, 2014).

1.5.3 3D robotics.

1- PX4 FMU

Cuenta con un procesador llamado Cortex-M4 de 68MHz, una memoria flash 1024KiB y una memoria RAM de 192KiB. La referencia del giróscopo y el acelerómetro con el que cuenta este piloto automático es el MPU6000, el cual es una integración de estos dos sistemas, el rango máximo de estos elementos es de hasta $\pm 2000^\circ/\text{seg}$ y $\pm 16g$ respectivamente. La referencia del magnetómetro con el que cuenta esta solución es la HMC5883L. Para medir la velocidad del aire se emplea el barómetro de referencia MS5611 el cual presenta una resolución en altitud de hasta 10 cm y funciona a un voltaje de 1.8-3.6V. Este piloto automático es de código abierto y cuenta con 8 salidas para servomotores, tiene unas dimensiones de 50mm x 36mm x 6mm, un peso de 9.62 gramos y tiene un precio de 200 USD (Campo, 2014).

2- 3DR Pixhawk

Cuenta con un procesador llamado 32-bit ARM Cortex M4 core de 168MHz, una memoria flash 2MBy una memoria RAM de 256KB. El acelerómetro y magnetómetro vienen en solo componente llamado ST Micro LSM303D, las medias máximas que estos sensores pueden tomar son de $\pm 16g$ y de 12 gauss respectivamente, la respuesta de estos elementos es de hasta 400 KHz y funcionan con un voltaje de 2.16-3.6V. La referencia del giróscopo con el que cuenta es el T Micro L3GD20. Al igual que el piloto automático PX4 FMU, para medir la velocidad del aire se emplea el barómetro de referencia MS5611. El 3DR Pixhawk es de código abierto y cuenta con 14 puertos de salida para servomotores, sus dimensiones son 81mm x 50mm x 15mm, pesa 28 gramos y tiene un precio de 280 USD (Campo, 2014).

3- AMP 2.6 set

Este piloto automático de software libre cuenta con un procesador llamado ATMEGA 2560 el cual es un componente arduino que puede ser reprogramable por el usuario y funciona a 16MHz, tiene una memoria flash de 2564KB. La referencia del giróscopo y el acelerómetro con el que cuenta este piloto automático, al igual que el PX4 FMU, es el MPU6000. Este piloto automático tiene capacidad para 8 Salidas y 8 Entradas, tiene unas dimensiones de 70mm x 45mm x 15mm, pesa 33 gramos y tiene un precio de 240 USD (Campo, 2014).

1.5.4 Cloud Cap Technology.

1- Piccolo nano autopilot system

Se le puede conectar por medio de un puerto serial RS232 una IMU de 3 ejes con 300°/sec en rata angular máxima y hasta 10g como medida máxima del acelerómetro. Cuenta con 14 puertos para la conexión de servomotores, las dimensiones de este piloto automático son de 46mm x 76mm, su peso es de 22 gramos y el precio es de aproximadamente 1000 USD. Este autopiloto es de código cerrado (Campo, 2014).

2- Piccolo SL

Este piloto automático de código cerrado cuenta con un giróscopo de 3 ejes con una tasa máxima de cambio de 300°/seg. Además posee un acelerómetro de 3 ejes que mide hasta 6g. La velocidad máxima que reporta el sensor de velocidad del aire es de 355Kph. El GPS que se le integra al Piccolo SL tiene una tasa de actualización de datos de 4 Hz, además posee 14 puertos configurables como entradas y salidas, sus dimensiones son de 131mm x 57mm x 19mm, su peso es de 110 gramos incluyendo el módulo de comunicaciones de 900 MHz, su precio es de aproximadamente 1200 USD (Campo, 2014).

1.6 Definición de la arquitectura de hardware.

Luego del análisis realizado se puede determinar que el autopiloto Pixhawk es una buena solución a utilizar debido a que, en comparación con los demás autopilotos, este posee hardware y software abierto, también es de bajo valor económico, lo que lo convierte en un componente más asequible, posee una amplia gama de sensores de elevadas prestaciones incluidos en la IMU, su procesador tiene una potencia de computo suficiente capaz de procesar hasta 252 millones de instrucciones por segundo.

1.6.1 Pixhawk.

El controlador de vuelo Pixhawk es una nueva evolución del sistema controlador de vuelo PX4. Pixhawk consiste en un controlador PX4-UMF y un PX4-IO integrado en una sola tarjeta con IO adicional, la memoria y otras funciones. Es altamente optimizado para proporcionar un control y automatización para el software de navegación (3D Robotics, 2014).

Los diferentes firmwares del proyecto Ardupilot implementan un Filtro de Kalman Extendido (EKF), el cual puede ser ejecutado en las plataformas de hardware con alta capacidad de cómputo, como por ejemplo Pixhawk. Este algoritmo es un observador óptimo que se utiliza para mejorar la estimación de las mediciones de los sensores, a partir de conocer la información estadística de los ruidos que afectan al sistema. Otra característica del algoritmo EKF es que es capaz de estimar las desviaciones en las lecturas del compás del vehículo y también estimar el campo magnético de la tierra. El algoritmo implementado del EKF (Extended Kalman Filter), estima un total de 22 estados con las ecuaciones subyacentes derivadas (Ardupilot, 2016).

Los ruidos estimados en los giroscopios y acelerómetros (EKF GYRO ruido y ACC_NOISE EKF) se utilizan para estimar el crecimiento de error en los ángulos, velocidades y posiciones calculadas a partir de datos de la IMU, estos errores estimados son capturados en una matriz llamada “Matriz de covarianza de Estado”. El algoritmo del filtro de Kalman extendido proporciona una forma de combinar o fusionar los de datos de la IMU, GPS, brújula y otros sensores para calcular una estimación más precisa y fiable de la posición y orientación del vehículo.

Pixhawk cuenta con un procesador avanzado de 32 bits ARM STM32F427 Cortex M4 Core con un acelerador ART (Adaptive real-time) que opera a una frecuencia máxima de 180MHz y que es capaz de procesar 225 millones de instrucciones de punto flotante por segundo (DMIPS), gracias al cual se logra un rendimiento de un estado de espera de ejecución del programa de cero. Incorpora memorias embebidas de gran velocidad (Hasta 2MB de memoria flash y 256 KB de SRAM) (STMicroelectronics, 2014).

El Cortex-M4 core caracteriza una unidad de punto flotante (FPU) de precisión simple, la cual soporta todas las instrucciones de procesamiento de datos de precisión simple y acelera el desarrollo del software utilizando herramientas de desarrollo de metalenguaje, mientras evita la saturación. Además implementa un set completo de instrucciones DSP que le permite un procesamiento de señal eficiente y la ejecución de complejos algoritmos; además, cuenta con una unidad de protección de memoria (MPU), la cual aumenta la seguridad de la aplicación debido a que es de gran ayuda para aplicaciones donde algún código crítico o

certificado deba ser protegido contra el mal funcionamiento de otras tareas, esto es usualmente manejado por el sistema operativo de tiempo real (RTOS).

Por otra parte Pixhawk cuenta con una IMU con sensores de altas prestaciones y de gran funcionalidad. Posee un giróscopo ST Micro L3GD20H de 16 bit que incluye un elemento sensor el cual es fabricado usando un proceso de micro maquinas dedicadas, desarrolladas por ST para producir sensores inerciales y actuadores, este sensor opera en un rango de temperaturas de -40°C a 85°C (InvenSense, 2015). También cuenta con un acelerómetro/magnetómetro ST Micro LSM303D de 14 bit. Los bloques magnéticos y acelerómetros pueden habilitarse o ser apagados de forma separadas. Además posee un acelerómetro/giróscopo de 3 ejes cada uno MPU 6000 y un módulo sensor de presión MEAS MS5611 que incluye un elemento medidor de presión linear y un convertidor A/D de 24 bit (Meas, 2012).

Por tanto atendiendo a todas las potencialidades que brinda Pixhawk se puede definir la nueva arquitectura de hardware. A pesar de que las entradas y salidas del Pixhawk no cumplen con las que son necesarias para el control y la comunicación con el HRC-AUV, la incorporación del hardware de bajo nivel daría solución a esta problemática. Con esta nueva plataforma no se tendría una unidad más potente de computo pero si con mayor funcionalidad y mayor desempeño.

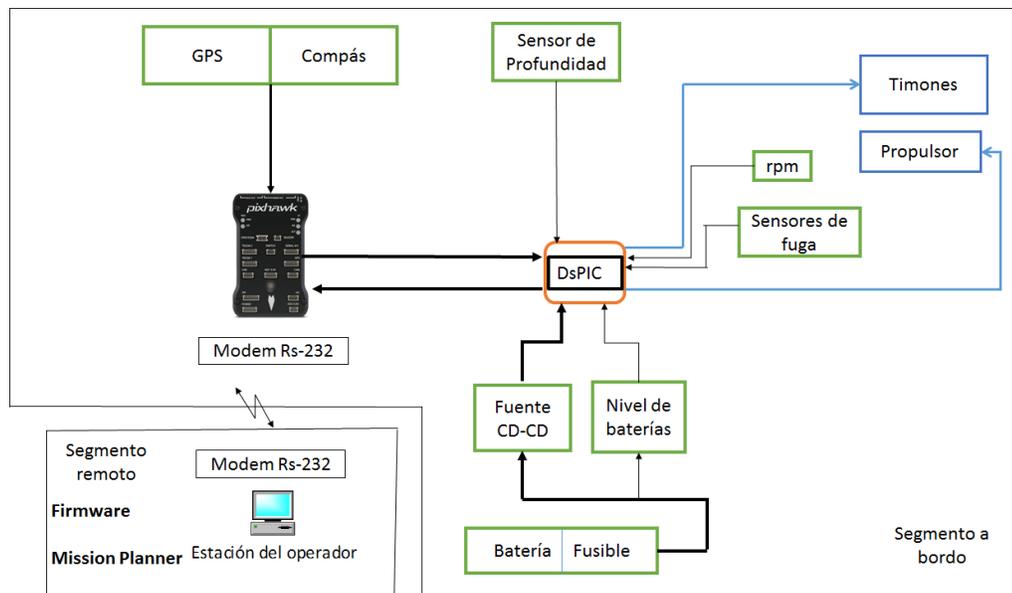


Figura 1.3 Estructura propuesta para el sistema.

