



Universidad de Valladolid



**ESCUELA DE INGENIERÍAS
INDUSTRIALES**

UNIVERSIDAD DE VALLADOLID

ESCUELA DE INGENIERIAS INDUSTRIALES

Grado en Ingeniería en Electrónica Industrial y Automática

**DISEÑO, DESARROLLO Y CONSTRUCCIÓN DE UN
ROBOT HEXÁPODO PARA INSPECCIÓN DE ÁREAS
DE DIFÍCIL ACCESO**

Autor:

Medina Rodríguez, Luis Carlos

Tutor:

**De la Fuente López, Eusebio
Departamento de Ingeniería de
Sistemas y Automática**

Valladolid, Octubre, 2024.

Diseño, desarrollo y construcción de un robot hexápodo para inspección de áreas de difícil acceso.

Diseño, desarrollo y construcción de un robot hexápodo para inspección de áreas de difícil acceso.

Resumen

Este TFG aborda el diseño y construcción de un robot multi-pata controlado a distancia, destinado a la inspección y rescate en zonas de difícil acceso, tales como derrumbes, áreas militares o cavidades.

Se trata de un robot hexápodo dotado de tres grados de libertad por extremidad, cuyo movimiento recae en servomotores, que le permiten desplazarse de forma estable sobre superficies irregulares. Cuenta además con control remoto y cámara integrada con retransmisión en tiempo real.

Surge con el objetivo de ser accesible a cualquier persona o entidad, gracias al limitado presupuesto y a la fácil replicabilidad de sus componentes, diseñados y producidos íntegramente a través de fabricación aditiva.

La integración del gran número de actuadores y sensores, así como la gestión de sus señales hace de este proyecto un gran reto multidisciplinar que pretende sentar las bases para desarrollos futuros que aporten nuevos puntos de vista y funcionalidades.

Palabras Clave

Robot, Hexápodo, Servomotor, Aditiva, Remoto.

Abstract

This TFG addresses the design and construction of a remotely controlled multi-legged robot intended for inspection and rescue operations in areas of difficult access, such as collapsed structures, military zones, or cavities.

The robot is a hexapod equipped with three degrees of freedom per limb, powered by servomotors, which allow it to move stably over uneven surfaces. It is also equipped with remote control and an integrated camera with real-time video transmission.

The project is designed to be accessible to any individual or organization, thanks to its low budget and the easy replicability of its components, which are entirely designed and produced through additive manufacturing.

The integration of a large number of actuators and sensors, as well as the management of their signals, makes this project a significant multidisciplinary challenge that aims to lay the groundwork for future developments, bringing new perspectives and functionalities.

Keywords

Robot, Hexapod, Servomotor, Additive, Remote.

Diseño, desarrollo y construcción de un robot hexápodo para inspección de áreas de difícil acceso.

Contenido

1. Introducción.....	1
1.1 Descripción general del proyecto.....	1
1.2 Objetivos del proyecto.....	1
2. Estado del arte.....	3
2.1 Robótica móvil.....	3
2.2 Robots multi-pata.....	3
3. Elementos clave y planificación.....	11
3.1 Análisis de requisitos y especificaciones.....	11
3.2 Tecnologías y componentes clave del robot.....	13
3.2.1 Cuerpo.....	13
3.2.2 Patas.....	13
3.2.3 Componentes electrónicos clave.....	13
3.3 Planificación del proyecto.....	15
4. Diseño electrónico y selección de componentes.....	17
4.1 Selección de los actuadores.....	17
4.1.1 Comparativa de servomotores.....	18
4.1.2 Elección del servomotor.....	21
4.2 Selección de la placa de control.....	23
4.2.1 Comparativa de placas de control.....	23
4.2.2 Elección de la placa de control.....	28
4.3 Selección de controladora de servomotores.....	29
4.3.1 Comparativa de controladoras de servomotores.....	29
4.3.2 Elección de la controladora de servomotores.....	32
4.4 Sensores de movimiento.....	33
4.5 Batería y gestión de energía.....	34
4.5.1 Comparativa de tipos de baterías.....	35
4.5.2 Elección del tipo de batería.....	36
4.5.3 Dimensionado de la batería.....	36
4.5.4 Comparativa de baterías LiPo.....	39
4.5.5 Elección de la batería de potencia.....	42

Diseño, desarrollo y construcción de un robot hexápodo para inspección de áreas de difícil acceso.

4.5.6	Elección de la batería de control.....	43
4.6	Selección de la cámara	44
4.6.1	Características básicas	44
4.6.2	Elección de la cámara	46
4.7	Selección de componentes electrónicos auxiliares	47
4.7.1	Multiplexor de señales	47
4.7.1	Regulador de tensión	49
4.7.2	Ventilador de refrigeración del regulador	51
4.8	Mando de control	51
4.9	Esquemas de conexión.	52
4.9.1	Esquema de alimentación.....	52
4.9.2	Esquema de control	53
5.	Diseño y construcción del robot.....	55
5.1	Diseño del robot.....	55
5.1.1	Cuerpo del robot.....	56
5.1.2	Patas del robot. Integración de los servos y potenciómetros.....	59
5.2	Disposición de los elementos electrónicos.....	65
5.3	Puntos de apoyo de las patas	67
5.4	Soporte de la pantalla	68
5.5	Fabricación del robot.....	69
5.6	Montaje del robot.....	77
6.	Control y programación	81
6.1	Configuración de la controladora de servos	81
6.2	Configuración y programación del ESP32.....	83
6.3	Configuración y funciones del mando de control.....	84
6.4	Control de movimiento del hexápodo	88
6.4.1	Control de los servomotores	88
6.4.2	Desplazamiento del robot.....	90
6.4.3	Giro del robot.....	95
6.4.4	Integración de las señales de feedback de las extremidades.....	96
7.	Pruebas y validación	99

Diseño, desarrollo y construcción de un robot hexápodo para inspección de áreas de difícil acceso.

7.1	Evaluación de la movilidad del robot	99
7.2	Verificación de la programación.....	100
7.2.1	Verificación del control a través del mando.....	100
8.	Conclusiones	103
8.1	Cumplimiento de objetivos.....	103
8.2	Posibles mejoras y líneas futuras	104
9.	Referencias y fuentes.....	107
9.1	Referencias	107
9.2	Referencias de imágenes.....	109
10.	Anexos.....	113
10.1	Diagramas y esquemas técnicos.....	113
10.1.1	Diagrama de conexiones.....	113
10.2	Fotos y vídeos del funcionamiento	115
10.3	Código fuente relevante.....	116
10.3.1	Código de Arduino	116

Diseño, desarrollo y construcción de un robot hexápodo para inspección de áreas de difícil acceso.

1. Introducción

1.1 Descripción general del proyecto

En el ámbito de la robótica y la automatización, los robots hexápodos destacan por su capacidad de moverse de manera estable sobre terrenos irregulares, replicando los movimientos de insectos y arácnidos. Este trabajo fin de grado se centra en el control y programación de un robot de diseño propio, construido a través de fabricación aditiva, con el objetivo principal de acceder e inspeccionar zonas de difícil acceso donde otros robots o métodos convencionales pueden fallar. Estas zonas pueden abarcar desde cavidades donde por las dimensiones de la entrada el ser humano no es capaz de acceder, derrumbes para la búsqueda de supervivientes o bien inspecciones de terrenos peligrosos con objetivo militar. Por ello, el robot deberá ser ligero para su transporte que podrá ser a pie hasta el lugar del despliegue y sus dimensiones deberán ser lo suficientemente reducidas como para poder entrar donde el ser humano no es capaz. Para ello se busca desarrollar una configuración de movimiento hexápodo, dotado de un sistema de control, que gobernará tanto las articulaciones, como una cámara que deberá transmitir imagen en tiempo real. Su control se realizará a través de un mando de control remoto.

1.2 Objetivos del proyecto

El objetivo principal de este proyecto es desarrollar un hexápodo robótico capaz de explorar y operar en entornos de difícil acceso, donde su capacidad para adaptarse a terrenos irregulares será fundamental. Además, se busca que sus componentes estructurales sean fácilmente replicables con el fin de poder sustituirlos en el probable caso de daño y que su coste de producción sea reducido. Para lograr este objetivo, se han definido los siguientes objetivos específicos:

Diseño y fabricación del cuerpo y las patas del hexápodo: Utilizando tecnologías de fabricación aditiva, se crearán las estructuras necesarias para asegurar la movilidad y estabilidad del robot, optimizándolas para superar obstáculos y terrenos complicados.

Integración y configuración de componentes electrónicos: Se seleccionarán y configurarán adecuadamente los sistemas de control del movimiento, garantizando una ágil comunicación y control de los actuadores para lograr un desplazamiento estable.

Desarrollo de software de control: Se programará el sistema de control para recibir y procesar las señales del mando, así como el envío de señales de vídeo,

Diseño, desarrollo y construcción de un robot hexápodo para inspección de áreas de difícil acceso.

permitiendo un control remoto preciso del hexápodo, adaptándose a las necesidades del terreno.

Pruebas y validación del sistema: Se evaluará la movilidad, estabilidad y capacidad del robot para operar en entornos difíciles, asegurando que cumple con los requisitos y especificaciones establecidos para acceder y moverse por estas áreas.

Este proyecto no solo busca culminar con la construcción de un robot hexápodo funcional capaz de operar en entornos complejos, sino también establecer una base sólida para futuros desarrollos y mejoras en el campo de la robótica móvil y la automatización. A través de una documentación detallada, se espera contribuir al conocimiento y avances en esta disciplina, fomentando la investigación y desarrollo de soluciones robóticas más eficientes y versátiles.

2. Estado del arte

2.1 Robótica móvil

La robótica móvil es una rama de la robótica que se centra en el desarrollo de robots capaces de moverse en su entorno. Estos robots pueden desplazarse de manera autónoma o controlada, adaptándose a diferentes tipos de terrenos y condiciones. Además, abarca una amplia variedad de aplicaciones, desde el reconocimiento de terrenos inexplorados, hasta la automatización de tareas industriales y domésticas.

Su desarrollo ha avanzado significativamente en las últimas décadas, gracias a los progresos en sensores, actuadores, sistemas de control y algoritmos de navegación. Los robots móviles utilizan diversos métodos de locomoción, como ruedas, orugas y patas, cada uno con sus ventajas y desventajas en función del entorno de aplicación.

2.2 Robots multi-pata

Los robots hexápodos, inspirados en la estructura y movimientos de los insectos, son una subcategoría de la robótica móvil que utiliza seis patas para desplazarse. Esta configuración les confiere una mayor estabilidad y capacidad para moverse sobre terrenos irregulares, en comparación con los robots con ruedas o con menor número de patas como los cuadrúpedos.

Los hexápodos son conocidos por su capacidad de adaptarse a diversas superficies y su robustez ante la pérdida de una o más patas. Estos robots emplean diversas técnicas de locomoción, como el movimiento alterno de las patas (gait alternante o tripod gate) y el movimiento de onda (wave gait), lo que les permite mantener el equilibrio y la estabilidad mientras se desplazan.

En los últimos años, los avances en la tecnología de servomotores, controladores y materiales han permitido el desarrollo de hexápodos más sofisticados y eficientes. Estos robots se utilizan en una variedad de aplicaciones, incluyendo exploración espacial, búsqueda y rescate, inspección de infraestructuras y entretenimiento.

A continuación, se muestra una breve descripción de los principales desarrollos de robots multi-pata en la historia reciente:

Diseño, desarrollo y construcción de un robot hexápodo para inspección de áreas de difícil acceso.

BigDog:



Figura 1. Robot BigDog. Fuente [1]

BigDog [1] es un robot cuadrúpedo, desarrollado por Boston Dynamics, que ha influido significativamente en el diseño de robots con patas debido a su avanzada tecnología de locomoción. Boston Dynamics también ha desarrollado otros robots cuadrúpedos, como el mostrado en la *Figura 2*, de dimensiones más reducidas para un uso más comercial dotado con un manipulador, llamado “Spot”.



Figura 2. Robot Spot. Fuente [2]

Gracias a sus sensores avanzados y actuadores hidráulicos, es capaz de mantener el equilibrio y moverse de manera autónoma en terrenos difíciles, como zonas empedradas, con pendiente o incluso sobre el hielo.

Como aplicaciones principales figuran el transporte de carga en terrenos difíciles para aplicaciones militares y de rescate.

Diseño, desarrollo y construcción de un robot hexápodo para inspección de áreas de difícil acceso.

SPIDAR:



Figura 3. Robot Spidar. Fuente [3]

SPIDAR [2] es un atípico robot cuadrúpedo desarrollado por el Instituto de Tecnología de Tokio. Está diseñado para operaciones en entornos peligrosos o inaccesibles para los humanos.

Posee una estructura ligera y flexible, propulsado por hélices, y sus patas están diseñadas para trepar y moverse sobre superficies irregulares. Tiene la capacidad de acceder hasta el área de trabajo por vía aérea para después operar en forma de cuadrúpedo.

Está enfocado para la exploración de áreas peligrosas, inspección de infraestructuras y operaciones de rescate.

T8X:



Figura 4. Robot T8X. Fuente [4]

Diseño, desarrollo y construcción de un robot hexápodo para inspección de áreas de difícil acceso.

El T8X [3] es un robot octópodo desarrollado por Robugtix. Es conocido por su apariencia realista de araña y sus movimientos fluidos.

Utiliza servomotores avanzados y un algoritmo de locomoción complejo para imitar los movimientos de una araña. Puede ser controlado mediante una aplicación móvil.

Sus objetivos principales son el entretenimiento, educación y demostraciones tecnológicas.

HEXA:



Figura 5. Robot Hexa. Fuente [5]

Hexa [4] es un robot hexápodo desarrollado por Vincross, una empresa de robótica china. Este robot está diseñado tanto para aplicaciones educativas como recreativas.

Viene equipado con una variedad de sensores, incluyendo cámaras y sensores de proximidad, y puede ser programado para realizar diferentes tareas.

Sus aplicaciones son la educación en robótica, investigación y desarrollo de aplicaciones personalizadas.

Diseño, desarrollo y construcción de un robot hexápodo para inspección de áreas de difícil acceso.

RHex:



Figura 6. Robot RHex. Fuente [6]

RHex (Robust Hexapod) [5] es un robot desarrollado por el laboratorio GRASP de la Universidad de Pensilvania. Es conocido por su diseño minimalista y su capacidad para moverse sobre terrenos extremadamente irregulares.

Utiliza patas flexibles y una locomoción de "gateo" que le permite adaptarse a obstáculos y superficies difíciles. Es capaz de subir escaleras, atravesar escombros y moverse en terrenos accidentados.

Ha sido diseñado para la exploración en terrenos complicados, búsqueda y rescate, y aplicaciones militares.

Diseño, desarrollo y construcción de un robot hexápodo para inspección de áreas de difícil acceso.

PhantomX Hexapod MKIII:



Figura 7. Robot PhantomX MKIII. Fuente [7]

Se trata de un kit de robot hexápodo desarrollado por Trossen Robotics [6]. Es popular en la comunidad de aficionados a la robótica y en aplicaciones educativas.

Se ha construido con servomotores de alta calidad y una estructura modular que permite a los usuarios personalizar y expandir sus capacidades.

Sus objetivos principales son la educación en robótica, investigación y proyectos de desarrollo personal.

Diseño, desarrollo y construcción de un robot hexápodo para inspección de áreas de difícil acceso.

MX-Phoenix:



Figura 8. Robot MX-Phoenix. Fuente [8]

MX-Phoenix [7] es un robot hexápodo desarrollado por Kåre Halvorsen, un ingeniero noruego, al que ha dotado de grandes cualidades a costa de una gran inversión en los componentes.

Este robot incorpora servomotores de altas prestaciones con un gran torque y realimentación. También han implementado un sistema de control comercial con múltiples funciones, que otorga un movimiento orgánico. Por último, las patas cuentan con un sensor de presión, para proporcionar otra realimentación y favorecer el movimiento estable en superficies irregulares.

Como principales aplicaciones figuran la investigación, exploración y aplicaciones militares.

En definitiva, han conseguido un robot con gran adaptabilidad para desplazarse por terrenos muy irregulares con un movimiento muy natural.

Diseño, desarrollo y construcción de un robot hexápodo para inspección de áreas de difícil acceso.

Conclusión

Ninguno de los robots mostrados hasta ahora cumple con todos los objetivos de este proyecto, que pretende desarrollar una funcionalidad y una capacidad para moverse por terrenos muy irregulares similar a este último, pero con un coste reducido y un menor tamaño.

El coste reducido permitiría que cualquier entidad pueda hacerse con una o varias unidades. Por ejemplo, dotaciones de bomberos. Esto permitirá generar también una plataforma de desarrollo para proyectos futuros, al contrario que los ejemplos anteriores, que pueden llegar a valer decenas de miles de euros en el caso de los desarrollos de Boston Dynamics, varios miles en el caso de Mx-Phoenix o con precios más asequibles, pero con una orientación más encaminada hacia el entretenimiento, con un diseño cerrado y sin funcionalidades de utilidad.

A su vez, su pequeño tamaño permitirá que pueda moverse por espacios reducidos, donde no podría llegar una persona, y al mismo tiempo lo convierte en fácilmente transportable.

3. Elementos clave y planificación

3.1 Análisis de requisitos y especificaciones

A continuación, se detallarán los principales requisitos y especificaciones del robot hexápodo desarrollado en este trabajo:

1. Requisitos funcionales:

- El hexápodo debe ser capaz de moverse en diferentes direcciones (adelante, detrás, izquierda, derecha), así como rotar sobre sí mismo.
- Debe poder superar obstáculos y adaptarse a terrenos irregulares, tales como oquedades, desniveles o escombros.
- Debe ser controlable de manera remota a través de un mando inalámbrico.
- Debe ser capaz de transmitir en tiempo real imágenes que proporcionen al operador toda la información necesaria para su manejo.

2. Especificaciones técnicas:

- **Dimensiones:** El cuerpo del hexápodo debe tener unas dimensiones que permitan albergar todos los componentes necesarios y proporcionar estabilidad, así como capacidad de superar obstáculos lo más grandes posibles, pero sin perder de vista la prioridad de que debe ser transportable, económico y capaz de entrar en cavidades pequeñas. Por ello deberá ajustarse lo máximo posible al tamaño de sus actuadores.
- **Peso:** El peso total, incluyendo la electrónica y la batería, debe ser lo suficientemente ligero para ser perfectamente transportable y no sobrecargar los actuadores elegidos. Además, cuanto mayor sea este, mayor será el consumo de los actuadores y menor la autonomía. Debería en todo caso ser no superior a los 3 Kg.
- **Actuadores:** Los motores deben tener un torque adecuado para mover las patas bajo las condiciones de carga previamente expuestas.
- **Sensores:** Debe contar con los sensores necesarios para conseguir un control fluido a pesar de los obstáculos. Se valorará

Diseño, desarrollo y construcción de un robot hexápodo para inspección de áreas de difícil acceso.

la inclusión de sensores de control de posición y de presión para las patas del robot.

- **Batería:** La capacidad de la batería debe permitir al menos una hora de operación continua.
- **Materiales:** Los materiales utilizados en la impresión 3D deben ser resistentes y ligeros.
- **Cámara:** El robot tiene que ser capaz de transmitir imágenes en tiempo real con buena definición para posibilitar su movimiento en remoto.

3. Requisitos de software:

- El software debe permitir el control preciso de cada servomotor para lograr un desplazamiento fluido.
- Debe ser capaz de recibir y procesar las señales del mando de control.
- Debe integrar la lectura de los sensores para mejorar sus funcionalidades.
- Debe incluir mecanismos de seguridad para evitar daños en caso de fallos.
- Debe ser capaz de transmitir imágenes en tiempo real a través de una cámara incorporada.

Este análisis proporciona una base sólida para el desarrollo del hexápodo, asegurando que todos los componentes y sistemas trabajen en conjunto para cumplir con los objetivos del proyecto. Al mantener un enfoque en la reducción de costos, se busca lograr un diseño funcional y eficiente, pero con un presupuesto accesible.

Diseño, desarrollo y construcción de un robot hexápodo para inspección de áreas de difícil acceso.

3.2 Tecnologías y componentes clave del robot

En este apartado se detallan las tecnologías y los componentes principales que serán utilizados en el desarrollo del hexápodo robótico, abarcando tanto la estructura física como los elementos electrónicos y de control. Toda la estructura del hexápodo será diseñada en CAD 3D y fabricada mediante fabricación aditiva.

3.2.1 Cuerpo

El diseño del cuerpo debe integrar los actuadores de rotación de las patas, así como toda la electrónica de control y la batería. Se optimizará la estructura para reducir al máximo el peso, mientras se mantiene la funcionalidad y robustez necesarias para soportar todos los componentes.

3.2.2 Patas

Cada pata debe estar compuesta de varias piezas articuladas, que proporcionen los grados de libertad necesarios. En este caso, se determina que se necesitarán como mínimo tres grados de libertad por pata para realizar los movimientos necesarios para el desplazamiento y para ser capaz de sortear obstáculos. En ellas deben estar integrados los motores, así como los sensores que otorguen la realimentación necesaria al sistema de control. Su diseño debe contemplar un punto de apoyo reducido, que permita al robot caminar sobre terrenos irregulares, sin comprometer la integridad de la estructura. Se evaluará la posibilidad de incluir un suplemento que permita aumentar la superficie de apoyo, para el caso de que el robot tenga que moverse por terrenos poco compactos o arenosos.

3.2.3 Componentes electrónicos clave

Circuito de control

Como todo robot, este debe estar gobernado por un microprocesador que gestione todas las señales de entrada y salida y actúe de nexo entre el operador y la máquina en sí. Se valorarán las diferentes opciones del mercado, así como la posibilidad de incluir circuitos auxiliares que funcionen de concentradores de señales, para el caso de los sensores o también de etapa de potencia para los actuadores, en el caso de que el microprocesador no sea capaz de gestionar simultáneamente todas las señales.

Actuadores

La parte móvil del robot viene definida por sus actuadores. En este caso, al igual que en los robots estudiados en el apartado anterior, lo más idóneo será

Diseño, desarrollo y construcción de un robot hexápodo para inspección de áreas de difícil acceso.

utilizar servomotores. Otra opción sería utilizar motores paso a paso, pero se descarta la opción debido al tamaño y a la necesidad de conocer en todo momento la posición de cada uno de los actuadores, así como gestionar toda esa información. En cuanto al número, se tomará el mínimo necesario. Teniendo en cuenta que se ha descrito en el apartado anterior que son imprescindibles tres grados de libertad por pata, se necesitarán 18 servomotores.

Sensores

Para facilitar el control de movimiento, se estudiará incluir en las patas sensores de posición de cada articulación, así como sensores de presión en las terminaciones de las patas, para detectar y adaptarse a los obstáculos que no se encuentren sobre el plano de partida.

Cámara

Como elemento clave para el control remoto del robot es imprescindible dotarlo de una cámara capaz de transmitir en tiempo real imágenes de lo que se encuentra ante el robot para poder anticiparse a los obstáculos y ser capaz de explorar el terreno.

Batería

Tal y como se ha indicado en el apartado anterior, es necesaria una batería de altas prestaciones, que permita dotar al robot de la autonomía requerida, así como suministrar la corriente necesaria para los motores y gestionar los picos de consumo que estos puedan requerir.

Mando de control remoto

Finalmente, para que el operador pueda gobernar el robot de manera remota, será necesario un control. Se valorarán las diferentes posibilidades.

Diseño, desarrollo y construcción de un robot hexápodo para inspección de áreas de difícil acceso.

3.3 Planificación del proyecto

La planificación del proyecto busca asegurar que todas las etapas se completen de manera ordenada y eficiente. A continuación, se describen las fases principales del proyecto:

1. Investigación y diseño:

- Investigación sobre robots hexápodos y sus componentes.
- Selección de los componentes del sistema de control.
- Diseño preliminar del cuerpo y las patas del hexápodo.
- Selección de materiales y tecnologías de impresión 3D.

2. Desarrollo del prototipo:

- Impresión 3D del cuerpo y las patas del hexápodo.
- Montaje inicial de las piezas impresas.
- Integración de los servomotores y los sensores.
- Ajustes y rediseño.

3. Integración de sistemas electrónicos:

- Configuración de los sistemas de control de movimiento del robot.
- Programación del sistema de control del robot integrando las señales proporcionadas por los sensores.
- Configuración del mando de control remoto e integración de las señales en la programación.
- Integración de la cámara.

4. Pruebas y validación:

- Pruebas iniciales de movilidad y estabilidad.
- Ajuste y calibración de los servomotores y la programación.
- Validación del control a través del mando.
- Evaluación de la realimentación proporcionada por los diferentes sensores.

5. Documentación y presentación:

- Elaboración de la documentación técnica y el informe final.
- Preparación de la presentación del proyecto.

Diseño, desarrollo y construcción de un robot hexápodo para inspección de áreas de difícil acceso.

4. Diseño electrónico y selección de componentes

Para arrancar con el diseño de la electrónica se ha comenzado por la elección de los actuadores, ya que conocer sus características se hace indispensable para una correcta elección del resto de elementos.

4.1 Selección de los actuadores

Como actuadores son opciones valorables los motores paso a paso y los servomotores.

Los primeros tienen la ventaja de ser más baratos. Sin embargo, no disponen de un control de lazo cerrado, muy necesario para la aplicación que se busca. Los servomotores incluyen un sensor de posición, generalmente un encoder o un potenciómetro, que proporciona retroalimentación continua al controlador. Esto permite un control en lazo cerrado del motor, ajustando su operación para alcanzar la posición deseada con precisión, ya que ajustan su salida en función de la retroalimentación del sistema para minimizar errores.

También son capaces de proporcionar un par constante en un rango de velocidades y un par elevado en los arranques, lo que es ideal para aplicaciones de alta carga, además de estar diseñados para ser compactos.

Por otra parte, al necesitar tres grados de libertad por pata serán necesarios dieciocho servomotores.

Se han descartado de salida opciones con costes muy elevados, como los utilizados en el robot anteriormente visto, Mx-Phoenix. Se trata de los *MX-64T* y *MX-106T* de *Dynamixel*, unos motores con unas prestaciones muy elevadas, al igual que su precio que se acerca a los 450€ a 700€ por unidad respectivamente. Además, su chasis de aluminio y sus dimensiones, aportarían cerca de 3Kg al peso total del robot, lo que haría que no fuera indicado para la aplicación.

Para seleccionar los más adecuados para el hexápodo robótico, es fundamental considerar varios factores, como torque, velocidad, precisión, durabilidad y costo. A continuación, se presenta una comparativa detallada de diferentes servomotores disponibles en el mercado.

Diseño, desarrollo y construcción de un robot hexápodo para inspección de áreas de difícil acceso.

4.1.1 Comparativa de servomotores

Servomotor SG90 de TowerPro



Figura 9. Servomotor SG90. Fuente [9]

Características:

- Torque: 1.8 kg/cm (4.8V)
- Velocidad: 0.12 s/60° (4.8V)
- Peso: 9 g
- Dimensiones: 23 x 12.2 x 29 mm
- Voltaje de operación: 4.8V a 6.0V

Es el más económico de las opciones contempladas. Muy ligero e ideal para aplicaciones donde el peso es un factor crítico [8].

Sin embargo, su bajo torque hace que no sea adecuado para aplicaciones en las que se requiere mover cargas pesadas.

También hay que considerar que los engranajes de su reductora son de plástico, lo que hace que sean menos robustos, y por lo tanto más propenso a averías en comparación con opciones metálicas, siendo una opción menos duradera.

Diseño, desarrollo y construcción de un robot hexápodo para inspección de áreas de difícil acceso.

Servomotor DS3218 de JX



Figura 10. Servomotor DS3218. Fuente [10]

Características:

- Torque: 20 kg/cm (6.8V)
- Velocidad: 0.16 s/60° (6.8V)
- Peso: 62 g
- Dimensiones: 40 x 20 x 40,5 mm
- Voltaje de operación: 4.8V a 6.8V

En contraposición a los anteriores, estos servomotores [9] tienen un alto torque, lo que lo hace adecuado para mover cargas pesadas, y su reductora está compuesta por engranajes metálicos, por lo que tienen una mayor durabilidad.

Por contra, tiene un coste bastante más elevado y es moderadamente pesado, aunque se compensa con el torque que es capaz de ejercer, por lo que es manejable para el hexápodo.

Diseño, desarrollo y construcción de un robot hexápodo para inspección de áreas de difícil acceso.

Servomotor MG995 de TowerPro



Figura 11. Servomotor MG995. Fuente [11]

Características:

- Torque: 13 kg·cm (4,8V) y 15kg/cm (6V)
- Velocidad: 0.17 s/60° (6V)
- Peso: 55 g
- Dimensiones: 40.7 x 19.7 x 42.9 mm
- Voltaje de operación: 4.8V a 7.2V

Su buen torque [10] lo hace adecuado para aplicaciones en las que se requiere un esfuerzo considerable.

En cuanto a su durabilidad, cuenta con engranajes metálicos al igual que el modelo anterior, lo que garantiza una larga vida útil.

La ventaja de este modelo de servomotores es que tienen un precio relativamente económico para sus prestaciones.

Si se busca un buen balance entre par, precio y peso, este motor es una buena opción.

Diseño, desarrollo y construcción de un robot hexápodo para inspección de áreas de difícil acceso.

Servomotor MG996R de TowerPro



Figura 12. Servomotor MG996R. Fuente [12]

Características:

- Torque: 9.4 kg·cm (4,8V) y 11kg/cm (6V)
- Velocidad: 0.17 s/60° (6V)
- Peso: 55 g
- Dimensiones: 40.7 x 19.7 x 42.9 mm
- Voltaje de operación: 4.8V a 7.2V

Se trata de una evolución del modelo anterior. Para este modelo han sacrificado algo de par y han mejorado la precisión en el movimiento.

Físicamente, las reductoras son muy similares, construidas en los mismos metales, por lo que cuenta con las mismas características de resistencia y durabilidad que el anterior.

Tiene un precio bastante similar, aunque algo más elevado. Si lo que se busca es una alta precisión en los movimientos, sería una buena opción.

4.1.2 Elección del servomotor

Después de evaluar las características, ventajas y desventajas de cada opción, el MG995 de TowerPro se destaca como la mejor elección para el hexápodo robótico. A continuación, se resumen las razones para esta elección:

- Torque adecuado: Con un torque de 15 kg/cm a 6V, el MG995 proporciona suficiente par para mover las patas del hexápodo de manera efectiva, incluso en terrenos irregulares.

Diseño, desarrollo y construcción de un robot hexápodo para inspección de áreas de difícil acceso.

- Durabilidad: Los engranajes metálicos aseguran que el servomotor pueda soportar cargas pesadas y uso prolongado sin fallos significativos.
- Costo: Aunque no es el servomotor más barato, el MG995 ofrece un excelente equilibrio entre precio y rendimiento, lo que es crucial para mantener el proyecto dentro de un presupuesto reducido sin sacrificar calidad.
- Compatibilidad y disponibilidad: El MG995 es ampliamente compatible con la mayoría de controladores y es fácilmente disponible en el mercado, facilitando su adquisición y reemplazo en caso de necesidad.

En conclusión, el servomotor MG995 de TowerPro ha sido seleccionado como la opción más adecuada para el hexápodo robótico debido a su combinación de alto torque, materiales, durabilidad, costo razonable y disponibilidad. Ante un coste muy similar frente al MG996R, ha primado el par a la precisión, dado que no es necesario un ajuste por debajo del grado de giro en los movimientos al realizar el avance. Esta elección asegura que el robot pueda operar de manera eficiente, cumpliendo con los requisitos del proyecto. En este caso, el conjunto de servomotores aporta algo menos de 1kg al peso total del robot.

