



UNIVERSI
POLITÈCN
DE VALÈN



Escuela Técnica Superior de Ingeniería del Diseño



Diseño e implementación de un cuadrúpedo con funcionalidades de bípedo

MEMORIA PRESENTADA POR:

Francesco Vurchio

Máster Universitario en Ingeniería Mecatrónica

DIRECTOR:

Juan José Serrano Martín

Septiembre, 2016

Índice

1. Introducción
2. Marco Teórico
3. Selección de componentes
 - *Servos*
 - *Modulo Microcontrolador STM32F334R8 Núcleo*
 - *Sensor de distancia (Ultrasonido)*
 - *Batería*
 - *Sensor de distancia (Infrarrojos)*
 - *Modulo relés*
 - *Modulo receptor de Bluetooth*
4. Estructura del Robot
5. Pruebas de levantamiento
6. Diseño del Robot
 - *Disposición de los servos*
 - *Esquema de funcionamiento y eléctrico del robot*
7. Prototipo básico
 - *Posición vertical (Posición perro sentado)*
 - *Posición vertical (Posición extendida)*
8. Programación
 - *Configuración de la tarjeta*
 - *Programación mediante el software uVisión 5*
 - *Modos de funcionamiento*
9. Prototipo final
 - *Estudio dinámico*
 - *Levantamiento del prototipo final*
10. Conclusiones y estudios futuros

INTRODUCCIÓN

El siguiente trabajo presenta los análisis, estudios y realización completa de un robot cuadrúpedo, capaz de levantarse y ejecutar funcionalidades de robot bípedos. En las páginas siguientes se mostrará en detalle todo el proceso de diseño del cuadrúpedo (estructura y componentes electrónicos) y sucesivamente se enfocará la parte de ensamblaje y programación de la tarjeta de control utilizada.

La idea principal es la mejora del robot cuadrúpedo clásico, estudiado durante el máster de Mecatrónica, buscando por un lado nuevas soluciones eficaces en la fase de funcionamiento ordinaria y por otro lado yendo más allá del concepto más común de cuadrúpedo, pasando a una visión más general de robot, capaz de moverse y actuar con características más similares a las de los robots humanoides.

El motivo que me ha llevado a enfrentarme a este proyecto es, a parte el interés y la pasión por la inteligencia artificial y la robótica en general, la extraordinaria posibilidad de poder aplicar y mejorar, en este tipo de trabajo, muchos de los conocimientos adquiridos durante el máster de Mecatrónica, en las distintas disciplinas que incorpora, desde la mecánica y la electrónica analógica y digital, hasta los sistemas embebidos y el control.

MARCO TEÓRICO

En este apartado se presentará toda la parte teórica sobre la que se ha basado la entera investigación, previa al trabajo, y la sucesiva mejora constante de las funcionalidades durante la creación del prototipo.

La parte teórica objeto del estudio previo ha sido prevalentemente enlazada a la del modelo clásico del cuadrúpedo para facilitar, en primer lugar, la realización de un cuadrúpedo completo capaz de andar en las dos direcciones y de girar. Sucesivamente el estudio se ha extendido a los robot bípedos, y el modelo inicial se ha rediseñado para añadir nuevas funcionalidades y permitir al cuadrúpedo levantarse y permanecer en posición vertical.

Empezando con la definición teórica de cuadrúpedo, un cuadrúpedo es un animal que se traslada caminando sobre cuatro extremidades (definición de Enciclopediadetareas.net).

En la naturaleza la mayoría de los animales tetrápodos actuales son cuadrúpedos y algunos de ellos son capaces de recorrer distancias más o menos cortas con solo dos extremidades apoyadas en el suelo (como por ejemplo los simios o los osos).

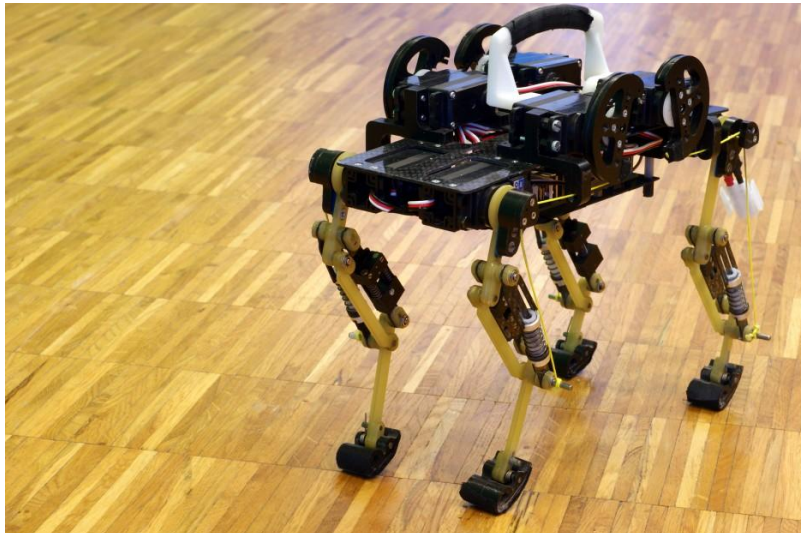
En la robótica claramente el término cuadrúpedo, así como por la especie animal, se refiere a un robot capaz de moverse soportado por cuatro extremidades; en la imagen siguiente se observa un ejemplo de robot cuadrúpedo:



Generalmente un robot cuadrúpedo posee dos o tres servo motores por cada pata, según los grados de libertad que se necesiten; los servos que permiten la traslación

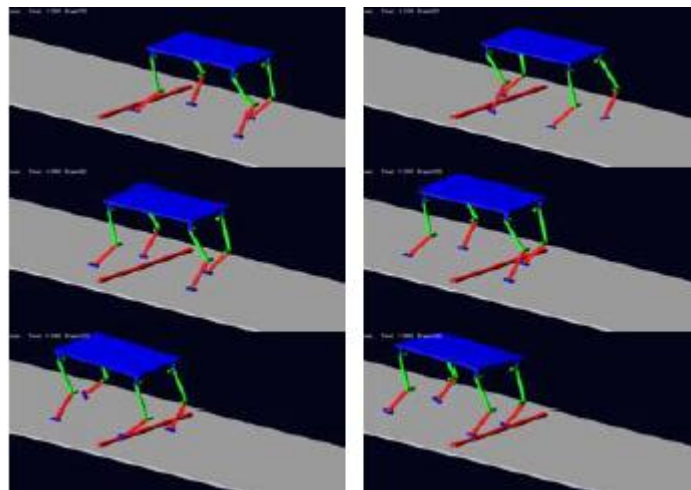
paralela al suelo de las patas normalmente están posicionados entre la pata y el núcleo central del robot.

Otra solución que se ha extendido en el mercado prevé dos servo motores por cada pata sin el servo de translación horizontal, como se puede apreciar en figura:

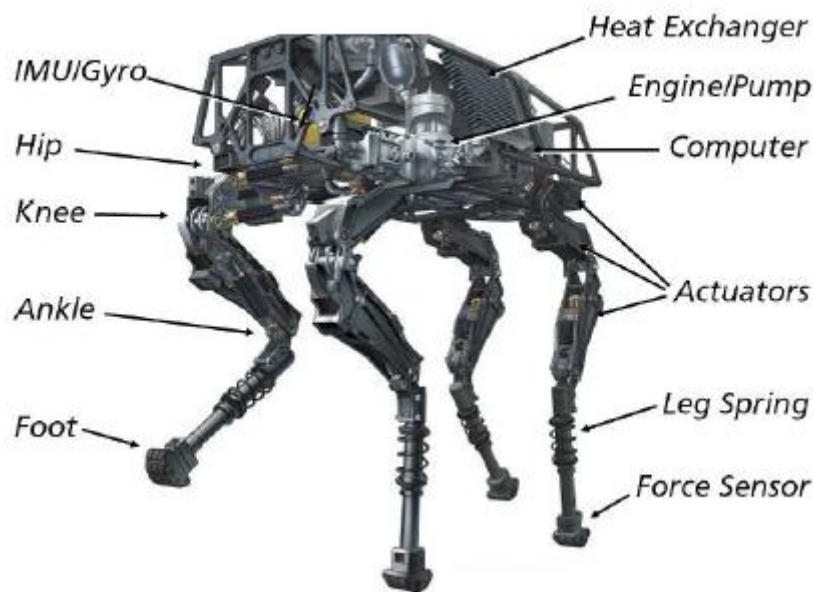


Este modelo de cuadrúpedo ha demostrado ser una solución muy óptima si se busca mucha estabilidad y más facilidad en el superar obstáculos y en subir escaleras o rampas muy pendientes; sin embargo los modelos de cuadrúpedo que permiten el desplazamiento horizontal de las patas parecen proporcionar más estabilidad, tanto durante los movimientos como en posición de descanso.

En la siguiente secuencia de imágenes se muestra un ejemplo de adelantamiento de obstáculo por parte de un robot inspirado biológicamente en un animal cuadrúpedo: El Robot Biosbot de la Universidad Tsinghua de Beijing.



Esta solución es muy interesante porque, además de permitir ahorrar un servo motor por cada pata, da al robot un enfoque más cercano al mundo animal, donde el motor del medio trabaja en la misma modalidad que la rodilla de un cuadrúpedo del mundo animal. Este tipo de robot es ampliamente estudiado por muchas de las mejores empresas a nivel mundial en el sector de la robótica, las cuales, aumentando notablemente el nivel de complejidad de las articulaciones, año tras año consiguen acercarse más a lo que son movimientos naturales propios del mundo animal.



En la figura anterior se observa una de las últimas creaciones de la prestigiosa *Boston Dynamics*, llamada BigDog. Este cuadrúpedo de última generación se propone simular al detalle los movimientos de un animal cuadrúpedo, optimizando la estabilidad frente a perturbaciones y en presencia de terrenos abruptos, gracias a un sistema muy avanzado de actuadores, sensores y amortiguadores.

En segundo lugar, el estudio se ha focalizado en los robots humanoides, capaces de desplazarse con solo dos extremidades tocando el suelo.

Actualmente en el mercado se comercializa una variedad muy amplia de robots bípedos cuyos movimientos e interacciones con el exterior pretenden emular en parte las humanas; desde los modelos dirigidos más hacia el mundo del juego y del entretenimiento, hacia otros más profesionales dotados de inteligencia artificial de gran espesor.

Gracias a las nuevas tecnologías, que permiten cada vez más velocidad de elaboración de los datos, más precisión en el control y más almacenamiento, en los últimos años

los avances en esta franja de la robótica han sido impresionantes; muchas de las mayores empresas de robótica en el mundo, como Boston Dynamics y Honda, han perfeccionado de manera sustancial las prestaciones de sus principales robots humanoides, alcanzando niveles de precisión y realismo increíbles. Un ejemplo de este avance es DRC-Hubo+, desarrollado por la empresa sur-coreana KAIST (Korea Advanced Institute of Science and Technology) en el 2015.