Capítulo 2

ARQUITECTURA DE HARDWARE DEL SISTEMA A BORDO.

En este capítulo se describirá lo relacionado a la implementación de la arquitectura de hardware del sistema. Primeramente se analizarán los requerimientos de E/S del sistema a implementar, así como los medios de cómputo que se utilizarán incluyendo las principales características que dieron lugar a su selección. Posteriormente se explicarán de forma detallada las interfaces de comunicación así como las de entrada-salida. De igual forma se describirán los sensores empleados por el sistema. Se describirá el diseño de la etapa de fuente de alimentación luego de realizar el análisis de los voltajes requeridos por los distintos elementos del sistema. Por último se realiza el diseño del hardware de bajo nivel con la combinación de cada uno de los elementos analizados.

2.1 Requerimientos del sistema.

Como se mencionó en el Capítulo 1, para el desarrollo del autopiloto es necesaria la implementación de un hardware de bajo nivel para establecer la comunicación entre los sensores y actuadores del submarino con el Pixhawk. El hardware de bajo nivel es imprescindible para gobernar las superficies de mando del vehículo, por lo que este, también tiene una alta responsabilidad en el control del AUV, y cualquier fallo que esté presente puede hacer colapsar totalmente el sistema.

Atendiendo a las exigencias del HRC-AUV, el hardware debe cumplir con determinados requerimientos para que su implementación sea fiable y que posea un funcionamiento adecuado. También es necesario tener en cuenta las características y requerimientos de los elementos que lo conforman: entradas, salidas, canales de comunicación y la unidad de cómputo que lo gobierne. A continuación se presentan los requerimientos de E/S del hardware.

Entradas:

- (Analógica) Lectura del nivel de baterías del motor.
- (Analógica) Lectura del sensor barométrico.
- (Analógica) Lectura del sensor de temperatura (Opcional).
- (Digital) Lectura de los 2 sensores de fuga.
- (Digital) Lectura el sensor rpm del motor.

Salidas:

- (Digital) Señal On/off para el propulsor.
- (Digital) Señal de mando vía SPI al convertidor digito-analógico.

Puertos:

- UART para comunicación con Pixhawk.
- SPI para comunicación con el DAC.

2.2 Procesador de la interfaz de bajo nivel.

Es necesario evaluar las plataformas de microcontroladores existentes en el mercado para el diseño de la interfaz de bajo nivel. Entre las principales plataformas de hardware basadas en microcontroladores se encuentran las placas arduino y los kit de desarrollo de Microchip basados en DsPIC. Arduino es una plataforma de desarrollo que su núcleo es un Microcontrolador de la línea Atmel, mientras que Microcontroladores, en general, son chips, es decir circuitos integrados que ofrecen un hardware embebido junto con varios periféricos también embebidos. Por ejemplo cuando hablamos de Timers, Osciladores internos, PWM, conversores A/D , módulos de captura, Puertos, Stacks TCP, o Usarts, Puertos GPIO, estamos hablando de periféricos de un Microcontrolador que están embebidos dentro del mismo. Para la selección se tomaron en cuenta los siguientes aspectos (Circelli, 2015).

Pre-Proyectos: Estudios de factibilidad o pruebas rápidas como estudio inicial de un proyecto más grande, Arduino ahorra costos y tiempo.

Escala de proyectos: Si un proyecto va a fabricarse en unidades masivas es importante considerar los costos. Cuando usamos un Arduino en un desarrollo, el producto final puede utilizar pocos pines y/o recursos de la placa que no se utilicen, es más barato entonces, comprar un Microcontrolador y diseñar el propio hardware.

Tipo de Aplicaciones especiales: Existen aplicaciones en donde las señales de tiempo real, requerimientos de velocidad, memoria interna y/o memoria externa, incluso librerías necesarias y no implementadas aún no satisfagan lo deseado. En estos casos habrá que estudiar el Microcontrolador internamente para modificar las librerías o, pasarse a otro microcontrolador.

En resumen el trabajar con Microcontroladores da otra apertura, otra imagen, otro grado más profundo en las aplicaciones y que implica un nivel de conocimiento superior. La desventaja es que con microcontroladores, hay que mantenerse más actualizado, nuevas herramientas, plataformas, compiladores, nuevos microcontroladores. Arduino es más simple y como posible desventaja se puede decir que ata a una arquitectura de diseño, lo cual puede ser ventaja para algunos y desventaja para otros (Circelli, 2015).

De acuerdo a los planteamientos anteriores, se decide optar por la utilización de un microcontrolador de la compañía “Microchip”, la cual fabrica microcontroladores de propósito general y componentes discretos, así como también placas de desarrollo para el trabajo con sus productos (Alvarez, 2014), esta compañía también proporciona una gran cantidad de documentación relacionada con sus dispositivos, además de poderosas librerías de software para desarrolladores que hacen más fácil y seguro la implementación de aplicaciones, sobre su plataforma de microcontroladores. Por esos motivos, “Microchip” resulta ser una solución fiable y económicamente factible para el desarrollo de este proyecto.

Para la arquitectura de hardware se seleccionó el dsPIC33FJ32MC204 fabricado por Microchip, ya que este cumple con los requerimientos mínimos necesarios para la aplicación, y a su vez es el más barato de su línea de fabricación. Este microcontrolador tiene como función general el procesamiento de los datos adquiridos desde los sensores y de manejar las interfaces de comunicación entre el vehículo y el Pixhawk. En lo que resta del epígrafe se contarán las principales características que presenta este chip.

2.2.1 Características de los DsPIC33FJxxMCx04.

El dsPIC33FJ32MC204 es un microcontrolador de gama alta, este tiene un encapsulado de 44 pines de montaje superficial, y se diferencia del resto de los PIC y DsPIC en que los micros de la serie 33 pueden redireccionar los periféricos hacia diferentes pines a través del programa de usuario. Alcanza un desempeño máximo de 40 millones de instrucciones por

segundo (MIPS), con un oscilador configurable para distintas frecuencias de ejecución. Presenta 32 KB de memoria flash, 2 KB de SRAM, con un set de instrucciones amplio y 7 niveles de prioridad de interrupción programable. Entre los periféricos incluidos presenta 3 temporizadores de 16 bit con posibilidades de formar 1 par de 32 bit, 4 mecanismos de captura de tiempo a la entrada y 2 a la salida de 16 bit cada uno, 6 canales de 16 bit de moduladores de ancho de pulso (PWM), así como un módulo de interfaz de codificador de cuadratura, lo que le permite capturar una amplia variedad de señales. Posee 9 entradas analógicas con un conversor que puede configurarse para una resolución de 12 bit a velocidad de 500 Ksps (kilo muestras por segundo) y para 10 bit a velocidad de 1.1 Msps, lo que satisface las necesidades de la aplicación. Presenta interfaces de comunicación serie que se emplean en la actualidad en distintos dispositivos en sistemas empotrados, dígame: un módulo SPI (Serial Peripheral Interface), un módulo UART (Universal Asynchronous Receptor Transmitter) y un módulo I²C (Inter-Integrated Circuit) (Microchip, 2007). En el transcurso de este capítulo se mostrarán los distintos usos dados a estos recursos.

2.2.2 Estructura general del hardware.

Luego del análisis realizado con respecto a los requerimientos del sistema y del desempeño y la funcionalidad del microcontrolador se puede definir una estructura general para el hardware a implementar. A continuación se muestra el diagrama funcional de enlace entre hardware y software para el sistema, el cual muestra el funcionamiento básico del mismo.

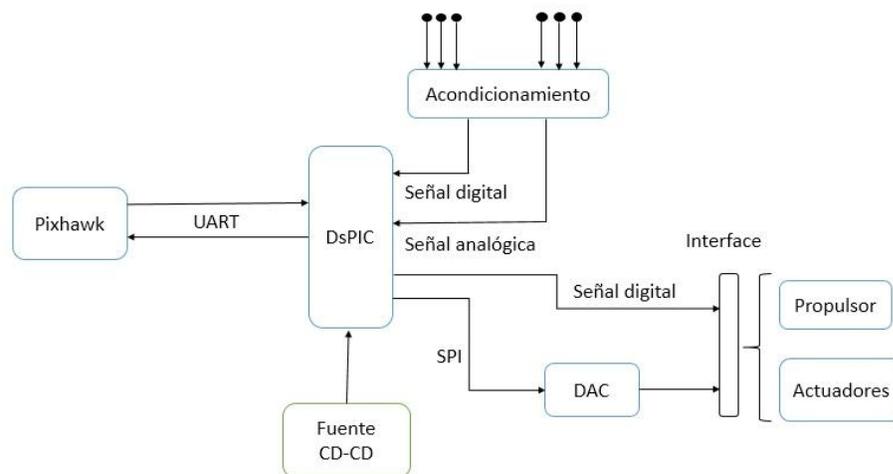


Figura 2.1 Diagrama funcional entre hardware y software del sistema.