Diseño, desarrollo y construcción de un robot hexápodo para inspección de áreas de difícil acceso.

4.2 Selección de la placa de control

Para la selección de la placa de control se han considerado varias opciones disponibles en el mercado, evaluando sus características, rendimiento y costo, teniendo en cuenta los requerimientos de entradas y salidas necesarios para controlar los actuadores escogidos y la posibilidad de incorporar sensores de control de posición y de presión para las patas del robot. A continuación, se presenta una comparativa detallada de las opciones más relevantes.

4.2.1 Comparativa de placas de control

Arduino Mega 2560

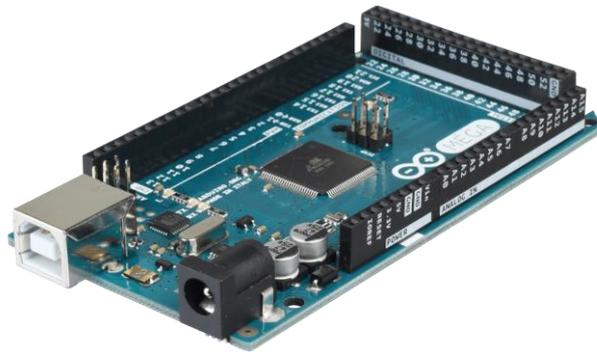


Figura 13. Arduino MEGA 2560. Fuente [13]

Características:

- Microcontrolador: ATmega2560.
- Frecuencia de reloj: 16 MHz.
- Memoria Flash: 256 KB.
- RAM: 8 KB.
- E/S: 54 digitales (15 con salida PWM), 16 canales ADC de 10 bits.
- Conectividad: USB tipo B, ICSP (In-Circuit Serial Programming).

Entre las ventajas de esta placa de control [11] destaca su gran número de pines, ya que cuenta con 54 pines digitales y 16 entradas analógicas. Esto permite que se puedan conectar gran cantidad de sensores y actuadores. También su gran capacidad de memoria, tanto flash como RAM, que permite almacenar y procesar programas más largos y complejos que sus versiones anteriores como el Arduino UNO o NANO. Por último, cuenta con una gran comunidad y soporte, lo que facilita el desarrollo y la resolución de problemas.

Diseño, desarrollo y construcción de un robot hexápodo para inspección de áreas de difícil acceso.

Como desventajas se puede destacar la falta de conectividad inalámbrica, que hace necesario un hardware adicional para conectarse vía Bluetooth o Wi-Fi. También su tamaño físico, que es más grande que otras placas y puede llegar a ser una limitación en aplicaciones con espacio reducido y por último su capacidad de memoria, que a pesar de ser mayor que sus predecesores, es sustancialmente inferior a las otras opciones que se plantean a continuación.

Arduino Due



Figura 14. Arduino DUE. Fuente [14]

Características:

- Microcontrolador: Atmel SAM3X8E ARM Cortex-M3.
- Frecuencia de reloj: 84 MHz.
- Memoria Flash: 512 KB.
- RAM: 96 KB.
- E/S: 54 digitales (12 con salida PWM), 12 canales ADC de 10 bits.
- Conectividad: USB On-The-Go, dos DAC (Digital to Analog Converter), dos TWI, tres UART, SPI.

Las ventajas de este tipo de placa es su alta capacidad de procesamiento, mucho mayor que la del Arduino Mega, lo que, sumado a su alta capacidad de memoria, el doble que la del Arduino Mega, le permite manejar tareas más demandantes y complejas.

Además, también cuenta con un gran número de entradas y salidas. 54 digitales y 12 analógicas.

La principal desventaja de este tipo de placa es que, igual que la anterior, carece de conectividad inalámbrica integrada, por lo que necesita módulos

Diseño, desarrollo y construcción de un robot hexápodo para inspección de áreas de difícil acceso.

adicionales para comunicación Wi-Fi o Bluetooth. También hay que tener en cuenta las dimensiones, ya que tiene el mismo tamaño que la placa anterior, y por último la complejidad de uso, ya que requiere conocimientos algo más avanzados de programación y manejo del entorno de desarrollo.

Raspberry Pi 4



Figura 15. Raspberry Pi 4. Fuente [15]

Características:

- Procesador: ARM Cortex-A72 quad-core 64-bit a 1.5 GHz.
- Memoria: Opciones de 2GB, 4GB, o 8GB de RAM.
- Almacenamiento: MicroSD.
- Conectividad: 2x USB 3.0, 2x USB 2.0, Ethernet, Wi-Fi 802.11ac, Bluetooth 5.0.
- E/S: 40 pines GPIO.
- Vídeo: Dual micro-HDMI con soporte para 4K.

Utilizar una Raspberry Pi 4 [12] como placa de control proporciona ventajas como tener conectividad inalámbrica integrada, tanto Bluetooth como Wi-fi y múltiples puertos USB, además de su capacidad de grabación de vídeo en 4K.

Además, tiene una mayor capacidad de procesamiento que las placas de control anteriores, pudiendo ejecutar sistemas operativos completos y aplicaciones complejas. Esta capacidad lo convierte en una opción muy versátil.

Diseño, desarrollo y construcción de un robot hexápodo para inspección de áreas de difícil acceso.

Sin embargo, al necesitar un sistema operativo y una configuración inicial, requiere mayor inversión de tiempo para su puesta en marcha y resolución de problemas que en otras placas directamente no existirían.

También tiene un consumo de energía relativamente alto, más alto que la mayoría de los microcontroladores, aunque para esta aplicación, teniendo en cuenta el consumo de los motores, esta no sería una gran desventaja.

ESP32



Figura 16. ESP32. Fuente [16]

Características:

- Microcontrolador: Tensilica Xtensa LX6 dual-core a 240 MHz.
- Memoria: 520 KB SRAM.
- Conectividad: Wi-Fi 802.11 b/g/n, Bluetooth v4.2 BR/EDR y BLE.
- E/S: 34 pines GPIO, 18 canales ADC de 12 bits, 2 DAC, SPI, I2C, UART.
- Periféricos: Sensores táctiles, sensor de temperatura, acelerador de criptografía.

Entre las ventajas de esta placa controladora [13] se encuentra la capacidad de ejecutar múltiples tareas simultáneamente gracias a su procesador dual-core de alta velocidad. Cuenta también con conectividad inalámbrica integrada, tanto Wi-Fi como Bluetooth, y aun así tiene un bajo costo, siendo muy accesible. Por último, cuenta con 34 pines GPIO, lo que permite conectar una gran variedad de sensores y actuadores.

La mayor desventaja de este tipo de placa es que requiere conocimientos algo más avanzados de programación, ya que la programación del ESP32 es más compleja que otras plataformas como Arduino, y que su comunidad en comparación no es tan extensa, aunque cada vez está cogiendo más importancia.

Diseño, desarrollo y construcción de un robot hexápodo para inspección de áreas de difícil acceso.

STM32 (Nucleo Board)

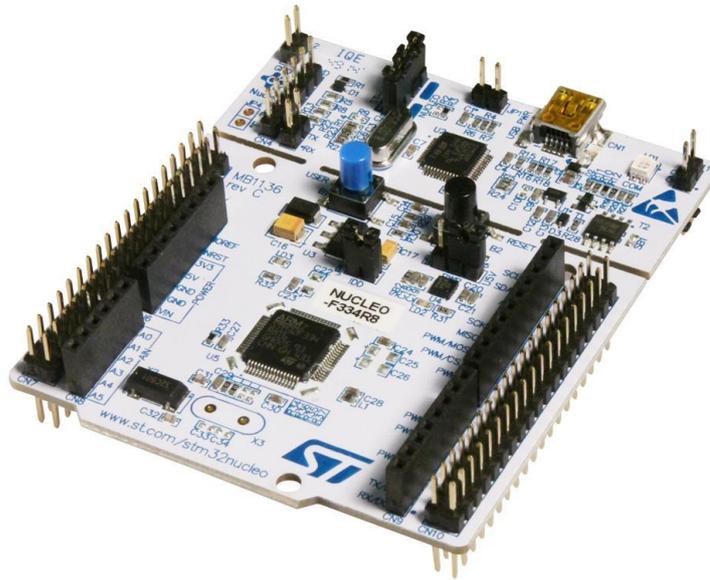


Figura 17. SMT32 (Nucleo Board). Fuente [17]

Características:

- Microcontrolador: ARM Cortex-M4 a 72 MHz.
- Memoria: 64 KB Flash, 20 KB SRAM.
- Conectividad: USB, UART, SPI, I2C, CAN.
- E/S: 37 pines GPIO.
- Periféricos: ADC de 12 bits, DAC de 12 bits, timers avanzados.

Las ventajas de este tipo de placas [14] son la alta eficiencia y rendimiento, lo que hace que tenga una excelente relación entre consumo de energía y capacidad de procesamiento.

Es también una placa muy flexible, al contar con una amplia gama de herramientas de desarrollo y bibliotecas, incluyendo STM32CubeMX y mbed, al igual que con una amplia gama de periféricos, como ADC y DAC de alta precisión.

Sin embargo, entre las desventajas de estas placas se encuentran la complejidad en la programación, que requiere un conocimiento más profundo de la arquitectura ARM y del entorno de desarrollo y al igual que en el caso anterior, la comunidad no es muy amplia.

Diseño, desarrollo y construcción de un robot hexápodo para inspección de áreas de difícil acceso.

4.2.2 Elección de la placa de control

La placa elegida es el *ESP32*, debido a su balance entre capacidad de procesamiento, conectividad inalámbrica integrada, número de pines I/O y costo. Su versatilidad y potencia convierten en una buena opción para el control del hexápodo.

Sin embargo, sus dieciséis de salidas PWM no son suficientes para controlar todos los servomotores, por lo que será necesaria la incorporación de una placa controladora de servos.

Por otra parte, el *ESP32* cuenta con dieciocho entradas analógicas, por lo que, en el caso de incorporar sensores para la lectura de la posición de los servos, se haría imprescindible también incluir un circuito multiplexor de señales para poder realizar un gran número de lecturas sin necesidad de ocupar los puertos del microcontrolador, debido a que, aunque el número sería suficiente, no quedarían puertos disponibles para añadir otros sensores.

Diseño, desarrollo y construcción de un robot hexápodo para inspección de áreas de difícil acceso.

4.3 Selección de controladora de servomotores

Como se ha indicado en el apartado anterior, las salidas del ESP32 no son suficientes para gestionar todas las señales para los dieciocho actuadores. Es por esto que se ha decidido añadir un circuito controlador de servomotores. Este tipo de circuitos funcionan como concentrador de señales de control, así como de etapa de potencia, pudiendo alimentar directamente los servos a través de él. Habitualmente disponen de puertos de comunicación, a través de los cuales intercambian los datos con el microcontrolador, sin necesidad de utilizar los pines GPIO.

4.3.1 Comparativa de controladoras de servomotores

Al igual que en el caso de las placas de control, para las controladoras de servomotores, se ha realizado un estudio de mercado, mostrándose a continuación las opciones más destacables:

Adafruit 16-Channel PWM/Servo Driver

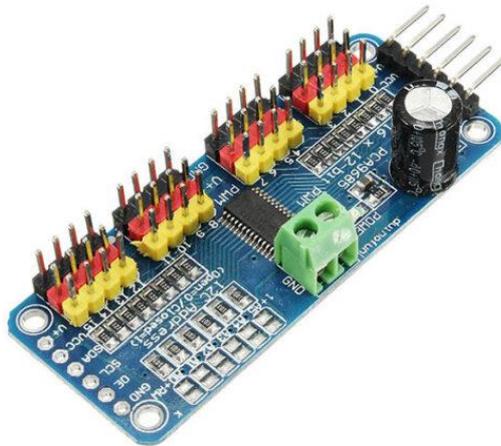


Figura 18. Adafruit 16-Channel Servo Driver. Fuente [18]

Características:

- Controlador: PCA9685.
- Canales: 16 canales de PWM/servo.
- Frecuencia PWM: 40-1000 Hz.
- Conectividad: I2C.
- Voltaje de operación: 2.3V a 5.5V.

Diseño, desarrollo y construcción de un robot hexápodo para inspección de áreas de difícil acceso.

Entre las principales ventajas de esta opción se encuentra su fácil integración, ya que al utilizar el bus I2C permite una fácil comunicación con los microcontroladores.

También es una controladora que permite conectar múltiples módulos en cadena para poder controlar mayor número de servomotores.

Es destacable también la amplia documentación y soporte con las que cuenta, incluyendo bibliotecas de software.

Como desventajas podemos encontrar que tiene un número limitado de canales, por lo que para conectar los 18 servos del hexápodo serían necesarios dos módulos, o bien un módulo y dos puertos adicionales de la placa de control.

Este cableado en cadena puede aumentar la complejidad del conexionado y la programación.

Lynxmotion SSC-32U USB Servo Controller



Figura 19. Lynxmotion SSC-32U Servo Controller. Fuente [19]

Características:

- Controlador: Atmel ATmega328P
- Canales: 32 canales de servo.
- Frecuencia PWM: Ajustable de 1.5 ms.
- Conectividad: USB, UART (TTL serial).
- Voltaje de operación: 6V a 9V.

Diseño, desarrollo y construcción de un robot hexápodo para inspección de áreas de difícil acceso.

Frente al anterior, esta controladora presenta la ventaja de tener mayor capacidad, contando con 32 canales, permitiendo el control de todos los servomotores y quedando espacio para expansiones futuras. También es fácilmente programable, siendo compatible con muchos lenguajes de programación y plataformas, ofreciendo además opciones de secuenciación y sincronización avanzadas.

La desventaja es que requiere un aprendizaje más profundo para aprovechar todas sus capacidades. Y también es más costosa que otras opciones de menor capacidad.

Pololu Mini Maestro 18-Channel USB Servo Controller



Figura 20. Pololu Mini Maestro Servo Controller. Fuente [20]

Características:

- Controlador: Microchip PIC18F25K50.
- Canales: 18 canales de servo.
- Frecuencia PWM: Ajustable hasta 20 ms (50 Hz) para servos estándar.
- Conectividad: USB, UART (TTL serial).
- Voltaje de operación: 5V a 16V.

Esta opción cuenta con suficientes canales, siendo ideal para controlar los 18 servomotores de un hexápodo sin necesitar módulos adicionales.

Su interfaz es fácil de utilizar, siendo la configuración intuitiva y por USB, permitiendo realizar ajustes precisos en cada servo.

Diseño, desarrollo y construcción de un robot hexápodo para inspección de áreas de difícil acceso.

También soporta secuencias, scripts de movimiento y realimentación analógica para cada canal.

Como desventajas podemos destacar la dependencia de software propietario, al requerir el uso de software específico de Pololu para la configuración y programación avanzada.

También su costo, que es algo más caro que otras opciones con menor número de canales.

4.3.2 Elección de la controladora de servomotores

Teniendo en cuenta las diferentes opciones, se ha seleccionado el *Pololu Mini Maestro 18-Channel USB Servo Controller* debido a su capacidad para controlar los 18 servomotores necesarios para el hexápodo sin requerir módulos adicionales. Si bien es cierto que el número de canales es demasiado ajustado, en el caso de requerir más canales para otros actuadores, siempre se puede integrar o bien otro controlador con suficientes canales, o bien conectarlos directamente a la placa de control. Su interfaz de configuración USB, funciones avanzadas y capacidad de realimentación la hacen ideal para este proyecto. Además, a pesar de necesitar software específico para su configuración, este es gratuito y se puede descargar de su página web.

4.4 Sensores de movimiento

Se ha valorado la opción de incluir sensores de rotación en las articulaciones con el fin de tener una realimentación de posición de cada pata. Los servomotores cuentan con esta realimentación a nivel interno, pero no disponen de una salida que se pueda leer. Es por ello que se ha decidido incluir unos pequeños potenciómetros analógicos.



Figura 21. Potenciómetro. Fuente [21]

Se ha elegido el encapsulado RM-063 vertical dada la aplicación. Era imprescindible que fueran lo más compactos posible, con los pines en posición vertical y que tanto la parte móvil como la fija tuvieran unas dimensiones y una forma que fuera fácilmente ensamblable sin necesidad de atornillar. Cuentan con un rango de movimiento de 180 grados, similar a los servomotores y el valor de resistencia elegido ha sido 100 kOhm con variación lineal. Se ha elegido un valor alto de resistencia dado que cuanto mayor sea este, menor será la corriente consumida. Dado que se van a alimentar a 3,3V, que será la tensión de alimentación y la tensión máxima de entrada analógica del ESP32, el consumo de corriente de todos los potenciómetros será de 0,6 mA, un valor prácticamente despreciable en comparación con el resto de componentes.

Como muchos potenciómetros cuenta con tres pines. Entre sus extremos se encuentra el total de la resistencia y es donde se debe conectar la alimentación, el VCC y el GND. El pin central es el que refleja la variación de resistencia. Este será el pin que irá conectado a la entrada analógica del microcontrolador, debido a que su valor variará entre VCC y GND en función de la posición en la que se encuentre.

En el apartado de diseño se puede comprobar la geometría que se ha desarrollado para su integración en la articulación sin apenas impacto.

Respecto a la resolución, los convertidores analógico-digital del ESP32 disponen de 12 bits, y por lo tanto 4096 valores diferentes (0-4095). Teniendo en cuenta que la tensión variará entre 0V y 3,3V, la resolución será de $3,3/4096 = 0,0008 V$, lo que significa que, por cada incremento de 0,0008V en la entrada analógica, se incrementará en 1 su lectura.

En cuanto a las distintas funcionalidades de los potenciómetros se encuentran por ejemplo el poder realizar lecturas de posiciones de las extremidades,

Diseño, desarrollo y construcción de un robot hexápodo para inspección de áreas de difícil acceso.

pudiendo dejar sus servos inactivos para moverla manualmente hasta la posición requerida y poder visualizar la posición concreta de las tres articulaciones de la pata. Como mejora de esta función, también se puede programar un modo en el cual se pueda ir definiendo las trayectorias que van a realizar las patas del robot, en función de las posiciones que se le vayan dando manualmente.

Concretamente la primera de ellas facilitará en gran medida la programación, debido sobre todo a que, dada la distribución hexagonal de las patas, cada una deberá realizar trayectorias distintas.

También se ha valorado la opción de colocar sensores de presión en los apoyos de las patas con el suelo, de manera que el controlador sepa en qué momento se ha apoyado la pata. Estos sensores añadirían una funcionalidad excelente, dado que el robot podría ir caminando sobre obstáculos de gran tamaño sin tener que modificar ninguna trayectoria, ya que se podría ir adaptando en tiempo real. Sin embargo, la programación de esta parte complicaría en gran medida el trabajo, por lo que se dejará propuesto como una de las posibles mejoras y continuación del proyecto.

4.5 Batería y gestión de energía

Debido al carácter remoto de la aplicación del hexápodo, se hace imprescindible la inclusión de baterías. La correcta elección de dichas baterías es importante para asegurar que el robot tenga una autonomía suficiente sin comprometer el peso y la eficiencia del sistema. Para que los altos requerimientos de corriente puntuales de los servomotores no perjudiquen la alimentación de la electrónica de control se ha decidido colocar dos baterías independientes. Una de ellas, con altas prestaciones para la alimentación de los servomotores, y otra que no requerirá más que una alimentación constante a una tensión fija con bajo requerimiento de corriente. Se comenzará con la elección de la batería de potencia o de alimentación de los motores. A continuación, se presenta una comparativa de diferentes tipos de baterías disponibles en el mercado.

Diseño, desarrollo y construcción de un robot hexápodo para inspección de áreas de difícil acceso.

4.5.1 Comparativa de tipos de baterías

Batería NiMH (Níquel-Metal Hidruro)

- **Características:**
 - Capacidad: Varía entre 1000 mAh y 5000 mAh.
 - Voltaje: Generalmente 7.2V a 8.4V.
 - Peso: Relativamente pesado en comparación con LiPo.
 - Durabilidad: Buena, pero con efecto memoria.

Entre las ventajas de este tipo de batería se encuentra que son bastante estables y poco propensas a incendiarse. Además, tienen un buen número de ciclos de carga y descarga antes de estropearse, por lo que su vida útil será mayor.

La desventaja más destacable es su peso, el mayor de los tres tipos evaluados. Esto repercute en el siguiente punto, su eficiencia, al tener una densidad de energía menor que los otros tipos.

Por último, tiene efecto memoria, por lo que necesita ser completamente descargada antes de ser recargada para mantener toda su capacidad de carga.

Batería LiPo (Polímero de Litio)

- **Características:**
 - Capacidad: Varía ampliamente, comúnmente entre 1000 mAh y 10000 mAh.
 - Voltaje: 3.7V por celda (7.4V, 11.1V, o 14.8V habitualmente).
 - Peso: Ligera, con alta densidad de energía.
 - Durabilidad: Muy buena, sin efecto memoria.

Las baterías LiPo cuentan con la ventaja de tener una alta densidad de energía, y por tanto una mayor eficiencia y mayor capacidad en un espacio y peso más reducidos.

Por este mismo motivo, son más ligeras para la misma capacidad, siendo ideales para aplicaciones donde el peso es crucial.

También es una ventaja que este tipo de baterías no tienen efecto memoria, lo que permite cargas parciales sin afectar la capacidad total.

Por contra, requieren un manejo más cuidadoso para evitar cortocircuitos que pudieran dañarla, pudiendo llegar en un caso extremo a su inflamación.

Diseño, desarrollo y construcción de un robot hexápodo para inspección de áreas de difícil acceso.

Además, generalmente son más caras que las baterías NiMH.

Batería LiFePO4 (Litio-Hierro-Fosfato)

- **Características:**
 - Capacidad: Generalmente entre 1500 mAh y 6000 mAh.
 - Voltaje: Normalmente 3.2V por celda (12.8V para packs de 4 celdas).
 - Peso: Moderado, más pesado que LiPo pero más ligero que NiMH.
 - Durabilidad: Excelente, con alta seguridad y sin efecto memoria.

Las baterías LiFePO4 son muy seguras y estables en comparación con las de Litio y cuentan con una mayor vida útil al tener mayor número de ciclos de carga y descarga y sin efecto memoria. Sin embargo, son más voluminosas que las LiPo y también tienen un mayor coste que las NiMH y que la mayoría de las baterías LiPo.

4.5.2 Elección del tipo de batería

Después de evaluar las características, ventajas y desventajas de cada tipo de batería, se ha decidido optar por una batería de polímero de litio (LiPo) por su alta densidad de energía, ligereza y eficiencia, que son cruciales para el desempeño del hexápodo. También se valora muy positivamente su gran capacidad de descarga, muy necesaria para que los servos puedan responder rápidamente a una gran solicitud de corriente ante cualquier estímulo externo o variación del terreno.

4.5.3 Dimensionado de la batería

Cálculo de la capacidad de la batería

Para garantizar una autonomía de aproximadamente una hora, es importante considerar el consumo energético de los servomotores. Según sus especificaciones y posteriores pruebas, se ha comprobado que cada servo MG995 consume aproximadamente 1A bajo carga total (en bloqueo). Considerando los 18 servomotores, se debe dimensionar toda la alimentación para un pico de 18A, el cual es altamente improbable que se dé, debido a que, en funcionamiento normal, sumando el peso del propio hexápodo al de una posible carga de algún equipo que pudiera transportar, los motores estarían muy por debajo de su capacidad máxima de carga, que sería de 15Kg/cm por motor.

Diseño, desarrollo y construcción de un robot hexápodo para inspección de áreas de difícil acceso.

Por lo tanto, para calcular la autonomía se estimará un valor de consumo, obtenido por medio de la experimentación con el prototipo, el cual mantiene un consumo inferior a 5 amperios. Teniendo en cuenta esto, se dimensionará la batería de acuerdo a la siguiente expresión:

$$\text{Autonomía [h]} = \frac{\text{Capacidad de la batería [mAh]}}{\text{Consumo promedio [mA]}}$$

Sabiendo que la autonomía requerida es aproximadamente una hora y que el consumo promedio es de 5 amperios,

$$\begin{aligned} \text{Capacidad de la batería [mAh]} \\ &= \text{Autonomía [h]} \cdot \text{Consumo promedio [mA]} \\ \text{Capacidad} &= 1 \cdot 5000 = 5000 \text{ mAh} \end{aligned}$$

Por tanto, se buscará una batería Li-Po con una capacidad de 5000 mAh.

En el caso de querer extender la autonomía, o bien de añadir algún otro equipo al hexápodo con gran consumo, siempre se podría cambiar la batería por otra con mayor capacidad o bien colocar dos baterías iguales en paralelo.

Cálculo de la tensión de la batería

Una vez obtenida la capacidad necesaria, se deberá decidir la tensión de operación. Teniendo en cuenta que la tensión de las baterías LiPo son múltiplos de la tensión de celda, 3.7V, se podrá optar por 7.4V, 11.1V o 14.8V.

Para estimar el voltaje de la batería, se debe evaluar las tensiones de los componentes electrónicos que se quieran alimentar desde ella, en este caso de los servomotores. Por lo tanto, como se van a alimentar a 7,2V y dado que se quiere que sea una alimentación estable y la celda LiPo tiene una fluctuación entre 4,2V y 3V aproximadamente, se deberá pasar por un regulador de tensión.

Se elegirá un regulador conmutado, debido a su mayor rendimiento y aprovechamiento de la energía, llegando a ser este mayor al 80%, frente al 40% a 60% de los lineales, en los cuales el exceso de energía se libera en forma de calor. Se trata de un regulador dc-dc, al que se le introducirá una tensión variable, como la de la batería, y en función de su modelo y configuración, se obtendrá a la salida otra tensión estable.

Existen tres tipos principales, que son los Buck o reductores, los Boost o elevadores y los Buck/Boost o reductores/elevadores. Su nombre obedece al funcionamiento del mismo, obteniendo en la salida de los Buck una tensión más baja que en la entrada, en los Boost más elevada, y en el caso de los Buck/Boost, este podrá trabajar de manera indistinta, pudiendo tanto elevar

Diseño, desarrollo y construcción de un robot hexápodo para inspección de áreas de difícil acceso.

como reducir la tensión de entrada. Por su mayor sencillez y menor precio, se elegirá un Buck o reductor, por lo que la tensión en todo momento deberá ser mayor a los 7,2V. Teniendo en cuenta que, con dos celdas, la tensión podría variar entre 8,4V y 6V, y con tres celdas la tensión variará entre 12,6V y 9V, se deberá elegir una LiPo de tres celdas. Por lo tanto, su valor nominal será de 11.1V.

Diseño, desarrollo y construcción de un robot hexápodo para inspección de áreas de difícil acceso.

4.5.4 Comparativa de baterías LiPo

A continuación, se presenta una comparativa de diferentes baterías de polímero de litio (LiPo) disponibles en el mercado, que se aproximen al máximo a los requerimientos calculados en el apartado anterior, con el objetivo de elegir una opción que ofrezca un buen equilibrio entre todos ellos.

Turnigy 5000mAh 3S 20C LiPo Pack



Figura 22. Batería Turnigy 5.0. Fuente [22]

- **Características:**
 - Capacidad: 5000 mAh
 - Voltaje: 11.1V (3S)
 - Peso: 360 g
 - Dimensiones: 166 x 50 x 20 mm
 - Tasa de descarga: 20C constante

Como una de las principales ventajas que tiene esta batería [15] es su balance entre peso y capacidad de descarga. La capacidad es la requerida según las especificaciones calculadas en el apartado anterior. Existen baterías similares con mayor capacidad o tasa de descarga, pero su peso aumenta considerablemente.

Como contraposición se encuentra su tamaño, que puede resultar algo voluminoso en comparación con otras soluciones.

Diseño, desarrollo y construcción de un robot hexápodo para inspección de áreas de difícil acceso.

ZIPPY Flightmax 5000mAh 3S1P 25C LiPo Pack



Figura 23. Batería Zippy Flightmax. Fuente [23]

- **Características:**
 - Capacidad: 5000 mAh
 - Voltaje: 11.1V (3S)
 - Peso: 427 g
 - Dimensiones: 155 x 49 x 26 mm
 - Tasa de descarga: 25C constante

La principal ventaja de esta batería [16] es su alta tasa de descarga, la cual permite un mayor rendimiento en aplicaciones de alta demanda. Tiene la misma capacidad que la batería mencionada en el apartado anterior, por lo que aportará una autonomía de una hora según los cálculos previos.

Como puntos negativos pueden destacarse su peso, algo mayor que las otras opciones, y su precio, también ligeramente mayor que otras baterías de capacidad similar.

Diseño, desarrollo y construcción de un robot hexápodo para inspección de áreas de difícil acceso.

Gens ace 5300mAh 3S 50C LiPo Pack



Figura 24. Batería Gens ace 5300. Fuente [24]

- **Características:**
 - Capacidad: 5300 mAh
 - Voltaje: 11.1V (3S)
 - Peso: 410 g
 - Dimensiones: 138 x 45 x 33 mm
 - Tasa de descarga: 50C constante

Esta batería [17] es compacta y ligera, por lo que es una buena opción para integraciones donde el espacio y el peso son críticos. Sin embargo, la principal característica diferenciadora de esta batería es que tiene una tasa de descarga muy alta, lo que la convierte en una opción excelente para aplicaciones que requieren picos altos de potencia.

Hay que tener en cuenta que debido a esta alta tasa de descarga tiene un precio más elevado, y también debido a ello, su vida útil puede verse reducida si no se manejan adecuadamente.

Diseño, desarrollo y construcción de un robot hexápodo para inspección de áreas de difícil acceso.

Tattu 4500mAh 3S 45C LiPo Pack



Figura 25. Batería Tattu 4500. Fuente [25]

- **Características:**
 - Capacidad: 4500 mAh
 - Voltaje: 11.1V (3S)
 - Peso: 350 g
 - Dimensiones: 135 x 42 x 31 mm
 - Tasa de descarga: 45C constante

Esta última opción [18] también cuenta con una tasa de descarga considerable. Y de igual forma también es ligera y compacta, sin embargo, ofrece una menor capacidad. Con la capacidad que aporta ya no se dispondría de una hora de autonomía.

En cuanto a la relación capacidad-precio tampoco es buena opción, al tener un precio algo elevado para la capacidad que ofrece.

4.5.5 Elección de la batería de potencia

Después de evaluar las características, ventajas y desventajas de cada opción, se ha decidido seleccionar la batería **Turnigy 5000mAh 3S 20C LiPo Pack**, por su excelente equilibrio entre capacidad, peso, tamaño y costo. Aunque hay opciones con más capacidad o tasa de descarga, aun con similares precios, el peso se incrementa notablemente, y esta batería asegura aproximadamente una hora de autonomía. La tasa de descarga es 20C, lo que implica que la corriente mantenida capaz de otorgar es de 100A ($20 \cdot 5000mA$), valor muy superior al requerido. Finalmente, el tamaño no supone una gran desventaja, a pesar de haber opciones más reducidas, debido a que ha primado más el peso que las dimensiones para esta aplicación en cuestión. Por último, es una batería fácilmente disponible en el mercado, lo que facilita su adquisición y reemplazo.

Diseño, desarrollo y construcción de un robot hexápodo para inspección de áreas de difícil acceso.