Esta nueva versión de Hubo, ganadora de la edición 2015 de la DARPA Robotics Challenge, enriquece la anterior versión DRC-Hubo del 2013 en casi todos los aspectos, entre todos la estabilidad y la velocidad (introduciendo un sistema de ruedas para el desplazamiento en superficies regulares) , además de la precisión de los movimientos y la capacidad de adaptarse a las distintas situaciones y condiciones del ambiente que lo circunda; todo esto es posible gracias a la rapidísima y precisa reconstrucción que Hubo es capaz de hacer del lugar a su alrededor, a través del complejo sistema de sensores y de la mejorada inteligencia artificial, acompañada de más grados de libertad asociados a las articulaciones que permiten movimientos más elaborados y dimensiones y peso más cercanos a los humanos (80 Kg por 1,75cm).

SELECCIÓN DE COMPONENTES

Después de un estudio inicial sobre los distintos tipos de robot cuadrúpedos presentes actualmente en el mercado, se han planteado varias posibles ideas de base para el prototipo inicial del proyecto. En general, los dos conceptos fundamentales que se han seguido y que han influido de manera sustancial sobre la elección de los componentes han sido:

- El peso: Se ha visto como el peso es de importancia crucial en todo tipo de robot así como su distribución en la estructura (pequeñas variaciones del baricentro varían de manera sustancial movimientos y funcionalidades del robot)
- Fuerza y precisión de los motores: La elección de buenos motores, capaces de generar un par relativamente elevado y que garanticen una muy buena precisión y repetitividad es la clave para que un robot sea algo más que un simple robot con funciones muy elementales.

Es evidente que también los motores juegan un papel fundamental en lo que va a ser el peso total y la distribución del mismo en el robot. Para ello se han efectuado varias pruebas, empezando con algunas muy básicas donde se han utilizado servo motores HDKJ S3090D

- Servos



Que presenta las siguientes características:

- Par generado: 9Kg*cm con 6 Voltios de alimentación, 10Kg*cm con 7,2 Voltios
- Velocidad: 0,16 seg/60° con 6 Voltios, 0,14 seg/60° con 7,2 Voltios

- Corriente sin carga: <500mA
- Tamaño: 40*20*40,5mm
- Longitud cable: 320mm
- Zona muerta: 4us
- Peso: 130g

Los motores utilizados para el prototipo final han sido los HSR-8498HB:



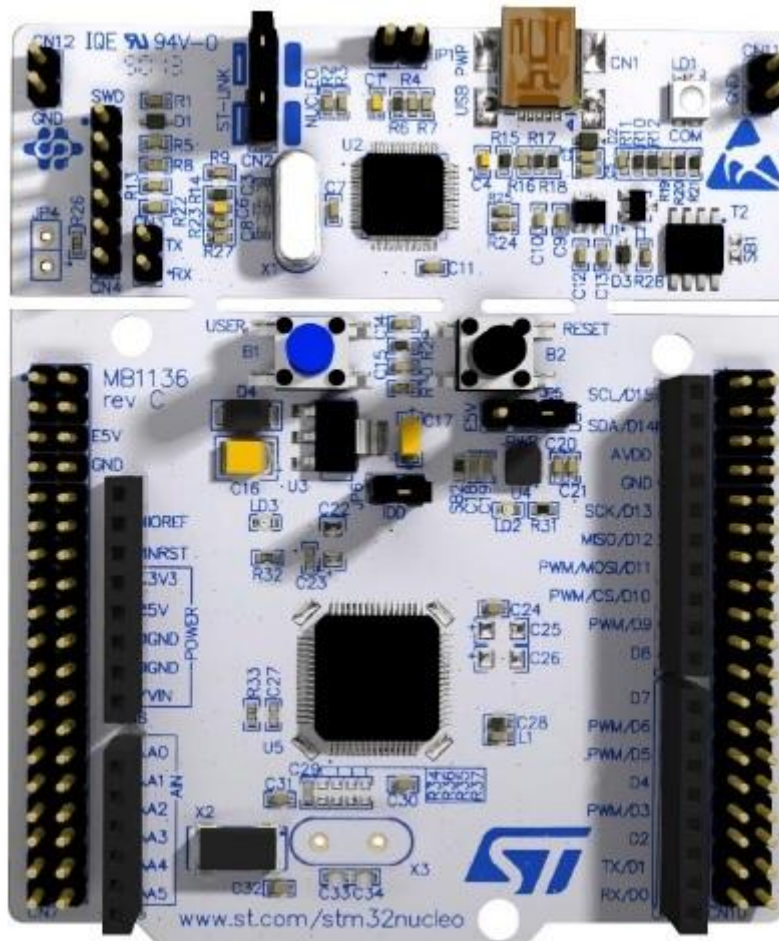
Con las siguientes características:

- Par generado: 6,7Kg*cm con 6 Voltios de alimentación, 8,1Kg*cm con 7,4 Voltios

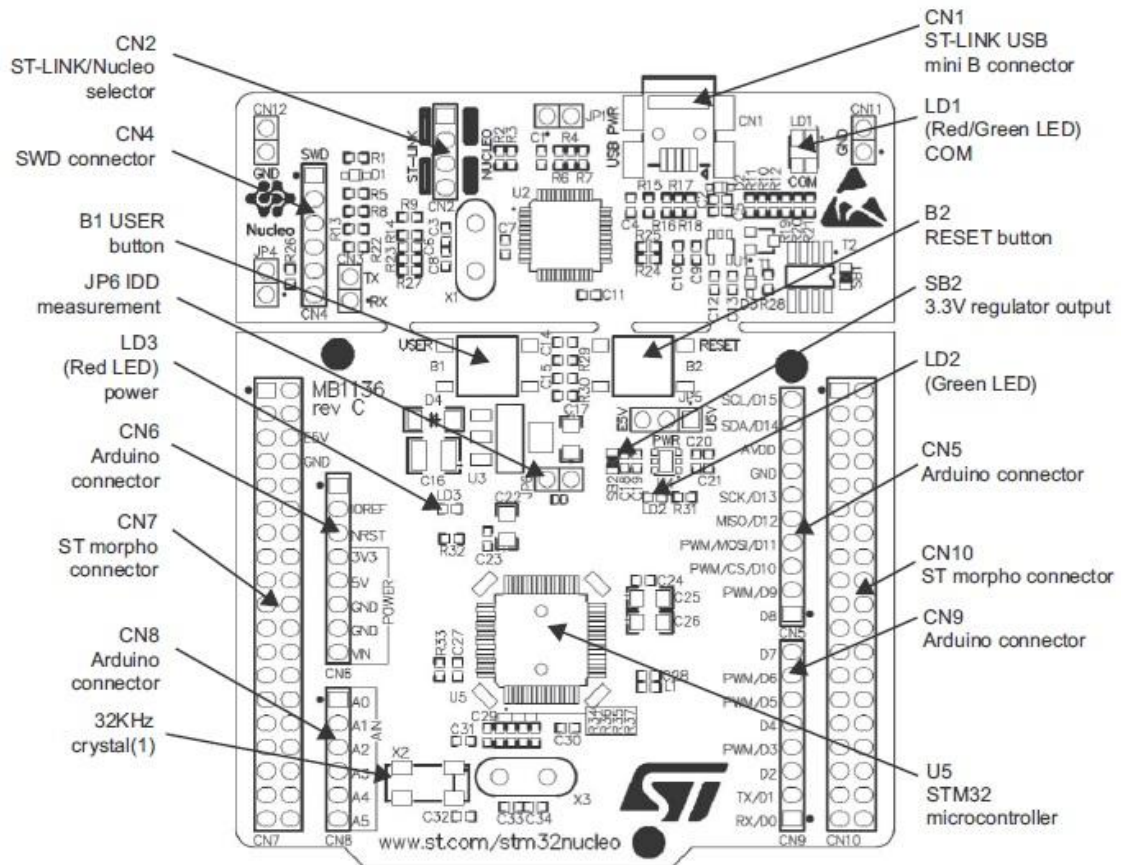
- Velocidad: 0,2 seg/60° con 6 Voltios, 0,18 seg/60° con 7,2 Voltios
- Corriente sin carga: <300mA
- Tamaño: 40*20*37mm
- Longitud cable: 400mm
- Zona muerta: 5useg
- Peso: 54,7g

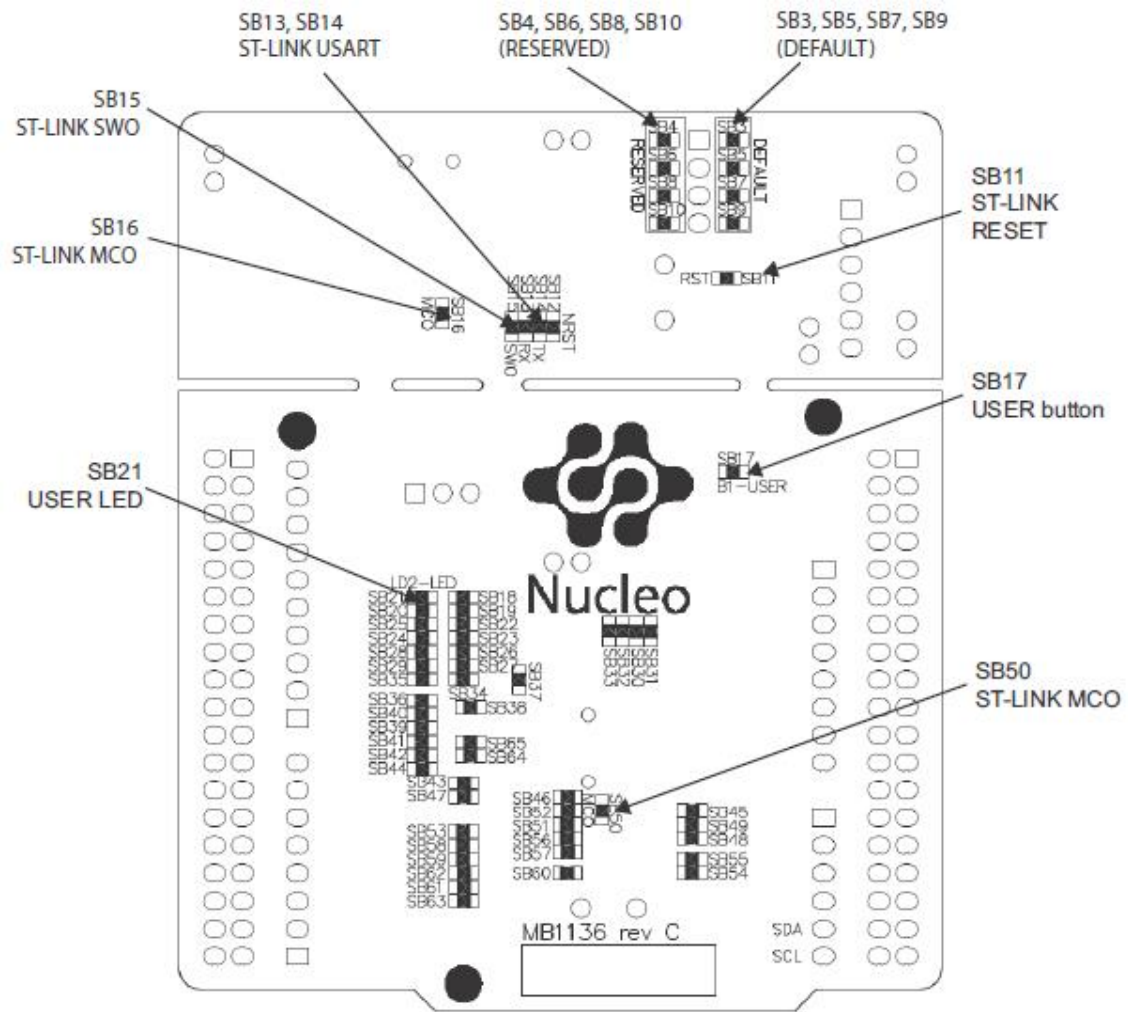
La tarjeta del microcontrolador es el corazón y el componente principal de un robot. Es indispensable para enviar las señales de control a los motores, para recibir y enviar señales analógicas y/o digitales a los sensores y actuadores y para monitorear el correcto funcionamiento de los algoritmos mediante los Leds de usuario. La tarjeta de control elegida para este proyecto ha sido la STM32 núcleo.