2.3 Interfaces de comunicación.

El microcontrolador utilizado presenta varios módulos de comunicación como son I²C, SPI y UART, en el hardware a implementar solo son utilizados los dos últimos, uno para el canal interno DsPIC-DAC, y otro para el externo Hardware-Pixhawk.

2.3.1 Interfaz SPI.

SPI (Serial Peripheral Interface) es una interfaz de comunicaciones, desarrollada por Motorola, para comunicar sus microcontroladores con otros chips periféricos, de una forma simple y de bajo costo. Al SPI también se le conoce como la interfaz de cuatro-hilos y puede ser utilizada para conectar diversos tipos de memorias, convertidores A/D y D/A, relojes de sistema a tiempo real y calendarios, pantallas LCD, sensores e incluso otros procesadores. El rango de componentes conectados que soporta el SPI es muy grande y creciente a lo largo del tiempo. En este hardware, dicho módulo fue utilizado como medio para la comunicación del canal interno entre el DsPIC y los convertidores digito-analógicos.

Como se mencionó anteriormente este tipo de interfaz utiliza cuatro líneas del microcontrolador que corresponden a: recepción, transmisión, selección y reloj. El bus SPI proporciona varios modos de funcionamiento, y no todos los modos necesitan el mismo número de señales para funcionar. Se puede conseguir una configuración mínima de comunicación utilizando sólo 3 señales funcionando en modo half-duplex, o una configuración de 4 señales en modo full-duplex.

SPI es un protocolo síncrono en el que todas las transmisiones están referidas a un reloj común, generado por el maestro, que en este caso es el microcontrolador. El periférico receptor, esclavo, usa este reloj para sincronizar la adquisición de la cadena de bit en serie. Esta señal también se encarga de fijar la velocidad a la que funcionará el bus y a la que se transmitirán los datos. La ventaja de ser un bus síncrono es que no hacen falta unas velocidades predeterminadas o estándares que deban cumplirse, sino que las velocidades las fijan los dispositivos que se están comunicando en un determinado instante de tiempo. La velocidad puede ser tan rápida como permita el dispositivo más lento que use la comunicación. Esta característica hace que si los dispositivos a comunicarse pueden trabajar a una alta frecuencia, la comunicación entre ellos será muy rápida. Se pueden alcanzar

velocidades de varios Mbit/s, lo que lo convierte en un bus de comunicación muy rápido (Vique, 2008).

La señal que transfiere datos desde el dispositivo maestro hacia los esclavos la podemos encontrar denominada de dos formas diferentes según el fabricante. Unos cuantos fabricantes la llaman MOSI, que significa Master Output Slave Input, mientras que otros fabricantes la llaman simplemente SDO, que significa Serial Data Out. El mismo caso se da para la señal que transmite los datos desde el esclavo hacia el maestro. Se puede llamar MISO (Master Input Slave Output) o SDI (Serial Data In). En la implementación del hardware la señal MISO no es utilizada. La transmisión y la recepción de los datos pueden ser de 8 o de 16 bits. El modo de funcionamiento es el mismo excepto el número de bits que son enviados y recibidos (Martín, 2013).

Finalmente, la señal SS es la señal de selección de esclavo (Slave Select), aunque también la podemos encontrar denominada como CS (Chip Select). Esta señal tiene la función de indicar con qué esclavo se está comunicando el maestro en cada momento. Cuando el maestro se comunica con un esclavo, para indicarlo pone a cero lógico la señal SS del esclavo con el que se comunica, mientras que cuando no se está comunicando con un esclavo pone la señal SS a 1 lógico. Esta característica hace que cada esclavo que se conecta a un mismo bus SPI necesite una señal propia de selección. La selección del chip hacia el periférico viene normalmente generada por una línea común de salida del microcontrolador maestro.

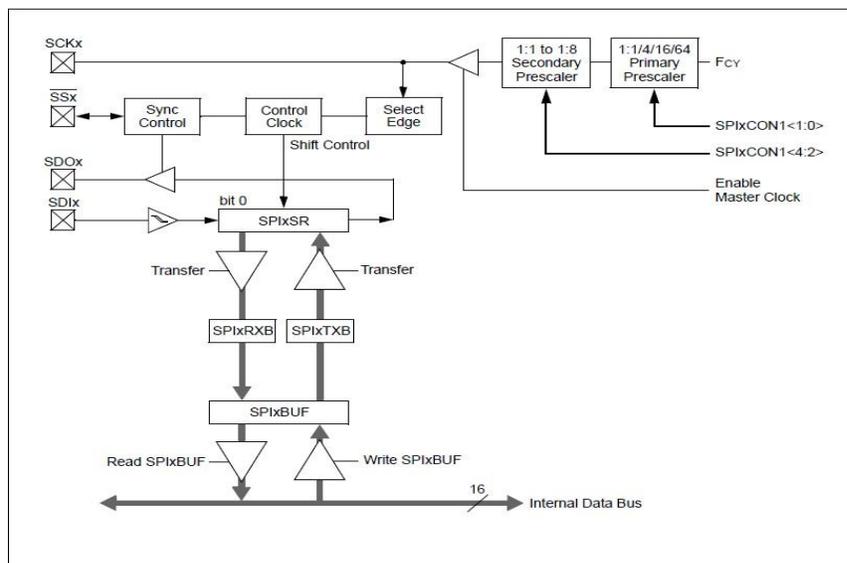


Figura 2.2 Diagrama de bloque del módulo SPI.

2.3.2 Interfaz UART.

El protocolo UART (Universal Asynchronous Receiver / Transmitter) es una interfaz de comunicación serie que se utiliza para comunicar dos dispositivos a través de cable. Permite la comunicación con dispositivos a distancias relativamente grandes, por encima de los 25m con gran fiabilidad, además de ser utilizadas en múltiples tipos de sensores y periféricos.

En el presente caso, la comunicación entre Pixhawk y DsPIC se produce mediante esta vía. Los mismos generan tramas con estructura similar a RS-232 excepto por los niveles lógicos. El microcontrolador implementa lógica TTL con niveles de 0 – 5V, mientras que RS-232 tiene lógica negada, bipolar y con niveles de voltaje que varían entre $\pm 3V$ y $\pm 25V$ (Guerra, 2010).

En esta interfaz los datos se transmiten asíncronamente entre los dos dispositivos, es decir, sin una señal de reloj que marque el ritmo de la transmisión. Por esto, la comunicación sólo se puede conseguir si ambos dispositivos están configurados a la misma velocidad y con las mismas características. La información se transmite en forma de bytes, junto con bits de start y stop para que el otro dispositivo pueda identificar donde empieza y donde termina un byte.

La transmisión de datos UART se realiza byte a byte siguiendo un protocolo concreto. La transmisión de cada byte empieza con un bit de start que pone la señal a nivel bajo. De esta manera el dispositivo al otro lado detecta el inicio de una transmisión, ya que cuando el puerto no transmite, las señales se mantienen a nivel alto. Seguidamente se inicia la transmisión de bits de datos, con la particularidad de que se puede definir la longitud de un byte entre 5 y 8 bits. Opcionalmente, al final del byte de datos, se puede enviar un byte de paridad para comprobar que éste se ha transmitido correctamente. El bit de paridad se puede configurar como par o impar, y lo que hace es comprobar si el número de bits a nivel alto del byte es par o impar. Para terminar la transmisión de un byte se envía una señal de stop a nivel alto que se puede configurar entre 1 bit, un bit y medio o dos bits (Martín, 2013).

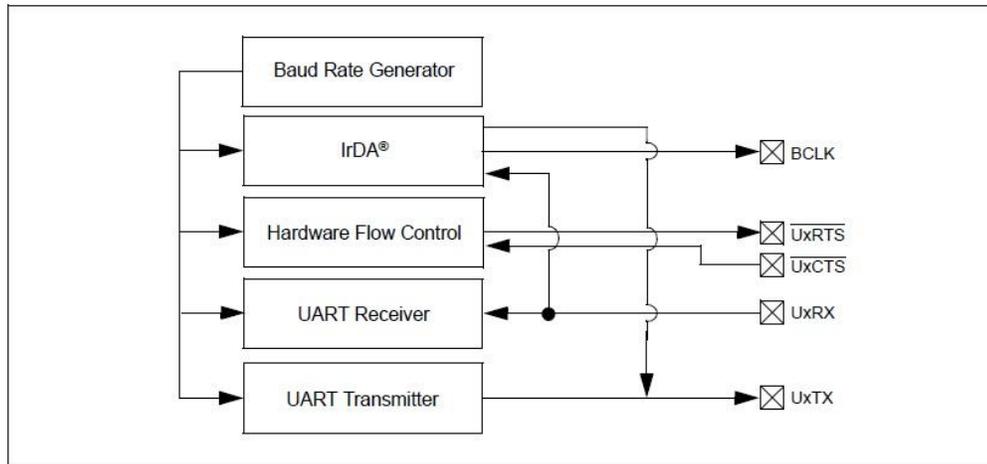


Figura 2.3 Diagrama de bloque del módulo UART.

2.4 Señal entrada-salida.

La arquitectura de hardware diseñada debe soportar la interacción con variables analógicas, ya que es el tipo de señal preponderante en el sistema. La mayor parte de los sensores empleados, utilizan este tipo de interfaz, así mismo la señal de gobierno a los timones es analógica. Por esta razón es necesario tener en cuenta requerimientos de enlace para su procesamiento digital (Guerra, 2010).