4.5.6 Elección de la batería de control

Como se ha indicado anteriormente se van a separar las alimentaciones para evitar interferencias y bajadas de tensión en las alimentaciones de los circuitos de control. Con este motivo se ha buscado una batería que deberá alimentar tanto el microcontrolador ESP32, como la placa concentradora de señales, junto con todos los potenciómetros, así como el controlador de servomotores. La tensión más adecuada será de 5V debido principalmente a que el ESP32 tiene un pin de alimentación (5V), cuyo regulador interno está dimensionado para 5V. También se podría colocar un regulador externo, pero no se ha creído necesario.

Además, los 5V también servirán para alimentar la controladora de servos, y los 3,3V que proporciona el ESP32 a través de su regulador interno, servirán también para alimentar tanto los circuitos multiplexores como todos los potenciómetros. Esto beneficiará a la hora de leer estos potenciómetros, dado que la tensión soportada por las entradas del ESP32 no es muy superior a los 3,3V. Por ello se podrá introducir directamente sin necesidad de reductores de señal.

Teniendo en cuenta que se necesitan 5V, se ha decidido utilizar un Power Bank, dado que están preparados para proporcionar esta tensión de forma estable y duradera. Se ha elegido concretamente uno con una capacidad de 5000mAh y una capacidad de descarga de 2A.



Figura 26. Power Bank 5000 mAh. Fuente [26]

El diagrama de alimentación final se podrá consultar en el apartado [4.9.1](#) dentro de la sección de esquemas de conexión.

4.6 Selección de la cámara

Dado que se pretende que el robot sea capaz de operar en entornos de difícil acceso o potencialmente peligrosos, es necesario evaluar varias características técnicas que aseguren un buen rendimiento en condiciones tanto de interior y exterior. Debido a que las posibles aplicaciones de este robot van desde asistencia en rescates en derrumbes hasta operaciones militares y exploración en áreas hostiles, la visibilidad y la captura de imágenes juegan un papel muy importante.

Las características de la cámara van a depender en gran medida del uso que se le quiera dar. Es por ello que en este apartado se va a seleccionar una cámara que cumpla con unos requisitos mínimos estándar y en el caso de aplicaciones más exigentes se deberá escoger una cámara con unas características más adecuadas a la aplicación en concreto que se le quiera dar.

4.6.1 Características básicas

Se deberán tener en cuenta diferentes factores en la selección de la cámara tales como la resolución, la conectividad, sus dimensiones, etc. A continuación, se hace una breve descripción de los más importantes.

Resolución

La resolución de la cámara es un factor importante para que el operador remoto pueda identificar obstáculos, estructuras y posibles víctimas con claridad.

Se recomienda una cámara con una resolución mínima de 720p, que proporciona un buen equilibrio entre calidad de imagen y transmisión de datos. Si la resolución es mayor, la velocidad de transferencia también deberá serlo para poder visualizar la imagen en tiempo real. En casos donde la identificación detallada es crucial, como en operaciones militares, una cámara con resolución mayor puede ser considerada, aunque su uso dependerá de la capacidad del sistema para manejar el mayor volumen de datos.

Ángulo

Otro aspecto a considerar es el del ángulo de visión, que influye directamente en la cantidad de área que la cámara puede cubrir sin necesidad de mover el dispositivo. Se recomienda un campo de visión mínimo de 90° para garantizar una cobertura amplia en entornos estrechos o confinados. Si la aplicación requiere un análisis situacional más amplio, una cámara de mayor amplitud (120° o más) sería ideal, aunque debe considerarse el potencial de distorsión en los bordes de la imagen.

Diseño, desarrollo y construcción de un robot hexápodo para inspección de áreas de difícil acceso.

Conectividad

La transmisión en tiempo real de las imágenes es esencial para el control remoto. La cámara seleccionada debe ser capaz de transmitir vídeo de alta calidad en tiempo real, lo cual requiere soporte para conexiones de alta velocidad, como por ejemplo Wi-Fi o tecnología de radiofrecuencia, dependiendo del entorno de operación.

La latencia de la transmisión debe ser mínima, para permitir un control ágil por parte del operador.

Peso y Tamaño

El peso y el tamaño de la cámara son consideraciones importantes, ya que influyen en la maniobrabilidad y la eficiencia energética de la araña robótica. Es preferible una cámara compacta y ligera que no sobrecargue el diseño mecánico ni el sistema de potencia del robot. Una cámara ligera y de dimensiones reducidas es ideal, siempre que cumpla con los demás requisitos funcionales.

Alimentación y Consumo Energético

El consumo energético de la cámara debe ser eficiente para maximizar la autonomía de la araña robótica.

La cámara puede tener incorporado su propio sistema de alimentación o bien alimentarse directamente del sistema de energía del robot, aunque esto podría comprometer la duración de la batería del robot.

Visión Nocturna e Infrarroja

Existen posibles aplicaciones en las que el robot puede ser desplegado en situaciones de baja o nula visibilidad, y en ellas, la capacidad de visión nocturna es indispensable. Para este propósito se debería seleccionar una cámara que incorpore tecnología de visión infrarroja (IR) para poder operar en condiciones de oscuridad. Más concretamente el infrarrojo cercano (el rango del espectro más cercano al espectro visible), permite la identificación de objetos y personas mediante el calor que emiten, lo cual es vital en operaciones de rescate donde la visibilidad puede estar comprometida por polvo, humo o escombros. Para ello, sin embargo, sería necesario dotar al robot de una fuente de luz bien visible o bien infrarroja para que además de poder detectar las fuentes de calor,

Resistencia y Durabilidad

Considerando los entornos adversos en los que operará la araña robótica, la cámara debe ser robusta y resistente a condiciones extremas, tales como

Diseño, desarrollo y construcción de un robot hexápodo para inspección de áreas de difícil acceso.

polvo, agua, golpes y vibraciones. Se recomienda una cámara con clasificación IP68, que asegure protección total contra el polvo y resistencia al agua, incluso en inmersión prolongada. Además, la carcasa de la cámara debe ser capaz de resistir impactos y caídas, lo cual es común en operaciones de rescate o en terrenos accidentados.

4.6.2 Elección de la cámara

Después de considerar todas las características técnicas y los requerimientos operativos se ha incorporado como cámara básica una cámara inicialmente pensada para drones, con una transmisión de imagen por radiofrecuencia a una pantalla de FPV (vista en primera persona), lo que garantiza una latencia reducida. Además, cuenta con resolución 720p y un ángulo de visión de 90°.

También incluyen sus propias baterías, tanto la cámara como la pantalla, con una duración aproximada de una hora, por lo que no reducirá la autonomía del hexápodo y convierte al sistema de visión en totalmente independiente del resto del robot.

En el caso de tener unos requerimientos mayores en función de la aplicación habrá que considerar el cambio de esta cámara por otra cuyas características se ajusten a la aplicación que se le quiere dar teniendo en cuenta los factores que se han descrito anteriormente.

En concreto, el conjunto seleccionado ha sido el *C4002 5.8G FPV MONITOR + 720P CAMERA SET*.



Figura 27. Sistema FPV C4002. Fuente [27]

Para su colocación se han diseñado adaptadores tanto para la cámara sobre el hexápodo, como para la pantalla sobre el mando de control. Se pueden ver en el apartado de diseño.

Diseño, desarrollo y construcción de un robot hexápodo para inspección de áreas de difícil acceso.

4.7 Selección de componentes electrónicos auxiliares

4.7.1 Multiplexor de señales

Tal y como se indicaba en el apartado de especificaciones técnicas, en el caso de que la controladora no dispusiera de tantas entradas analógicas como son necesarias para la realimentación de todas las articulaciones, sería necesario agregar un multiplexor de señales que permita ir leyendo las posiciones de todos los motores de forma ordenada. Como la controladora elegida no cuenta con entradas suficientes, para solucionar este problema se ha seleccionado un multiplexor de dieciséis canales, gobernado por cuatro bits digitales que irían conectados a sendas salidas del ESP32. La salida de datos del multiplexor se conectaría a una entrada analógica que será la que leerá la señal de los potenciómetros en función de los bits de selección del propio multiplexor.

El circuito elegido es el siguiente:

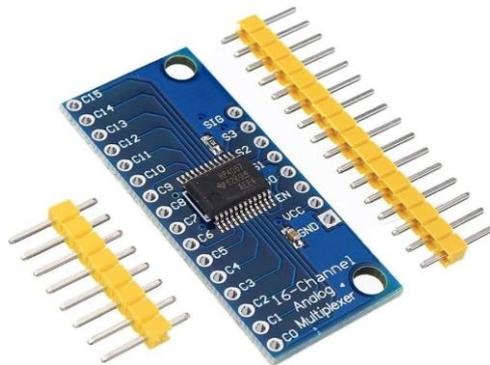


Figura 28. Multiplexor de señales. Fuente [28]

Se trata de una placa basada en el circuito integrado 74HC4067. Dispone de 16 canales, una alimentación con un rango entre 2 y 6 voltios, un pin de *Enable* a nivel bajo, cuatro pines de selección del canal y un pin de salida de la señal, en este caso, analógica.

Teniendo en cuenta que se dispone de 18 salidas que deben ser leídas y tan sólo 16 canales, se puede, o bien trabajar tan sólo con esos 16 canales, y las dos señales restantes conectarlas directamente a otras dos entradas analógicas del ESP32, o bien, utilizando tres multiplexores, asociarlos para, usando tan sólo un bit más de selección, obtener un multiplexor de 32 canales. Como el coste de estos multiplexores es muy bajo, se ha decidido tomar esta última opción. De esta forma el esquema de conexiones con el ESP32 quedará más limpio y robusto y a la hora de programar, el código también lo será, ya que tan sólo necesitamos cinco salidas digitales que funcionarán como bits de selección y una entrada analógica, dejando el resto para futuras ampliaciones.

Diseño, desarrollo y construcción de un robot hexápodo para inspección de áreas de difícil acceso.

Además, con este fin, también quedarán las catorce señales de entrada restantes de los multiplexores. El esquema de asociación será el siguiente:

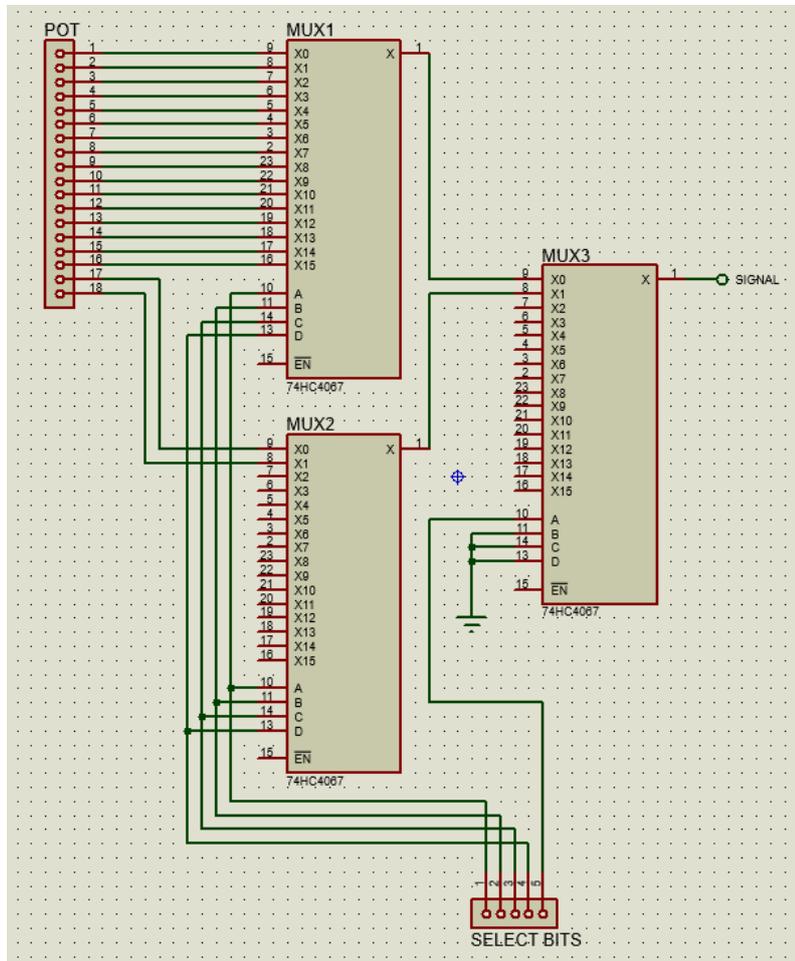


Figura 29. Asociación de multiplexores. Elaboración propia

SIGNAL será la salida con la señal analógica elegida, SELECT BITS los bits de selección de la entrada activa y POT las señales provenientes de los potenciómetros de las 18 articulaciones.

A continuación, se muestra una imagen del circuito ya soldado de la asociación de multiplexores descrita anteriormente:

Diseño, desarrollo y construcción de un robot hexápodo para inspección de áreas de difícil acceso.

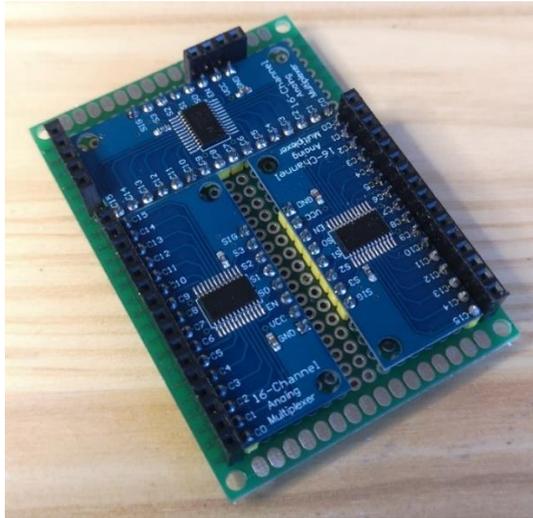


Figura 30. Asociación de multiplexores. Elaboración propia

El conexionado se ha realizado por la parte inferior de la placa de prototipado y se han colocado varias tiras de pines hembra donde se conectarán las entradas y salidas. Las tiras de 16 pines izquierda y derecha son las 32 entradas de sensores, los cinco pines de la zona superior izquierda son los bits de selección, y los cuatro pines de la parte superior son Enable, VCC, GND y la señal de salida.

4.7.1 Regulador de tensión

Como se comentó en el apartado del dimensionado de la batería, se ha elegido usar un regulador conmutado reductor de tensión, o Buck. Como condiciones iniciales se tiene que el rango de tensión de entrada tiene que ser mayor o igual al rango de tensiones de la batería, que como se dijo estará entre 9V y 12,6V. Además, la salida tiene que ser de 7,2V, por lo que lo más sencillo será que sea regulable. Habitualmente la regulación viene determinada por un potenciómetro analógico, aunque existen también sistemas digitales. Finalmente se debe tener en cuenta los altos requerimientos de corriente que tienen los servos, por lo que el regulador tendrá que soportar esta corriente, o por lo contrario podría dañarse. Teniendo en cuenta estas condiciones, se ha elegido el siguiente regulador de tensión:

Diseño, desarrollo y construcción de un robot hexápodo para inspección de áreas de difícil acceso.

XL4016E1 DC-DC Step Down Buck Converter



Figura 31. Regulador de tensión. Fuente [29]

Se trata de un circuito reductor [19] con una tensión de entrada entre 5V y 40V y una tensión de salida entre 1,2V y 36V, regulado por un potenciómetro analógico. Es una integración de AZ Delivery y está basado en el regulador conmutado XL4016. Cabe destacar que, al tratarse de un Step Down, en ningún caso podrá proporcionar una tensión superior a la de entrada. Tiene un rendimiento aproximado del 95% según fabricante y cuenta con dos disipadores de calor para poder evacuarlo de forma eficiente, debido a que cuenta con una potencia de 200W, pudiendo soportar corrientes de 8A sin ventilación y hasta 12A con una ventilación adecuada. Es por ello que, se dejará preparada la implementación de un ventilador sobre la estructura para refrigerar los disipadores en caso de que fuera necesario, para lo cual cuenta además con una salida de alimentación para dicho ventilador. Por último, tiene integrado un pequeño fusible para evitar posibles daños por una sobre corriente. En el caso de aumentar los requerimientos de corriente por las posibles aplicaciones del robot, se podrían colocar dos reguladores en paralelo con una pequeña protección contra corrientes inversas en ambas salidas.

La salida del ventilador no se encuentra conectada por defecto, debido a que cuenta con tres pads SMD como selector, para poder alimentar el ventilador o bien a la salida regulada o bien a 12V de corriente continua a través de un regulador interno. Se ha tomado la decisión de colocar el puente con los 12V fijos, para poder conectar un ventilador de esta tensión.

Diseño, desarrollo y construcción de un robot hexápodo para inspección de áreas de difícil acceso.

4.7.2 Ventilador de refrigeración del regulador



Figura 32. Ventilador de refrigeración. Fuente [30]

En el caso de que se considere necesario se colocará en la parte superior del regulador un ventilador de 12V de 40x40mm con un suficiente flujo de aire, motor brushless y cuyo nivel de sonido es tan sólo 24dB, para asegurar la correcta refrigeración del regulador.

4.8 Mando de control

En último lugar se procede a seleccionar el mando de control. Para controlar de manera remota el robot será necesario un mando que disponga de varios pulsadores, uno o dos joysticks por la facilidad en el manejo y que disponga de conectividad inalámbrica Wi-Fi, Bluetooth o radiofrecuencia, en función sobre todo de la distancia a la que se quiera controlar el robot.

Como en este caso la funcionalidad es de prototipo y en todo caso, se considera que 10 metros de radio es distancia suficiente para un primer uso de robot y dado que el microcontrolador elegido cuenta con antena integrada, se ha decidido optar por un controlador de PS3. Dispone de conexión Bluetooth, varias horas de autonomía, dieciséis pulsadores digitales, dos joysticks analógicos y otros seis de los pulsadores digitales que también pueden funcionar como analógicos. Existe una buena oferta en el mercado, a un precio muy reducido, es un controlador de sobra testado y estable y, además, la amplia comunidad tras este tipo de proyectos, ha desarrollado una biblioteca para ESP32 y Arduino, facilitando la tarea de su configuración y recepción de señales.

Diseño, desarrollo y construcción de un robot hexápodo para inspección de áreas de difícil acceso.



Figura 33. Mando de control PS3. Fuente [31]

En el apartado [6.3](#) se muestra la configuración necesaria del mando, así como la biblioteca utilizada.

4.9 Esquemas de conexión.

Para finalizar el apartado de la electrónica, se muestran a continuación dos diagramas en los que se indican las conexiones eléctricas del robot. El primero muestra las alimentaciones, mientras que el segundo, muestra las conexiones de control. Cabe aclarar que la cámara no se ha incluido en ninguno de los dos esquemas, debido a que dispone tanto de alimentación, como de transmisión de señal independiente.

4.9.1 Esquema de alimentación

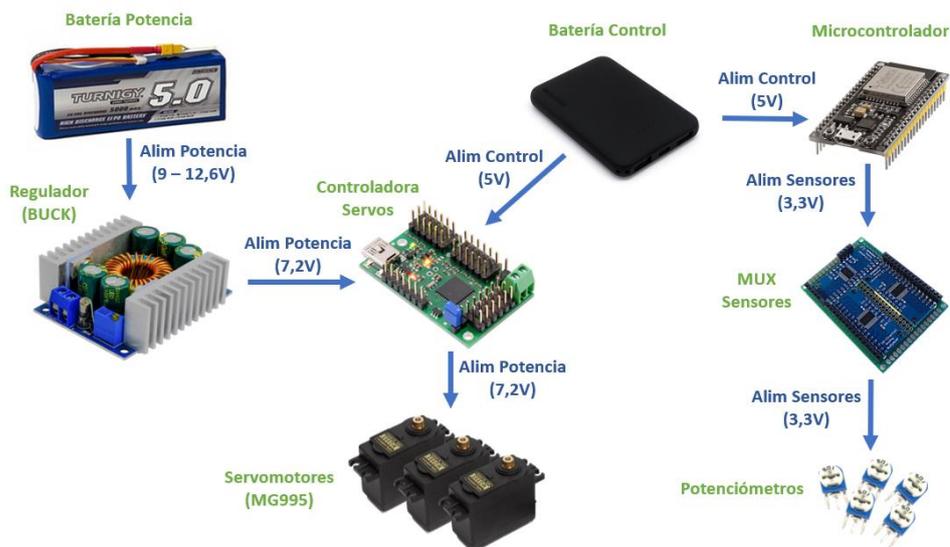


Figura 34. Esquema de alimentaciones. Elaboración propia

Diseño, desarrollo y construcción de un robot hexápodo para inspección de áreas de difícil acceso.

4.9.2 Esquema de control



Figura 35. Esquema de señales de control. Elaboración propia

Diseño, desarrollo y construcción de un robot hexápodo para inspección de áreas de difícil acceso.

Diseño, desarrollo y construcción de un robot hexápodo para inspección de áreas de difícil acceso.

5. Diseño y construcción del robot

En esta sección se describe detalladamente el proceso de diseño y construcción del hexápodo robótico, desde la concepción inicial hasta la integración final de todos los componentes. Se utilizan herramientas de diseño asistido por ordenador (CAD) para crear los modelos 3D del robot y sus componentes, y se lleva a cabo la fabricación mediante fabricación aditiva a través de deposición de material fundido o FDM, debido a su factor económico, a la inmediatez y a ser fácilmente replicable.

5.1 Diseño del robot

El diseño se ha llevado a cabo por medio del software CAD Catia V5 [20]. En primer lugar, se han tenido en cuenta todos los componentes que tenía que albergar el robot, dejando espacio en el cuerpo para localizar las diferentes placas electrónicas, así como la batería, y en las patas se han integrado tanto los servomotores como los sensores de posición.

En segundo lugar, ha sido necesaria una amplia investigación sobre los robots del mercado, así como de los diferentes insectos y arácnidos que pudieran dar una idea sobre las geometrías más óptimas, así como la distribución de las articulaciones de cada pata.

Una vez recopilada toda la información se ha comenzado a modelar cada componente, tanto electrónico como estructural para finalizar por integrarlo todo en el ensamblaje del conjunto que se muestra a continuación.

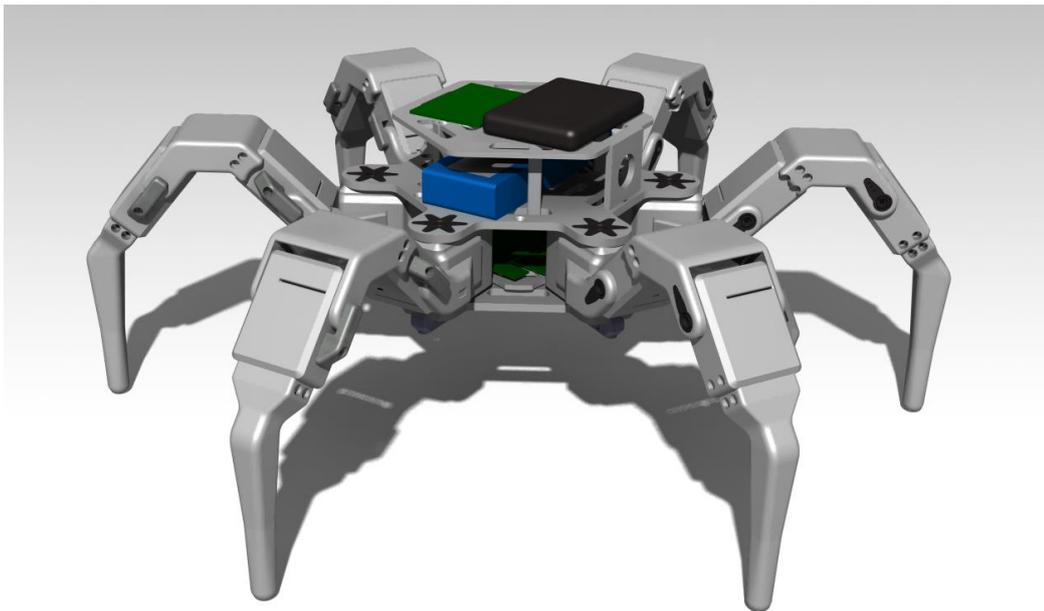


Figura 36. Ensamblaje final del robot. Elaboración propia

Diseño, desarrollo y construcción de un robot hexápodo para inspección de áreas de difícil acceso.

Para los componentes electrónicos, se han descargado de repositorios online (como GRABCAD.com, [21]), así como de las páginas web de los fabricantes, haciendo varios ajustes sobre cada modelo para incluir las tolerancias necesarias para su encaje en las piezas fabricadas.

Por este motivo, la solución que se ha tomado, ha sido la de recrear unas décimas de milímetro cada componente, para que, una vez fabricadas las piezas, el ajuste fuera óptimo. Se ha realizado de esta manera, dado que habitualmente las medidas de los componentes en los modelos 3D, están en sus dimensiones nominales, y en la fabricación aditiva, en función de la máquina, la boquilla de extrusión, la configuración y el polímero utilizado, las piezas muestran unas pequeñas expansiones volumétricas que se deben tener en cuenta. Estas configuraciones se muestran en el apartado de fabricación.

5.1.1 Cuerpo del robot

El cuerpo del hexápodo es el núcleo que aloja la mayoría de los componentes electrónicos y proporciona una estructura sólida para la fijación de las patas.

El cuerpo se compone de dos bases en paralelo con forma principalmente hexagonal, de manera que las patas queden equidistantes y uniformemente distribuidas, para que se puedan adaptar mejor a las irregularidades del terreno. También se valoró una distribución rectangular, en la cual se situarían tres patas por lado. Esta distribución permitiría un movimiento más sencillo, debido a que, a la hora de avanzar, las patas deberían hacer los mismos movimientos, aunque de forma alterna. Sin embargo, se descartó la idea por la mayor estabilidad y mejor distribución de pesos en la configuración en hexágono, a pesar del incremento de la dificultad a la hora de realizar el control en el movimiento. Además, al ser simétrico, puede realizar el desplazamiento en cualquier dirección, por lo que, en el caso de encontrar un obstáculo insalvable, se puede tomar otra dirección inmediatamente sin necesidad de girar la orientación del robot.

Para aligerar al máximo la estructura, se han realizado vaciados como se observa en las imágenes, que reducen el material aportado sin comprometer la integridad estructural. De hecho, estas pequeñas aberturas funcionan de refuerzo, a pesar de que pueda parecer lo contrario, debido a que la base no tiene un relleno del 100%, sino que tiene una estructura interna de celdillas. Sin embargo, en estas aberturas, sus perímetros son continuos al 100%, ejerciendo de refuerzo estructural desde su capa inicial a la final, lo que junto a su distribución uniforme impide que la pieza pueda flexar en ninguna dirección.

Diseño, desarrollo y construcción de un robot hexápodo para inspección de áreas de difícil acceso.

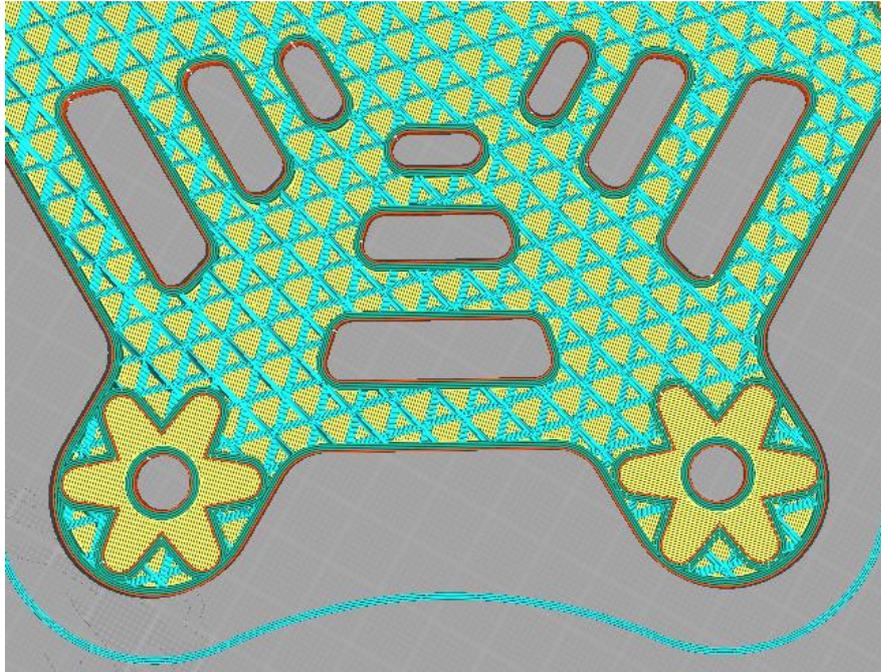


Figura 37. Detalle de patrón de relleno. Elaboración propia

La imagen representa un corte horizontal de la base, en que se puede apreciar el plástico que va depositando la máquina, con las zonas sólidas en amarillo, las estructuras interiores en azul claro y los perímetros sólidos en verde y naranja (perímetro exterior).

A continuación, se ve un detalle de estos perímetros.

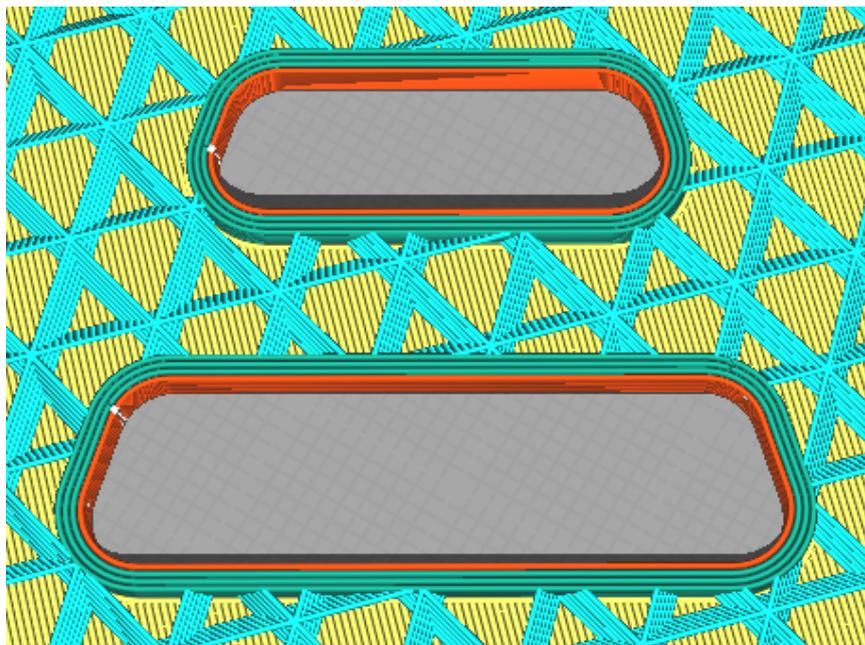


Figura 38. Detalle de perímetros en los vaciados. Elaboración propia

Diseño, desarrollo y construcción de un robot hexápodo para inspección de áreas de difícil acceso.

Las geometrías con forma de estrella de seis puntas, son las zonas donde encajan los acoples de servomotor con esta misma forma. Se ha elegido este tipo de pieza debido a su distribución uniforme de pesos, dado que será la pieza que deberá soportar gran parte del peso del robot, el cual será transmitido por cada extremidad desde el punto de apoyo del suelo hasta la base. Teniendo en cuenta que la extremidad puede tomar cualquier orientación, esta geometría de distribución uniforme de las cargas, asegurará la estabilidad sin permitir que se pueda mover o dañar. Este acople va encajado a la base y, por medio del tornillo, se fija al servo que se introduce desde el otro lado. De esta forma, se consigue que el servo pueda rotar utilizando el propio eje del motor de eje de giro para la pata. Este eje se complementa con un pequeño tornillo junto con el potenciómetro por su lado opuesto, dando más robustez a la unión y haciendo que ambas bases del cuerpo del robot queden fijas entre sí sin necesidad de componentes extra.