- *Modulo Microcontrolador STM32F334R8 Núcleo*



En las figuras siguientes se presenta el layout de arriba y el layout de abajo de la tarjeta:





Se ha preferido utilizar la tarjeta Núcleo STM32F334R8 en lugar de la Discovery F3 utilizada en el proyecto de clase principalmente por razones legadas al peso y a las dimensiones.

La Discovery F3 presenta un peso y unas dimensiones superiores debido a los componentes extra añadidos para facilitar su uso, como por ejemplo la rueda de LEDs muy útil para comprobar el correcto funcionamiento del giróscopo y el pequeño mando con cinco pulsadores. A pesar de la utilidad de estos componentes, en este proyecto no resultaban de fundamental importancia; se ha preferido optar por la tarjeta más pequeña y ligera posible capaz de controlar todos los servos necesarios, sin perder mucho en términos de velocidad y con un número suficiente de salidas/entradas digitales; así como se observará en el apartado de la programación.

En la imagen siguiente se presentan las dimensiones de la tarjeta Núcleo STM32F334R8.

- ST-LINK/V2-1 debugger/programmer (COM port virtual y almacenamiento datos)
- Compatibilidad con Arduino
- Tres LEDs: LD1 de comunicación, LD2 de usuario y LD3 de alimentación
- Alimentación flexible: USB VBUS o externa
- *Sensor de distancia (ultrasonido)*

El sensor de distancia elegido es el módulo ultrasónico HC - SR04, cuyo rango de trabajo está comprendido entre 2 cm - 400 cm, presentando pues una zona muerta de 2 cm.

La precisión de la medida oscila y puede alcanzar los 3 mm.

Los componentes internos del sensor son los transmisores ultrasónicos, el receptor y el circuito de control.

El modulo presenta cuatro pines:

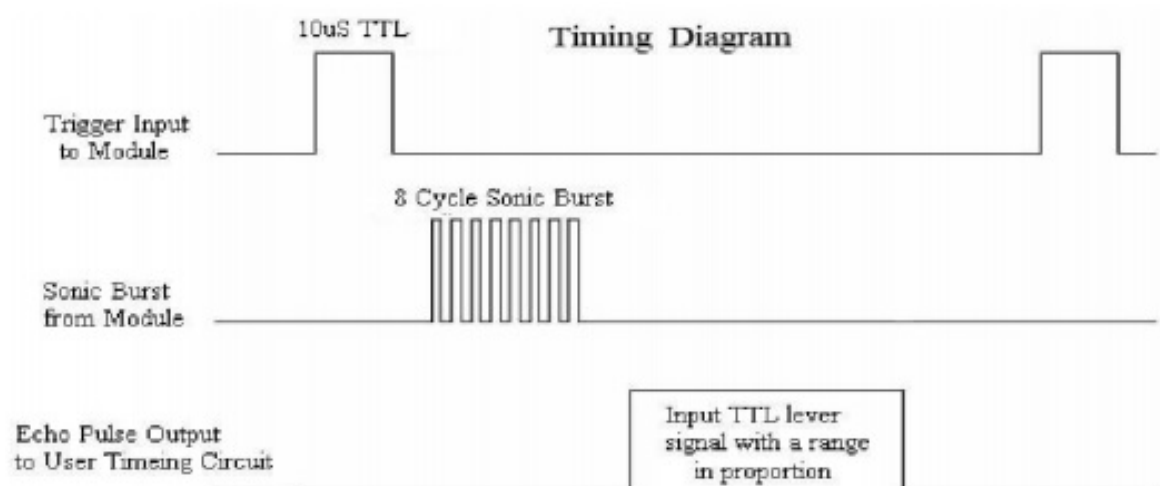
Dos de alimentación (Vcc = 5V y GND), el Trig y el Echo.

El principio básico de trabajo es:

Una vez que el módulo esté alimentado, espera una señal de trigger de 10us a través del pin *Trig*; una vez recibido este pulso de 10us, el módulo envía una señal por el pin *Echo* cuyo ancho de pulso es proporcional a la distancia medida por el sensor; la proporcionalidad es dada por la siguiente relación:

Distancia = (Periodo pulso × velocidad del sonido (340M / S) / 2),





Las características principales de este sensor son:

- Tipología I/O: Digital
- Voltaje: 5 V
- Consumo de corriente: 15mA
- Frecuencia de trabajo: 40Hz
- Distancia medida: de 2cm a 4m
- Angulo medido: 15°
- Dimensiones: 45x20x15mm

- **Batería**

La batería utilizada es de tipo Lipo (Polimero de Litio), que garantiza una capacidad suficiente para alimentar el cuadrúpedo a régimen durante más de una hora y una corriente de descarga máxima superior a la que necesitan en el arranque los diez servos motores empleados en el proyecto.

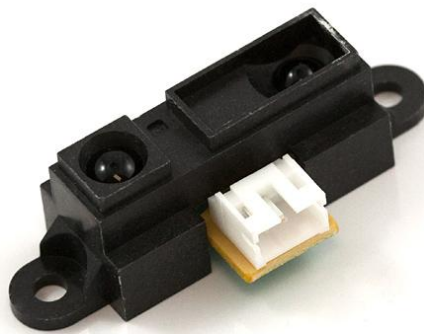


- Capacidad: 1800 mAh
- Voltaje: 7,4V
- Máxima descarga continua: 40C (72A)
- Máximo pico de descarga: 80C (144A)

- Peso: 111,9 g

- *Sensor distancia (Infrarrojos)*

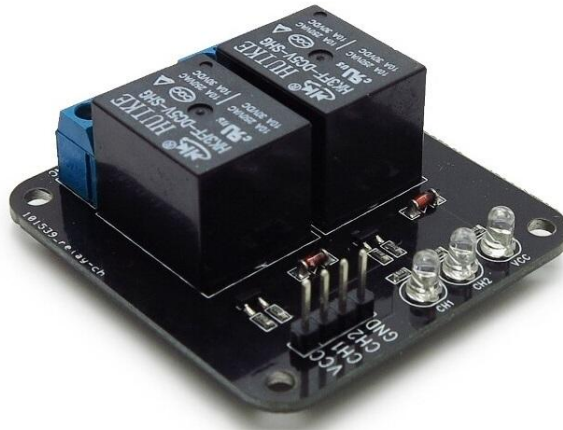
El sensor de distancia secundario utilizado es el SHARP GP2Y0A21YK0F; posicionado en la parte baja del cuerpo del cuadrúpedo y empleado solo en la modalidad bípedo.



- Distancia medida: entre 10 y 80 cm
- Tipología I/O: Analógico
- Dimensiones: 29,5x13x13,5 mm
- Consumo corriente: 30 mA
- Voltaje: De 4,5 a 5,5 V

- *Módulo Relés*

Además de desconectar la batería Lipo de alimentación de forma manual, a través de un interruptor, se ha predispuesto el prototipo final de un módulo con dos relés, para desconectarla y conectarla de forma automática, cuando sea previsto por el programa.

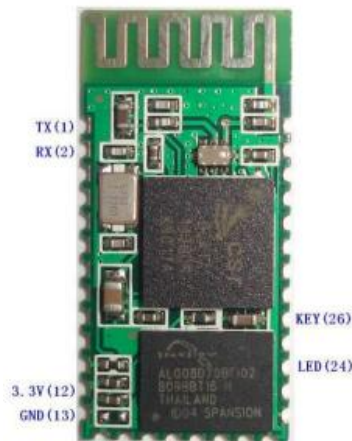


Los relés presentes en el módulo son unos HUIKE HK3FF-DC5V-SHG y tienen las siguientes características:

- Voltaje: 5V
- Corriente máxima de contacto: 10A
- Tensión máxima de contacto: 250V en AC y 30V en DC
- Dimensiones: 1,9x1,5x2,0 cm

- *Módulo receptor Bluetooth*

Se ha utilizado un módulo Bluetooth para el radio control del robot.
El módulo receptor de Bluetooth utilizado es el EGBT-046S.