Para el sistema de adquisición analógica el DsPIC cuenta con un ADC (Analog to Digital Converter), este utiliza el método de conversión de aproximaciones sucesivas. Este convertidor puede ser configurado para funcionar el modo 12-bit o en 10-bit. Los voltajes de referencia del convertidor pueden ser ajustados desde pines externos o fijados por software, en el hardware la referencia negativa está conectada al pin VSS y la positiva al pin VDD. Esto fija los niveles de entrada de los canales de 0 – 5v, además se utiliza el modo de 10-bit para la conversión, lo que nos da un peso por cada LSb (Low Significant bit) de 3,22mV (Alvarez, 2014).

2.4.1 Acondicionamiento de entradas analógicas.

El microcontrolador posee nueve entradas analógicas con un rango de señal de 0 – 5v, de las cuales se utilizan tres en los distintos tipos de sensores y se dejan dos como reserva completando un total de cinco interfaces disponibles (AI0 – AI4). Cada etapa tiene un acondicionamiento analógico conformado por un diodo zener de 5.1v con función de saturar

la entrada de señal contra sobre voltaje y un filtro pasivo RC de primer orden paso bajo con la función principal en la eliminación del efecto de “*aliasing*”. La constante de tiempo para cada uno de estos corresponde a la relación RC y se ajustó a una frecuencia de corte de 20 rad/seg debido a que la dinámica que experimentan estos sensores es lenta producto de la construcción física del vehículo.

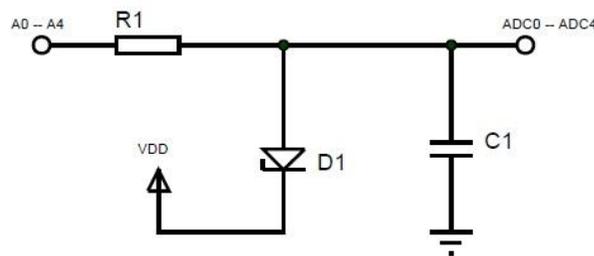


Figura 2.4 Sistema de acondicionamiento analógico.

2.4.2 Salidas analógicas.

Es necesario implementar dos salidas analógicas para el gobierno de los timones de dirección del vehículo, ya que los “*drivers*” utilizados para el manejo de estos requieren este tipo de interfaz en la entrada y con un rango de $\pm 10\text{v}$. Para esto se diseñó un (DAC) con dos salidas analógicas disponibles, este está conformado por dos etapas, la primera un convertor dígito-analógico, y la segunda una ganancia para elevar el nivel de la señal de salida (Guerra, 2010). Para los convertidores se utilizaron los circuitos integrados MCP4821. Los mismos presentan una interfaz de entrada digital serie vía SPI, la configuración de cada módulo convertidor se realiza por software. La segunda etapa está constituida por un amplificador operacional LM741 en modo no inversor con una ganancia de magnitud 2, la misma amplifica la salida del convertor D/A, para colocarla en el nivel de señal deseado, acorde al rango de entrada de los “*drivers*”, que para el HRC-AUV es de $\pm 10\text{v}$.

Tabla 2.1. Especificaciones para el MCP4821

Parámetro	Rango	Unidad
Salida analógica	0 - 4.95	v
Resolución	12	Bit
Voltaje de alimentación	2.5-5.5	v

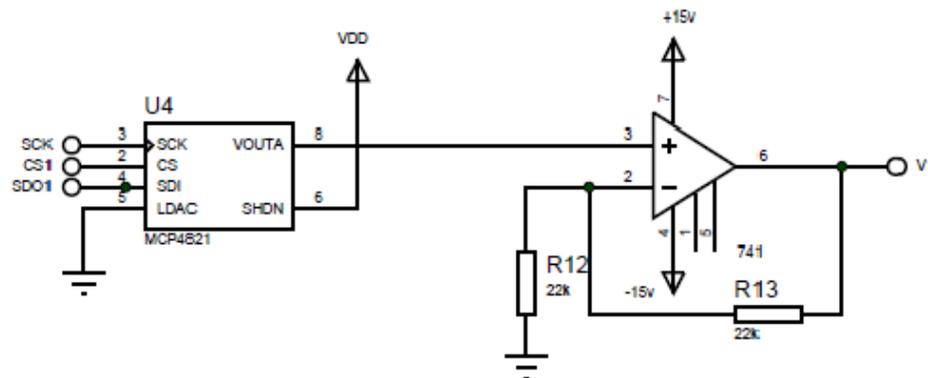


Figura 2.5 Interface de salidas analógicas.

2.5 Sensores.

La arquitectura del sistema presenta una serie de sensores que la convierten en una estructura multisensorial. Tomando en cuenta los requerimientos mencionados en el inicio de este capítulo se pueden definir los sensores a utilizar para de esta forma determinar las modificaciones pertinentes que le corresponden a las entradas analógicas del hardware.

2.5.1 Presión barométrica.

Para la medición de profundidad se utilizó el sensor de presión barométrica Cerabar T PMP131 del fabricante *Hendres-Hausser* el cual es un medidor de presión absoluta. Este sensor se encuentra actualmente instalado en el vehículo. La interface de salida del transductor es un lazo de corriente, por lo que se requiere convertir el mismo a una señal de voltaje que pueda ser adquirida por medio de un canal analógico. Por lo cual se utilizó un conversor sencillo basado en una resistencia de 250Ω en serie a la señal logrando un rango

en voltaje de 1 - 5v para una resolución de 1.6 v/Bar (Guerra, 2010). En la figura 2.6 se muestra el proceso de conversión.

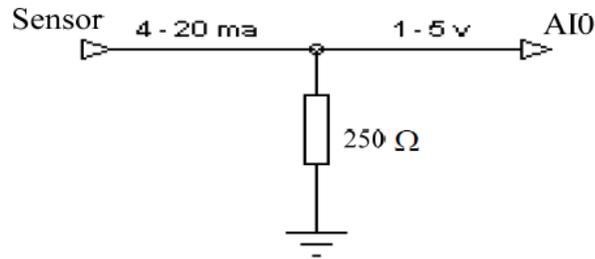


Figura 2.6: Conversor de corriente a voltaje.

De esta forma se obtiene un rango de profundidad máximo de 25m. Para la adquisición se utilizó la interface A0.

2.5.2 Nivel de baterías.

La medición del nivel de baterías está determinada por la necesidad de conocer el estado energético de la fuente de suministro del vehículo, así mismo es un parámetro fundamental para la estimación del torque del motor de propulsión. Por medio de un divisor resistivo se mide el voltaje directamente en la fuente de alimentación principal (48v) en el cual la señal queda reducida con una relación de 11:1 como se ilustra en la figura 2.7 y logra insertarse en el rango de 0 – 5v para poder ser medida a través de las entradas analógicas en el procesador (Guerra, 2010). Esta señal se adquiere a través de la interface A2.

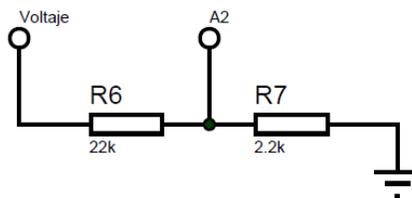


Figura 2.7 Divisor resistivo.

2.5.3 Temperatura.

Para esta medición se utilizó el mismo sensor de temperatura MCP9700, que es empleado en la arquitectura actual del HRC-AUV, fabricado por *Microchip*. La señal de salida de este sensor es obtenida por la entrada analógica A1.

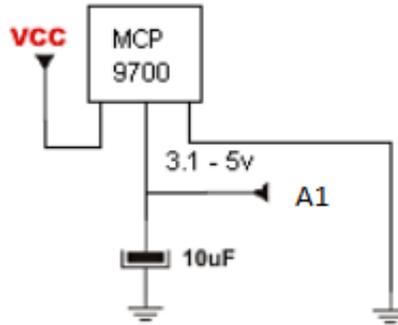


Figura 2.8 Sensor de temperatura.

2.5.4 Presencia de agua.

Estos sensores se encuentran distribuidos en ambos extremos del vehículo “popa y proa”, presentando una interface digital (On-Off), su constitución física es similar al de un interruptor ordinario con un retardo de tiempo, por lo que se le debe incluir un acondicionamiento digital para poder adquirirlo por entradas digitales de la unidad DsPIC. Los sensores de presencia de agua son muestreados por el hardware mediante las interfaces digitales DI/1 y DI/2 que corresponden al sensor en proa y en popa respectivamente, estos son activos en bajo, por lo que presentan una lógica negada (Guerra, 2010).

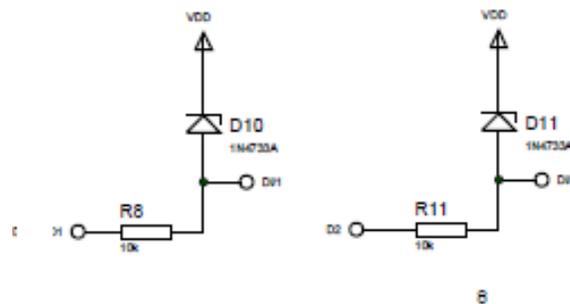


Figura 2.9 Conexionado para sensores de presencia de agua.