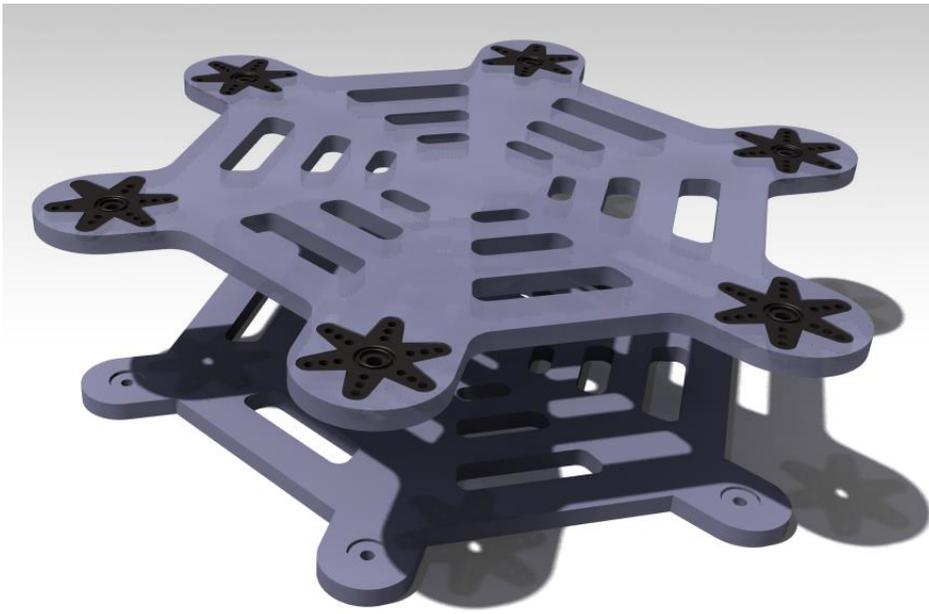


Figura 39. Bases del cuerpo del robot. Elaboración propia

Diseño, desarrollo y construcción de un robot hexápodo para inspección de áreas de difícil acceso.

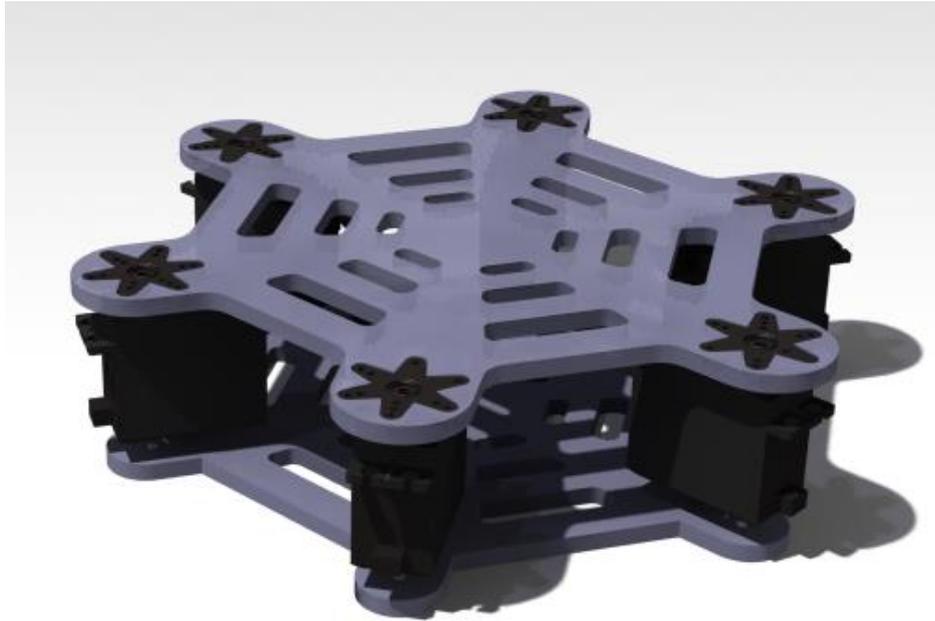


Figura 40. Bases con servomotores articulación 1. Elaboración propia

5.1.2 Patas del robot. Integración de los servos y potenciómetros

Cada una de las seis patas del hexápodo consta de múltiples segmentos articulados, proporcionando tres grados de libertad (3 DOF) por pata. Esto permite movimientos complejos y adaptabilidad a terrenos irregulares.

Esto implica que cada pata constará de tres segmentos principales, que definen sus grados de libertad. Cada segmento está compuesto por varias piezas con el fin de facilitar su fabricabilidad (DFM o Design for manufacturing), así como su ensamblaje (DFA o Design for assembly). Estas piezas van unidas con tornillos M3, arandelas y tuercas.

A continuación, se pasará a describir cada uno de los segmentos.

En primer lugar, el *segmento 1* será el que realice la función del hombro o de rotación de las patas. Es el que incluye los servomotores vistos en el apartado anterior y aporta robustez al conjunto de la base del robot.

Diseño, desarrollo y construcción de un robot hexápodo para inspección de áreas de difícil acceso.

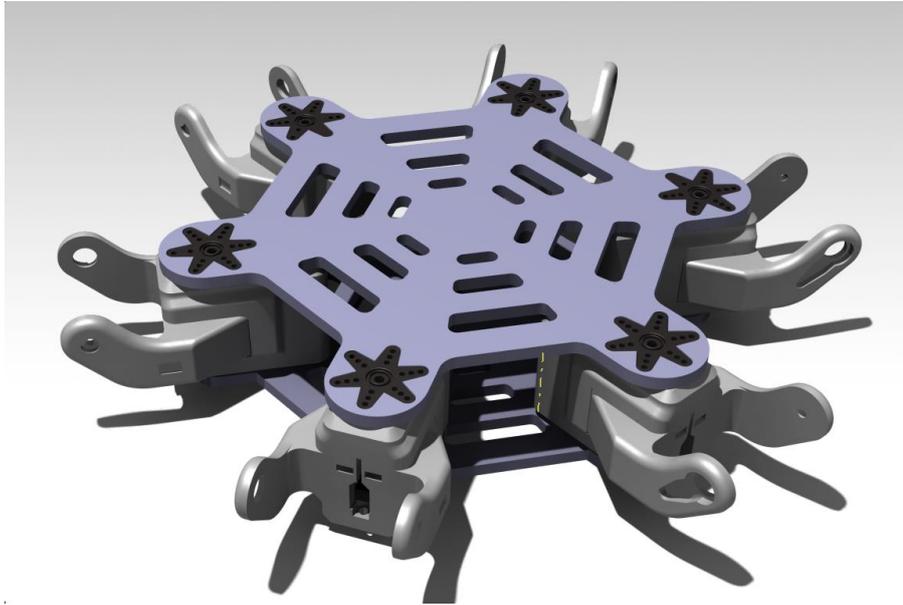


Figura 41. Cuerpo del robot con articulación 1 completa. Elaboración propia

Este segmento consta de cuatro partes, dos cubiertas para el servomotor y dos brazos. Uno de los brazos tiene en negativo la geometría de otro de los acoples de los servomotores para conectar el servomotor del siguiente segmento. El otro brazo cuenta con un pequeño orificio para el eje y un vaciado con la forma del potenciómetro. De esta forma, la parte móvil del potenciómetro encajará perfectamente en él. Se ha elegido este tipo de acople para el motor, debido al poco espacio que se dispone, y a que, dada la dirección del esfuerzo que tiene que ejercer el motor, no es necesario añadir más superficie de contacto.

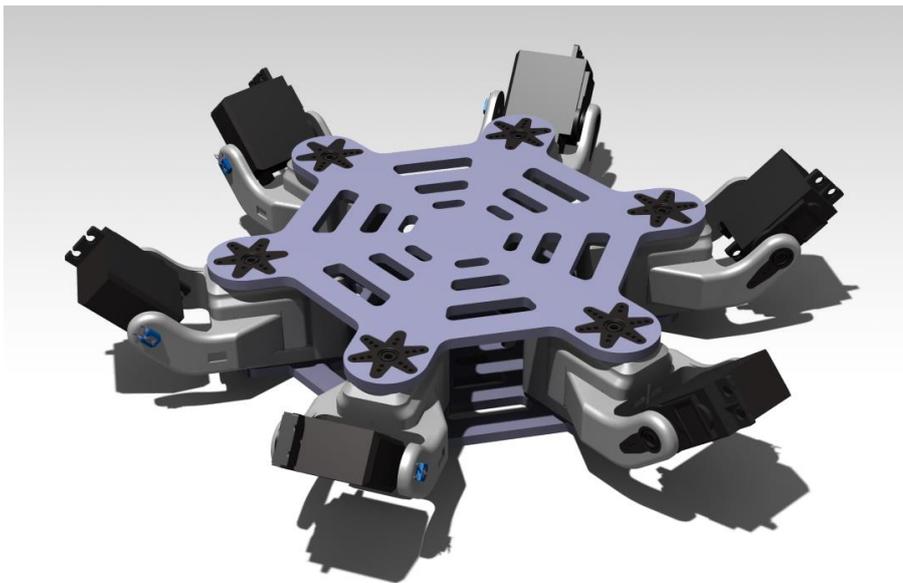


Figura 42. Cuerpo con servomotores de articulación 2. Elaboración propia

Diseño, desarrollo y construcción de un robot hexápodo para inspección de áreas de difícil acceso.

En cuanto al segmento número dos, será el que integra el servomotor mostrado en la imagen anterior.

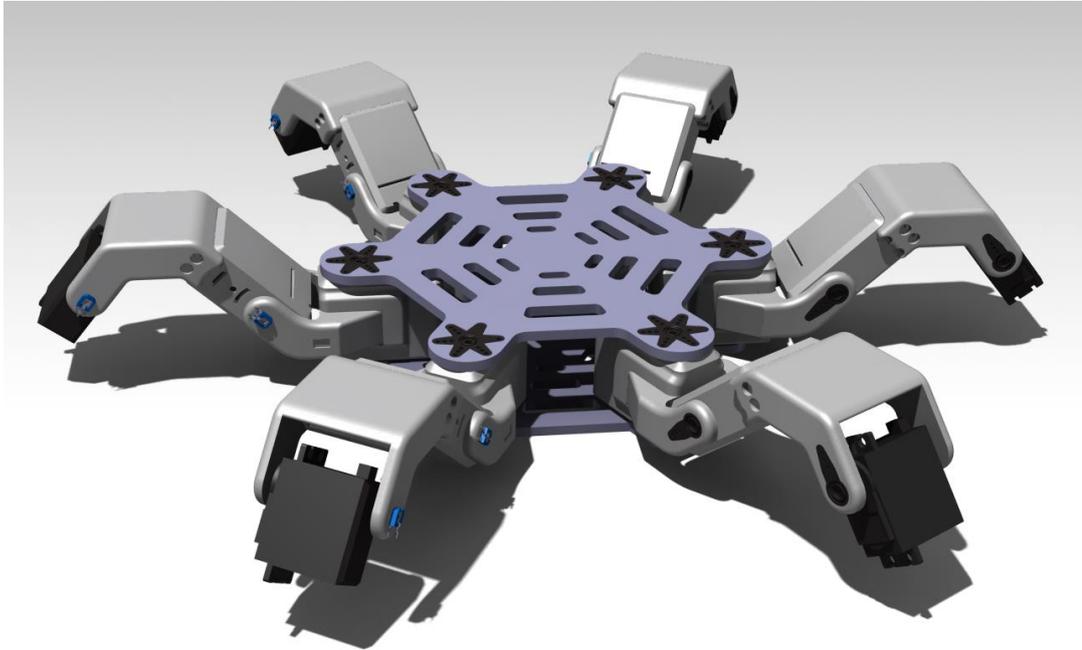


Figura 43. Cuerpo con servomotores de articulación 3. Elaboración propia

Como se puede ver en la figura, este segmento se compone de otros cuatro elementos. Dos cobertores para los motores y otros dos que serán los brazos en ángulo. Sus funciones son exactamente las mismas que en el segmento anterior, con la salvedad del ángulo que forman. Este ángulo es necesario para conseguir que los puntos de apoyo de las patas, queden lo más alejadas posible, confiriendo al robot de la máxima estabilidad, sin sobrepasar el par máximo que pueden mover los motores, debido a que cuanto más alejado esté el punto de apoyo del eje del motor, mayor momento de fuerza ejercerá sobre dicho eje, y, por consiguiente, mayor par necesitará el motor para vencer este momento.

Finalmente, el tercer segmento es el encargado de cubrir el tercer motor de cada extremidad, o muñeca en su símil humano, y proporcionar los puntos de apoyo al hexápodo. Está compuesto por tan sólo tres piezas, los dos cobertores del motor y la estructura de contacto con el suelo. Se ha diseñado una pata intercambiable para que en el caso de querer modificar estos puntos de apoyo o de añadir algún tipo de sensor, tan sólo sea necesario modificar el diseño de esta pieza, refabricarla y ensamblarla.

Diseño, desarrollo y construcción de un robot hexápodo para inspección de áreas de difícil acceso.

El montaje de los servomotores se ha realizado, de manera que su posición central, esté en el ángulo que se muestra en las imágenes. Por lo tanto, cada motor será capaz de mover su articulación 90 grados hacia cada lado.

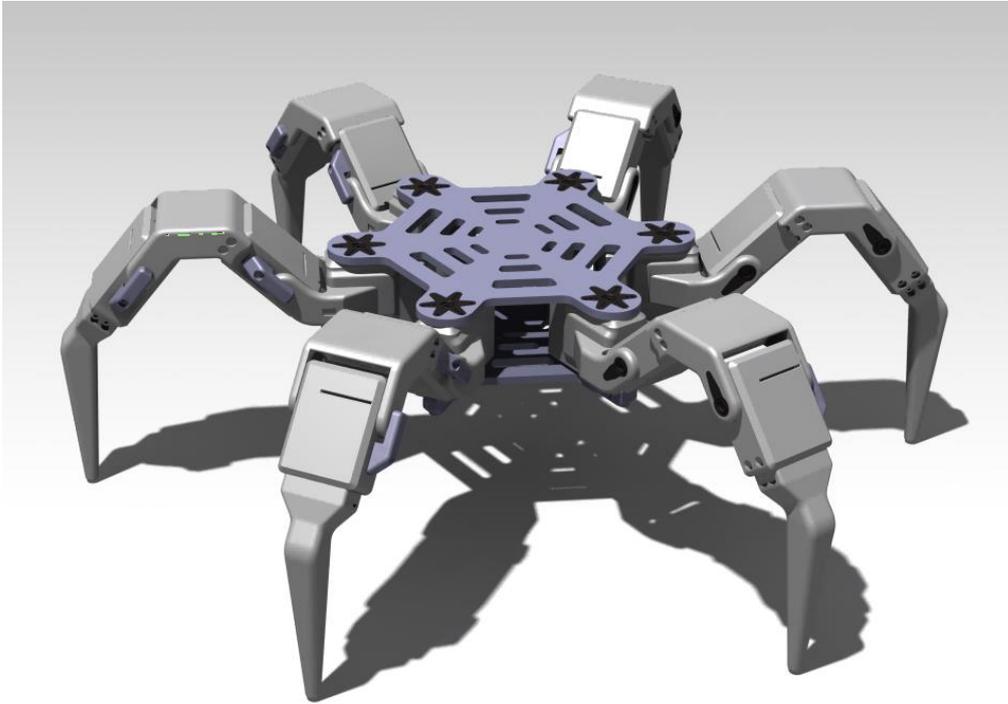


Figura 44. Cuerpo con extremidades completas. Elaboración propia

En la imagen anterior se pueden ver ya los potenciómetros cubiertos por la pieza que les permite rotar, a medida que se mueve el motor. Se muestra un detalle de su montaje a continuación.

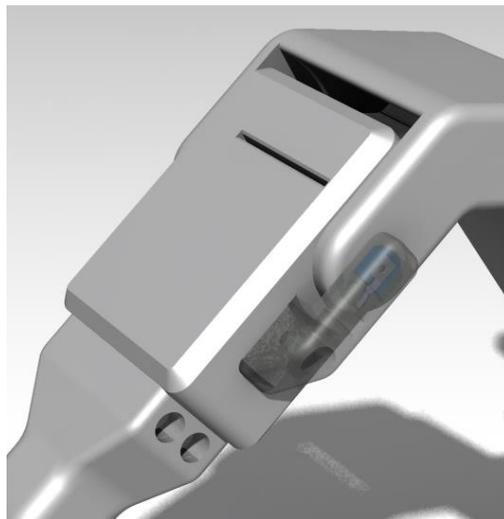


Figura 45. Detalle del montaje de los cubre potenciómetros. Elaboración propia

Diseño, desarrollo y construcción de un robot hexápodo para inspección de áreas de difícil acceso.

Como se puede observar en la imagen, el potenciómetro en color azul, se encuentra envuelto por el cobertor. La parte móvil del potenciómetro queda perfectamente encajada en el brazo del segmento anterior, que sería la parte fija de la articulación. La parte “fija” del potenciómetro queda abrazada por el cobertor como se muestra en la siguiente imagen.

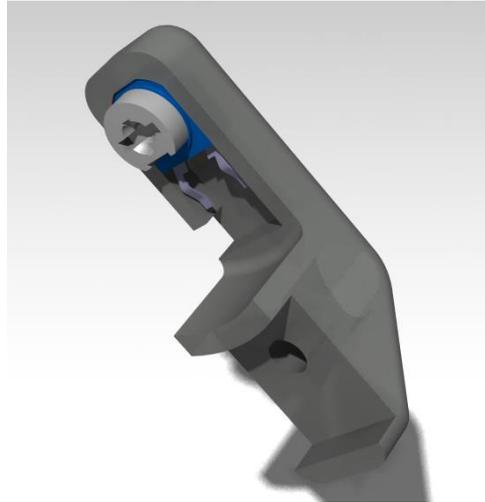


Figura 46. Detalle del montaje de los potenciómetros. Elaboración propia

De esta forma, cuando rote el servomotor sobre su eje, cuyo eje comparte con el potenciómetro, hará rotar dicho potenciómetro variando así la resistencia y por ello el voltaje. Ambos ejes, del potenciómetro y servomotor están completamente alineados y es el cobertor que se encuentra unido a la cubierta del servomotor a través de dos tetones y un tornillo, el que hace que se mantenga solidario a la parte fija del potenciómetro.

Estos potenciómetros tienen un ángulo de aproximadamente 180 grados, y a pesar de que los servomotores no permiten un giro mayor a los 180 grados, estas piezas actúan de límite físico para que la articulación no pueda girar más de este ángulo con el fin de evitar daños.

Para la colocación de los componentes electrónicos que se mostrarán en el apartado 5.2 ha sido necesario diseñar una base extra que irá situada sobre el cuerpo ya descrito. Además, se ha aprovechado para que los pilares que soportan dicha base extra, actúen de tope para la colocación de la batería de potencia. También se ha aprovechado para colocar un interruptor circular empotrado en una nueva pieza para la alimentación de los motores como se muestra a continuación:

Diseño, desarrollo y construcción de un robot hexápodo para inspección de áreas de difícil acceso.

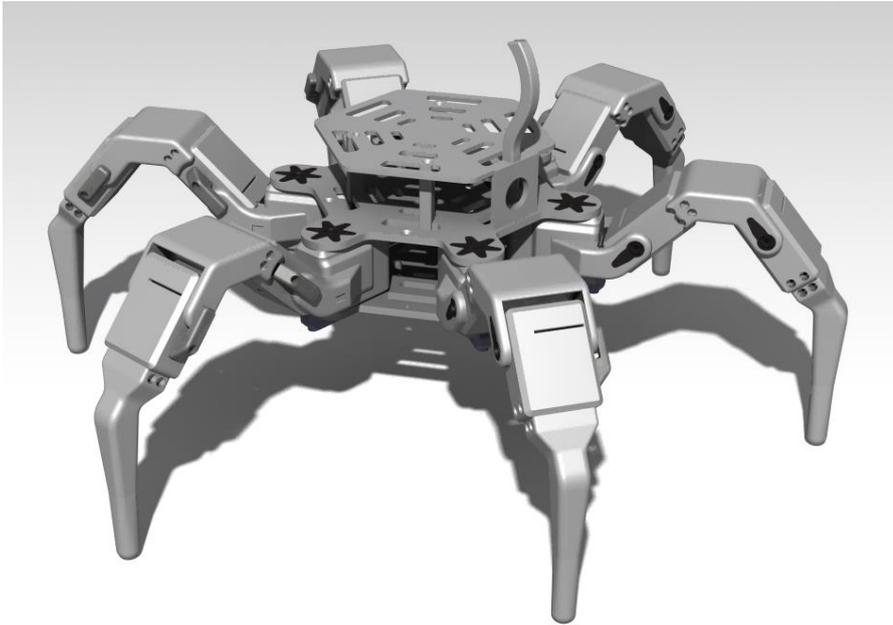


Figura 47. Ensamblaje de la estructura completa. Elaboración propia

Para la colocación de la cámara en la parte superior se ha diseñado un soporte que aportará la altura y el ángulo suficiente para que la cámara pueda enfocar a la parte delantera del robot para poder controlar el desplazamiento de forma remota. Se trata del perfil curvo que sobresale hacia arriba, situado sobre el hueco del interruptor.

Por último y para facilitar la colocación de los componentes de la capa inferior del robot, se ha diseñado una placa en la que posicionarán y atornillarán los componentes electrónicos de forma que se eliminará el riesgo de cualquier contacto eléctrico no deseado.

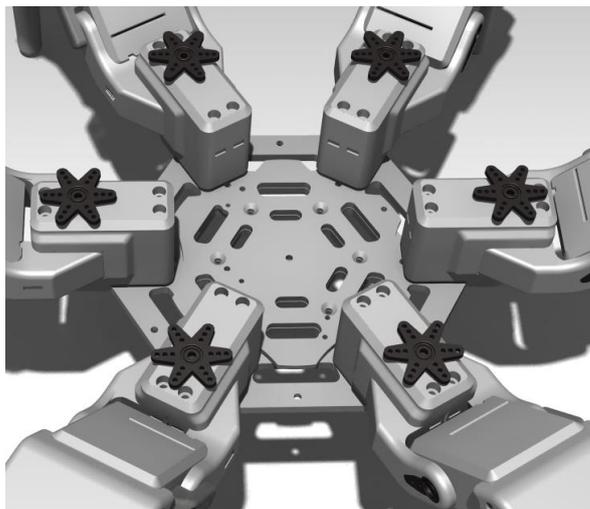


Figura 48. Detalle de la placa de componentes. Elaboración propia

Diseño, desarrollo y construcción de un robot hexápodo para inspección de áreas de difícil acceso.

5.2 Disposición de los elementos electrónicos

La disposición de los componentes electrónicos es crucial para la funcionalidad y el mantenimiento del robot. Se ha tenido especial cuidado en la ubicación y fijación de cada elemento para asegurar su protección y accesibilidad. Además, se ha intentado mantener suficientemente alejados los componentes que pueden causar interferencias, como el regulador de tensión y los controladores, además de utilizar cable apantallado para la comunicación de las señales entre el microcontrolador y la controladora de servos.

A continuación, se explica brevemente la colocación de cada elemento electrónico, organizados por su localización en tres capas:

Capa inferior. Señales

Esta capa es la localizada entre las dos bases hexagonales que conforman el cuerpo del robot.

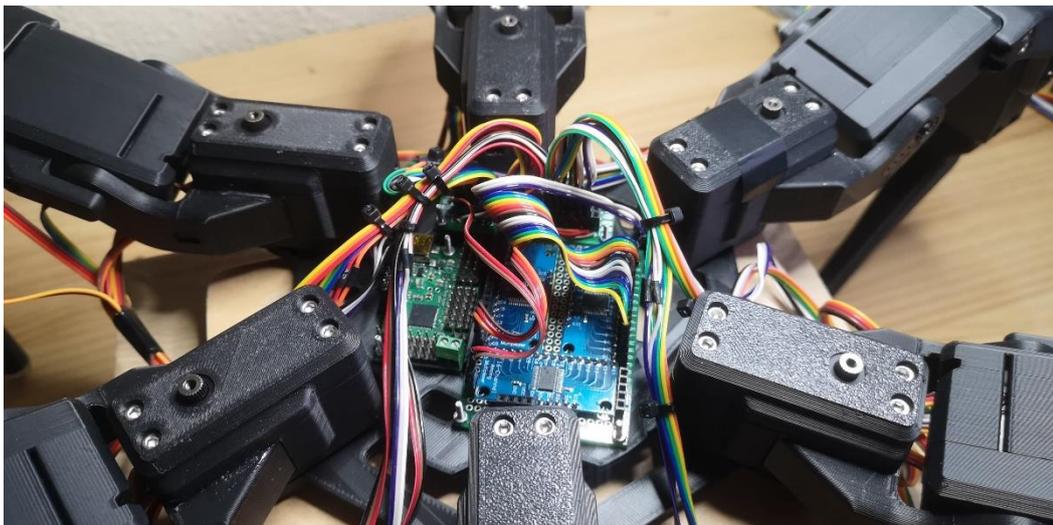


Figura 49. Colocación de la electrónica en la capa inferior. Elaboración propia

En ella se han localizado el controlador de servos Minimaestro, así como la concentradora de señales de los sensores. Para ello se ha diseñado una pequeña base, donde poder atornillar cada placa y que queden aseguradas. Se han decidido situar en esta capa debido al gran número de cables, procedentes de las patas que tienen que ir conectados a ellos. Por cada pata salen tres conjuntos de tres cables provenientes de cada servomotor, así como otros tres conjuntos de tres cables provenientes de los potenciómetros. Todos ellos entran y van conectados entre estas dos bases.

Diseño, desarrollo y construcción de un robot hexápodo para inspección de áreas de difícil acceso.

Capa intermedia. Potencia

Sobre la base superior del cuerpo se sitúa la capa dedicada a la potencia, en la cual se sitúa tanto la batería de alimentación de los servomotores, como el regulador de tensión, que irá situado en la parte delantera. Además, en la parte trasera se encuentra el interruptor de alimentación. De esta forma, los cables de potencia no estarán cercanos a las placas de control.

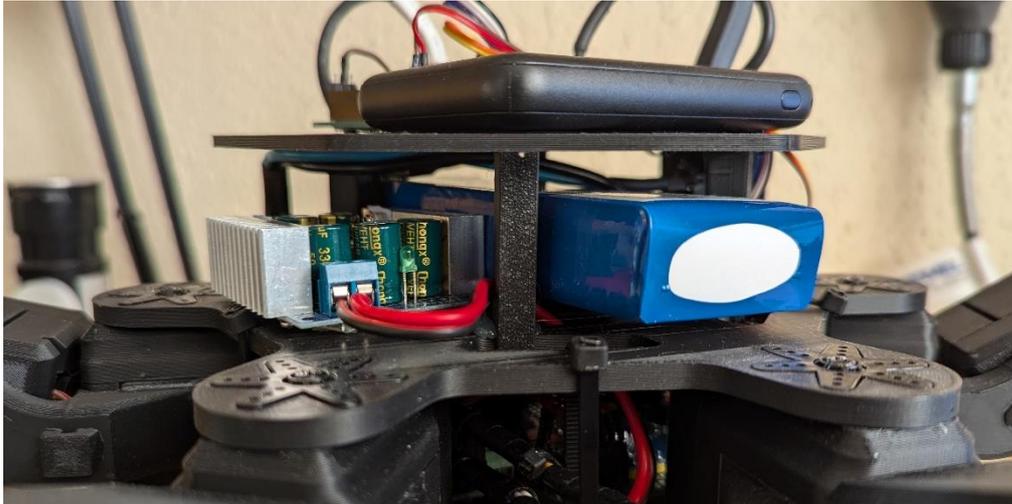


Figura 50. Elementos de la capa intermedia. Elaboración propia

Capa superior. Control

Por encima de la capa anterior se ha diseñado otra base, sobre la cual se sitúa tanto la batería que alimenta los circuitos de control, como el microcontrolador ESP32. De esta forma la antena de comunicación del ESP32 se encuentra en el punto más exterior del hexápodo, favoreciendo la limpieza en las señales.

La única desventaja de esta colocación es que los cables de comunicación entre el microcontrolador y las placas de señales de la capa inferior deben atravesar la capa de potencia. Es por ello que para transportar la señal del microcontrolador al controlador de servos, se ha utilizado cable apantallado, para evitar posibles interferencias en las comunicaciones.

Las señales de los potenciómetros, al trabajar en baja frecuencia no tendrán este tipo de problemas.

Diseño, desarrollo y construcción de un robot hexápodo para inspección de áreas de difícil acceso.



Figura 51. Elementos de la capa superior. Elaboración propia

Finalmente, en esta capa también se ha situado el soporte sobre el cual va colocada la cámara de visión en tiempo real.

5.3 Puntos de apoyo de las patas

Los puntos de apoyo de las patas son críticos para la estabilidad y movilidad del hexápodo. Cada pata debe tener un punto de apoyo que permita una buena tracción en diferentes tipos de terreno. Para el prototipo, teniendo en cuenta que es principalmente testado en interiores donde las superficies son muy lisas, se han utilizado unos toques de silicona con forma semiesférica, que aportan agarre y evitan que las patas deslicen sobre las superficies.

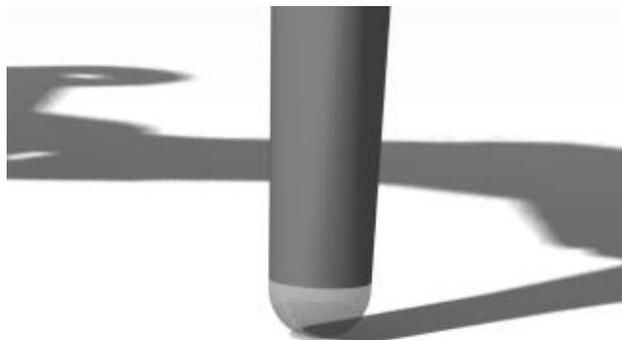


Figura 52. Patas de silicona. Elaboración propia

Diseño, desarrollo y construcción de un robot hexápodo para inspección de áreas de difícil acceso.

Está pensado que en el futuro estos apoyos sean intercambiables y se puedan encontrar diferentes puntos de apoyo en función de las superficies donde vaya a operar el robot.

5.4 Soporte de la pantalla

Para poder sostener la pantalla se ha diseñado una pieza intermedia, partiendo de un modelo de adaptador para mando de PS3 encontrado en el repositorio online Thingiverse [22]. Esta pieza encaja perfectamente tanto con el adaptador como con el acople del que disponía la pantalla, y se ha unido con tornillos y tuercas M3. De esta forma ha quedado perfectamente integrado tal y como se muestra a continuación. Además, se puede montar y desmontar del mando sin dificultad.



Figura 53. Pantalla con soporte. Elaboración propia

Diseño, desarrollo y construcción de un robot hexápodo para inspección de áreas de difícil acceso.



Figura 54. Pantalla ensamblada en el mando. Elaboración propia

5.5 Fabricación del robot

En este apartado se pasa a describir el proceso de fabricación, desde que se termina con el diseño de los modelos 3D hasta la obtención de las piezas para su montaje.