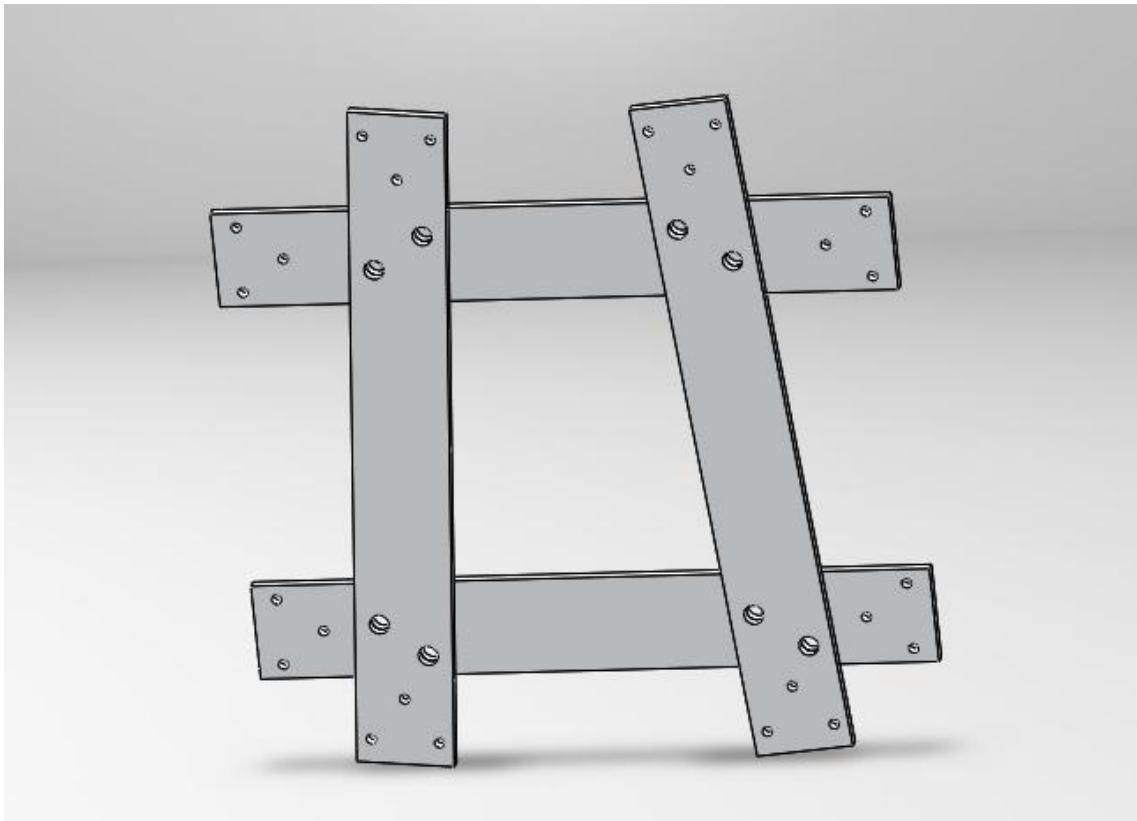
Las características son las siguientes:

- Radio Chip: CSR BC417
- Memoria: External 8Mbit Flash
- Potencia de Output: -4 to +6dbm Class 2
- Sensibilidad: -80dbm Typical
- Bit Rate: EDR, up to 3Mbps
- Interface: UART
- Dimensiones: 27x13 mm
- Tensión: De 3,1 a 4,2 Voltios

- Corriente maxima: 40mA

ESTRUCTURA DEL ROBOT

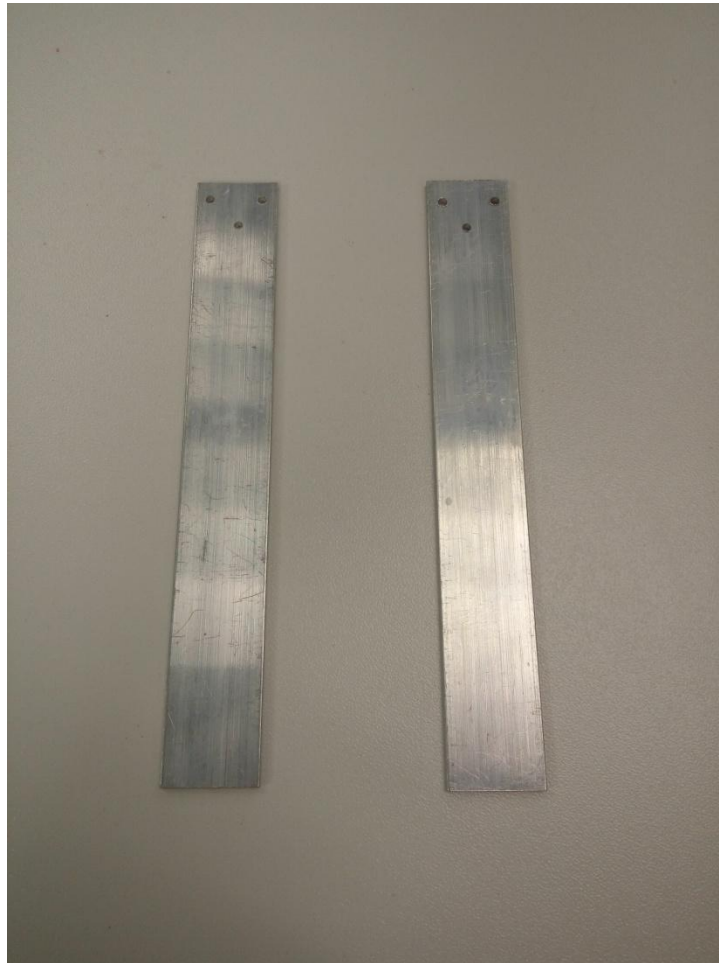
La estructura inicial del prototipo del robot es la siguiente:



Technical drawing of a mechanical part, likely a bracket or support structure, showing dimensions in millimeters (mm). The part consists of two vertical plates and two horizontal plates. Key dimensions include:

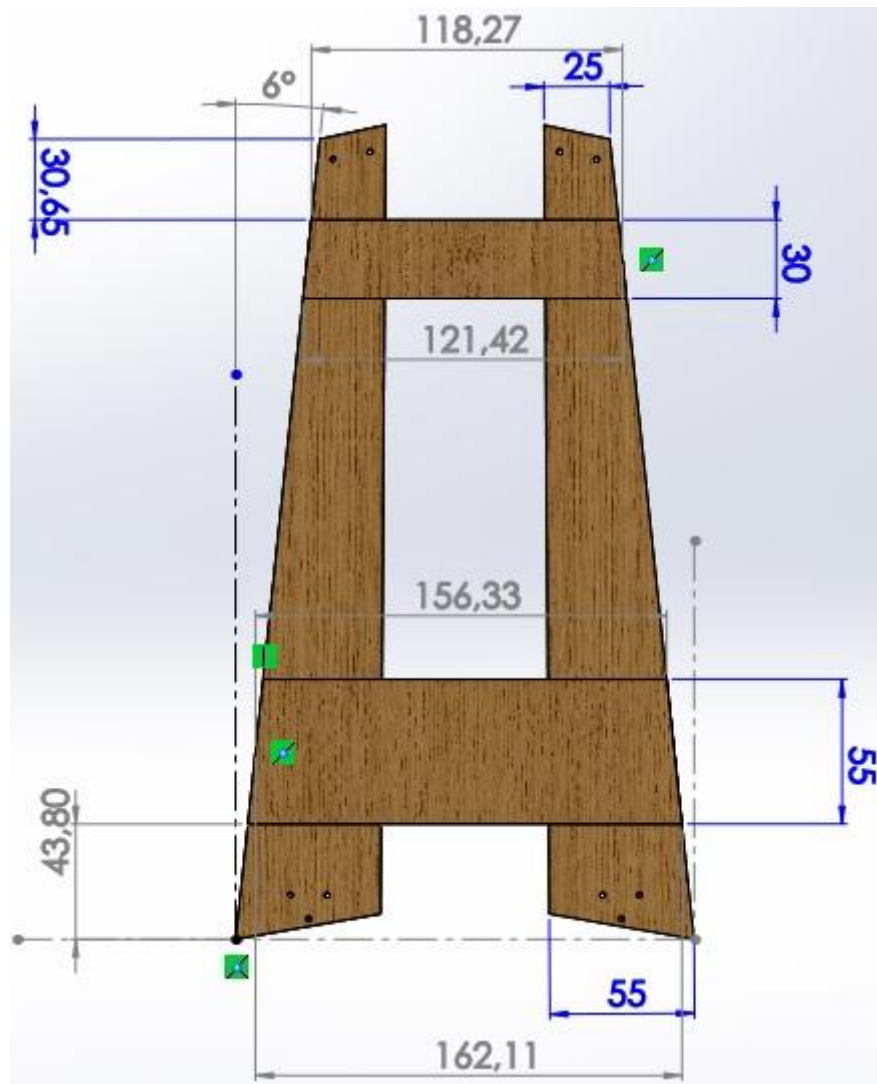
- Overall width: 140 mm
- Overall height: 20 mm
- Horizontal plate thickness: 3 mm
- Vertical plate thickness: 5 mm
- Horizontal plate width: 12 mm
- Vertical plate width: 14 mm
- Horizontal plate width (inner): 9 mm
- Vertical plate width (inner): 5 mm
- Angle between vertical plates: 12.75°

El material utilizado en este primer prototipo está compuesto por láminas de aluminio muy finas:

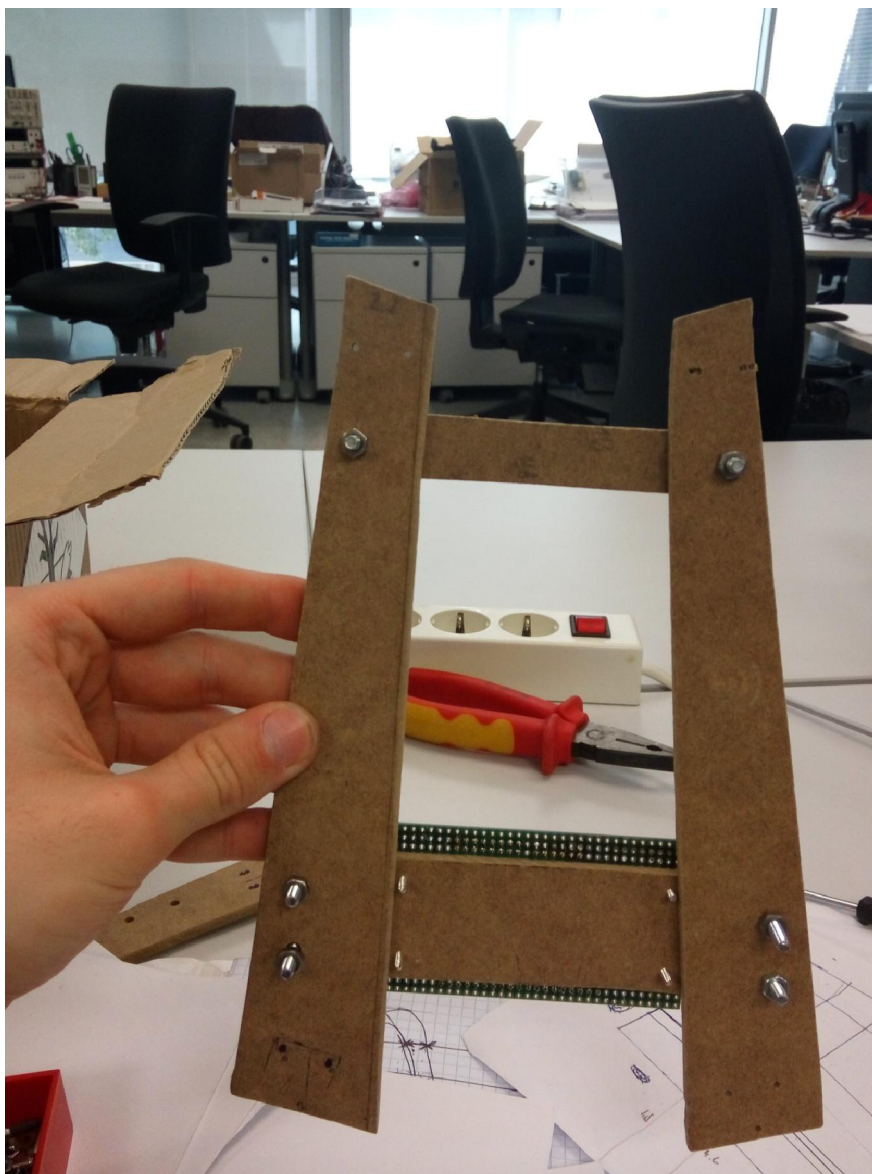


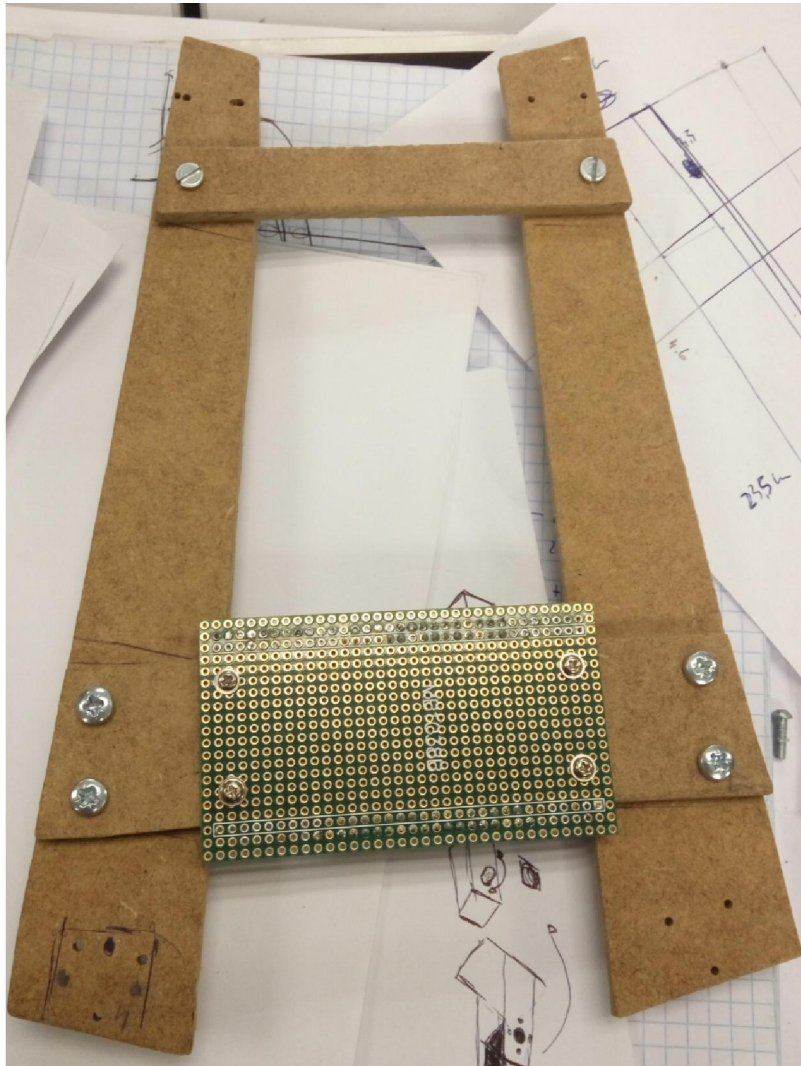
Sucesivamente se ha mejorado el prototipo, optando por una mejor estabilidad y más espacio para poder añadir otros componentes al robot, como el módulo Bluetooth, los sensores de distancia, un soporte de alimentación para los sensores y el módulo Bluetooth, del módulo de relés para desconectar la batería, un interruptor para la desconexión manual, de todo el cableado y de posibles nuevos componentes.

Aunque se haya mantenido la forma del prototipo inicial, las dimensiones han sido aumentadas a más del doble de las anteriores (30cm * 16,5cm en la parte trasera y 12,5cm en la parte delantera). El material elegido para el prototipo final ha sido un tablero de madera de fibra de densidad media; un justo equilibrio entre peso y robustez, capaz, a diferencia del aluminio, de dar al robot más flexibilidad en los movimientos.

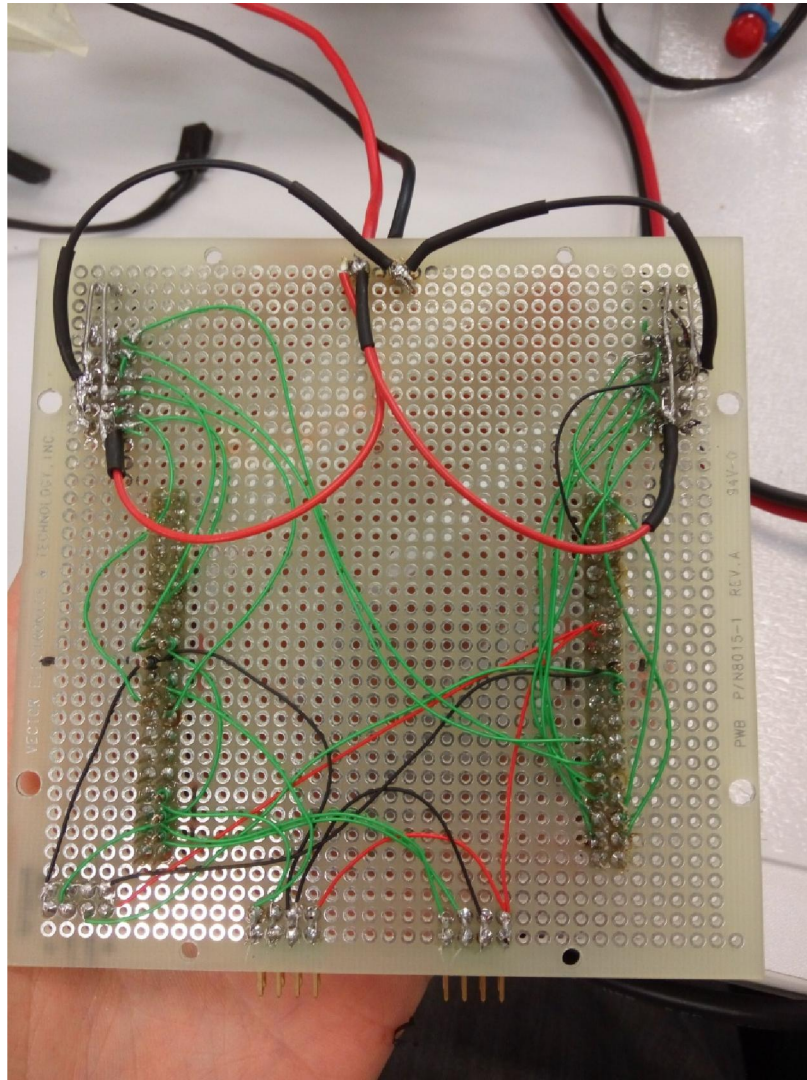


El montaje de la base se aprecia en las imágenes siguientes:



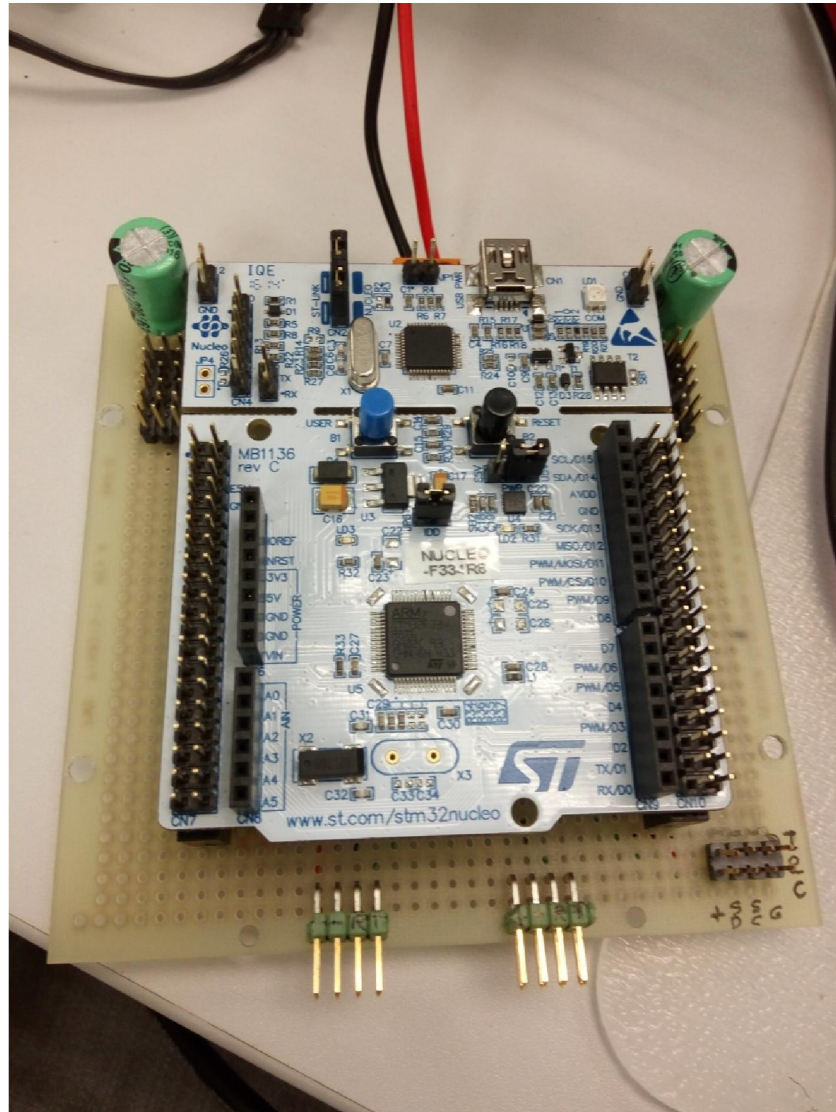


En realidad el soporte de la tarjeta de la figura ha sido rápidamente sustituido por el de la imagen siguiente, que permite organizar mejor las salidas pwm para los servos y además dispone de 2 entradas seriales muy útiles para el control en radio frecuencia:

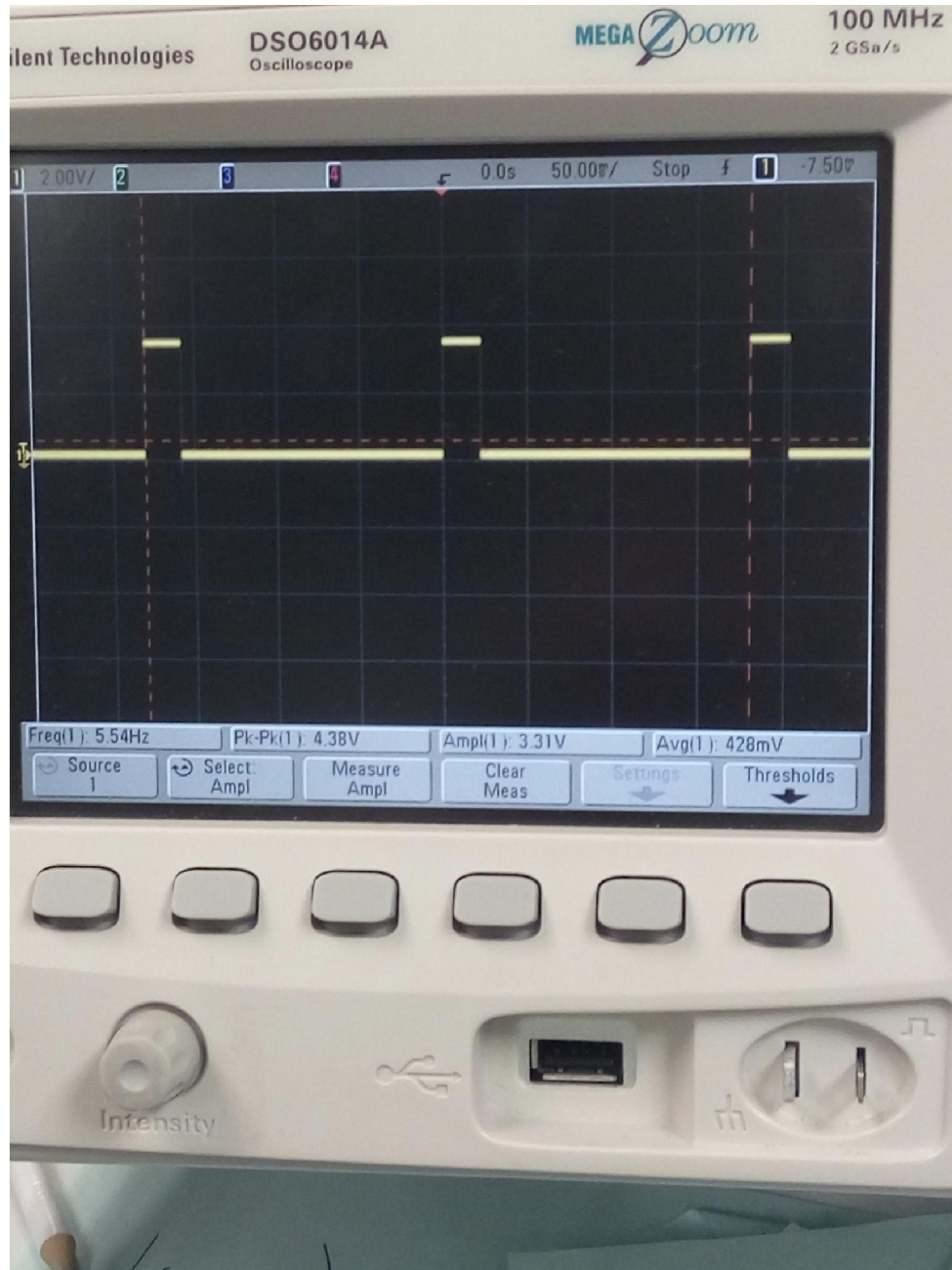


En la siguiente imagen se muestra el soporte más la tarjeta con dos condensadores de 220 microFaradios conectados entre el positivo y la masa de los servos; estos condensadores son indispensables para estabilizar la tensión en los servos en todo momento, también en la fase de arranque de los mismos, durante la cual presentan fuertes picos de corriente.

En la parte inferior de la imagen se aprecian dos entradas seriales, conectadas a la USART1 y 2 de la tarjeta:



En la figura sucesiva se observan las señales PWMs generadas por la tarjeta y medidas en el osciloscopio:

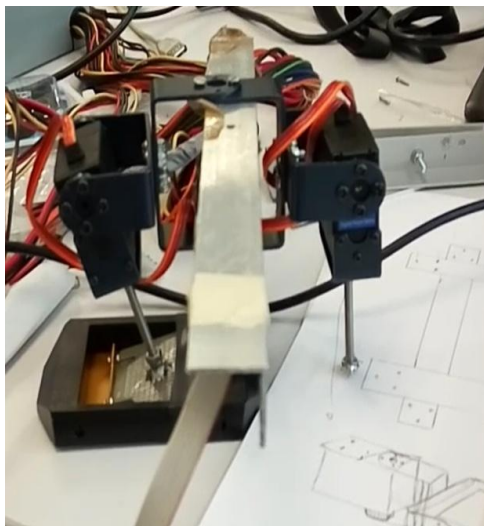


En realidad la frecuencia de la señal de control de los servos generada ha sido de 50Hz, correspondiente a 20ms de periodo, del cual se ha utilizado un pulso entre 0,56ms y 2,44ms de ancho, para los servos HSR-8498HB, así como se explicará más detalladamente en la sección dedicada a la programación.

PRUEBAS DE LEVANTAMIENTO

Después de haber analizado la estructura del robot y los componentes utilizados, se pasa a reconstruir cronológicamente las pruebas de funcionamiento efectuadas. Se evidencia que las pruebas no se han hecho después de la selección de todos los materiales, ni después de haber elegido la estructura del prototipo final del cuadrúpedo, sino paralelamente a estas dos fases y la fase del diseño final. Esto se debe a que, gracias a los resultados de las pruebas, se han analizado las distintas posibilidades e ideas de diseño, siempre buscando obtener los mejores resultados y analizando los pros y contras de cada solución.

Las primeras pruebas que se han hecho para simular el levantamiento del robot han sido finalizadas con motores HDKJ S3090D.



En estas pruebas, como se puede observar, se ha conseguido levantar un servo-motor mediante otro apoyado al suelo, soportado por una base rectangular de dimensiones: 10,2cm * 6,6cm.

En las primeras pruebas se ha intentado levantarlo con los dos motores en posición paralela entre sí y perpendicular a la mesa; el resultado ha sido un movimiento opuesto al deseado (base de apoyo deslizada hacia atrás, con las dos extremidades aún en contacto con la mesa). En las sucesivas pruebas se ha dado al servo-motor encargado del movimiento de levantamiento un ángulo no nulo con respecto a la vertical, de manera que el baricentro total fuese desplazado más hacia atrás, haciendo resultar la parte delantera más ligera con respecto a la trasera y permitiendo entonces

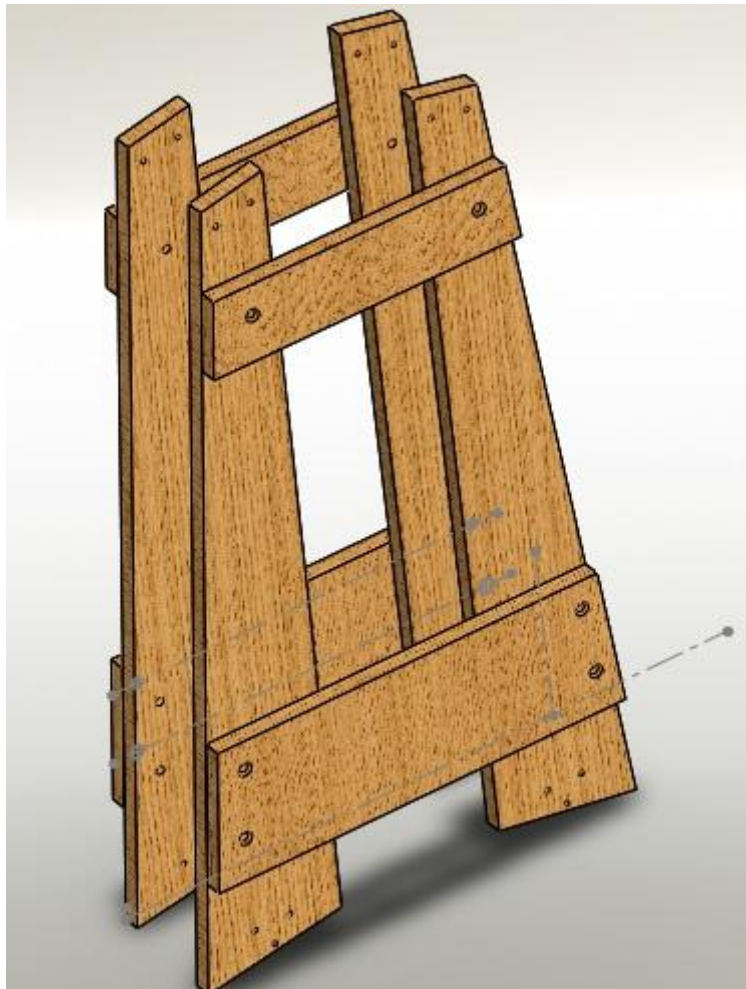
su levantamiento; las dimensiones de la base de apoyo han resultado ser las correctas para este caso, ya que con una superficie de apoyo inferior no se ha conseguido alcanzar una estabilidad suficiente para llevar a cabo el movimiento completo.

Estas pruebas iniciales han sido muy útiles para determinar algunos parámetros clave en el proceso de levantamiento del robot. Es evidente que las dimensiones de la estructura del robot van a ser mucho mayores que estas, sin embargo se ha considerado que la componente mayor del peso, situada en la parte anterior del cuadrúpedo, sea la de los servo-motores y que la única diferencia principal sea la distancia de estos con respecto al perno, la cual generará un momento proporcional a esa distancia que se opondrá al movimiento de levantamiento.

DISEÑO DEL ROBOT

El primer estudio sobre el levantamiento ha sido la clave para poder empezar el diseño final del robot, ya que ha dado una idea de la fuerza efectiva de los servos, relacionados con sus propio peso, y también indicaciones muy útiles sobre los movimientos y la superficie de apoyo necesaria para llevar a cabo la acción de levantamiento y el sucesivo estacionamiento en posición vertical.