2.5.5 Medición de las RPM del motor de propulsión.

Para el sensor de rpm se utilizaron dos sensores con el objetivo de implementar los codificadores ópticos. Uno de ellos es basado en el integrado CNY70 y el otro se implementó

a partir de un diodo de emisión infrarroja y un foto-transistor (Díaz, 2012), este sensor se encuentra actualmente como parte de la arquitectura de hardware instalada en el vehículo. La señal de salida del sensor es digital y es adquirida por la entrada DI/0, donde el DsPIC procesa el tren de pulsos, con una frecuencia proporcional a la velocidad con que rota el motor, y calcula las rpm del motor de propulsión.

2.6 Fuente de alimentación.

La fuente alimentación es fundamental para el correcto funcionamiento del hardware. Esta es la encargada de la regulación de voltaje para cada uno de los distintos tipos de elementos del sistema de hardware así como sensores y otros dispositivos.

El voltaje de entrada para el hardware es de 15v provenientes de un juego de baterías, las cuales son un subgrupo del banco de alimentación del motor de propulsión, con voltaje total de 48v, esta situación implica altos picos de corriente en el arranque y por consiguiente una caída de voltaje. Esto podría tener efectos perjudiciales en el equipamiento electrónico, por esta razón para eliminar este transiente se utilizó a la entrada del bloque de fuente un circuito conformado por un capacitor de alto valor de capacitancia para contrarrestar este efecto (Guerra, 2010).

El hardware contiene elementos digitales que funcionan a diferentes voltajes. El DsPIC trabaja con un voltaje de 3.3v, mientras que los circuitos integrados MCP4821 se alimentan con 5v. Para la regulación de 5v se utilizó el circuito integrado LM7805 que es un regulador de voltaje positivo de 5v con tres terminales. Este entrega una corriente máxima 1A, la cual es suficiente para la parte digital del hardware, pero los servomotores pueden llegar a consumir picos de corriente repetitivos y no repetitivos superiores a 1A, y esto causaría un descenso en los niveles de voltaje provocando el rest de los elementos digitales y haciendo colapsar toda la aplicación. Para resolver este problema se le añadió un transistor y un resistor externo al integrado con la configuración de regulador de corriente, de forma tal que valores de corrientes demandados por el sistema y superiores a los 500mA sean drenados por el transistor, esto también evita calentamientos no deseados en el regulador principal ya que se limita la potencia disipada por el mismo (Alvarez, 2014). Para la regulación de 3.3v se utiliza el regulador 17-33, este regulador es conectado en cascada al 7805 para que la potencia disipada sea menor. Al tener esta fuente una demanda de corriente pequeña, no es necesario

acoplar el regulador e corriente adicional. En la figura se muestra el esquema completo de la fuente.

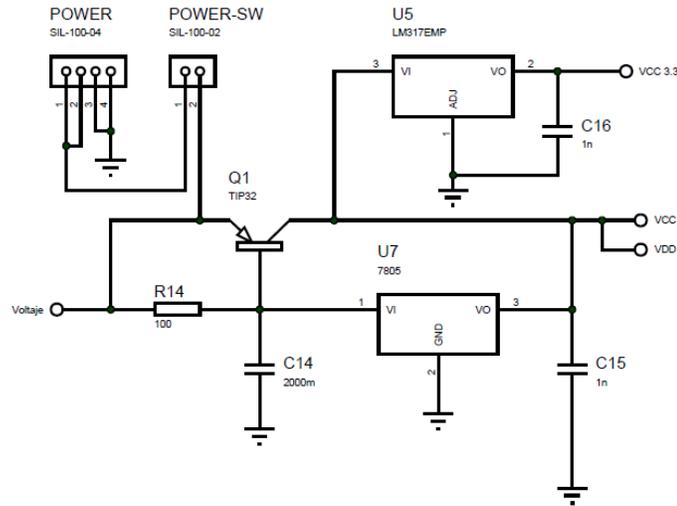


Figura 2.11 Sistema regulador de alimentación.

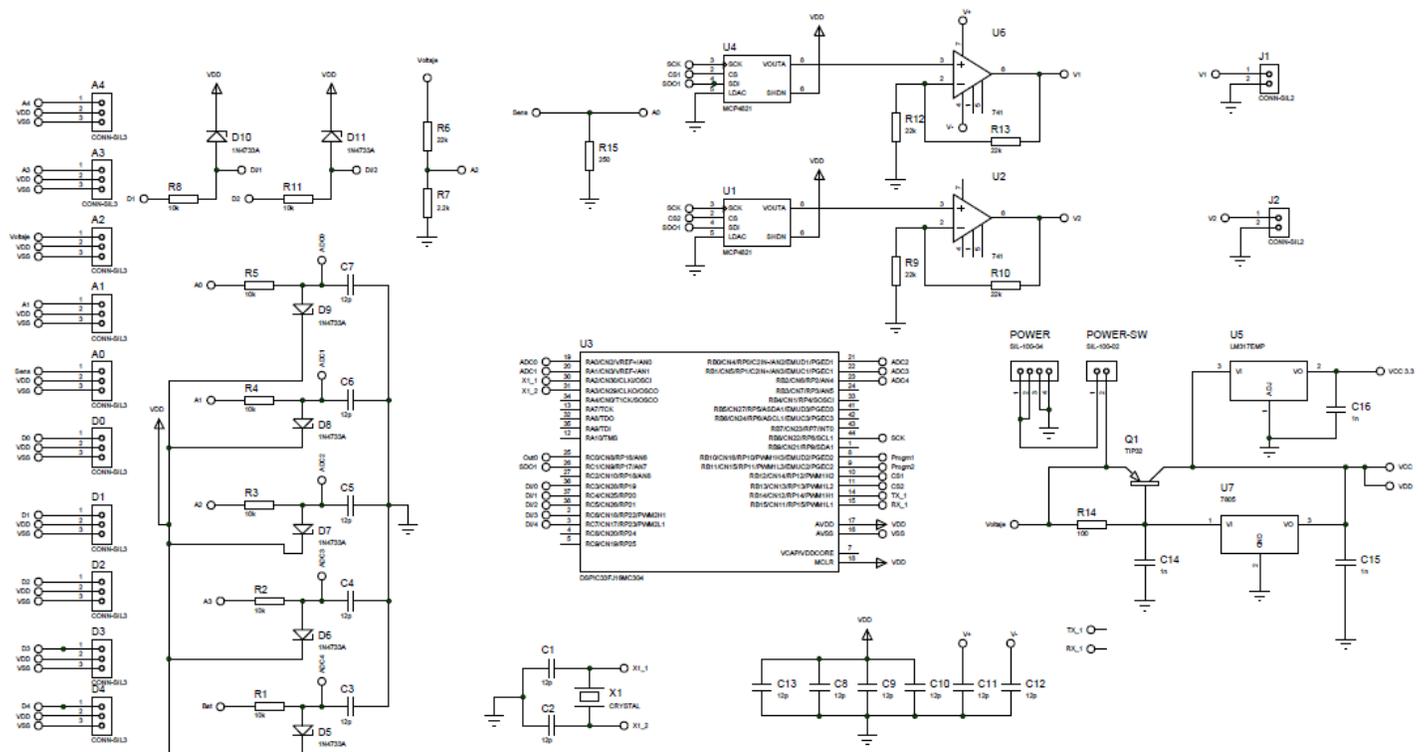


Figura 2.12 Esquema del hardware de bajo nivel.

2.7 Consideraciones finales del capítulo.

En este capítulo después de realizar un análisis de los requerimientos y prestaciones que debe tener el sistema, así como las consideraciones a tomar en cuenta para el diseño, se llevó a cabo el proceso de selección del microcontrolador para el hardware. En el capítulo quedan comentados los distintos módulos que componen el sistema, los cuales son: adquisición y salidas, comunicación y fuente.

Capítulo 3

SOFTWARE DEL SISTEMA A BORDO.

En este capítulo se aborda lo relacionado con el software de bajo nivel implementado para lograr el flujo de datos coherentes entre el Pixhawk y el HRC-AUV. Se describe la estrategia de programación utilizada para establecer protocolos de comunicación, adquisición de señales y configuración de los módulos utilizados por el sistema. Además se realiza un análisis económico de costos relacionado a la implementación de la arquitectura en general.

3.1 Software de Bajo nivel.

Hasta el momento se ha analizado todo lo relacionado con la arquitectura de hardware de bajo nivel, ya en este punto es posible abordar la estrategia de programación empleada en el sistema para establecer y dar solución a la problemática planteada en este trabajo, la cual hace referencia a la comunicación y las E/S del mismo.

El software de bajo nivel es ejecutado en el microcontrolador y está implementado en lenguaje C a bajo nivel. Para su implementación fueron aprovechadas varias librerías brindadas por Microchip que refieren al desarrollo de aplicaciones en tiempo real en sistemas empujados. La ejecución de este software se basa en la planificación de tareas del sistema y en protocolos para la comunicación.

Para el correcto funcionamiento del software es necesario conocer el funcionamiento del hardware, para ello es necesario analizar la figura 2.1 mostrada en el capítulo anterior, la cual muestra las principales funciones del software que son la adquisición y la comunicación.

3.2 Planificación de tareas.

Para la implementación las tareas mencionadas en el epígrafe anterior, el microcontrolador cuenta con varias interrupciones programables, las cuales son utilizadas para manejar los hilos de ejecución de la aplicación. En la figura 3.1 se ilustra en diagrama de flujo para la planificación de tareas del software.