Tecnología

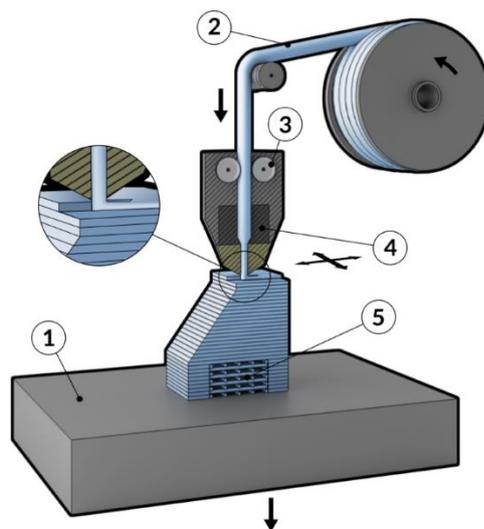


Figura 55. Diagrama de tecnología FDM. Fuente [32]

Diseño, desarrollo y construcción de un robot hexápodo para inspección de áreas de difícil acceso.

La tecnología utilizada para la fabricación de todos los componentes ha sido fabricación aditiva, y en concreto la deposición de material fundido o FDM [23]. Esta tecnología consiste en tomar una bobina del polímero elegido (2), de un diámetro uniforme, habitualmente de 1,75 o 2,85mm, y tras ser empujado por el motor del extrusor (3), fundirlo a través de un canal calefactado (4) e irlo depositando en finas capas sobre una base de fabricación (1).

Material



Figura 56. Bobina PLA+. Fuente [33]

El material elegido ha sido PLA+ del fabricante SUNLU. El PLA [24] es uno de los materiales más utilizados para esta tecnología, por su buen funcionamiento. Es bastante económico, tiene un bajo punto de fusión, da menos problemas que otros materiales en cuanto a su fabricabilidad y además se trata de un material biodegradable, al estar formulado a través de compuestos orgánicos.

Es un material sin embargo que cuenta con una dureza considerable y bajo grado de elasticidad. Puede ser conflictivo para algunas aplicaciones, pero concretamente para la fabricación de cobertores e incluso estructuras es muy adecuado.

Concretamente el material utilizado ha sido una reformulación llamada comercialmente PLA+, el cual proporciona algo más de resistencia y alguna mejora en cuanto a la fabricabilidad, como la velocidad de impresión. Tiene una buena cohesión entre capas, por lo que no es muy probable que las piezas se rompan.

Diseño, desarrollo y construcción de un robot hexápodo para inspección de áreas de difícil acceso.

Impresora



Figura 57. Creality Ender 3 Max Neo. Fuente [34]

El equipo utilizado para fabricar las piezas ha sido una Creality Ender 3 Max Neo. Se trata de un fabricante muy conocido por producir máquinas económicas y con una buena relación entre prestaciones y precio. Se trata de un equipo con un volumen de fabricación de 300x300x320 mm, considerablemente mayor a la gran mayoría del mercado que cuenta con una base habitualmente de 200x200. Estos centímetros extras proporcionan la posibilidad de realizar piezas mayores y con mayor resistencia, debido a que se pueden fabricar como un solo elemento, en lugar de dividirlo en varias partes.

Además, para producciones algo mayores, como es el caso de los componentes del hexápodo, permite la realización de lotes más grandes, reduciendo tiempos de producción.

Tolerancias

Un aspecto muy importante a tener en cuenta a la hora de fabricar componentes es la tolerancia. Esta tolerancia depende tanto del equipo, como del polímero, como de las opciones de laminación. Además, es un aspecto imprescindible a tener en cuenta cuando los componentes tienen que encajar entre sí, o bien si hay que encajar otros elementos dentro de dichos componentes. Ambas situaciones han tenido lugar en el robot en estudio.

Además, se ha ajustado de forma precisa para poder realizar piezas como por ejemplo los cobertores de los potenciómetros, debido a que el ajuste tiene que ser perfecto para que no haya holguras entre el potenciómetro y la pieza, dado que su buen funcionamiento depende de este ajuste. Lo mismo sucede con los componentes que tenían que encajar con los acoples de los servomotores, ya que no van pegados ni atornillados, y es muy importante que no existan holguras.

Diseño, desarrollo y construcción de un robot hexápodo para inspección de áreas de difícil acceso.

Para poder ajustar estas tolerancias lo principal es realizar una serie de piezas de test inicialmente, realizando medidas y controlando estas expansiones. Habitualmente suele haber expansiones horizontales de entre 2 y 3 décimas de milímetro en cada pared. Esto es propio de la tecnología, debido a que la boquilla por la que sale el plástico fundido es circular con un diámetro en este caso de 0,4mm. Esta boquilla se aproximará a la capa previamente depositada una distancia de 0,2 o 0,3 mm en función de la altura de capa elegida. Sin embargo, al aproximarse tanto, este plástico es aplastado formando unas pequeñas crestas entre una capa y la siguiente generando este sobre espesor. Esto hay que tenerlo en cuenta a la hora de diseñar, para aplicar estos offsets en función de si es una pieza interior o exterior y que el ajuste sea óptimo.

En el caso de geometrías que requieran un ajuste muy fino, además de la realización de estos tests iniciales, será conveniente realizar piezas de prueba, dado que la propia geometría también puede hacer variar estas expansiones. Una vez hecha la prueba se puede volver al programa de diseño para ajustar más aún las dimensiones de las piezas.

También existen opciones en los softwares de laminado para ajustar esta expansión horizontal.

Soportes

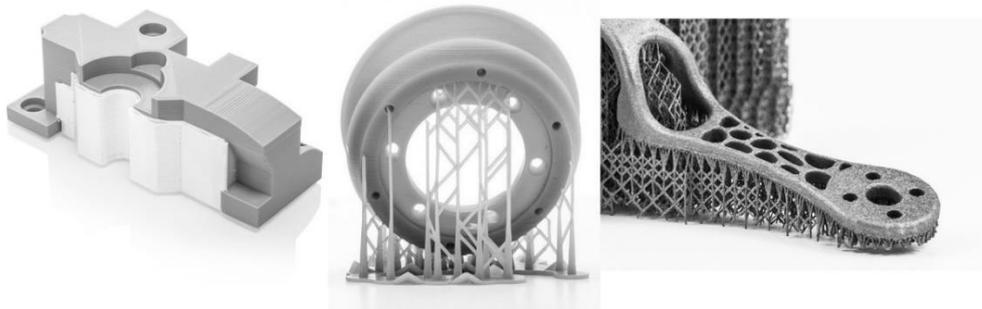


Figura 58. Estructuras de soporte. Fuente [35]

Esta tecnología de fabricación no permite directamente la creación de voladizos en piezas de más de unos pocos milímetros. Como la deposición del material fundido se realiza siempre sobre la capa anterior, cuando hay una superficie en voladizo, es decir, sin material por debajo, el plástico fundido tiende a descolgarse. Este factor depende en gran medida de la máquina, de los parámetros, temperaturas y velocidades, pero en mayor o menor medida es un problema que afecta en general a la tecnología.

Pues bien, para poder fabricar este tipo de geometrías en primer lugar se puede orientar la pieza de forma que se pueda minimizar estas áreas en voladizo. Sin

Diseño, desarrollo y construcción de un robot hexápodo para inspección de áreas de difícil acceso.

embargo, en el caso de no poder eliminarlas del todo, se deberán utilizar unas estructuras de soporte temporales, que se fabricarán para poder conformar esas áreas y una vez fría la pieza se puedan retirar sin mayor problema. Esto no siempre se puede realizar sin consecuencias, pero dada la gran comunidad en este sector, las opciones de los laminadores para crear este tipo de estructuras de soporte, ha mejorado notablemente, permitiendo crear geometrías que apenas utilizan plástico y que se pueden retirar sin problema, dejando la pieza prácticamente intacta.

Sin embargo, otra opción a la hora de diseñar componentes propios es el llamado DFM o Design for manufacturing. Esto significa, conocer perfectamente los requerimientos del método de fabricación y realizar un diseño acorde a ello. En este caso, visualizando la orientación que va a tener la pieza en la máquina según es diseñada, se pueden minimizar las zonas de voladizo o eliminarlas introduciendo chaflanes o paredes en ángulo para no tener que hacer una pared en horizontal.

Además, conociendo las limitaciones de las opciones de laminado, en este caso se han realizado una serie de soportes por diseño, con un bajo espesor para facilitar su retirada, que reforzarán los propios soportes que genera el software de laminado en las zonas más críticas.

Orientación de los componentes en la base de fabricación

Una vez diseñados los componentes es el momento de proceder al laminado. El primer paso será el de orientar estos componentes en la base de fabricación. Una vez conocido el factor de los soportes, deberán tenerse en cuenta también las propiedades de la tecnología en cuanto a las direcciones de fabricación. Por las propias características de esta tecnología, no es posible obtener componentes plenamente isótropos. Dado que se fabrica por capas, dentro de cada capa (direcciones x e y) la resistencia será muy alta, pero la cohesión entre capas (dirección z) será mucho menor, por lo que siempre será más fácil que se rompa una pieza entre capas que en otra dirección. Es por ello que es muy importante conocer los esfuerzos a los que va a estar sometidos una pieza para poder decidir su orientación en la fabricación y así optimizar su funcionamiento.

Opciones de laminado

Decidida la orientación de los componentes llega el último paso previo a la fabricación, el laminado. Para poder preparar las piezas para su producción, existen gran variedad de programas que realizan esta función. Se denominan así puesto que son los encargados de descomponer los sólidos en láminas, que obedecerán a las capas que se irán fabricando gradualmente durante la impresión. Estas láminas no son más que la visualización que proporciona este

Diseño, desarrollo y construcción de un robot hexápodo para inspección de áreas de difícil acceso.

tipo de softwares de las trayectorias de deposición que va a ir realizando el extrusor de la máquina a través de toda la fabricación, descomponiéndose en capas, y dentro de estas, en la trayectoria dentro de cada capa. Al fin y al cabo, es un archivo G-CODE, compuesto de una consecución de órdenes en lenguaje máquina, que puede interpretar el controlador de la impresora, que le indicarán en todo momento las temperaturas de los diferentes componentes, velocidades y aceleraciones de los desplazamientos, así como todas las trayectorias que tendrá que ir siguiendo para conseguir el conformado de los productos.

Por regla general se han mantenido la mayor parte de la configuración del laminado por defecto, pero se han ajustado ciertas variables para mejorar los resultados, como por ejemplo aumentar el número de perímetros exteriores a 3, aumentar el relleno al 40% para mejorar su resistencia, elevar la temperatura de extrusión a 210 °C para mejorar la cohesión entre capas y una serie de mejoras estéticas para que el acabado sea mejor.

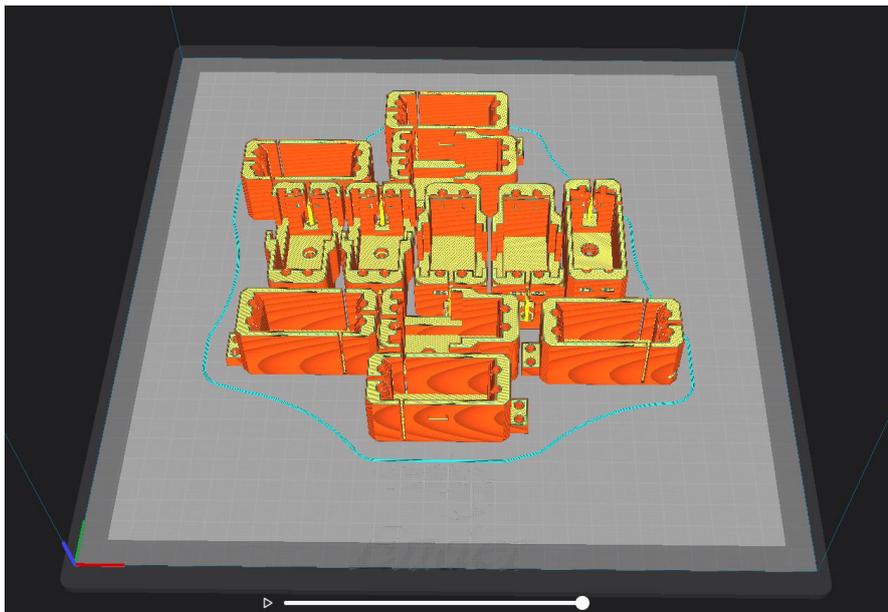


Figura 59. Lote de fabricación laminado. Elaboración propia

Diseño, desarrollo y construcción de un robot hexápodo para inspección de áreas de difícil acceso.

Fabricación

Por último, se ha procedido a la fabricación de los lotes de componentes del robot. Se han dividido los lotes en piezas de similar orientación y requerimientos. Se ha realizado un total de diez impresiones, cuyo tiempo de fabricación total aproximado ha sido de unas 120 horas a una media de 12 horas por fabricación.

Estos tiempos se pueden modificar sustancialmente en función de los parámetros de relleno, perímetros exteriores o capas sólidas inferior o superior. Además, depende de las velocidades configuradas que dependerán a su vez en gran parte de las posibilidades del equipo, por lo que estos tiempos podrían reducirse notablemente en el caso de contar con otro equipo de características superiores.

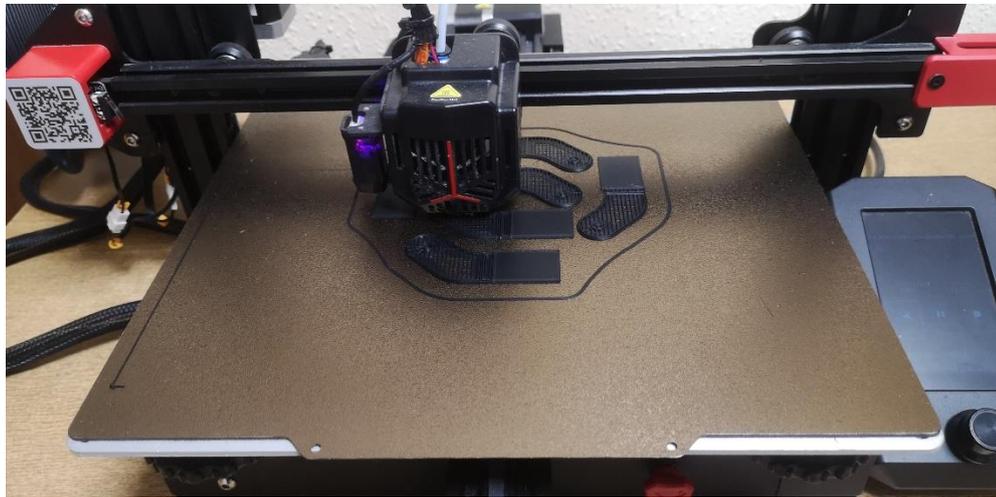


Figura 60. Fabricación en estado inicial. Elaboración propia

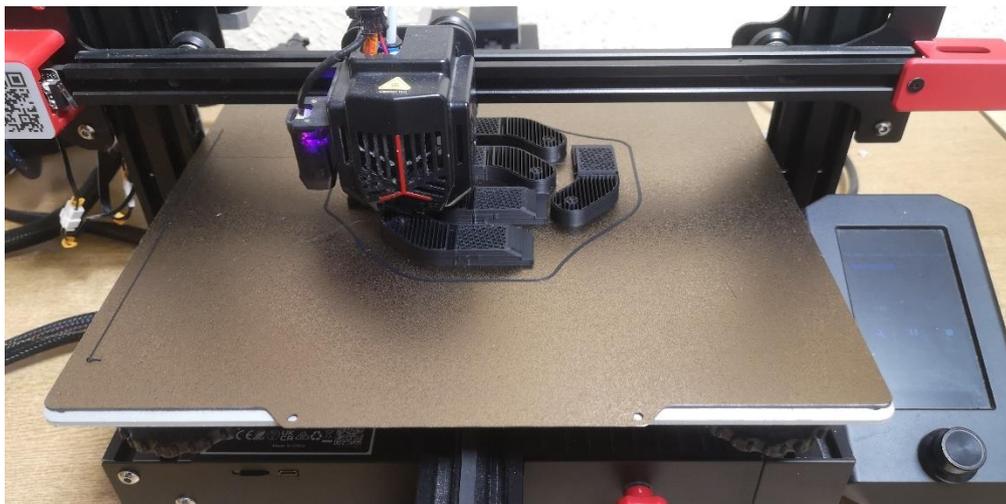


Figura 61. Fabricación en estado intermedio. Elaboración propia

Diseño, desarrollo y construcción de un robot hexápodo para inspección de áreas de difícil acceso.



Figura 62. Lote de fabricación finalizado. Elaboración propia



Figura 63. Pieza con soporte intacto (izq.) y con el soporte retirado (de). Elaboración propia

Diseño, desarrollo y construcción de un robot hexápodo para inspección de áreas de difícil acceso.



Figura 64. Lotes de piezas correspondientes a las patas. Elaboración propia

5.6 Montaje del robot

Una vez fabricada la totalidad de las piezas, se ha procedido a su montaje. En primer lugar, se ha introducido cada servomotor dentro de su cobertor, teniendo en cuenta que se debe incluir en varios de ellos una tuerca para la colocación del potenciómetro y un tornillo que hará de eje. Una vez cerrados los cobertores se han comenzado a montar las patas uniendo cada segmento con el siguiente por medio de los “brazos” que articulan estas uniones en los ejes de los servos. Finalmente se colocan los potenciómetros con sus cobertores y se concluye con el montaje de las patas.

Una vez montadas se situarán sobre la base inferior, se incluirá toda la electrónica con el conexionado de servomotores y sensores. Para que los cables queden más ordenados se ha decidido colocar cobertores en espiral para organizar los cables por conjuntos de cada pata, por un lado, los de los motores y por el otro los de los sensores. Se cerrará colocando la base superior mediante los acoples de los servomotores de giro de cada pata y se colocarán sus tornillos, quedando con ello la estructura del robot cerrada.

Para continuar, se deberá colocar la batería de potencia con el regulador de tensión y el interruptor y finalmente en la capa superior, la batería de control, la placa de control y la cámara.

En cuanto a la tornillería, se han utilizado tornillos con cabeza Allen métrica 3, de acero inoxidable DIN912, además de tuercas y arandelas. Se muestra a continuación el listado de la tornillería necesaria para el montaje.

Diseño, desarrollo y construcción de un robot hexápodo para inspección de áreas de difícil acceso.

Tornillos Patas	
M3x20	102
M3x10	24
M3x8	12
M3x6Av	12
M3x6	6

Tornillos cuerpo	
M3x10Av	14

Tuercas cuerpo	
M3	14

Tuercas patas	
M3	138

Arandelas patas	
M3	6

Figura 65. Listado de tornillería. Elaboración propia

Una vez montadas las patas se deberá proceder a la calibración de la posición de los servomotores. Para ello se deberán desatornillar los acoples, extraerlos ligeramente, posicionar por medio de la controladora de servos todos los servomotores en cero grados y a continuación volver a colocarlos y atornillarlos. De esta manera, se podrán ajustar de una forma sencilla y rápida.

Para finalizar quedaría conectar el resto de cables de control y alimentación y finalizar por embridar los cables sobre los soportes situados en la parte inferior del segmento 2 de cada pata, de tal forma que no pueda haber ningún problema con ellos.

Diseño, desarrollo y construcción de un robot hexápodo para inspección de áreas de difícil acceso.

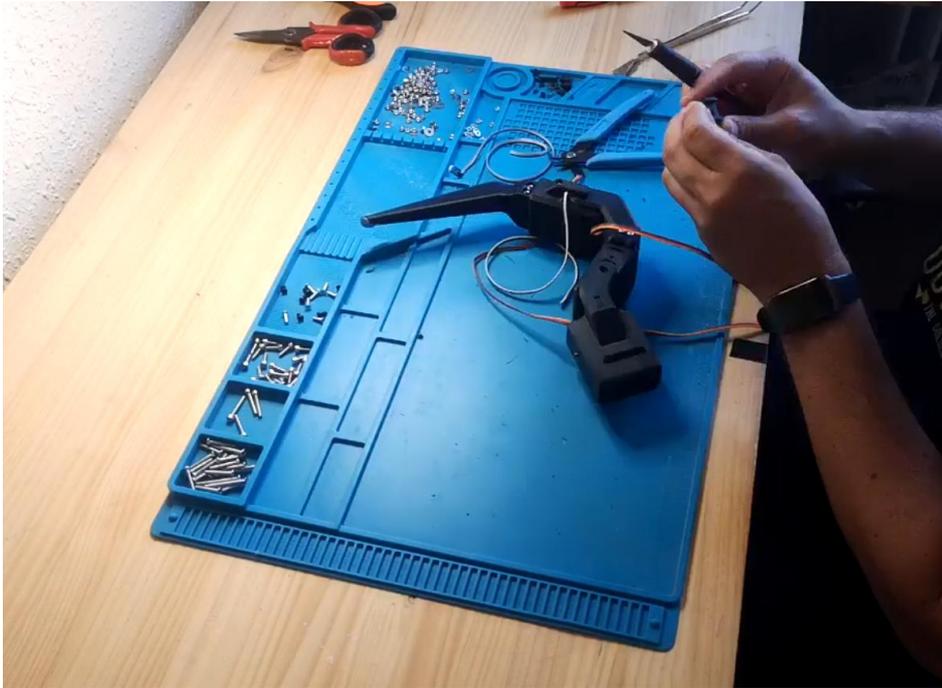


Figura 66. Montaje del robot. Elaboración propia

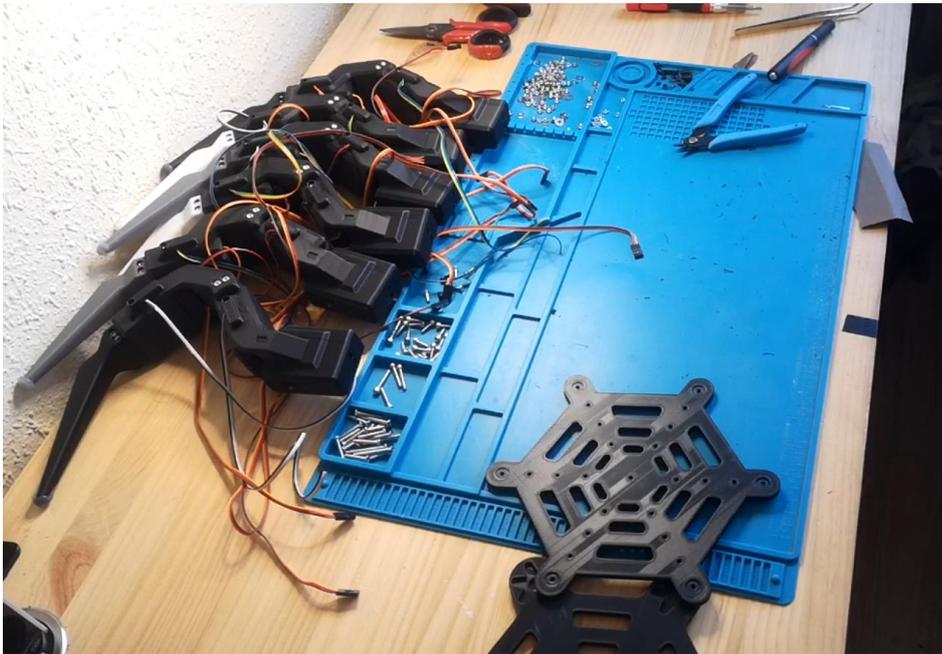


Figura 67. Montaje del robot. Elaboración propia

Diseño, desarrollo y construcción de un robot hexápodo para inspección de áreas de difícil acceso.

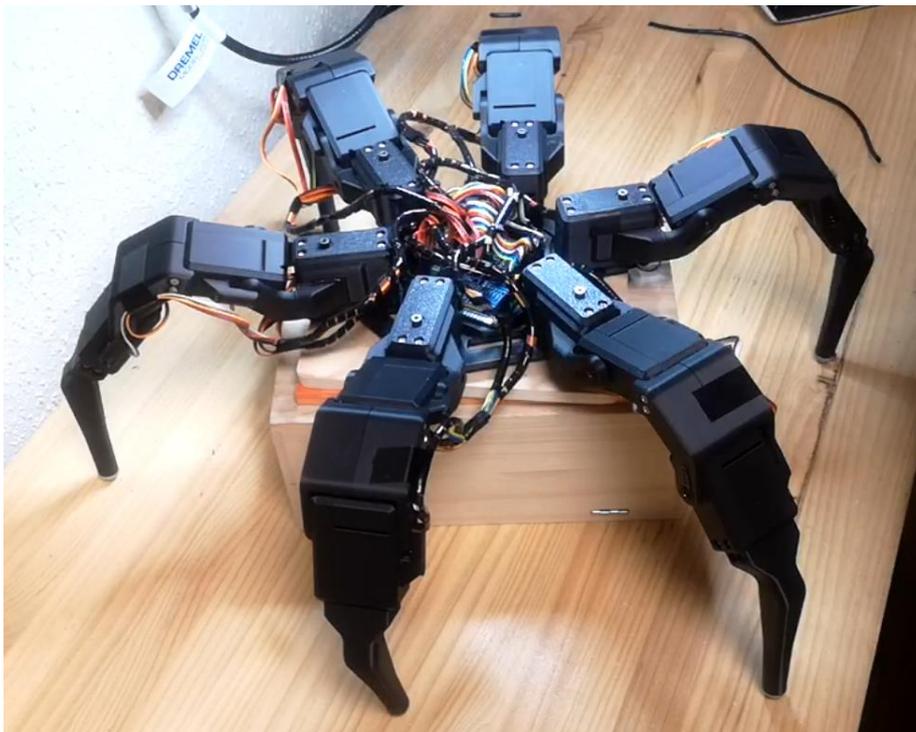


Figura 68. Montaje del robot. Elaboración propia



Figura 69. Montaje del robot. Elaboración propia

6. Control y programación

6.1 Configuración de la controladora de servos

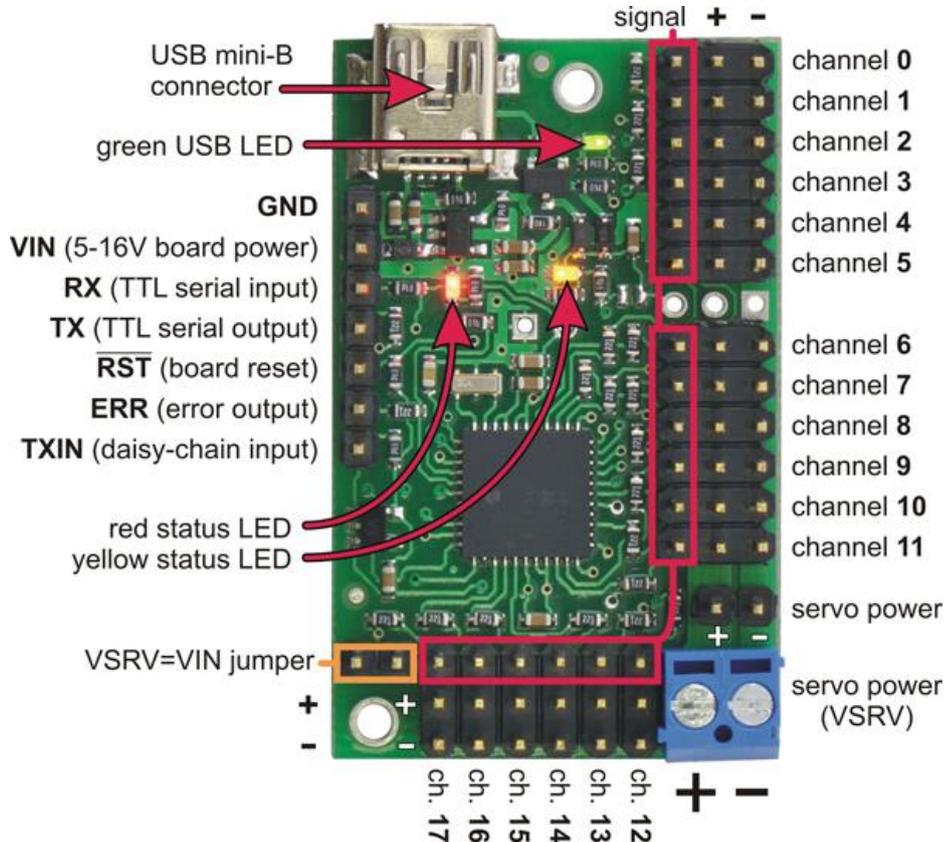


Figura 70. Controladora Pololu Mini-Maestro. Fuente [36]

En primer lugar, se va a configurar la controladora de servos. Esta controladora actúa de concentradora tanto de señales como de alimentaciones para todos los servos. Cuenta con la alimentación de potencia exterior, a través de un puerto de dos vías (Servo Power VSRV), que alimenta directamente todos los servos. A su vez, su microprocesador gestiona todas las señales de control de los 18 servos (signal). Esta placa cuenta con tres modos de control.

1.- USB a través de conexión directa con Pc. A través de este modo, por medio de una interfaz para Windows o Linux sencilla, se puede manipular velocidad, posición y aceleración de cada servo en tiempo real. Sin embargo, aunque es un modo interesante a la hora de hacer pruebas, a la hora de controlar numerosos servos de manera simultánea se vuelve inoperativo. También es posible almacenar una secuencia de posiciones desde este modo. Se posiciona el conjunto de servomotores en la configuración requerida y se van almacenando puntos, que pueden ser renombrados y reordenados. De esta forma, de una manera muy visual se puede ir generando una secuencia de

Diseño, desarrollo y construcción de un robot hexápodo para inspección de áreas de difícil acceso.

movimiento que luego se puede reproducir en bucle para así crear un funcionamiento autónomo. Sin embargo, este modo tampoco resulta interesante debido a la imposibilidad de adaptar y modificar el movimiento en tiempo real.

2.- Scripts internos. El segundo modo de funcionamiento es a través de scripts internos. Una pequeña evolución del modo anterior. Permite introducir código a través de la interfaz de PC con las configuraciones de todos los servos, generar las secuencias y ejecutarlo en bucle. Sin embargo, al igual que en el anterior caso, tampoco resulta interesante debido a la imposibilidad de modificar el comportamiento preestablecido del robot.

3.- UART TTL. Finalmente, el modo elegido ha sido el de comunicación serie en tiempo real. Resulta mucho más interesante para la aplicación en concreto que los anteriores. Desde una placa de control con comunicación serie, (Rx y Tx), se van enviando instrucciones que determinan las posiciones, velocidades y aceleraciones de cada servo en tiempo real. Existen una serie de instrucciones predeterminadas por el fabricante que han sido utilizadas en las diferentes funciones del programa de control para mover el robot [25] dentro de los manuales, en la sección 5.e del documento “PololuMaestroGuiaDeUsuario.pdf” y de “maestro.pdf”.

Finalmente, para dejar configurada la controladora para poderla utilizar en este modo, es necesario entrar en la interfaz de PC a través de la aplicación “Pololu Maestro Control Center”.

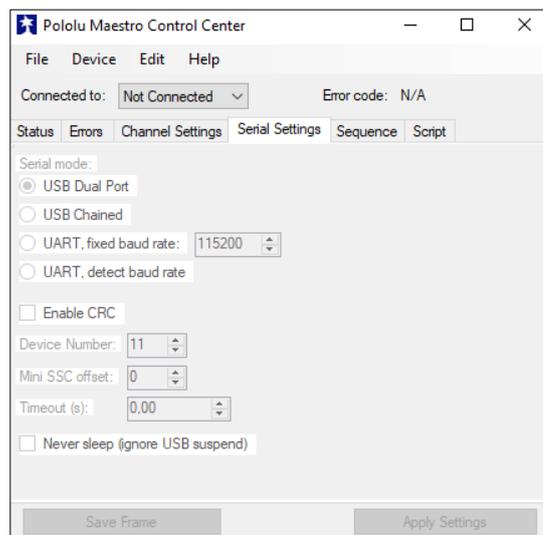


Figura 71. Pololu Maestro Control Center. Elaboración propia

En la pestaña *Serial Settings*, se deberá elegir *UART, fixed baud rate* con la velocidad que se requiera para la comunicación con la placa de control (en este caso se ha seleccionado 9600 baudios). A continuación, se deshabilitará el

Diseño, desarrollo y construcción de un robot hexápodo para inspección de áreas de difícil acceso.