Como ya se ha indicado en el apartado de la estructura, el modelo creado ha sido el siguiente:



Donde se pueden observar las dos capas, la superior y la inferior del cuadrúpedo, que constituyen la estructura completa, con las características siguientes:

Propiedades de masa de Ensamblaje3		
Configuración: Predeterminado		
Sistema de coordenadas: -- predeterminado --		
Masa = 283.59 gramos		
Volumen = 506409.97 milímetros cúbicos		
Área de superficie = 176602.81 milímetros cuadrados		
Centro de masa: (milímetros)		
X = 67.12		
Y = 105.38		
Z = 244.16		
Ejes principales de inercia y momentos principales de inercia: (gramos * milímetros cuadrados)		
Medido desde el centro de masa.		
Ix = (-0.08, 1.00, 0.00)	Px = 890698.72	
Iy = (0.04, 0.00, 1.00)	Py = 2253833.77	
Iz = (1.00, 0.08, -0.04)	Pz = 2800643.90	
Momentos de inercia: (gramos * milímetros cuadrados)		
Obtenidos en el centro de masa y alineados con el sistema de coordenadas de resultados.		
Lxx = 2788100.53	Lxy = -148891.34	Lxz = 21513.03
Lyx = -148891.34	Lyy = 902383.26	Lyz = 2776.85
Lzx = 21513.03	Lzy = 2776.85	Lzz = 2254692.58
Momentos de inercia: (gramos * milímetros cuadrados)		
Medido desde el sistema de coordenadas de salida.		
lxx = 22842909.03	lxy = 1857170.68	lxz = 4669231.59
lyx = 1857170.68	lyy = 19085526.49	lyz = 7299520.97
lzx = 4669231.59	lzy = 7299520.97	lzz = 6681911.82

Resultando la masa total de la estructura de 283,59g.

Una vez obtenida una idea mucho más clara también sobre lo que va a ser la forma del robot y la disposición de los servos, se ha decidido, en primer lugar, pasar de los motores HDKJ S3090D a los más precisos HSR-8498HB, ya que en las propias pruebas de levantamiento se ha podido comprobar la poca sensibilidad y suavidad en el movimiento de los HDKJ S3090D, además de una mayor dificultad en encontrar el rango de trabajo adecuado; la suavidad y la precisión del movimiento son de vital importancia en este tipo de contexto, a pesar de una fuerza, a paridad de tensión,

inferior en los nuevos motores:

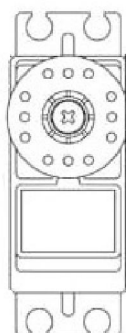
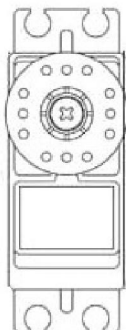
	6 Voltios	7,4 Voltios
HDKJ S3090D	9Kg.cm	10Kg.cm
HSR-8498HB	6,7Kg.cm	8,1Kg.cm

Este factor, añadido a la masa total del robot y al momento mucho más grande generado por los servo motores en los extremos anteriores, ha condicionado algunas decisiones en el diseño del robot, tal como será presentado sucesivamente.

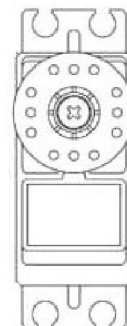
- *Disposición de los servos*

Volviendo al modelo básico de la estructura, como resultado de todos los estudios anteriores, se han dispuesto los servos de la siguiente forma:

Patas posteriores



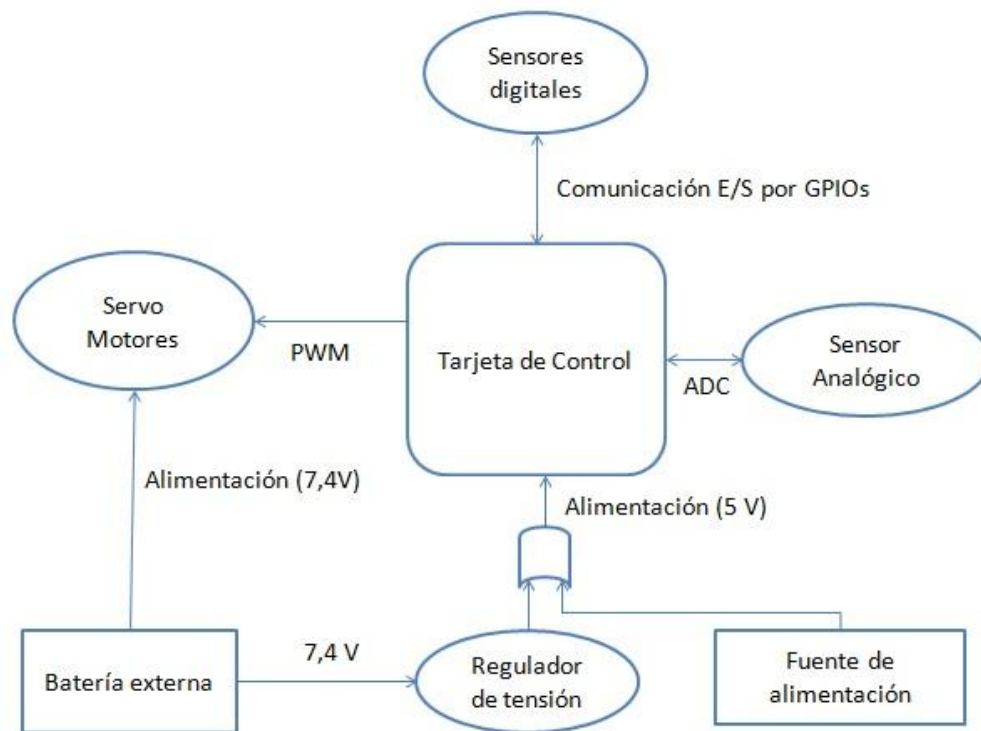
Patas anteriores



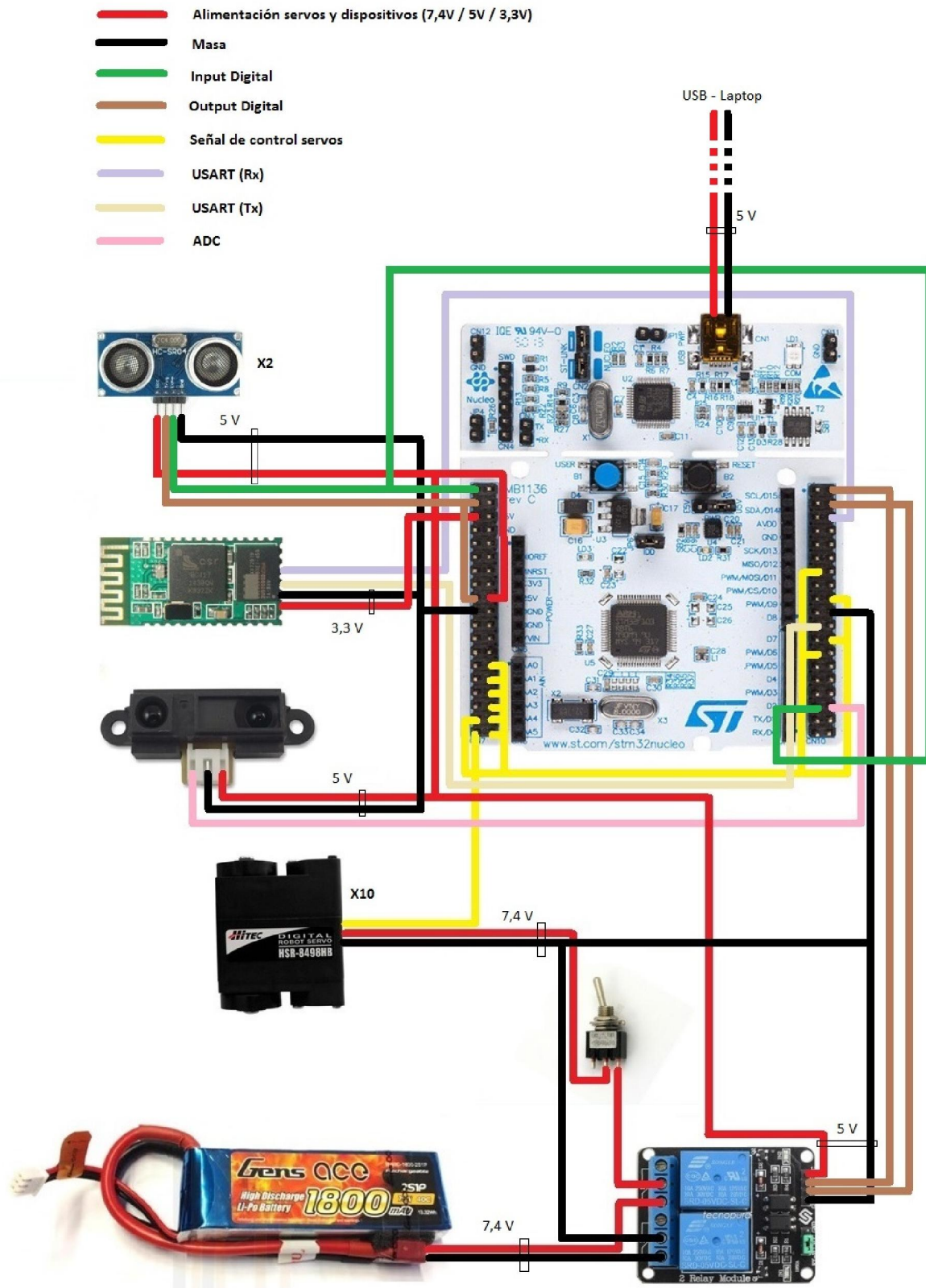
La disposición de los servo motores en la imagen anterior permite al robot funcionar como un cuadrúpedo cualquiera, donde el servo superior se ocupa de desplazar la pata paralelamente al suelo y el servo inferior de levantarla para permitir el movimiento de desplazamiento; además de esto, los dos servo motores en línea de las patas posteriores, permiten al robot andar durante el funcionamiento como bípedo, es decir levantar las dos piernas y completar el paso entero sin desplazamiento lateral alguno.