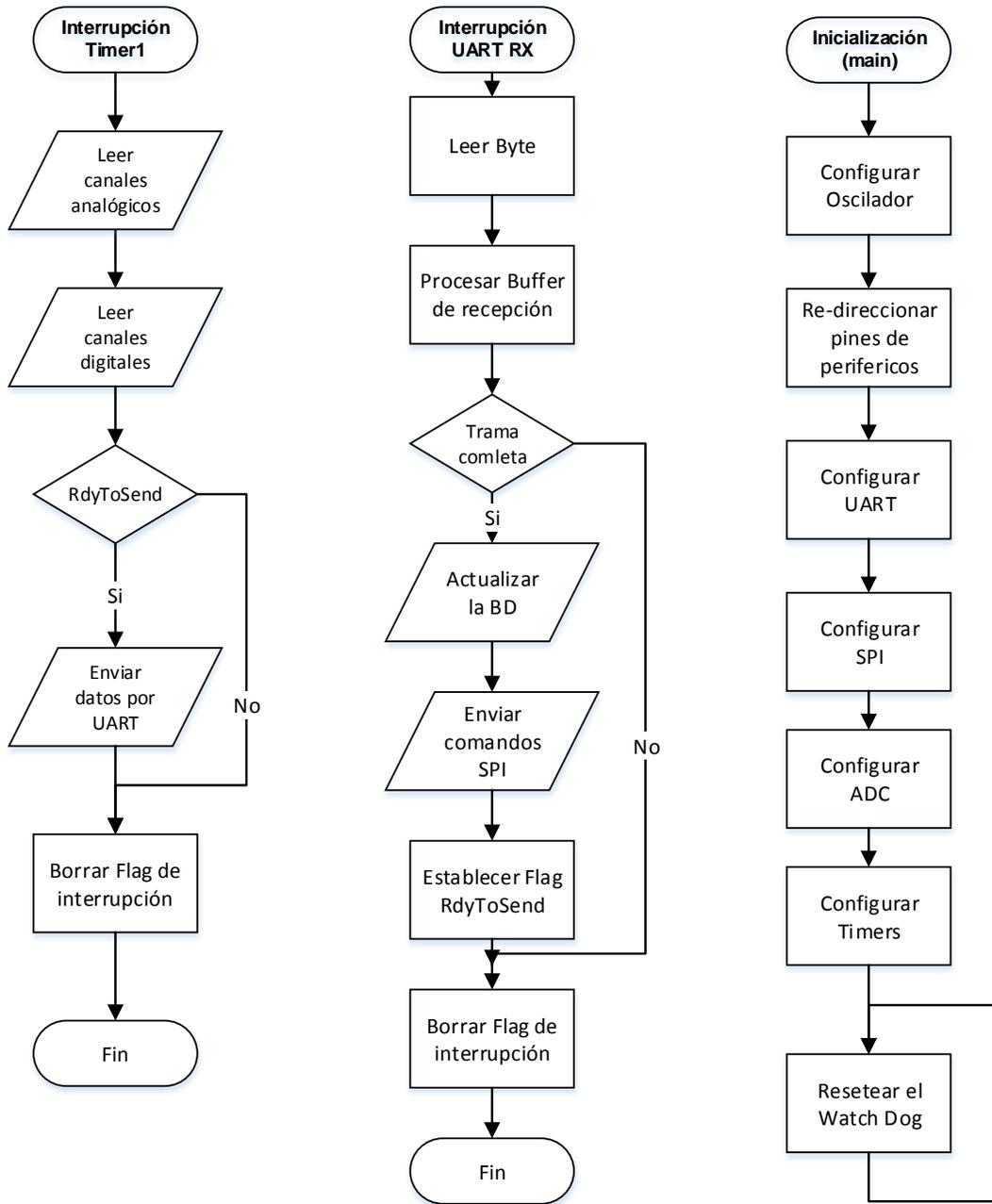


Figura 3.1 Esquema de tareas del dsPIC.

En una computadora normal, el sistema operativo es el encargado de distribuir los tiempos de cómputo y los recursos entre las aplicaciones y las tareas. Mientras que en los DsPIC, para implementar la característica multitarea, se utiliza la programación cooperativa a base de interrupciones, en este caso son los propios procesos los que ceden el uso del procesador en intervalos regulares (Ruben, 2013).

En total se utilizaron tres ciclos para manejar las tareas de inicialización, comunicación y adquisición. En este caso el periodo de muestreo se fijó para 20ms, de esta forma el sistema no se carga prácticamente. En el transcurso de este capítulo se describirán cada uno de los procesos empleados en la arquitectura de software.

3.3 Inicialización.

El proceso de inicialización corresponde a la configuración de los componentes internos del sistema y de sus periféricos. Este proceso se realiza una sola vez en el lazo principal del programa (*main*) y del mismo depende el correcto funcionamiento del sistema. Entre los componentes a inicializar se encuentran el oscilador, módulo de comunicación SPI, módulo de comunicación UART, interfaces analógicas, temporizadores y los pines de E/S del microcontrolador.

Para llevar a cabo esta tarea se implementaron dos clases además de la clase principal *main*, estas son la clase *System* donde se implementaron las funciones para configurar los módulos de comunicación serie directamente a cada bit de sus registros respectivos, y la clase *SetupHardware* donde fue configurado el oscilador, el módulo ADC y los temporizadores, también en esta clase se redireccionaron los pines de los periféricos, característica que brinda la familia del DsPIC empleado. La tabla a continuación muestra las especificaciones principales de estas configuraciones.

Tabla 3.1 Especificaciones de configuración.

Parámetro	Configuración	Aplicación
Oscilador	Frecuencia = 80 MHz Fcy = 40 MHz	Oscilador del Proceso
SPI	Modo 16 bit Modo Master	Comunicación con DAC
UART	Baud Rate = 115200	Comunicación con Pixhawk
Convertidor A/D	Resolución 10 bit	Medición de las variables analógicas
Timer 1	Modo 16 bit T = 20 ms	Ciclo principal de muestreo
Timer 2	Modo <i>Gate</i>	Medición de ancho de pulso para el sensor RPM

3.4 Comunicaciones.

Como se mencionó en el capítulo anterior fue necesario, para establecer la comunicación entre los canales internos y externos del sistema, implementar los módulos UART y SPI, ambos requieren una configuración que especifique el modo de trabajo de cada uno de acuerdo a las frecuencias, paquetes de datos, finalidad, etc. A continuación se muestra de forma más detallada la configuración de los mismos.

3.4.1 Comunicación UART.

La implementación del módulo UART se realizó siguiendo un protocolo el cual es adoptado por todas las aplicaciones del GARP y ha demostrado un índice de funcionamiento adecuado, el cual se basa en una estructura de paquetes, donde cada paquete cuenta con un encabezado, un campo de datos y una cola donde se inserta el chequeo de integridad.

Con la implementación de este protocolo se puede enviar y recibir hasta un total de 255 bytes de datos, 127 integer de 16-bit, 63 *float* o 31 *double*, esto significa un aprovechamiento del ancho de banda del canal del 97.7%, 95.4%, 91.3% y 83.7% respectivamente (Alvarez, 2014), en el caso que ocupa este trabajo, los datos que se manejan son *int*, *float*, *char*, por lo que en cada caso se utiliza el canal de forma óptima. La tabla muestra la estructura de los mensajes.

Tabla 3.2 Estructura de mensajes

Preámbulo			Longitud	Identificador	Datos	Chequeo
<i>0xFA</i>	<i>0xAA</i>	<i>0x00</i>	<i>0x00...0xFF</i>	<i>0x00...0xFF</i>	Nbytes	\sum Datos

El sistema del identificador de mensaje es utilizado para definir el campo de datos que le sigue, o sea, cada mensaje con diferente objetivo poseerá su propio identificador. Por otra parte el sistema de comprobación de integridad de los datos consiste en realizar una sumatoria de los datos del campo de datos, despreciando el desbordamiento de la variable, lo que genera un número diferente para cada paquete de datos y es enviado al final del mismo. Cuando se realiza la comprobación simplemente se calcula nuevamente la sumatoria y este valor es comparado con el último dato del paquete, en caso de ser iguales la trama es correcta, si estos valores son diferentes significa que los datos están corruptos y la trama debe ser despreciada. Este mecanismo no es capaz de determinar cuál es el dato no válido, simplemente que hubo un error en la transmisión, este algoritmo falla cuando son intercambiados los bits de la misma posición entre diferentes bytes, ya que al ser una sumatoria, cuando se suma un determinado valor incorrecto y posteriormente se resta esa misma cantidad el resultado de la operación no se altera (Alvarez, 2014).

Mediante este módulo se actualizan los datos correspondiente a las señales adquiridas desde los sensores enviándolos hacia el Pixhawk, proceso que es inicializado cada 20ms luego de haberse ejecutado los comandos de adquisición, a su vez es el encargado de recibir los datos de la señal de mando para los actuadores del vehículo, generando una interrupción en el programa para proceder al procesamiento y envío de dichos datos. La trama de datos que es enviada con las variables de los sensores posee una longitud de 18 *bytes*, mientras que la

longitud de la trama recibida posee una longitud de 9 *bytes*. La siguiente tabla muestra la correspondencia de estos datos en el campo *Datos*.

Tabla 3.3 Correspondencia de los datos en las tramas.