CRC (Control de Redundancia Cíclica) para la detección de errores. Este es un requisito de la biblioteca `<PololuMaestro.h>`, utilizada para la programación del ESP32, debido a que no contempla estos bits de seguridad. Además, evitándola, se consigue una mayor velocidad de comunicación, y el error, en el caso de que ocurriera, sería rápidamente solucionado, debido a que los servomotores trabajan con posiciones absolutas, por lo que dicho error sería puntual y no se acumularía, corrigiéndose con la siguiente instrucción. Una vez configurado, se pulsa el botón de “*Apply Settings*” y la placa estará lista para recibir comandos por puerto serie. MiniMaestro cuenta con tres posibles protocolos de comandos utilizables: COMPACT, POLOLU Y MINISSC. En concreto la biblioteca utilizará el protocolo COMPACT, por lo que será este el que determine los comandos que se deben utilizar.

Además de los pines que se han descrito anteriormente, la controladora cuenta con otras conexiones. Dispone de un jumper para activar o desactivar la alimentación común, para control y potencia. En este caso el jumper estará desconectado, ya que la alimentación de los motores se hará a una tensión que el control no soportaría.

También dispone de otros tres pines como son RESET, ERROR y el “DAISY-CHAIN INPUT”. RESET será una entrada para reiniciar la controladora, ERROR un pin de salida que se podrá a nivel alto en el caso de que la controladora haya detectado un error, y por último el “DAISY-CHAIN INPUT” se trata de un pin de control para la conexión de dispositivos en cadena. Ninguno de ellos tendrá utilidad para este proyecto, por lo que no se han utilizado.

6.2 Configuración y programación del ESP32

En cuanto a la configuración del ESP2 se realizará directamente con el código de programación, que se realizará a través del IDE (Entorno de desarrollo integrado) de Arduino. Esto proporciona una interfaz amigable y útil a la hora de programar el ESP32. Además, facilita la labor de la instalación y actualización de los drivers y bibliotecas. Para la configuración del IDE ha sido necesario la instalación de los drivers de *esp32* de *Espressif Systems*, así como la descarga e instalación de la biblioteca *PololuMaestro* de *Pololu*. Esta biblioteca proporciona algunos ejemplos de utilización, así como las funciones básicas para el envío de las tramas por el puerto Serie.

Una vez descargado e instalado todo, se pasará a la programación propiamente dicha. Dentro del código se debe incluir la biblioteca `<PololuMaestro.h>`, como se ha indicado, y se debe configurar el puerto serie, que se comunicará con el Mini-Maestro, al cual se debe aplicar la misma velocidad de transferencia que en la configuración anterior (9600 baudios). Para completar la comunicación,

Diseño, desarrollo y construcción de un robot hexápodo para inspección de áreas de difícil acceso.

se deberán interconectar físicamente los pines RX y TX de ambas placas. (TX de ESP32 con RX de Pololu MiniMaestro y RX de ESP32 con TX de Pololu MiniMaestro). En este caso se han utilizado los pines 16 como RX y 17 como TX, correspondientes al puerto U2 del ESP32.

Para la conexión del multiplexor de señales de entrada se necesitarán en primer lugar cinco pines digitales de salida, que representarán los cinco bits de selección de la entrada del multiplexor, conectados a los pines 27, 26, 25, 33 y 32, que corresponderán a los bits de S0 a S4. Además, la placa cuenta con otros cuatro pines, dos de los cuales serán de alimentación que irán conectados a 3,3V y GND y otros dos, que serán Enable y la señal de salida. Enable se ha conectado al puerto 35, que funcionará como salida digital para poder habilitar y deshabilitar la placa. Por defecto estará habilitada. Por último el pin de lectura que será una entrada analógica y estará conectada al puerto 34.

En los siguientes apartados se hablará de las funciones tanto de control remoto como de movimiento del robot. En ellas se describirán y enumerarán ciertas funciones y procedimientos del código de control. Para cualquier consulta, este se encontrará íntegro en los anexos.

6.3 Configuración y funciones del mando de control

Para la programación del mando se ha utilizado una biblioteca disponible desde el IDE de Arduino. Para su configuración se han seguido las instrucciones del desarrollador de la biblioteca (Jeffrey Van Pernis) localizada en Github [26].

En primer lugar, se deberá averiguar la dirección MAC (Media Access Control) del mando que se va a utilizar. Para ello, se deberá descargar e instalar un software llamado SixAxisPairTool, mediante el cual, conectando el mando vía USB al PC en el que esté instalado dicho software, se puede acceder a leer la dirección MAC de su hardware, así como modificarla. En este caso, sólo será necesario visualizarla y copiarla al portapapeles.

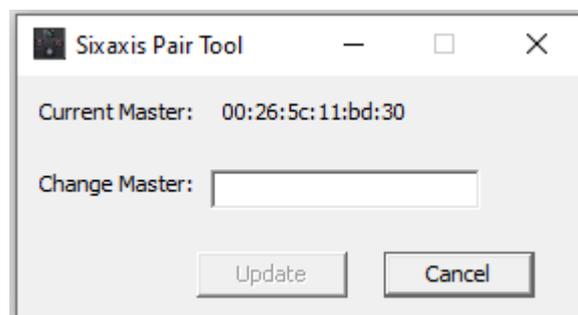


Figura 72. Cuadro de visualización de la dirección MAC. Elaboración propia

Diseño, desarrollo y construcción de un robot hexápodo para inspección de áreas de difícil acceso.

Será el código hexadecimal mostrado en el campo “*Current Master*”.

Una vez averiguada la MAC, se deberá acudir al IDE de Arduino, donde se deberá descargar la biblioteca *PS3 Controller Host* <*Ps3Controller.h*>. Una vez descargada y tras incluir la cabecera en el programa principal, se podrán utilizar todas sus funciones.

Para poder visualizar todos los controles, se puede abrir y cargar en el ESP32 el ejemplo *PS3DataNotify.ino* en el cual se podrán encontrar los comandos necesarios para leer cada pulsador o joystick, y a través del cual se podrá mostrar por pantalla los valores que recibe el ESP32 del mando. Para que funcione con el mando elegido, tan sólo se deberá recuperar la dirección MAC conseguida con el software anterior e introducirla al comienzo del código con el comando “*Ps3.begin("00:26:5c:11:bd:30");*”, donde el código en Hexadecimal entre las comillas será dicha dirección MAC.

Finalizado este paso se podrá conectar el mando pulsando el botón central y el programa, a través de su monitor de señales comenzará a mostrar todas las pulsaciones que se realicen en el mando.

Una vez comprobado el funcionamiento del mando, se han incluido las líneas de comandos necesarias en una nueva página del código del robot para su control.

Cabe destacar que el modo de funcionamiento de las lecturas del mando se realizará por medio de interrupciones. De esta manera, con un sistema de banderas que se irán levantando y bajando a medida que se pulsen o se levanten los botones en el control remoto, se irán integrando dichas señales en el flujo normal del programa. Es necesario integrarlo de esta forma para que se pueda realizar cada movimiento en su momento justo, sin interrumpir cada secuencia de movimientos, dada la gran complejidad de estas secuencias.

A continuación, se enumeran los controles de mando que serán utilizados, así como sus funcionalidades:

Diseño, desarrollo y construcción de un robot hexápodo para inspección de áreas de difícil acceso.



Figura 73. Controles de mando PS3. Fuente [37]

Home Button: Será el botón de conexión del mando. Mediante este pulsador, se conectará el mando al microcontrolador del robot cada vez que este se arranque, para poder proceder a su control. Para poder comprobar que la conexión ha sido exitosa, se iluminará el led L1, situado en la parte superior izquierda del mando.

Left Joystick: Joystick de desplazamiento. Se trata del control que indicará al robot en qué dirección deberá avanzar. Se ha configurado para que pueda avanzar en dirección perpendicular a cada lateral del hexágono base, tomando como la parte trasera el lateral donde se sitúa el conmutador de conexión de potencia de los motores.

Right Joystick: Joystick de giro. En este joystick tan sólo se utilizará el eje X, a través del cual se indicará al robot que deberá rotar hacia la derecha o hacia la izquierda. Se ha configurado el programa para que cada rotación, gire el robot unos 45 grados en la dirección que se esté indicando.

R1 Button: Aumento de la velocidad de movimientos. Este pulsador es el encargado de indicar al robot que se debe aumentar la velocidad en las secuencias de movimientos. Cabe aclarar que no aumentará la velocidad de desplazamiento de los motores en sí, sino que reducirá los tiempos de espera entre cada movimiento dentro de las secuencias. De esta forma, se aumentará

Diseño, desarrollo y construcción de un robot hexápodo para inspección de áreas de difícil acceso.

de forma efectiva la velocidad del robot sin ser necesario modificar la velocidad de los motores. Este valor tiene un rango de seguridad, fuera del cual no es posible modificarlo a través de este control, que variará entre los 300 milisegundos y 1 segundo entre cada movimiento.

L1 Button: Reducción de la velocidad de movimientos. De forma análoga al pulsador anterior, este se encargará de la función contraria. Por su parte, reducirá la velocidad efectiva del robot, aumentando el tiempo entre sus movimientos dentro de cada secuencia. Cuenta con el mismo rango de funcionamiento que el pulsador anterior.

6.4 Control de movimiento del hexápodo

En primer lugar y para que sirva como base de las explicaciones de los movimientos que se realizarán a continuación en este apartado, se hará una breve explicación del funcionamiento de los servomotores.

6.4.1 Control de los servomotores

Los servomotores tienen un rango de funcionamiento de 180 grados, que se ha configurado de -90° a 90° para mayor comodidad a la hora de programar, siendo cero la posición de reposo. Cuentan con un control de posición absoluto, es decir, para controlarlo no es necesario conocer el punto en el que se encuentra cada motor. Tan sólo se le debe dar la posición de destino y será la electrónica interna del servomotor, que consta de un pequeño circuito con un comparador, el que decidirá en qué dirección y a qué velocidad deberá moverse.

Esto es así debido a que el circuito de control de los servomotores, proporciona una velocidad más elevada si el ángulo de destino está más alejado y una velocidad menor si está más cercano. Sin embargo, más adelante se explicará cómo la controladora de servomotores sí que puede ejercer un control en la velocidad y la aceleración de los mismos.

A nivel de control externo, que es lo que se puede ejercer, los servomotores cuentan con tres pines, dos de alimentación por los que se alimentará a 7,2V y uno de control. Por ese pin de control se debe enviar un tren de impulsos constante al servomotor o por el contrario dejará de ejercer fuerza en el eje. La frecuencia de estos pulsos deberá ser siempre fija de 50 Hz, o lo que es lo mismo, su periodo deberá ser de 20 milisegundos. Lo que determinará la posición del servomotor será la relación entre el tiempo que está el pin a nivel alto y el que está a nivel bajo.

A continuación, se puede observar de manera gráfica el funcionamiento en un servo genérico:

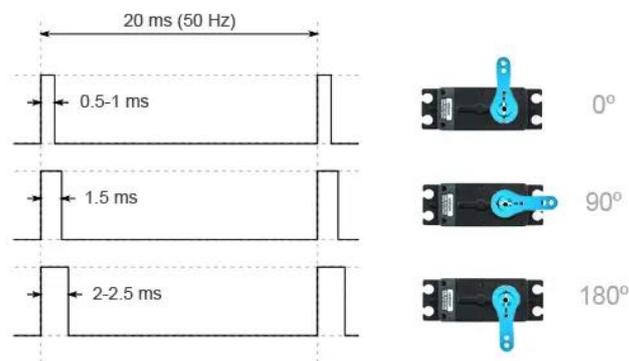


Figura 74. Señales de control de un servo standard. Fuente [38]

Diseño, desarrollo y construcción de un robot hexápodo para inspección de áreas de difícil acceso.

Para este caso en particular, es la biblioteca `<PololuMaestro.h>` la que se encargará de enviar las posiciones a los servomotores, y tan sólo se debe configurar en el programa principal el parámetro de microsegundos máximos y mínimos que debe estar la señal a nivel alto. Para estos servomotores, la posición máxima tendrá un valor de 500 microsegundos y la mínima de 2500 μ s. Estos valores han sido extraídos de la documentación y testados posteriormente con los propios servomotores.

En el programa, el parámetro deberá introducirse multiplicado por cuatro, por requerimientos de la propia biblioteca, por lo que, los valores que se encontrarán en la configuración del programa serán 2000 y 10000 respectivamente. Una vez configurados los valores máximo y mínimo, estos pueden ser utilizados en una función `map()` para convertir una posición en grados entre 0 y 180, en el valor de microsegundos que se debe enviar a las funciones de posicionamiento de los servos. Estas funciones y sus parámetros serán explicadas en apartados siguientes.

Para que los servos no dejen de ejercer fuerza en ningún momento, será la controladora de servomotores la que, de forma autónoma, envíe constantemente las posiciones a todos los servos. Una vez que se envíe una nueva posición, la controladora actualizará la posición de ese servomotor y seguirá enviándola indefinidamente hasta que se mande otra.

Tal y como se ha indicado en los párrafos anteriores, a pesar de que es el servomotor de manera interna el que configura su velocidad en función de la posición objetivo, la controladora de servomotores, realiza una serie de operaciones para poder configurar una velocidad y una aceleración para cada servomotor, independientemente de la posición de destino. Para realizar estos ajustes, lo que hará será, en lugar de enviar directamente el ángulo de destino al servomotor, irá enviando una secuencia de ángulos, en función de la velocidad y aceleraciones elegidas.

En el caso de querer que el servo tenga una velocidad muy baja, tal y como se ha indicado anteriormente, la única forma de conseguirlo de manera externa es enviar una posición muy cercana como destino al servomotor. Esta es la operación que realizará la controladora. En lugar de enviar directamente los pulsos que indiquen el destino, enviará en función de la velocidad configurada, una serie de posiciones cercanas en sentido creciente al motor, para que se vaya desplazando de forma regular, manteniendo la velocidad, hasta alcanzar el destino objetivo.

En el caso de la aceleración, la operación que deberá realizar la controladora a nivel interno será, en lugar de enviar una serie de posiciones equiespaciadas y cercanas unas de otras, enviará una distribución de posiciones variando la

distancia entre las posiciones objetivo para que vaya aumentando o disminuyendo la velocidad en función de la aceleración elegida.

Hay que tener en cuenta que estos ajustes los realiza la controladora a nivel interno y en ningún caso el usuario detectará estas variaciones, dado que la frecuencia de la señal es de 50Hz, o lo que es lo mismo, el periodo de esta señal es de 20 ms. Para hacerse una idea más gráfica, esto significa que cada segundo, el servo debe recibir 50 señales que le indiquen el ángulo de destino. Es por ello que la controladora cuenta con un margen suficiente para realizar estos ajustes sin que sean perceptibles.

6.4.2 Desplazamiento del robot

Como se citó anteriormente, existen varias posibles estrategias de movimiento para un hexápodo. Se pueden diferenciar principalmente cuatro estrategias, siendo dos de ellas bastante similares entre sí [27].

La primera es la llamada *gait alternate* o *tripod gait*, que como su propio nombre indica, alterna las patas en contacto con el suelo, manteniendo siempre tres, a modo de trípode, asegurando la estabilidad del hexápodo. El movimiento se realiza moviendo simultáneamente tres de las patas, dejando las otras tres inmóviles. Una vez concluido el movimiento de estas, se pasaría a mover las opuestas, y así sucesivamente generando el desplazamiento. Si bien es cierto que no se percibe como un movimiento orgánico, como el que realizaría un insecto, es un movimiento que además de ser más sencillo de ejecutar desde el punto de vista del control, asegura siempre una estabilidad, teniendo al menos tres puntos de apoyo, separados entre sí, en contacto con el suelo.

Una variación de este modo es el *triple gait*. Al igual que el anterior, va alternando los apoyos dejando siempre al menos tres en contacto con el suelo. Sin embargo, el movimiento de las tres patas que deben moverse cada vez, no se realiza de manera simultánea, sino que aplica un pequeño offset entre los momentos de activación de cada una, generando un movimiento más orgánico y visualmente más atractivo.

Otra técnica de movimiento sería el *wave gait* o *movimiento de onda*. Este movimiento consiste en ir moviendo cada extremidad de forma independiente y ordenada, de manera que se va creando un efecto de ola en el movimiento global. En primer lugar, desplaza las tres patas de uno de los lados de adelante a atrás, para a continuación comenzar a desplazar las del otro lado, en el mismo orden. Es un tipo de movimiento que no tiene mucho interés a nivel desplazamiento, debido sobre todo a que lo ralentiza, teniendo que esperar

Diseño, desarrollo y construcción de un robot hexápodo para inspección de áreas de difícil acceso.

para realizar un nuevo movimiento de cada pata, a haber movido el resto previamente.

Finalmente se encuentra el *ripple gait*. Este movimiento es el que realizan los animales, que es una combinación de los movimientos de onda con el movimiento alterno. Consiste en tres pasos. Sigue un patrón cíclico donde para cada paso, dos de las patas se mueven para dar el siguiente, teniendo en cuenta que no se encuentren en el mismo lado del robot, mientras las otras cuatro permanecen en contacto con el suelo. El orden de los pares de patas en movimiento es indiferente y se puede elegir indistintamente siempre que se cumplan las condiciones citadas. Es un movimiento algo más lento que el tripod gait.

A continuación, se puede ver una representación aclaratoria del orden de las patas en los movimientos que se han explicado anteriormente. Hay que tener en cuenta que L1, L2 y L3 representa las tres patas izquierdas, de adelante hacia atrás, mientras que R1, R2 y R3 son las tres patas derechas en el mismo orden. Cada columna muestra en negro las patas que deben avanzar en cada consecución de movimientos.

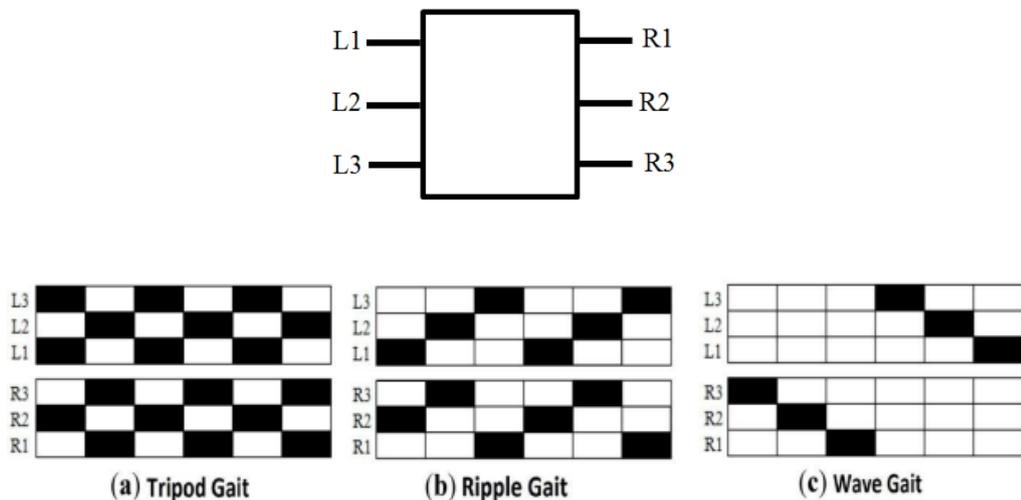


Figura 75. Diagrama de movimientos de un hexápodo. Fuente [39]

Las explicaciones del movimiento se realizarán tomando el Tripod Gait como modelo, debido a que es más sencillo de visualizar y a que ha sido el elegido para el movimiento del robot. El resto de desplazamientos se realizarían de una manera análoga, sólo que utilizando otro orden en las secuencias de movimientos tal y como se indica en el diagrama anterior.

En el caso del robot en desarrollo, la nomenclatura de las patas será distinta a la descrita en el diagrama, debido a su distribución de patas en hexágono. Las

Diseño, desarrollo y construcción de un robot hexápodo para inspección de áreas de difícil acceso.

extremidades se numerarán en orden creciente, comenzando por la extremidad trasera derecha, en sentido contrario a las agujas del reloj como se muestra en el siguiente diagrama.

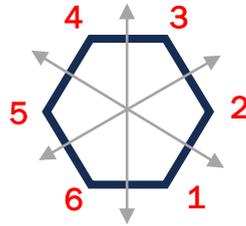


Figura 76. Numeración de extremidades y sentidos de desplazamiento. Elaboración propia

A pesar de su morfología, por necesidades de la programación se tomará uno de los laterales del hexágono como parte frontal (lateral 3-4), pero, sin embargo, el robot podrá avanzar en cualquier otra dirección, siempre perpendicular a dichos laterales del hexágono del cuerpo, como indican las flechas grises.

Antes de entrar de lleno en la explicación del movimiento, cabe recordar que cada extremidad cuenta con tres articulaciones. Se han numerado en orden creciente comenzando por la parte más cercana al núcleo del robot y siguiendo hacia la más alejada. Por lo tanto, la articulación número uno, será la que aporta la rotación a la pata y cuyo servomotor se encuentra entre ambas bases hexagonales. A continuación, se encontraría la articulación número dos, cuya posición de salida se encuentra desfasada 45 grados hacia la vertical, y por último la articulación número tres, que sería la más alejada del robot, y la que finalizaría en el punto de apoyo que contactará con el suelo.

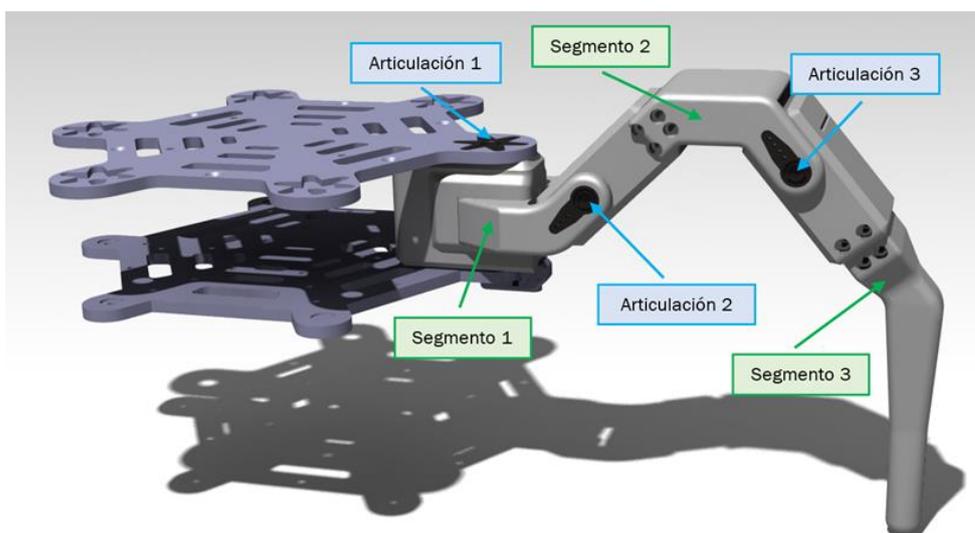


Figura 77. Nomenclatura de los segmentos y articulaciones de la extremidad. Elaboración propia

Diseño, desarrollo y construcción de un robot hexápodo para inspección de áreas de difícil acceso.

En el caso del Tripod Gait, el desplazamiento del robot se realiza moviendo las extremidades en conjuntos de tres, manteniendo siempre las otras tres en el suelo. Los conjuntos serán, por un lado, las patas pares (2, 4 y 6) y por el otro las impares (1, 3 y 5). Cuando un conjunto esté desarrollando una parte del movimiento, el otro debe estar desarrollando el contrario.

En cuanto al movimiento de las extremidades, el diagrama de las secuencias mostrado en la *figura 75*, sólo refleja el avance de las patas. Sin embargo, el desplazamiento está compuesto de dos movimientos bien diferenciados: el avance de las patas y el desplazamiento o empuje del robot. En primer lugar, para el caso del avance, la pata debe adelantarse sin rozar el suelo. Para eso se tendrá que buscar una trayectoria que levante la extremidad, avance y finalmente baje el punto de apoyo.

Una vez concluida esta secuencia, se dispondrá de todas las patas sobre el suelo de nuevo, pero con las recién desplazadas, en una posición adelantada. Pues bien, en este punto es cuando se deben levantar las contrarias para permitir el desplazamiento del robot. Una vez que estén en el aire, las que se habían desplazado hacia adelante previamente, pasarán a desplazarse hacia atrás, pero esta vez manteniéndose siempre en contacto con el suelo. Esto empujará el cuerpo del robot hacia delante, De esta forma, a medida que estas se retrasen, el cuerpo del robot se adelantará.

Dada la configuración en hexágono, las trayectorias que deben realizar las patas para generar un desplazamiento son distintas unas de otras. Sin embargo, se pueden agrupar por parejas. En primer lugar, se encontrarán las patas delanteras (3 y 4), en segundo lugar, las intermedias (2 y 5) y para terminar las traseras (1 y 6). Estas parejas de extremidades llevan entre sí unas trayectorias similares, sólo diferenciadas en el signo del ángulo de la articulación 1. Sin embargo, también diferirá el momento en que se debe mover cada una.

Para la configuración de las trayectorias de cada una de las extremidades, ha facilitado la tarea la utilización de los potenciómetros que actúan de encoder en cada una de las articulaciones. Conectados a través de la placa de multiplexores al ESP32, se ha generado un procedimiento para mostrar por pantalla los ángulos en los que se está posicionando en todo momento. Este proceso se explica en el apartado 6.4.4.

Para ello, con la alimentación de los servomotores apagada, se ha ido moviendo manualmente la pata a las posiciones requeridas, captando los ángulos de cada uno de los tres servomotores de cada extremidad para alcanzar dicha posición. Se han almacenado estos datos y con ellos se han ido generando las trayectorias.

Diseño, desarrollo y construcción de un robot hexápodo para inspección de áreas de difícil acceso.

Ha sido necesario generar estas trayectorias tres veces, una por cada pareja de patas, y con ellas, se han obtenido las restantes invirtiendo los signos de los servomotores de giro (servo 1).

Como se ha podido ver en la explicación anterior, se ha procedido de un modo similar a la toma de puntos de los brazos robóticos industriales, los cuales, en uno de sus modos de configuración, pueden dejar sus articulaciones muertas para que sea el programador el que mueva manualmente el brazo hasta alcanzar el punto deseado. Una vez alcanzado el punto se almacenaría esta posición, para que una vez en funcionamiento, el propio brazo replique estas trayectorias entre puntos. De la misma manera se han ido guardando posiciones, que no son más que conjuntos de tres ángulos que corresponden a los tres servos de cada pata.

De esta forma se realiza la programación por trayectorias, lo cual simplifica en gran medida la tarea.

A continuación, se puede observar un ejemplo de las trayectorias que deberían seguir las patas, en este caso las impares, para realizar un desplazamiento hacia adelante:

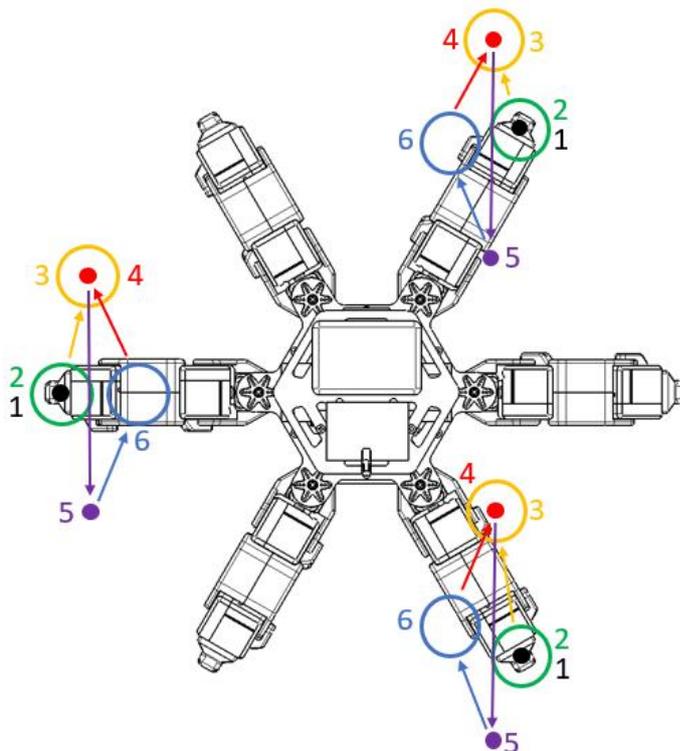


Figura 78. Diagrama de movimientos de las extremidades en el avance. Elaboración propia

Diseño, desarrollo y construcción de un robot hexápodo para inspección de áreas de difícil acceso.

Se ha representado con puntos, las posiciones en las que el robot tiene que estar en contacto con el suelo y circunferencias en las posiciones que tiene que estar levantado.

En primer lugar, partirá de la posición de reposo (1), en la cual todas las patas del hexápodo se encuentran en perfecta simetría. A continuación, se levantará (2) y avanzará hacia adelante (3) para finalmente bajar (4) a un punto adelantado. Mientras las patas pares se levantan, esta realizará el desplazamiento hacia atrás (5) y continuará con una secuencia de levantamiento - avance (6) y bajada (4), para posicionarse de nuevo en el punto de partida de un nuevo desplazamiento hacia atrás. Esto se repetirá indefinidamente mientras reciba la orden de avanzar (puntos 4, 5 y 6).

Se puede observar que, a pesar de ser movimientos muy similares, hay pequeñas variaciones entre los tipos de pata delantera, intermedia y trasera. La delantera adelanta mucho menos el punto de apoyo 4, produciendo a continuación un desplazamiento similar al de la pata intermedia. La trasera, por su parte, adelantará más que las anteriores ese punto 4, para que el punto 5 no quede demasiado alejado del cuerpo, generando más momento.

Para poder realizar el avance en cualquier dirección perpendicular a los lados del hexágono, el factor de forma del hexápodo, que ha complicado el desplazamiento, teniendo de calcular diferentes trayectorias para cada pata, ha facilitado en gran manera este desplazamiento omnidireccional.

Teniendo en cuenta que la configuración motriz del robot es totalmente simétrica, se deduce que al igual que se puede desplazar en dirección perpendicular al segmento 3-4, definido como “adelante”, también se puede elegir otro segmento y avanzar perpendicularmente a él.

Para ello se ha desarrollado una función en el programa, *WrapAround()*, la cual, en función de la posición del joystick izquierdo que se reciba, a la hora de posicionar una pata, se definirá la dirección del movimiento, dividida en seis secciones, y se enviará la configuración a una u otra pata.

6.4.3 Giro del robot

En este caso, la configuración en hexágono también facilita este movimiento. Para realizar una rotación sobre el eje vertical del robot se seguirá el mismo proceso que en el desplazamiento. Por un lado, se deberán levantar conjuntos de tres patas (pares o impares), manteniendo el otro conjunto en contacto con el suelo. En primer lugar, se levantarán las patas impares, se realizará una rotación de estas en dirección contraria al sentido del giro, mientras que las que están en contacto con el suelo, en este caso las pares, rotarán en el mismo sentido que se quiere dar. De esta forma, con las patas en contacto con el

Diseño, desarrollo y construcción de un robot hexápodo para inspección de áreas de difícil acceso.

suelo, se habrá realizado la mitad del giro. Una vez en esta posición, las patas levantadas bajarían hasta tocar de nuevo con el suelo, quedando estas adelantadas y preparadas para realizar la otra mitad del giro.