Sentido de comunicación	Numero de Bytes	Variables
Transmisión	Byte 0 – Byte 3	Sensor RPM
Transmisión	Byte 4	Presencia de agua
Transmisión	Byte 5	Presencia de agua
Transmisión	Byte 6 – Byte 9	Sensor Temperatura
Transmisión	Byte 10 – Byte 13	Sensor Profundidad
Transmisión	Byte 14 – Byte 17	Nivel de baterías
Recepción	Byte 0	Propulsión
Recepción	Byte 1 – Byte 4	Actuador 1
Recepción	Byte 5 – Byte 8	Actuador 2

3.4.2 Comunicación SPI.

Para comenzar a utilizar este tipo de comunicación es imprescindible tener en cuenta algunas especificaciones como son: si se trabaja en modo esclavo la línea SCK debe ser de salida, determinación de la polaridad del reloj, momento de lectura del dato (puede ser al final o a mitad del impulso del reloj), etc.

Primeramente es necesario configurar el modo en que es enviada la trama de datos, siguiendo el protocolo establecido para la comunicación con el MCP4821, proceso mencionado en el capítulo anterior. La figura muestra el registro de comandos para este módulo, donde el bit 15 corresponde a la selección del canal de salida DAC_A o DAC_B, el bit 13 corresponde a la

ganancia del voltaje de salida, el bit 12 corresponde a la habilitación del buffer de salida, mientras que los bits del 11 al 0 corresponden al paquete de datos.

Upper Half:							
W-x	W-x	W-x	W-0	W-x	W-x	W-x	W-x
$\overline{A/B}$	—	\overline{GA}	\overline{SHDN}	D11	D10	D9	D8
bit 15							bit 8
Lower Half:							
W-x	W-x	W-x	W-x	W-x	W-x	W-x	W-x
D7	D6	D5	D4	D3	D2	D1	D0
bit 7							bit 0

Figura 3.2 Registro de comandos de escritura para el MCP4821.

Por otra parte, para lograr el funcionamiento del bus SPI se utilizan una serie de registros de configuración, que son principalmente los registros SPIxCON y SPIxSTAT. El registro SPIxCON proporciona la interfaz necesaria para controlar la operación del módulo y configurarlo para diferentes modos de operación. El registro SPIxCON es el encargado de especificar, entre otras características, la velocidad del reloj, la selección de modo maestro o esclavo, la elección de una comunicación *byte* o *word* y la operación de los pins. El registro SPIxSTAT indica varias condiciones de estado tales como Receive Overflow, Transmit Buffer Full y Receive Buffer Full, y, además, contiene un bit que habilita y deshabilita el módulo (Martín, 2013).

R/W-0	U-0	R/W-0	U-0	U-0	U-0	U-0	U-0
SPIEN	—	SPISIDL	—	—	—	—	—
bit 15							bit 8
U-0	R/C-0	U-0	U-0	U-0	U-0	R-0	R-0
—	SPIROV	—	—	—	—	SPITBF	SPIRBF
bit 7							bit 0

a)

U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	—	DISSCK	DISSDO	MODE16	SMP	CKE ⁽¹⁾
bit 15							bit 8
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
SSEN	CKP	MSTEN	SPRE<2:0>			PPRE<1:0>	
bit 7							bit 0

b)

Figura 3.3 Registros a) SPISTAT y b) SPICON.

Analizando la figura anterior se pueden señalar los bits más importantes para la configuración de los registros de estado y control. En el caso del registro SPISTAT, el bit (SPIEN=1) es el encargado de habilitar el módulo SPI, el bit (SPISIDL=0) permite el funcionamiento del módulo en estado IDLE4. Los otros bits del registro son el indicador de desbordamiento en la recepción <SPIROV> y los bits de estado de los buffers de transmisión <SPITBF> y recepción <SPIRBF>.

En el registro SPICON hay que destacar los bits encargados para establecer la frecuencia, en caso de modo Maestro, SPRE<2:0> y PRE<1:0>, cuya función es realizar un preescalado. Los bits <CKP>, <SMP> y <CKE> tiene como función sincronizar el reloj (SCK) con los datos a transmitir o a recibir. El bit <MODE16>permite la selección de la comunicación por byte (8 bits) o por palabra (16 bits). Por ultimo destacar el bit <MSTEN>con el cual se puede seleccionar el modo como Maestro o como Esclavo (Ruiz, 2010).

La activación de este módulo se produce mediante la interrupción asignada por el receptor Rx del módulo UART del DsPIC. Una vez ejecutado el comando se envían las tramas de datos hacia los convertidores digito-analógico, ajustando el valor del CS para cada uno en dependencia del dato a enviar para lograr una correcta comunicación.

3.5 Adquisición.

Para la adquisición de las señales analógicas se utilizó el módulo ADC del DsPIC. Luego de su conversión, los datos son tratados y almacenados en una arreglo tipo *char* para posteriormente ser enviados al Pixhawk por medio del módulo UART. Estos datos deben ser actualizados en un período de muestreo constante y sincrónico con el resto de los procesos por lo que es gobernado por el Timer implementado que se ejecuta cada 20ms, mientras tanto cada vez que se actualizan los datos, estos son reescritos en el arreglo mencionado anteriormente. De igual manera se ejecuta la adquisición para las señales digitales.

Cada dato de una variable analógica corresponde a un *float* que es equivalente a 4 *bytes*, mientras que cada dato de una variable digital ocupa solamente 1 *byte*, excepto el que pertenece al sensor rpm que corresponde a un float, por lo que las variables de las señales analógicas ocupan un total de 12 *bytes* y las señales digitales 6 *bytes*. Estos valores son necesarios a tomar en cuenta a la hora de configurar la trama de datos que será enviada.

Para el proceso de medición de la velocidad del motor (rpm) se implementó un *Timer* en modo *Gate*, esta característica mide el tiempo en que es ejecutado un pulso, tanto subida como bajada. Como fue mencionado en el capítulo anterior, la frecuencia del tren de pulsos adquirido por la entrada correspondiente a este sensor es proporcional a la velocidad del motor, por tanto una vez determinada esta frecuencia se puede calcular el valor en rpm con la siguiente ecuación.

$$Velocidad = \frac{1}{2} \frac{F}{2\pi} \times 60 \quad (\text{RPM})$$

3.6 Análisis económico.

El costo de producción del HRC-AUV es mucho menor que el de los AUVs que se comercializan actualmente en el mundo. El HRC-AUV es un vehículo de bajo costo que ha sido totalmente diseñado y construido en Cuba. Todo el equipamiento de hardware y sensores instalados clasifican en la gama baja que proporcionan sus fabricantes. La solución brindada en esta investigación sin dudas contribuye a disminuir el valor de adquisición de los componentes del HRC-AUV, lo que demuestra que es posible realizar en Cuba aplicaciones como esta, para la que solo sería posible su implementación en países desarrollados (Tkachova, 2012).

A pesar de que el mayor porcentaje de costo corresponde a la carcasa, los elementos de propulsión y actuadores, el hardware que representa una pequeña parte del costo total, es el que determina mayoritariamente el índice de desempeño y funcionalidad que tendrá este vehículo en el momento de realizar las misiones, por tanto para alcanzar una conclusión sobre la factibilidad económica de la nueva arquitectura de hardware propuesta es necesario realizar una comparación entre dichos componentes en ambas.

Los principales componentes utilizados en la arquitectura de hardware definida por el CIDNAV y el GARP y que son reemplazados en la arquitectura de hardware propuesta por el kit completo de Pixhawk son: PC-104 modelo PCM-3362 de la firma Advantech, (IMU): MTi de la firma Xsens, GPS: XL12 de la firma Garmin (Martínez et al., 2013) y Modem X24-019-DK.

Estos componentes poseen un costo total aproximado de 5946 USD (Advantech, 2016, Gómez, 2009, Tkachova, 2012), sumado a un aproximado del costo de diseño de la unidad

DsPIC, de 80 USD. Por otra parte, la nueva arquitectura de hardware está conformada ,como anteriormente se mencionó, por el módulo Pixhawk con un costo de 200 USD y su kit de accesorios con GPS y Compas incluidos que alcanza un valor aproximado de 90 USD (Robotics, 2016), mientras que el hardware de bajo nivel posee un valor aproximado de 40 USD.

Por lo que este tipo de inversión resulta económicamente factible para nuestro país y abre enormes posibilidades para el desarrollo nacional de vehículos autónomos tanto subacuáticos como de otros tipos.

3.7 Consideraciones finales del capítulo.

En este capítulo luego de analizar las características del software de bajo nivel implementado se llevó a cabo el análisis de los principales hilos del programa de manera detallada. Se mostró la configuración realizada para los distintos módulos y se establecieron los protocolos de comunicación y la conformación de las tramas de datos. Finalmente se realizó un análisis económico para la arquitectura de hardware propuesta.

CONCLUSIONES Y RECOMENDACIONES

Conclusiones

El desarrollo científico de nuevas tecnologías para AUVs ha sido un punto de mira por parte de un gran número de investigadores, debido al gran número de aplicaciones y soluciones de control y navegación que se pueden llevar a cabo de forma más rápida y exacta. Con la realización de este trabajo se arribaron a las siguientes conclusiones.

- Entre las diferentes plataformas de hardware analizadas, Pixhawk resulta una opción muy prometedora para el desarrollo de aplicaciones para vehículos autónomos, entre ellos los subacuáticos, ya que presenta las mayores capacidades de integración electrónica y un soporte firme y creciente por parte de la comunidad de desarrolladores.
- Con el uso de un autopiloto de código abierto tipo Pixhawk y agregándole interfaces de E/S adicionales se puede extender su funcionalidad a otros tipos de vehículos, tales como vehículos autónomos de gran envergadura.
- Con el empleo de microcontroladores y elementos de hardware discretos se puede diseñar interfaces de E/S compatibles con plataformas de alto nivel, permitiéndole extender las funcionalidades de la misma.
- Con la utilización de herramientas para aplicaciones de bajo nivel para microcontroladores se pueden lograr el manejo de diferentes señales así como comunicaciones de una tarjeta de E/S con un autopiloto de altas prestaciones para mejorar las capacidades de comunicación y control del mismo.

Recomendaciones

- Llevar a cabo el desarrollo del autopiloto propuesto en esta investigación para validar su funcionamiento mediante pruebas experimentales.
- Realizar un análisis de los sensores que son actualmente utilizados en otros proyectos de AUVs para así optimizar la calidad del proceso de adquisición de datos.
- Tomar en cuenta el posible uso de señales digitales vía serie para el control de los actuadores del sistema para eliminar componentes en el sistema empotrado y eliminar parámetros negativos de las señales analógicas como el ruido.

REFERENCIAS BIBLIOGRÁFICAS

3D ROBOTICS, I. 2014. Pixhawk Kit User Guide. 8.

ADVANTECH. 2016. *PCM-3362 - Intel® Atom™ N450 PC/104-Plus SBC, VGA, LVDS, Ethernet, USB, COM, SATA, Onboard Flash - Advantech* [Online]. Available: http://www.advantech.com/products/1-2jkltu/pcm-3362/mod_2b69da4c-d506-4aea-8cc3-f6e58e3cbc87.

AERORECORD. 2014. *Pixhawk autopiloto 32 bits* [Online]. Available: <http://www.aerorecord.es/pixhawk-autopiloto-32/>.

ALVAREZ, H. D. 2014. *Arquitectura de Hardware y Software basada en Smartphone para el control de UAV*. Universidad Central “Marta Abreu” de Las Villas.

ARDUPILOT. 2016. *Pixhawk Overview* [Online]. ArduPilot Dev Team. Available: <http://ardupilot.org/copter/docs/common-autopilots.html>.

CAMPO, D. C. 2014. *Estudio de pilotos automáticos enfocados en la configuración de ala volante para vehículos aéreos no tripulados*. Universidad EAFIT.

CIRCELLI, G. 2015. *Microcontroladores o Arduino, ¿no es lo mismo?* [Online]. Available: <https://panamahitek.com/>.

CRUZ, N. A. 2011. *AUTONOMOUS UNDERWATER VEHICLES*, Croatia, InTech.

DÍAZ, J. J. 2012. *Actualización de la arquitectura de hardware del CIDNAV-AUV*. Universidad Central “Marta Abreu” de las Villas.

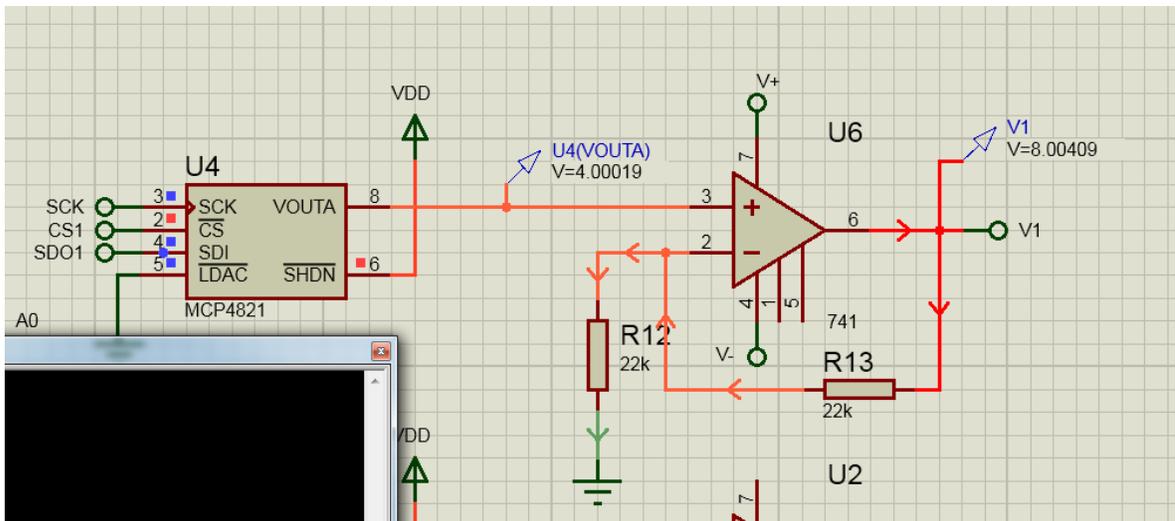
GÓMEZ, A. D. 2009. *Diseño de la arquitectura de hardware de un Vehículo Submarino Autónomo*. Universidad Central “Marta Abreu” de Las Villas.

- GUERRA, C. E. 2010. *Diseño e implementación de hardware y software de bajo nivel para vehículo submarino autónomo*. Universidad Central “Marta Abreu” de Las Villas.
- INVENSENSE, I. 2015. *InvenSense Online Store* [Online]. Available: <http://store.invensense.com/ProductDetail/>.
- MARTÍN, R. C. 2013. *DESARROLLO DE UNA LIBRERÍA PARA MICROCONTROLADOR dsPIC33F PARA LA LECTURA DE TARJETAS MICROSD EN UN DISPOSITIVO INERCIAL*. Universidad Politécnica de Catalunya.
- MARTÍNEZ, A., RODRIGUEZ, Y., HERNÁNDEZ, L., GUERRA, C., LEMUS, J. & SAHLI, H. 2013. Diseño de AUV.Arquitectura de hardware y software. *Revista Iberoamericana de Automática e Informática industrial*.
- MARTINEZ, Y. 2012. *Estrategias de Control de Profundidad para un Vehículo Autónomo Subacuático*., Universidad Central “Marta Abreu” de Las Villas.
- MEAS. 2012. *Precision Micro Barometer Module* MS5611* [Online]. Available: <http://www.meas-spec.com/downloads/MS5611-01BA03.pdf>.
- MICROCHIP 2007. dsPIC33FJ32MC202/204 and dsPIC33FJ16MC304 Data Sheet. Microchip Technology Inc.
- ROBOTICS, D. 2016. *3DR Pixhawk* [Online]. Available: <https://store.3dr.com/t/pixhawk>.
- RUBEN, J. 2013. *Programación Multitarea en Microcontroladores* [Online]. Available: <http://www.geekfactory.mx/>.
- RUIZ, D. V. 2010. *Estado del arte en robótica submarina*. Universidad Politécnica de Valencia.
- STMICROELECTRONICS 2014. STM32F427xx STM32F429xx, Datasheet.
- TKACHOVA, A. F. 2012. *Estrategia de control no convencional para un vehículo autónomo subacuático*. Universidad Central “Marta Abreu” de Las Villas.
- VIQUE, F. I. 2008. *Prototipo electrónico para la valoración de amplificadores operacionales*., UNIVERSITAT POLITÈCNICA DE CATALUNYA.

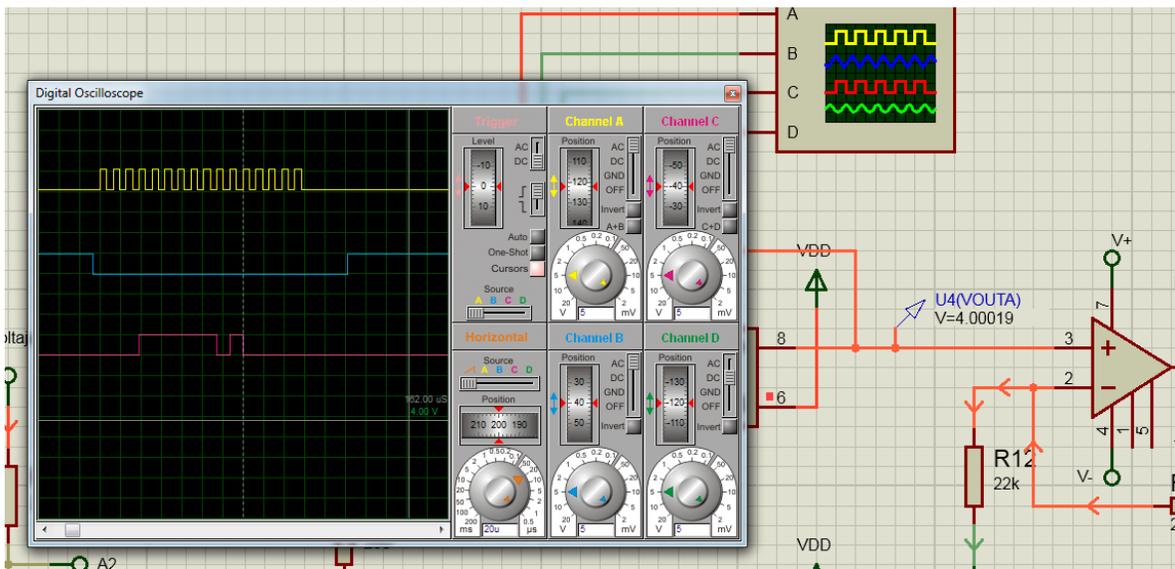
WAKITA, N., HIROKAWA, K., ICHIKAWA, T. & YAMAUCHI, Y. 2010. Development of Autonomous Underwater Vehicle (AUV) for Exploring Deep Sea Marine Mineral Resources. *Mitsubishi Heavy Industries Technical Review*.

ANEXOS

Anexo I Convertidor DAC.



Anexo II Transmisión SPI.



Anexo III Transmisión de datos mediante el módulo UART.