A continuación, se realizaría la operación contraria, levantando las patas pares, y recuperando la posición de salida, mientras que las impares rotarían también a posición cero, manteniendo el contacto con el suelo. De esta forma se completaría un giro en un sentido o en otro, en función del lado hacia el que esté moviendo el joystick el operador.

Cabe aclarar que, para realizar los movimientos del giro, a la hora de levantar las patas, serán solamente los servomotores número 2 y 3 de cada extremidad los que se moverán. Y para realizar las rotaciones o giros, lo harán tan sólo los servomotores número 1. De esta forma los movimientos son más sencillos que en el caso del avance, debido a que los puntos no son combinaciones de los tres servos de cada pata.

Se ha configurado la rotación de tal forma que por cada vez que se indique al robot que tiene que girar, lo haga en un ángulo fijo de 45 grados, quedando después otra vez en posición cero, dispuesto a realizar un nuevo giro o cualquier desplazamiento.

Finalmente, en el programa principal se ha establecido una prioridad en los movimientos, siendo mayor la de los desplazamientos y dejando estas rotaciones en segundo plano. Se ha establecido de esta forma, dado que los desplazamientos se pueden realizar en cualquier dirección, por lo que no es imprescindible realizar una rotación para seguir avanzando independientemente de los obstáculos del terreno.

La rotación sólo se podrá llevar a cabo si no se está moviendo el joystick izquierdo de desplazamiento.

6.4.4 Integración de las señales de feedback de las extremidades

Como se ha citado en apartados anteriores, estas señales de realimentación de los potenciómetros han sido de vital importancia a la hora de programar las trayectorias de las patas del robot. De esta forma, dejando los servomotores sin alimentación, se puede mover la extremidad manualmente y por medio del IDE y el Monitor Serial, el cual muestra los mensajes que se mandan imprimir desde el programa, se visualiza los distintos valores que van tomando los potenciómetros.

Se ha realizado una función para la calibración de dichos potenciómetros, debido a que las características eléctricas de cada uno no son completamente idénticas al resto. Se pueden apreciar diferencias entre las resistencias y las posiciones de cada uno de ellos. Por ello, el procedimiento de calibración, va

Diseño, desarrollo y construcción de un robot hexápodo para inspección de áreas de difícil acceso.

moviendo los servomotores de las patas, de una en una, colocándolas en lo que se considera el límite del movimiento, en este caso unos 80 grados hacia cada lado de la posición cero. De esta forma se reduce en 20 grados el alcance, pero se evitan las zonas límite tanto de los servomotores como de los potenciómetros que podrían acarrear problemas de lectura.

Para ello se almacenaría en primer lugar la lectura de los tres potenciómetros de una pata en posición cero, para a continuación comenzar una secuencia recorriendo todos los puntos “máximos” de los servomotores y almacenando sus puntos.

Una vez almacenados los puntos, se podrán utilizar estos máximos para, a través de una regla de tres, o lo que es lo mismo, la función *map()*, convertir estas lecturas analógicas en un número entero que representará el ángulo entre -80 y 80 grados.

Esta función es la utilizada en la función *lee_canal()*, la cual envía a la placa de multiplexores los bits de selección del canal que se quieren leer, lee el pin de entrada analógica los datos y devuelve este valor analógico de 12 bits, o lo que es lo mismo, de 0 a 4095 en una escala con un rango entre -90 y +90 grados, a pesar de que realmente nunca vaya a alcanzar estos límites.

Esos valores son los enviados impresos a través del puerto Serial, el cual representa físicamente el cable USB y se visualizará a través del Monitor Serial del Arduino IDE.

Se valoran nuevas funcionalidades para estas señales de realimentación, que serán citadas en el apartado [8.2](#) de mejoras o líneas futuras.

Diseño, desarrollo y construcción de un robot hexápodo para inspección de áreas de difícil acceso.

7. Pruebas y validación

A continuación, se hará una breve evaluación del movimiento, estabilidad, así como las funciones del robot. Cabe destacar que estas evaluaciones, han sido un ciclo que se ha ido realizando a medida que se iba realizando el desarrollo y la programación del robot, siendo un proceso largo y minucioso. A continuación, se muestran las evaluaciones finales.

7.1 Evaluación de la movilidad del robot

En primer lugar, se ha testado el movimiento del robot; tanto su capacidad para desplazarse, para rotar, como para mantenerse inmóvil. A pesar de la dificultad de la coordinación en todos los movimientos en cuanto al desplazamiento, uno de los mayores problemas que han surgido en su estabilidad ha sido una vibración en los servos de las articulaciones 1 principalmente, que muchas veces se trasladaba a otras articulaciones y recordaba a un control de posición mal ajustado, ya que se generaba una inestabilidad que no era capaz el propio motor de detener.

Se hicieron infinidad de pruebas en cuanto a interferencias y señales por si fueran las causantes de esta vibración. Sin embargo, finalmente se llegó a la conclusión de que era un problema de inercias de los propios servomotores. Al estar colgando toda la pata de ese servomotor, contaba con una gran inercia. Pues bien, al realizar el servomotor algún pequeño ajuste de ángulo de manera automática, esta inercia hacía que se desplazara más de lo debido, lo cual debía corregir hacia el lado contrario, etc. De esta forma se desestabilizaba y generaba estas oscilaciones en cadena.

Se llegó a la conclusión de que el problema se podría solucionar añadiendo una ligera fricción en la articulación. Se había buscado que se redujera tanto su fricción, que eso generaba el propio problema. Por ello se colocaron una serie de bridas alrededor del cuerpo del robot, entre la base inferior y la intermedia, de tal forma que las acercaran unas décimas de milímetro, generando más fricción entre las piezas. La prueba fue todo un éxito y no sólo se acabó con las oscilaciones en esas articulaciones sino en el resto también. Para una solución más definitiva se diseñaron y fabricaron una serie de tensores para sustituir las bridas con tornillos y tuercas.

Una vez solucionada esta cuestión, el reposo no presenta mayor dificultad.

En cuanto al movimiento, se han realizado diferentes ajustes en cuanto a los puntos de las trayectorias de avance para lograr un avance recto, relativamente rápido, pero sin riesgo de inestabilidad.

En cuanto a su estabilidad de la base a la hora de desplazarse no es perfecta, pero es correcta. Podría ser mejorable a través de un control de movimiento

Diseño, desarrollo y construcción de un robot hexápodo para inspección de áreas de difícil acceso.

más complejo que implementara movimientos combinados en cada pata con control de velocidad en cada articulación, pero para ello se debería utilizar un controlador mucho más complejo con gran capacidad de computación y alguna aplicación como Matlab para poder ir realizando los ajustes en tiempo real.

Otro ajuste que ha sido necesario, ha sido identificado a la hora de testar el robot sobre superficie irregular, como tierra o hierba. Ha sido notable que la altura necesaria para desplazar las patas sin rozamiento contra la superficie debía ser mayor, dado que sobre una superficie regular no hay nada que sobresalga de dicha superficie, cosa que sí sucede en las irregulares.

Actualmente se desplaza como se definió en los objetivos, en cualquier dirección perpendicular a los laterales del hexágono del cuerpo, pudiendo pausar el movimiento para continuarlo sin problema. También puede cambiar de dirección de avance, para lo cual pasará de forma automática por la posición inicial para comenzar el nuevo desplazamiento. Y finalmente puede rotar sobre su centro en cualquier dirección en giros enteros de 45 grados.

7.2 Verificación de la programación

La programación se ha verificado principalmente realizando pruebas de movimiento, realizando todas las posibles combinaciones de controles desde el mando, sobre superficie regular e irregular, y comprobando que se llevan a cabo satisfactoriamente.

7.2.1 Verificación del control a través del mando

Prioridad del avance sobre la rotación

Se ha comprobado satisfactoriamente que en todo momento el Joystick izquierdo tendrá prioridad sobre el derecho, ignorando el robot las señales que se le envíen desde el control de giro, siempre y cuando el desplazamiento esté teniendo lugar. Una vez concluido el desplazamiento rotará sin problema.

Reducción y aumento de la velocidad

Se ha evaluado el aumento y reducción de la velocidad con los pulsadores R1 y L1 del control y efectivamente se realiza sin ningún problema, dentro del rango de seguridad programado.

Zonas muertas de los Joystick

Se ha establecido una zona “muerta” del área del joystick que impide que cualquier pequeño movimiento involuntario de los dedos acabe generando un movimiento en el robot. Para ello se ha establecido esta zona que ha funcionado correctamente. Inicialmente esta área era menor, pero se tuvo que aumentar debido a que al soltar el joystick y este volver a su posición central,

Diseño, desarrollo y construcción de un robot hexápodo para inspección de áreas de difícil acceso.

había veces que rebotaba ligeramente, recibiendo el controlador una señal de avanzar en la dirección contraria. Era momentáneo y por ello no comenzaba a avanzar, pero sí que llegaba a colocarse el robot en la posición inicial para ello. Por ello se aumentó y este error desapareció.

Diseño, desarrollo y construcción de un robot hexápodo para inspección de áreas de difícil acceso.

Diseño, desarrollo y construcción de un robot hexápodo para inspección de áreas de difícil acceso.

8. Conclusiones

8.1 Cumplimiento de objetivos

A continuación, se revisará el listado de los principales objetivos del proyecto del apartado 1.2 de la presente memoria y se evaluará el grado de cumplimiento logrado con ellos:

Diseño y fabricación

En cuanto al diseño y la fabricación se buscaba crear las estructuras necesarias para asegurar la movilidad y estabilidad del robot, optimizándolas para operar en terrenos complejos.

Se da por cumplido el objetivo, habiendo desarrollado una estructura sólida, pero al mismo tiempo ligera, a través de fabricación aditiva, tecnología que permite generar sólidos con diferentes densidades internas creando cuerpos aparentemente continuos con un interior aligerado. Esta estructura es capaz de albergar todos los componentes mecánicos y electrónicos necesarios y sus patas son ágiles y esbeltas, permitiendo al robot penetrar a zonas de difícil acceso.

Integración y configuración de componentes electrónicos

El objetivo era seleccionar y configurar satisfactoriamente todos los sistemas electrónicos para lograr un control estable.

También se da por cumplido. Se ha integrado gran número de componentes electrónicos comerciales y donde ha sido necesario se ha desarrollado hardware electrónico propio para conectar todos los sensores y actuadores del robot. El resultado ha sido una integración limpia, a pesar del gran número de conexiones que comunican los sensores y actuadores de las patas con los distintos sistemas de control. Además, la comunicación entre los diferentes circuitos electrónicos se realiza de forma precisa, habiendo llegado a generar diferentes planos para la colocación sin interferencias de los diferentes componentes electrónicos y utilizar cable apantallado para el transporte de la señal de las comunicaciones.

Desarrollo del software de control

Se buscaba realizar el programa de control, el cual recibiera y procesara las señales del mando y enviara las señales de vídeo, así como permitiera al robot desplazarse.

El objetivo se ha alcanzado debido a que la comunicación entre el mando y el microcontrolador es perfecta. También se ha desarrollado todo el sistema de

Diseño, desarrollo y construcción de un robot hexápodo para inspección de áreas de difícil acceso.

movimiento que permite al robot desplazarse y rotar. Por último, se ha dotado al robot de un sistema de visión en primera persona a través de la cámara integrada en el robot y la pantalla de visualización colocada sobre el mando.

Pruebas y validación del sistema

Se buscaba realizar las pruebas necesarias para asegurar que el robot cumple con los requisitos planteados en el proyecto.

Se considera que las pruebas han sido satisfactorias desde el punto de vista del desplazamiento y rotación del robot, envío de señal de vídeo y control remoto. Sin embargo, al ser un proyecto tan complejo, ha quedado mucho margen de mejora en varios aspectos que se evaluarán en el apartado siguiente. Sin embargo, en general el funcionamiento ha sido satisfactorio.

Objetivo final

Como objetivo final se buscaba desarrollar un hexápodo robótico, robusto, con componentes fácilmente replicables, que fuera capaz de desplazarse por terrenos complejos y cuyo desarrollo sentara las bases para futuros desarrollos y mejoras en el campo de la robótica móvil.

Para concluir, se considera el objetivo cumplido, habiendo logrado un robot plenamente móvil, con buena comunicación con el control y transmisión de vídeo y con un presupuesto muy reducido. Sin embargo, el actual hardware y software no permite al robot moverse por terrenos con grandes desniveles, tales como escalones. Para ello habría sido necesario implementar sensores de contacto en las patas que haría más complejo aún más tanto hardware como software. Sin embargo, el robot ha quedado preparado para esta mejora, incluyendo puertos extra en las placas concentradoras de señales que han quedado disponibles para su futuro desarrollo. Además, el microcontrolador ESP32 cuenta con infinidad de puertos y funcionalidades no utilizadas, por lo que sería factible y relativamente sencillo utilizar este desarrollo como base móvil para incluir gran número de sensores y actuadores que pudieran llevar a nuevos proyectos y desarrollos en el futuro.

8.2 Posibles mejoras y líneas futuras

Como ya se ha indicado anteriormente, es un proyecto de gran envergadura, que además propicia la inclusión de nuevos sensores y actuadores para añadir funcionalidades, las cuales han quedado preparadas para ser añadidas. A continuación, se realiza una breve descripción de algunas de las modificaciones que se le podrían hacer al robot para mejorar sus características o funcionalidades.

Diseño, desarrollo y construcción de un robot hexápodo para inspección de áreas de difícil acceso.

Suplemento en las patas con mayor superficie para terrenos arenosos

Se considera diseñar y fabricar un suplemento para colocar sobre los puntos de apoyo de las patas con el fin de poder caminar sobre superficies arenosas, en las que las patas actuales se hundirían clavándose probablemente sobre el terreno.

Sensores de presión en las terminaciones de las articulaciones

Tal y como se ha descrito en apartados anteriores, se ha planificado la inclusión de algún tipo de sensor de presión, el cual cuando apoye la pata en el suelo, mande una señal para indicar que ha llegado a la superficie, y así interrumpir en esa posición el movimiento de la extremidad, originando una nueva superficie de contacto. De esta forma se podría modificar la programación para que el robot sea capaz de pasar por encima de terreno completamente irregular o con escalones. La placa concentradora de sensores donde están conectados todos los potenciómetros ha quedado preparada para incluirlos.

Cámara con visión artificial para control autónomo o reconocimiento de caras u objetos

En el caso de las mejoras con la cámara las posibilidades se vuelven infinitas, teniendo en cuenta el gran número de cámaras con aplicaciones que hay actualmente en el mercado. Además, se podría desarrollar software propio para realizar una navegación autónoma o añadir el reconocimiento de caras u objetos para los rescates. También se puede añadir funcionalidad de visión nocturna para espacios o trabajos con poca luminosidad.

Servomotores para mover la cámara

Se plantea también o bien colocar un servomotor, con un radio de acción de 360 grados, o bien un pan-tilt, que haría que la cámara pudiera apuntar en cualquier dirección, tanto rotando como cabeceando. Sería posible controlarlo desde el mismo controlador, sustituyendo alguno de los controles, como por ejemplo el giro.

Encoder en las articulaciones

Otra mejora que se propone es la de sustituir los potenciómetros de las extremidades por encoder rotativos para mejorar la precisión y resolución. De esta forma se podrían incluir unos sensores de pequeño tamaño, quedando integrados de forma similar a los potenciómetros y mejorando su funcionalidad.

Diseño, desarrollo y construcción de un robot hexápodo para inspección de áreas de difícil acceso.

Almacenamiento externo para configuraciones

Se plantea también que las calibraciones, bien de potenciómetros o bien de encoders, puedan ser almacenadas en un archivo de texto. Para ello sería necesario un circuito electrónico muy accesible que sería un lector de tarjetas MicroSD o SD. Estos lectores se conectan y comunican directamente con el ESP32, teniendo pines dedicados a dicha función, por lo que sería una mejora bastante sencilla de llevar a cabo. Una vez realizada la calibración, se podría grabar la matriz de calibración en el fichero de texto, para que cada vez que se encienda el robot, se pueda leer sin tener que volver a calibrar cada articulación.

Control desde aplicación móvil

Una mejora que tampoco tendría mucha dificultad sería la de conectar el teléfono con el microcontrolador aprovechando la antena bluetooth o wifi, utilizando alguna app de control para manejarlo, sin tener que recurrir al mando de control remoto.

Inclusión de sensores

También se propone la inclusión de otro tipo de sensores, dependiendo de la aplicación, como por ejemplo detectores de metales en el caso de querer rastrear minas, sensores de ultrasonidos para detección de supervivientes o incluso un altavoz para poder reproducir mensajes de audio para que los puedan oír los posibles supervivientes.

Fabricación con tecnología SLS

Finalmente se plantea una mejora en cuanto a la resistencia de los componentes que sería cambiar la tecnología dentro de la fabricación aditiva por SLS o sinterizado láser selectivo, tecnología que permite realizar piezas en diferentes poliamidas, con una isotropía mucho menor y mayor resistencia en sus componentes. Se incrementaría ligeramente el peso del hexápodo, así como su coste, pero con unos motores algo más potentes, la mejora sería sustancial.

Tal y como se ha explicado al comienzo de este apartado las posibilidades que abre el proyecto son infinitas, por lo que tan sólo quedará limitado por la imaginación de la persona que decida trabajar sobre ello.

9. Referencias y fuentes

9.1 Referencias

- [1] “BigDog, the Rough-Terrain Quadruped Robot”. Accedido el 12 de febrero de 2024. [En línea]. Disponible: <https://www.sciencedirect.com/science/article/pii/S1474667016407020>
- [2] J. O. Pérez. “SPIDAR, el robot anfibio aeroterrestre de la Universidad de Tokio”. WIRED. Accedido el 29 de febrero de 2024. [En línea]. Disponible: <https://es.wired.com/articulos/spidar-el-robot-anfibio-aeroterrestre-de-la-universidad-de-tokio>
- [3] “T8X — Robugtix”. Robugtix. Accedido el 10 de marzo de 2024. [En línea]. Disponible: <https://www.robugtix.com/t8x>
- [4] “Robot HEXA de Vincross”. Vincross. Accedido el 5 de marzo de 2024. [En línea]. Disponible: <https://www.vincross.com/>
- [5] “Summary of the RHex robot platform”. Summary of the RHex robot platform. Accedido el 17 de febrero de 2024. [En línea]. Disponible: <https://www.rhex.web.tr/>
- [6] “Trossen Robotics PhantomX AX Metal Hexapod MK III — TECHNOLOGY x2”. TECHNOLOGY x2. Accedido el 15 de marzo de 2024. [En línea]. Disponible: https://www.technologyx2.com/proj_robot_hexapod/2019/11/9/trossen-robotics-phantomx-ax-metal-hexapod-mk-iii
- [7] “GitHub - KurtE/Phantom_Phoenix: Lynxmotion Phoenix code ported to Trossen PhantomX”. GitHub. Accedido el 24 de febrero de 2024. [En línea]. Disponible: https://github.com/KurtE/Phantom_Phoenix
- [8] “SG90 datasheet(pdf)”. ALLDATASHEET.ES - Sitio de Búsqueda de Datasheet, Sitio de Búsqueda de Datasheet de Componentes Electrónicos y Semiconductores y otros semiconductores. Accedido el 25 de abril de 2024. [En línea]. Disponible: <https://www.alldatasheet.es/datasheet-pdf/pdf/1572383/ETC/SG90.html>
- [9] “15DS3218”. Accedido el 2 de mayo de 2024. [En línea]. Disponible: <https://aliseoperations.com/shop/rc-hobby/servomotors/dsservo-ds3218mg-20kg-waterproof-digital-servo-270/>
- [10] “MG995 datasheet, PDF”. ALLDATASHEET.COM - Electronic Parts Datasheet Search. Accedido el 18 de abril de 2024. [En línea]. Disponible: <https://www.alldatasheet.com/view.jsp?Searchword=Mg995%20datasheet&>

Diseño, desarrollo y construcción de un robot hexápodo para inspección de áreas de difícil acceso.

[amp;gad_source=1&gclid=CjwKCAjw9eO3BhBNEiwAoc0-jQIgiHALPg4tYZYspIV-KdszKxEsBUuRwvyjH0ar7soG3xt0Qxi-phoCcfsQAvD_BwE](https://store.arduino.cc/en-es/products/arduino-mega-2560-rev3?gad_source=1&gclid=CjwKCAjw9eO3BhBNEiwAoc0-jQIgiHALPg4tYZYspIV-KdszKxEsBUuRwvyjH0ar7soG3xt0Qxi-phoCcfsQAvD_BwE)

[11] “Arduino Mega 2560 Rev3”. Arduino Official Store. Accedido el 20 de marzo de 2024. [En línea]. Disponible: https://store.arduino.cc/en-es/products/arduino-mega-2560-rev3?gad_source=1&gclid=CjwKCAjw9eO3BhBNEiwAoc0-jdGMcfakUeVyvqwpYpGUI_ztSCUmeVDI6AY_frwx2sPgnpecyA2ArhoCjzMQAvD_BwE

[12] “Raspberry Pi 4”. RASPBERRYPI.COM. Accedido el 27 de marzo de 2024. [En línea]. Disponible: <https://www.raspberrypi.com/products/raspberry-pi-4-model-b/>

[13] “ESP32 Wifi + Bluetooth en un solo lugar”. Programarfácil Arduino y Home Assistant. Accedido el 3 de abril de 2024. [En línea]. Disponible: <https://programarfácil.com/esp8266/esp32/>

[14] “STM32 32-bit Arm Cortex MCUs - STMicroelectronics”. STMicroelectronics. Accedido el 10 de abril de 2024. [En línea]. Disponible: <https://www.st.com/en/microcontrollers-microprocessors/stm32-32-bit-arm-cortex-mcus.html>

[15] “Turnigy 5000mAh 3S 20C LiPo Paquete con XT-60”. Hobbyking. Accedido el 17 de mayo de 2024. [En línea]. Disponible: https://hobbyking.com/es_es/turnigy-5000mah-3s-20c-lipo-pack-xt-90.html?srsId=AfmBOoqbEQ1awr77SNMB_f6WC25TnlWTDyWQJn3bRML7fKB8Y00X5eGX

[16] “ZIPPY Flightmax 5000mAh 3S1P 25C”. Hobbyking. Accedido el 25 de mayo de 2024. [En línea]. Disponible: https://hobbyking.com/en_us/zippy-flightmax-5000mah-3s1p-25c.html?srsId=AfmBOopHLyuBUuw_dN8Mj0RSfdpWR5LxaEF908RUN4SqSEG_1bMjgWWWh

[17] “Lipo”. EliteHobby. Accedido el 3 de junio de 2024. [En línea]. Disponible: <https://elitehobby.eu/es/baterias/1309-lipo-gens-ace-bashing-series-5500mah-111v-50c-3s.html>

[18] “TAA37004S45X6”. Aeromodelismo Serpa. Accedido el 10 de junio de 2024. [En línea]. Disponible: <https://aeromodelismoserpa.es/tattu/17329-tattu-3700mah-148v-45c-4s1p-lipo-battery-pack-with-xt60.html>

[19] “Convertidores Buck/Boost – Step up/Step Down”. Accedido el 9 de mayo de 2024. [En línea]. Disponible: <https://solectroshop.com/es/blog/convertidores-buckboost-step-upstep-down-n82>

Diseño, desarrollo y construcción de un robot hexápodo para inspección de áreas de difícil acceso.

[20] “Manual Catia v5 | PDF”. Scribd. Accedido el 25 de julio de 2024. [En línea]. Disponible: <https://es.scribd.com/document/770057671/Manual-Catia-v5>

[21] “GrabCAD making additive manufacturing at scale possible”. GrabCAD Making Additive Manufacturing at Scale Possible. Accedido el 25 de junio de 2024. [En línea]. Disponible: <https://grabcad.com/>

[22] “Thingiverse - Digital Designs for Physical Objects”. Thingiverse. Accedido el 2 de julio de 2024. [En línea]. Disponible: <https://www.thingiverse.com/>

[23] “Guía técnica de impresión 3d”. E3dplusvet. Accedido el 9 de julio de 2024. [En línea]. Disponible: <https://www.e3dplusvet.eu/wp-content/docs/01A1-ES-RES.pdf>

[24] “Plástico PLA: qué es y para qué se utiliza | IBP Uniuoso”. Uniuoso. Accedido el 17 de julio de 2024. [En línea]. Disponible: https://www.uniuoso.com/blog/plastico-pla-que-es-y-para-que-se-utiliza/#Principales_caracteristicas_del_plastico_PLA

[25] “Datasheet Minimaestro controller”. POLOLU. Accedido el 3 de agosto de 2024. [En línea]. Disponible: <https://www.pololu.com/docs/pdf/OJ40/maestro.pdf>

[26] “GitHub - jvpernis/esp32-ps3: Control your ESP32 projects with a PS3 controller!” GitHub. Accedido el 10 de agosto de 2024. [En línea]. Disponible: <https://github.com/jvpernis/esp32-ps3>

[27] CORE Scholar - Wright State University. Accedido el 18 de junio de 2024. [En línea]. Disponible: https://corescholar.libraries.wright.edu/cgi/viewcontent.cgi?article=3006&context=etd_all

[28] “Spider Robot”. University of Evansville. Accedido el 18 de agosto de 2024. [En línea]. Disponible: <https://www.evansville.edu/majors/computerengineering/downloads/projects-2019/spider-robot-report.pdf>

9.2 Referencias de imágenes

Referencias de imágenes

Fuente [1]: <https://robotsguide.com/robots/bigdog/>

Fuente [2]: <https://bostondynamics.com/products/spot/>

Fuente [3]: <https://interestingengineering.com/innovation/spider-like-robot-ready-to-change-robotics>

Diseño, desarrollo y construcción de un robot hexápodo para inspección de áreas de difícil acceso.

Fuente [4]: <https://www.robugtix.com/t8x>

Fuente [5]: <https://www.vincross.com/collections/hexa/products/hexa>

Fuente [6]: <https://kodlab.seas.upenn.edu/past-work/rhex/research-rhex/>

Fuente [7]: <https://www.interbotix.com/Robotic-Hexapod>

Fuente [8]: <https://laughingsquid.com/3d-printed-hexapod-robot/>

Fuente [9]: https://tienda.bricogeek.com/3972-thickbox_default/micro-servo-miniatura-sg90.jpg

Fuente [10]: <https://www.tiendatec.es/maker-zone/servos/2110-dsservos3218-servo-digital-20kg-180-8472496026123.html>

Fuente [11]: <https://www.tiendatec.es/electronica/servos/585-servomotor-digital-mg995-8405851490017.html>

Fuente [12]: https://vishaworld.com/products/mg996r-digital-metal-gear-high-torque-servo-motor-180-degree-rotation?_pos=1&_sid=769ddc751&_ss=r

Fuente [13]: <https://tienda.bricogeek.com/arduino/306-arduino-mega-2560.html>

Fuente [14]: <https://www.circuitos-electricos.com/arduino-due/>

Fuente [15]: <https://raspi.tv/2019/raspberry-pi-4-launches-with-bcm2711-quad-core-cortex-a72-64-bit-soc-running-at-1-5ghz-with-dual-4k-display-capability>

Fuente [16]: <https://www.aranacorp.com/fr/product/nodemcu-esp32/>

Fuente [17]: <https://in.element14.com/stmicroelectronics/nucleo-f070rb/stm32-onboard-st-link-v2-1-dev/dp/2467272>

Fuente [18]: <https://www.multanelectronics.com/product/adafruit-pca9685-12-bit-pwm-servo-driver/>

Fuente [19]: https://wiki.lynxmotion.com/info/wiki/lynxmotion/download/servo-erector-set-system/ses-electronics/ses-modules/ssc-32u/WebHome/lynxmotion_ssc-32u_usb_user_guide.pdf

Fuente [20]: <https://www.pololu.com/product/1354>

Fuente [21]: <https://www.laserse.com/p/rm063-multiturn-potentiometer-103-102-101-104-202-100r-200r-500r-1k-2k-3k-5k-10k-20k-50k-100k-200k-500k-1m-adjustable-resistors/>

Diseño, desarrollo y construcción de un robot hexápodo para inspección de áreas de difícil acceso.

Fuente [22]: https://hobbyking.com/en_us/turnigy-5000mah-3s-20c-lipo-pack-xt-90.html

Fuente [23]: https://hobbyking.com/es_es/zippy-flightmax-5000mah-3s1p-25c.html?store=es_es

Fuente [24]: <https://www.peckamodel.cz/b-50c-5000-3s1p-trx-s-p-gens-ace-lipo-3s-5000-mah-50c-3s1p-short-size-lipo-s-xt-60>

Fuente [25]: <https://www.professional-multirotors.com/product/tattu-4500mah-22-2v-25c-lipo-pack/>

Fuente [26]: <https://hallmiba.com/p/fritid/mobiltillbehor/laddare-kablar/powerbank-5000-mah.html>

Fuente [27]: <https://cosmos.bluesoft.com.br/produtos/7382499719629-c4002-5-8g-fpv-monitor-720p-camera-set-for-jjrc-h16-yizhan-tarantula-x6-rc-model-remote-control-helicopter-drone-spare-parts>

Fuente [28]: https://www.amazon.es/dp/B09Z27VVRL?ref=ppx_yo2ov_dt_b_fed_asin_title

Fuente [29]: https://cdn.shopify.com/s/files/1/1509/1638/files/XL-4016E1_Datasheet.pdf?v=1692779019

Fuente [30]: <https://rambal.com/impresoras-accesorios/1933-ventilador-brushless-8010-12v.html>

Fuente [31]: <https://gamesandgears.pk/product/sony-playstation-3-wireless-controller-black/>

Fuente [32]: <https://www.manufacturingguide.com/sv/fused-deposition-modeling-fdm>

Fuente [33]: <https://www.sunlu.com/es-es/products/3kg-pla-plus-matte-pla-filament-bundle-buying-1kg-roll>

Fuente [34]: <https://www.crealityofficial.co.uk/products/creality3d-ender-3-max-neo-3d-printer-official-store>

Fuente [35]: <https://www.3dsculplab.xyz/Blog/post/What-are-support-structures-in-3D-printing>

Fuente [36]: <https://www.pololu.com/docs/pdf/OJ40/maestro.pdf>

Fuente [37]: https://a.aliexpress.com/_EJphoD3

Fuente [38]: <https://www.tecnosalva.com/control-de-un-servomotor-con-arduino/>

Diseño, desarrollo y construcción de un robot hexápodo para inspección de áreas de difícil acceso.

Fuente [39]:

https://corescholar.libraries.wright.edu/cgi/viewcontent.cgi?article=3006&context=etd_all

Diseño, desarrollo y construcción de un robot hexápodo para inspección de áreas de difícil acceso.

10. Anexos

10.1 Diagramas y esquemas técnicos

10.1.1 Diagrama de conexiones

Circuito concentrador de señales a través de multiplexores

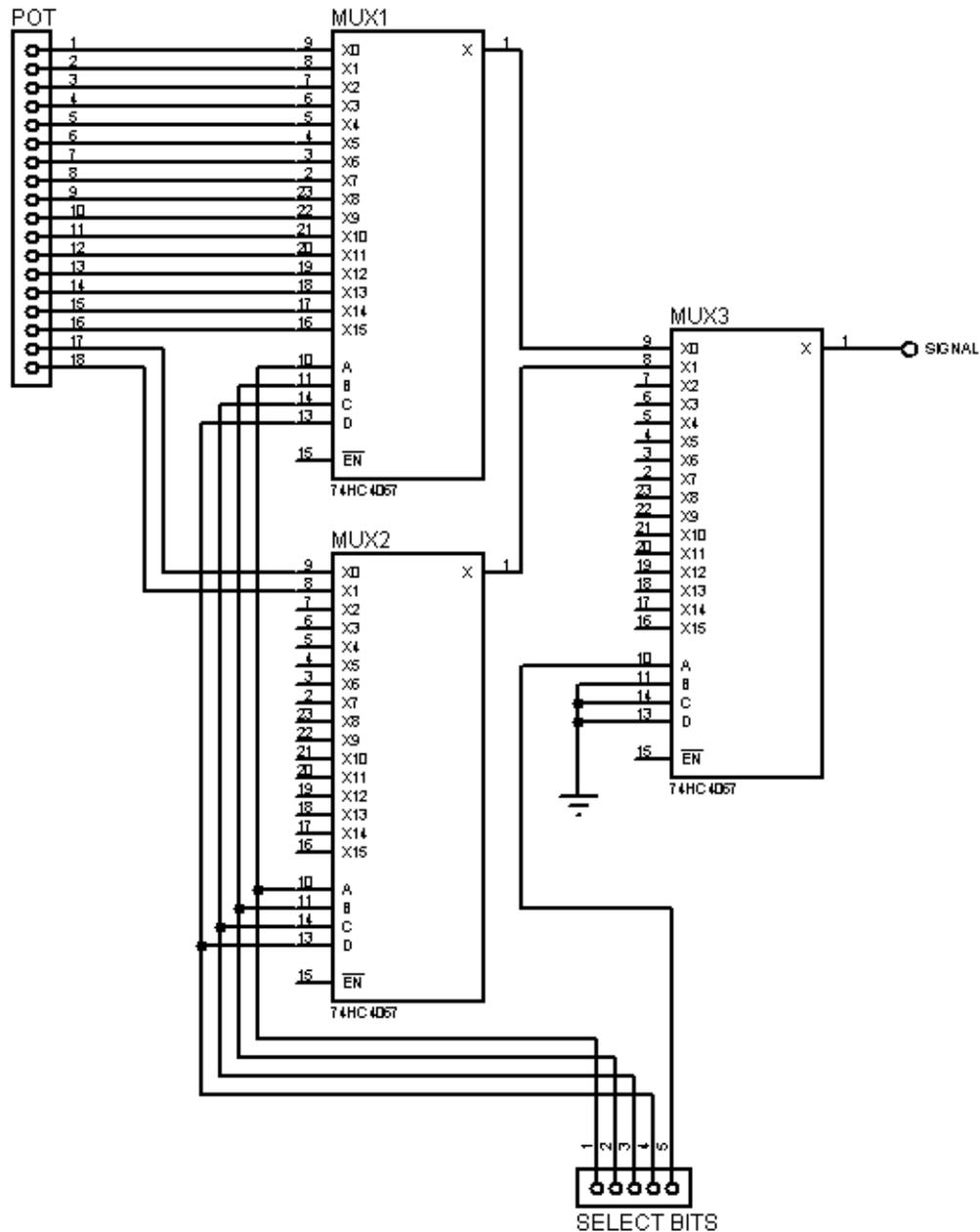


Figura 79. Circuito concentrador de señales. Elaboración propia

Diseño, desarrollo y construcción de un robot hexápodo para inspección de áreas de difícil acceso.

Circuito de alimentación y control

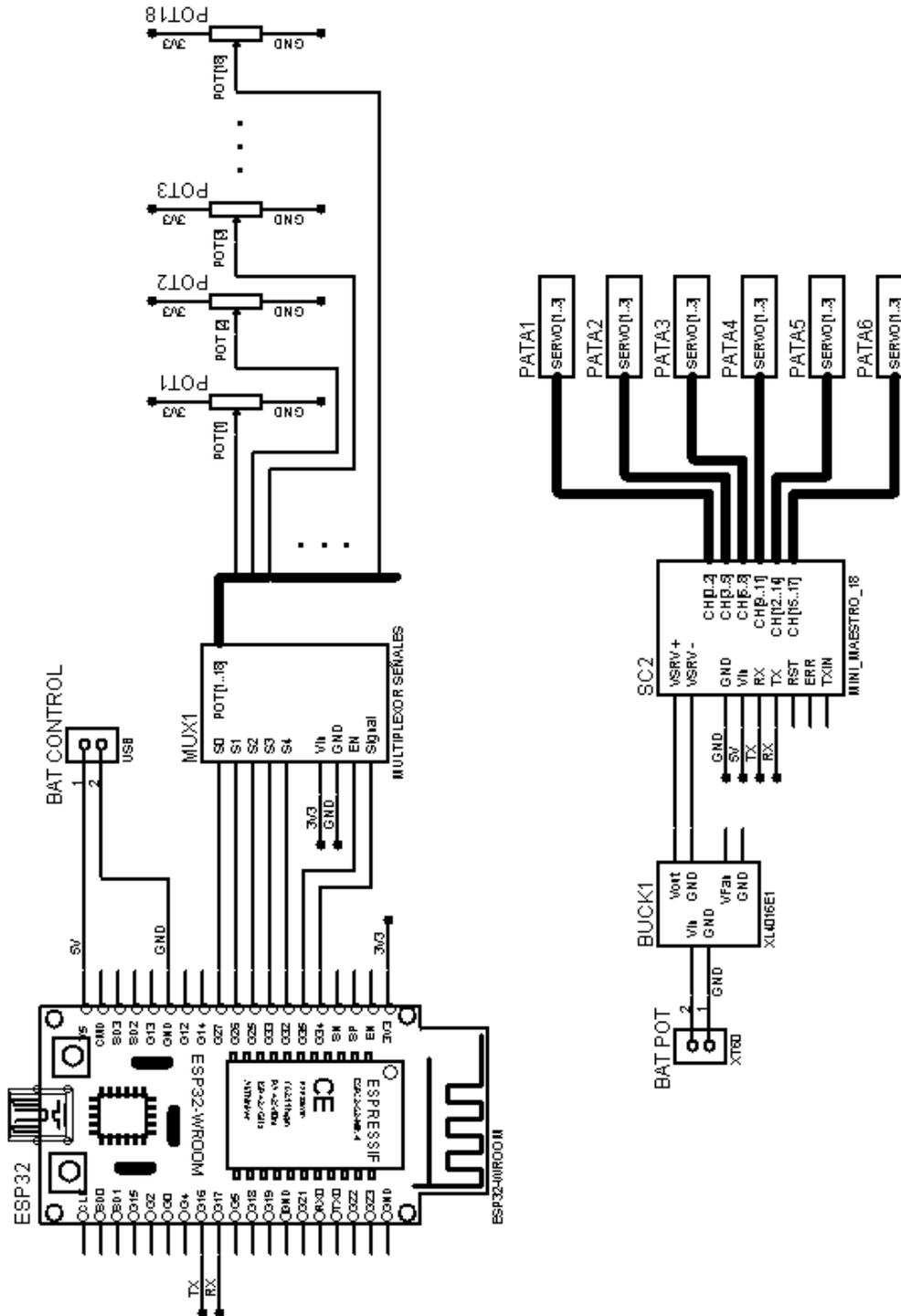


Figura 80. Circuito de alimentación y control

Diseño, desarrollo y construcción de un robot hexápodo para inspección de áreas de difícil acceso.

10.2 Fotos y vídeos del funcionamiento

Se adjuntan en un fichero ZIP junto al documento y se proporciona a continuación un enlace para acceder a ellos y poder descargarlos:

https://drive.google.com/drive/folders/1wYKr1ulKa9Hxnk02IV_z2n5eF53NjFZu?usp=drive_link



Diseño, desarrollo y construcción de un robot hexápodo para inspección de áreas de difícil acceso.

10.3 Código fuente relevante

Se adjuntan los programas a continuación. Además, se podrán encontrar en formato “ino” en un fichero ZIP junto al documento.

10.3.1 Código de Arduino

Programa principal

```
1  #include <PololuMaestro.h>
2  #include <Ps3Controller.h>
3  //----- Controladora Servos -----
4  #define micros_min 10048
5  #define micros_max 2048
6  #define RXD2 16
7  #define TXD2 17
8  #define NUMSERVOS 18
9  #define ACC 30
10 #define VEL 80
11 //----- Multiplexor -----
12 #define pin_lectura 34
13 #define enable_mux 35
14 #define s0_mux 27
15 #define s1_mux 26
16 #define s2_mux 25
17 #define s3_mux 33
18 #define s4_mux 32
19
20 #define zona_muerta 80
21
22 int select_mux[5];
23 uint16_t posiciones[NUMSERVOS];
24 uint16_t pos_pata[3];
25
26 int seccion = 0, seccion_ant = 0;
27
28 int delays = 300;
29 int player = 0;
30 int battery = 0;
31 //----- Banderas -----
32 bool marcha = false;
33 bool estoy_en_cero = false;
34 bool vete_a_cero = false;
35 bool gira_der = false;
36 bool gira_izq = false;
37 bool primer = true;
38
39 MiniMaestro driver(Serial2);
40
41 void setup() {
42     Serial.begin(9600); //Iniciación serial
43     //comunicación con PC
44     Serial2.begin(115200, SERIAL_8N1, RXD2, TXD2); //Iniciación serial
45     //Controladora Servos
46     pinMode(s0_mux, OUTPUT);
47     pinMode(s1_mux, OUTPUT);
48     pinMode(s2_mux, OUTPUT);
49     pinMode(s3_mux, OUTPUT);
50     pinMode(s4_mux, OUTPUT);
51     Ps3.attach(notify);
52     Ps3.attachOnConnect(onConnect);
53     Ps3.begin("00:26:Sc:11:bd:30"); //Conecta el mando con la MAC entre comillas
54     Serial.println("Ready.");
55
56     inicializa_servos();
57     estoy_en_cero = true;
58     player = 4;
59 }
60
61 void loop() {
62
63     if (!Ps3.isConnected())
64         return;
65
66     if (marcha) { //si marcha está activo (prioritario)
67         Serial.println("Avanza tripod");
68         avanza_tripod(delays);
69     } else if (gira_der) { // si no está activo marcha y si está gira_der
70         Serial.println("Gira Derecha");
71         gira_derecha(delays);
72     }
```

Diseño, desarrollo y construcción de un robot hexápodo para inspección de áreas de difícil acceso.

```
72     } else if (gira_izq) { // si no están los anteriores y si está gira_izq
73       Serial.println("Gira Izquierda");
74       gira_izquierda(delays);
75     }
76
77     if (vete_a_cero) { //si está activa la bandera vete_a_cero
78       Serial.println("Ir a pos cero");
79       ir_a_pos_cero();
80       seccion = 2;
81     }
82   }
83 }
```

Diseño, desarrollo y construcción de un robot hexápodo para inspección de áreas de difícil acceso.

Procedimientos básicos

```
1 //----- Posiciona un servo en concreto -----
2 void posiciona(int pata, int tramo, int angulo) {
3     uint16_t microsegundos = map(angulo, -90, 90, micros_min, micros_max);
4     int canal = (3 * (pata - 1)) + (tramo - 1);
5     driver.setTarget(canal, microsegundos);
6 }
7
8 //----- Posiciona varios servos en orden de canal -----
9 void posiciona_en_serie(int numero_de_servos, int canal_inicio, uint16_t
10 *p_posiciones) {
11     uint16_t envio[NUMSERVOS];
12     int final = canal_inicio + numero_de_servos;
13     for (int i = canal_inicio; i < final; i++) {
14         envio[i] = map(p_posiciones[i], 0, 180, micros_min, micros_max);
15     }
16     driver.setMultiTarget(numero_de_servos, canal_inicio, envio);
17     delay(1000);
18 }
19 //----- Posiciona los tres servos de una pata -----
20 void posiciona_en_serie_pata(int canal_inicio, uint16_t *p_pos_pata) {
21     uint16_t envio[3];
22     for (int i = 0; i < 3; i++) {
23         envio[i] = map(p_pos_pata[i], 0, 180, micros_min, micros_max);
24     }
25     driver.setMultiTarget(3, canal_inicio, envio);
26 }
27
28 //----- Mueve la pata en función de las entradas y de la sección del joystick -----
29 void posiciona_pata(int pata, int servo_1, int servo_2, int servo_3) {
30
31     switch (seccion) {
32         case 1:
33             pata = wrapAround(pata, -1);
34             break;
35         case 2:
36             //nada
37             break;
38         case 3:
39             pata = wrapAround(pata, 1);
40             break;
41         case 4:
42             pata = wrapAround(pata, 2);
43             break;
44         case 5:
45             pata = wrapAround(pata, 3);
46             break;
47         case 6:
48             pata = wrapAround(pata, -2);
49             break;
50     }
51
52     pos_pata[0] = servo_1 + 90;
53     pos_pata[1] = servo_2 + 90;
54     pos_pata[2] = servo_3 + 90;
55     posiciona_en_serie_pata(((pata - 1) * 3), pos_pata);
56 }
57
58 //----- Inicializa posiciones de los servos -----
59 void inicializa_servos() {
60     for (int i = 0; i < NUMSERVOS; i++) {
61         posiciones[i] = 90;
62         driver.setAcceleration(i, ACC); //0 sin limitación rango(1-255)
63         driver.setSpeed(i, VEL); //0 sin limitación rango(1-255)
64     }
65     posiciona_en_serie(18, 0, posiciones);
66     estoy_en_cero = true;
67 }
68
69 //----- Mueve todas las patas a posición de salida -----
70 void ir_a_pos_cero() {
71     levanta_pares();
72     delay(delays);
73 }
```

Diseño, desarrollo y construcción de un robot hexápodo para inspección de áreas de difícil acceso.

```
73     pos_cero_pares();
74     delay(delays);
75     levanta_impares();
76     delay(delays);
77     pos_cero_impares();
78     delay(delays);
79     vete_a_cero = false;
80     estoy_en_cero = true;
81 }
82
83 //----- Rotar la pata en función de la dirección -----
84 int wrapAround(int pata, int desviacion) {
85     int result = (pata + desviacion - 1) % 6 + 1;
86     if (result <= 0) {
87         result += 6;
88     }
89     return result;
90 }
91
92 //----- Modificar la velocidad de los servos -----
93 void mod_vel(int vel_nueva) {
94     for (int i = 0; i < NUMSERVOS; i++) {
95         driver.setSpeed(i, vel_nueva);
96     }
97 }
98
99 //----- Modificar la aceleración de los servos -----
100 void mod_ac(int ac_nueva) {
101     for (int i = 0; i < NUMSERVOS; i++) {
102         driver.setAcceleration(i, ac_nueva);
103     }
104 }
```

Diseño, desarrollo y construcción de un robot hexápodo para inspección de áreas de difícil acceso.

Movimientos

```
1  //----- Levantar una pata -----
2  void levanta(int pata) {
3      posiciona_pata(pata,0,45,-45);
4  }
5
6  //----- Posicionar en cero una pata-----
7  void pos_cero(int pata) {
8      posiciona_pata(pata,0, 0, 0);
9  }
10
11 //----- Puntos "avanza" de cada pata -----
12 void avanza(int pata) {
13     switch (pata) {
14         case 1:
15             posiciona_pata(pata,17, 45, -45);
16             break;
17         case 2:
18             posiciona_pata(pata,23, 45, -45);
19             break;
20         case 3:
21             posiciona_pata(pata,3, 45, -45);
22             break;
23         case 4:
24             posiciona_pata(pata,-3, 45, -45);
25             break;
26         case 5:
27             posiciona_pata(pata,-23, 45, -45);
28             break;
29         case 6:
30             posiciona_pata(pata,-17, 45, -45);
31             break;
32     }
33 }
34
35 //----- Puntos de posición "empuja" de cada pata -----
36 void empuja(int pata) {
37     switch (pata) {
38         case 1:
39             posiciona_pata(pata,-3, -8, 14);
40             break;
41         case 2:
42             posiciona_pata(pata,-12, 0, 5);
43             break;
44         case 3:
45             posiciona_pata(pata,-12, 1, -23);
46             break;
47         case 4:
48             posiciona_pata(pata,12, 1, -23);
49             break;
50         case 5:
51             posiciona_pata(pata,12, 0, 5);
52             break;
53         case 6:
54             posiciona_pata(pata,3, -8, 14);
55             break;
56     }
57 }
58
59 //----- Puntos de posición "empujal" de cada pata -----
60 void empujal(int pata) {
61     switch (pata) {
62         case 1:
63             posiciona_pata(pata,8, 2, -10);
64             break;
65         case 2:
66             posiciona_pata(pata,0, 0, 0);
67             break;
68         case 3:
69             posiciona_pata(pata,-1, 0, -10);
70             break;
71         case 4:
72             posiciona_pata(pata,1, 0, -10);
73             break;
```

Diseño, desarrollo y construcción de un robot hexápodo para inspección de áreas de difícil acceso.

```
74     case 5:
75         posiciona_pata(pata,0, 0, 0);
76         break;
77     case 6:
78         posiciona_pata(pata,-8, 2, -10);
79         break;
80     }
81 }
82
83 //----- Puntos de posición "empuja2" de cada pata -----
84 void empuja2(int pata) {
85     switch (pata) {
86         case 1:
87             posiciona_pata(pata,-3, -8, 14);
88             break;
89         case 2:
90             posiciona_pata(pata,-12, 0, 5);
91             break;
92         case 3:
93             posiciona_pata(pata,-12, 1, -23);
94             break;
95         case 4:
96             posiciona_pata(pata,12, 1, -23);
97             break;
98         case 5:
99             posiciona_pata(pata,12, 0, 5);
100            break;
101        case 6:
102            posiciona_pata(pata,3, -8, 14);
103            break;
104    }
105 }
106
107 //----- Puntos de posición "levanta_avanza" de cada pata -----
108 void levanta_avanza(int pata) {
109
110     switch (pata) {
111         case 1:
112             posiciona_pata(pata,2, 36, -38);
113             break;
114         case 2:
115             posiciona_pata(pata,0, 45, -45);
116             break;
117         case 3:
118             posiciona_pata(pata,-4, 38, -37);
119             break;
120         case 4:
121             posiciona_pata(pata,4, 38, -37);
122             break;
123         case 5:
124             posiciona_pata(pata,0, 45, -45);
125             break;
126         case 6:
127             posiciona_pata(pata,-2, 36, -38);
128             break;
129     }
130 }
131
132 //----- Puntos de posición "baja_avanza" de cada pata -----
133 void baja_avanza(int pata) {
134     switch (pata) {
135         case 1:
136             posiciona_pata(pata,17, 6, -26);
137             break;
138         case 2:
139             posiciona_pata(pata,23, 0, 0);
140             break;
141         case 3:
142             posiciona_pata(pata,3, -9, 13);
143             break;
144         case 4:
145             posiciona_pata(pata,-3, -9, 13);
146             break;
```

Diseño, desarrollo y construcción de un robot hexápodo para inspección de áreas de difícil acceso.

```
147     case 5:  
148         posiciona_pata(pata,-23, 0, 0);  
149         break;  
150     case 6:  
151         posiciona_pata(pata,-17, 6, -26);  
152         break;  
153     }  
154 }  
155  
156
```

Diseño, desarrollo y construcción de un robot hexápodo para inspección de áreas de difícil acceso.

Tripod

```
1  //----- Procedimientos de llamada a cada pata para cada movimiento
2  -----
3  void levanta_impares() {
4      levanta(1);
5      levanta(3);
6      levanta(5);
7  }
8  void pos_cero_impares() {
9      pos_cero(1);
10     pos_cero(3);
11     pos_cero(5);
12 }
13 void avanza_impares() {
14     avanza(1);
15     avanza(3);
16     avanza(5);
17 }
18 void empuja_impares() {
19     empuja(1);
20     empuja(3);
21     empuja(5);
22 }
23 void empuja_1_impares(){
24     empujal(1);
25     empujal(3);
26     empujal(5);
27 }
28 void empuja_2_impares(){
29     empuja2(1);
30     empuja2(3);
31     empuja2(5);
32 }
33 void levanta_avanza_impares() {
34     levanta_avanza(1);
35     levanta_avanza(3);
36     levanta_avanza(5);
37 }
38 void baja_avanza_impares() {
39     baja_avanza(1);
40     baja_avanza(3);
41     baja_avanza(5);
42 }
43 void levanta_pares() {
44     levanta(2);
45     levanta(4);
46     levanta(6);
47 }
48 void pos_cero_pares() {
49     pos_cero(2);
50     pos_cero(4);
51     pos_cero(6);
52 }
53 void avanza_pares() {
54     avanza(2);
55     avanza(4);
56     avanza(6);
57 }
58 void empuja_pares() {
59     empuja(2);
60     empuja(4);
61     empuja(6);
62 }
63 void empuja_1_pares(){
64     empujal(2);
65     empujal(4);
66     empujal(6);
67 }
68 void empuja_2_pares(){
69     empuja2(2);
70     empuja2(4);
71     empuja2(6);
72 }
```

Diseño, desarrollo y construcción de un robot hexápodo para inspección de áreas de difícil acceso.

```
73 void levanta_avanza_pares() {
74     levanta_avanza(2);
75     levanta_avanza(4);
76     levanta_avanza(6);
77 }
78 void baja_avanza_pares() {
79     baja_avanza(2);
80     baja_avanza(4);
81     baja_avanza(6);
82 }
83
84 //----- Procedimiento de avanzar en modo "tripod gait" -----
85 void avanza_tripod(int delay servo) {
86
87     if (estoy_en_cero == true) { //si está en posición cero posiciona las patas
88         impares adelante
89         estoy_en_cero = false;
90
91         levanta_impares();
92         delay(delay servo);
93
94         avanza_impares();
95         delay(delay servo);
96
97         baja_avanza_impares();
98         delay(delay servo);
99     }
100     while (marcha && !vete_a_cero ) { //continúa avanzando mientras marcha esté activa
101         y no se le mande a cero
102
103         levanta_pares();
104         delay(delay servo);
105
106         empuja_impares();
107         delay(delay servo);
108
109         baja_avanza_pares();
110         delay(delay servo);
111
112         levanta_avanza_impares();
113         delay(delay servo);
114
115         empuja_pares();
116         delay(delay servo);
117
118         baja_avanza_impares();
119         delay(delay servo);
120
121     }
122 }
```

Diseño, desarrollo y construcción de un robot hexápodo para inspección de áreas de difícil acceso.

Giros tripod

```
1 void gira_derecha(int delayservo) {
2   gira(-22, delayservo); //define sentido de rotación para las articulaciones 1
3 }
4
5 void gira_izquierda(int delayservo) {
6   gira(22, delayservo); //define sentido de rotación para las articulaciones 1
7 }
8
9 void gira(int giro, int delayservo) {
10
11   posiciona_pata(1, 0, 45, -45); //levanta impares
12   posiciona_pata(3, 0, 45, -45);
13   posiciona_pata(5, 0, 45, -45);
14   delay(delayservo);
15
16   posiciona(1, 1, giro); //gira articulación 1
17   posiciona(3, 1, giro);
18   posiciona(5, 1, giro);
19   posiciona(2, 1, -giro);
20   posiciona(4, 1, -giro);
21   posiciona(6, 1, -giro);
22   delay(delayservo);
23
24   posiciona_pata(1, giro, 0, 0); //baja impares
25   posiciona_pata(3, giro, 0, 0);
26   posiciona_pata(5, giro, 0, 0);
27   delay(delayservo);
28
29   posiciona_pata(2, -giro, 45, -45); //levanta pares
30   posiciona_pata(4, -giro, 45, -45);
31   posiciona_pata(6, -giro, 45, -45);
32   delay(delayservo);
33
34   posiciona(1, 1, 0); //gira articulación 1 a posición inicial
35   posiciona(3, 1, 0);
36   posiciona(5, 1, 0);
37   posiciona(2, 1, 0);
38   posiciona(4, 1, 0);
39   posiciona(6, 1, 0);
40   delay(delayservo);
41
42   posiciona_pata(2, 0, 0, 0); //baja pares
43   posiciona_pata(4, 0, 0, 0);
44   posiciona_pata(6, 0, 0, 0);
45   delay(delayservo);
46
47   estoy_en_cero = true;
48 }
```

Diseño, desarrollo y construcción de un robot hexápodo para inspección de áreas de difícil acceso.

Mando

```
1  float angleRadians = 0;
2  float angleDegrees = 0;
3  int8_t x = 0;
4  int8_t y = 0;
5  int8_t x_giro = 0;
6  int8_t y_giro = 0;
7
8
9  void notify() {
10
11     //----- Botones traseros L1 y R1 -----
12     if (Ps3.event.button_down.l1) {
13         if (delays < 1000) {
14             delays += 100;
15         }
16     }
17
18     if (Ps3.event.button_down.r1) {
19         if (delays > 300) {
20             delays -= 100;
21         }
22     }
23
24     //----- Joystick Izquierdo -----
25     if (abs(Ps3.event.analog_changed.stick.lx) +
26         abs(Ps3.event.analog_changed.stick.ly) > 2) {
27
28         x = Ps3.data.analog.stick.lx;
29         y = Ps3.data.analog.stick.ly;
30
31         if (abs(x) > zona_muerta || abs(y) > zona_muerta) //Damos un margen para saber
32             que se está pulsando el joystick
33         {
34             angleRadians = atan2(-y, x);
35             angleDegrees = angleRadians * (180.0 / PI);
36
37             // Ajustar el ángulo para que esté en el rango de 0 a 360 grados
38             if (angleDegrees < 0) {
39                 angleDegrees += 360;
40             }
41             //Divido entre 60 para obtener en cual de las 6 secciones de 60 grados nos
42             encontramos
43             seccion = angleDegrees / 60 + 1;
44             marcha = true;
45             Serial.println("Marcha true");
46             if (seccion != seccion_ant) {
47                 if (estoy_en_cero == false) {
48                     vete_a_cero = true;
49                     Serial.println("vete_a_cero_joystick_izq");
50                 }
51                 seccion_ant = seccion;
52             }
53             Serial.print("La seccion es: ");
54             Serial.print(seccion);
55             Serial.println();
56         } else {
57             marcha = false;
58             Serial.println("Marcha false");
59         }
60     }
61
62     //----- Joystick Derecho -----
63     if (abs(Ps3.event.analog_changed.stick.rx) +
64         abs(Ps3.event.analog_changed.stick.ry) > 2) {
65
66         x_giro = Ps3.data.analog.stick.rx;
67         y_giro = Ps3.data.analog.stick.ry;
68
69         if (((x_giro < -zona_muerta) || (x_giro > zona_muerta)) && (marcha == false)) {
70             if (estoy_en_cero == false) {
71                 vete_a_cero = true; //vete a cero
72                 Serial.println("vete_a_cero_joystick_der");
73             }
74         }
75     }
76 }
```

Diseño, desarrollo y construcción de un robot hexápodo para inspección de áreas de difícil acceso.

```
70     if (x_giro > zona_muerta) {
71         gira_der = true; //activo girar dcha
72         gira_izq = false;
73     } else if (x_giro < -zona_muerta) {
74         gira_izq = true; //activo girar izq
75         gira_der = false;
76     }
77
78     } else {
79         gira_der = false;
80         gira_izq = false;
81     }
82 }
83
84 //----- Estados de la batería -----
85 if (battery != Ps3.data.status.battery) {
86     battery = Ps3.data.status.battery;
87     Serial.print("The controller battery is ");
88     if (battery == ps3_status_battery_charging) Serial.println("charging");
89     else if (battery == ps3_status_battery_full) Serial.println("FULL");
90     else if (battery == ps3_status_battery_high) Serial.println("HIGH");
91     else if (battery == ps3_status_battery_low) Serial.println("LOW");
92     else if (battery == ps3_status_battery_dying) Serial.println("DYING");
93     else if (battery == ps3_status_battery_shutdown) Serial.println("SHUTDOWN");
94     else Serial.println("UNDEFINED");
95 }
96 }
97
98 void onConnect() {
99     Serial.println("Connected.");
100 }
101
```

Diseño, desarrollo y construcción de un robot hexápodo para inspección de áreas de difícil acceso.

Lecturas

```
1  int pot[18][4]; //matriz con 4 filas (lectura, min, posición cero, max) y 18
    columnas (canales)
2  int lectura = 0;
3
4  void calibra_potenciómetros() {
5      int valor_min = 1;
6      int valor_max = 3;
7      int canal_min; //numero de canal mínimo en cada pata
8
9      //----- posiciona todos los servos en posición central
10     for (int i = 0; i < NUMSERVOS; i++) {
11         posiciones[i] = 90;
12     }
13     posiciona_en_serie(18, 0, posiciones);
14
15     //----- Almacena los valores de todos los servos en posición cero
16     int k = 0;
17     for (int i = 1; i < 7; i++) {
18         for (int j = 1; j < 4; j++) {
19             pot[k][2] = lee_canal(i, j);
20             k++;
21         }
22     }
23
24     //----- Recorre todas las patas haciendo la secuencia de calibración y
    almacenamiento de valores
25     for (int pata_pot = 1; pata_pot < 7; pata_pot++) {
26         canal_min = 3 * (pata_pot - 1); //se trata del canal de la articulación 1 de
    cada pata
27         posiciona_pata(pata_pot, 0, -90, 90);
28         delay(1000);
29         pot[canal_min + 1][valor_min] = lee_canal(pata_pot, canal_min + 2);
30         pot[canal_min + 2][valor_max] = lee_canal(pata_pot, canal_min + 3);
31         posiciona_pata(pata_pot, 90, -90, 90);
32         delay(1000);
33         pot[canal_min + 0][valor_max] = lee_canal(pata_pot, canal_min + 1);
34         posiciona_pata(pata_pot, -90, -90, 90);
35         delay(1000);
36         pot[canal_min + 0][valor_min] = lee_canal(pata_pot, canal_min + 1);
37         posiciona_pata(pata_pot, 0, -90, 90);
38         delay(500);
39         posiciona_pata(pata_pot, 0, -90, -90);
40         delay(500);
41         pot[canal_min + 2][valor_min] = lee_canal(pata_pot, canal_min + 3);
42         posiciona_pata(pata_pot, 0, 90, -90);
43         delay(1000);
44         pot[canal_min + 1][valor_max] = lee_canal(pata_pot, canal_min + 2);
45     }
46 }
47
48 int lee_canal(int pata, int art) {
49     int lectura;
50     int canal;
51     int aux;
52     int binario[5];
53     canal = 3 * (pata - 1) + (art - 1); //convierte número de pata y articulación en
    número de canal entre 0 y 18
54
55     //----- binariza canal y lo mete en un vector llamado binario -----
56     aux = canal;
57     for (int i = 0; i < 4; i++) {
58         binario[i] = aux % 2;
59         aux = aux / 2;
60     }
61     binario[4] = aux;
62
63     //----- envía a los multiplexores los bits de selección desde el vector -----
64     digitalWrite(s0_mux, binario[0]);
65     digitalWrite(s1_mux, binario[1]);
66     digitalWrite(s2_mux, binario[2]);
67     digitalWrite(s3_mux, binario[3]);
68     digitalWrite(s4_mux, binario[4]);
69 }
```

Diseño, desarrollo y construcción de un robot hexápodo para inspección de áreas de difícil acceso.

```
70     lectura = analogRead(pin_lectura);  
71     lectura = map(lectura, 0, 4095, 90, -90);  
72  
73     return lectura;  
74 }
```