- *Esquema de funcionamiento y eléctrico del robot*

A continuación se observa el esquema de funcionamiento del robot:



En la imagen siguiente se representa el esquema eléctrico con los componentes reales:



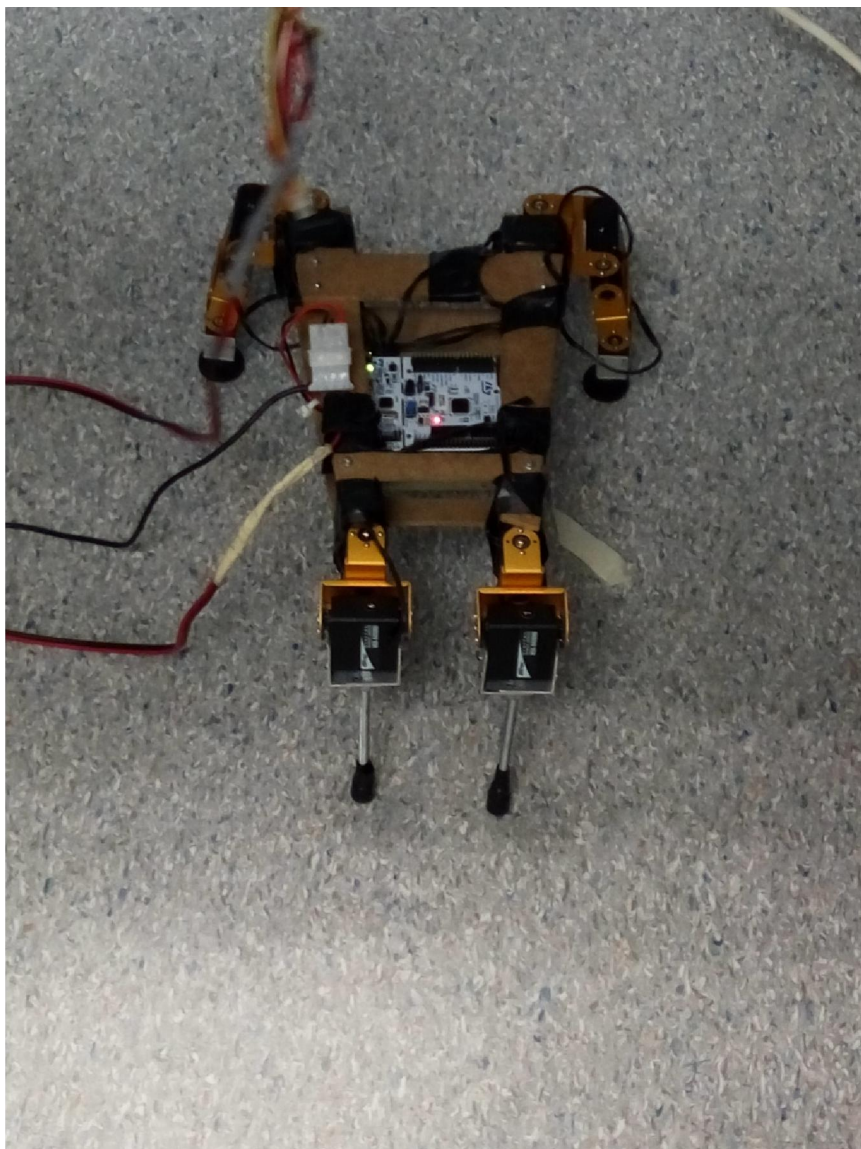
En el esquema de funcionamiento del robot se aprecian las varias interacciones entre los componentes que forman el sistema; a partir de la batería, que suministra con una buena condición de carga los 7,4 voltios necesarios para alimentar los servo motores, seguida por los relés para el control electrónico de la alimentación y un interruptor para el control manual. En la leyenda se aprecian con distintos colores los diferentes tipos de señales y en rojo y en negro respectivamente la señal de alimentación y la masa; hay que evidenciar también que con el color rojo se representan todas las distintas señales de alimentación, no solo la de la batería de 7,4 voltios, si no también las de 5 voltios y 3,3 voltios generadas por la tarjeta, para alimentar sensores y modulo Bluetooth. Para la alimentación de la tarjeta se ha utilizado en primer lugar una salida USB de ordenador portátil, así como indicado en el esquema eléctrico, sucesivamente se ha estudiado la posibilidad de alimentar la tarjeta directamente por la batería Lipo a través de un regulador de tensión, que convierte una tensión de entrada de 7,4 voltios (mediamente) en una de 5 voltios de salida, siendo este el voltaje necesario para alimentar la tarjeta Nucleo STM32.

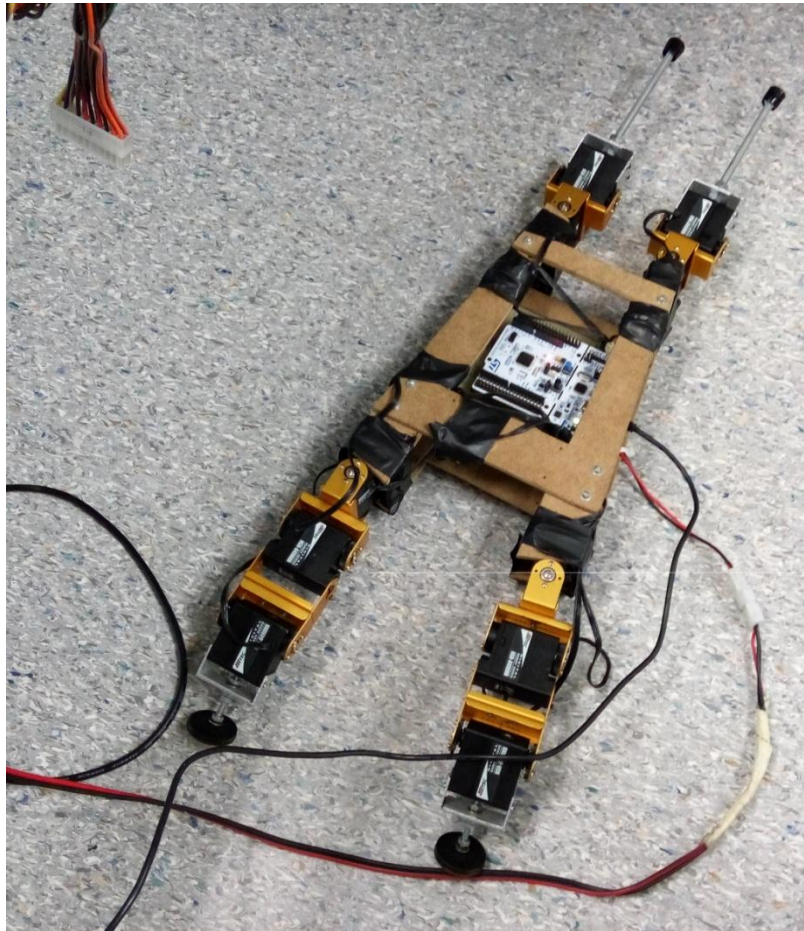
Por último se evidencia que para simplificar la visualización y hacer el esquema eléctrico más compacto se ha preferido utilizar un único componente por cada tipo de elemento (como por ejemplo en el caso de los servo motores), precisando a lado el número de elementos presentes en el montaje real.

PROTOTIPO BASICO

En este capítulo se observa el aspecto del prototipo básico del cuadrúpedo sin sensores, sin batería y sin las modificaciones y las medidas adoptadas sucesivamente a partir de este modelo de base, con los estudios siguientes.

A seguir se presentan las pruebas de movimiento y levantamiento finalizados sobre este prototipo.



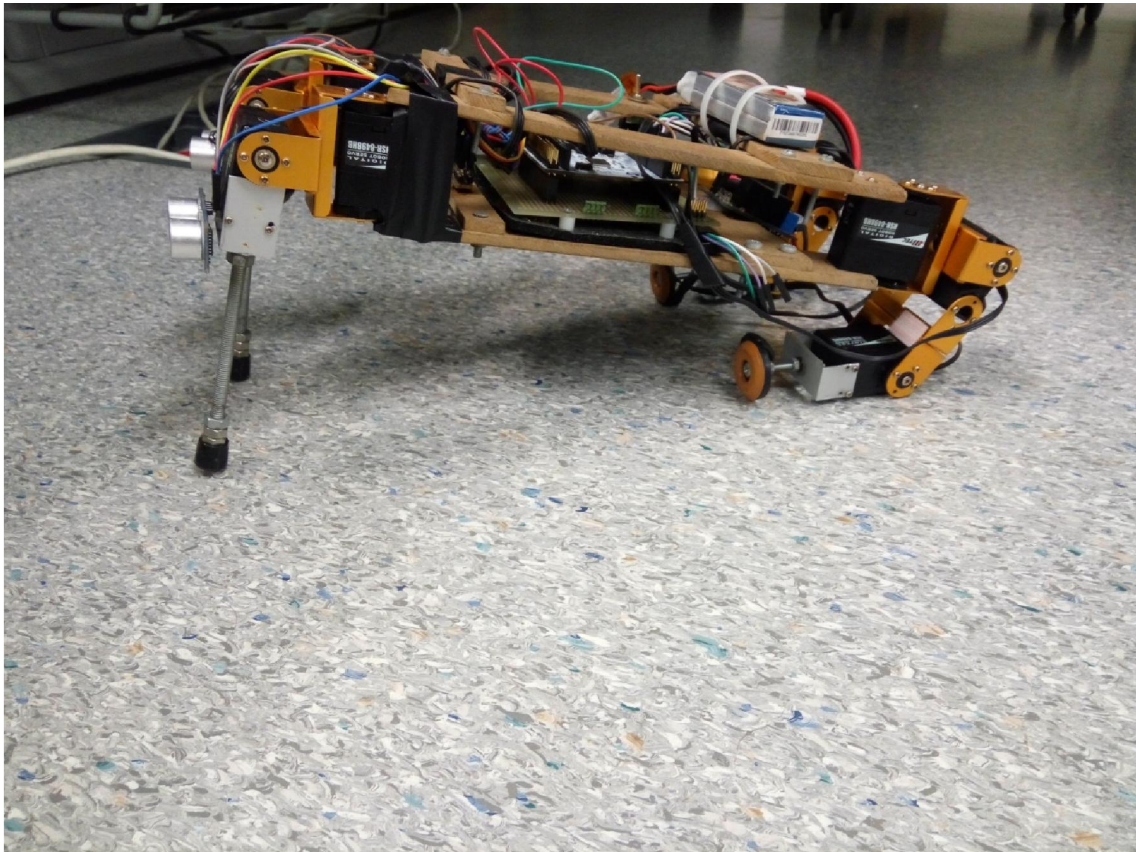


El modelo básico visto en las imágenes anteriores era capaz de andar como cuadrúpedo, moviéndose con discreta velocidad y precisión aunque los movimientos han sido mejorados modificando las partes mecánicas e implementando un código C de alto nivel, que permitiese una programación más abstracta y sencilla y con funciones capaces de generar distintos modos y velocidades de funcionamiento del robot, así como se analizará en detalle en la sección de programación.

El paso sucesivo ha sido el de buscar una solución para poder llevar el cuadrúpedo a una posición vertical de forma estable, librando las dos patas delanteras de la función de apoyo en el suelo. Esto permite utilizar los cuatro servos delanteros para otras funciones, como por ejemplo la de coger y transportar piezas, interactuar con una superficie elevada con respecto a la del suelo, medir distancias con distintos ángulos y en distintas direcciones (instalando sensores de distancia encima de dichas patas), sacar fotos, grabar vídeos en todas las posiciones, etc.

- *Posición vertical (posición perro sentado)*

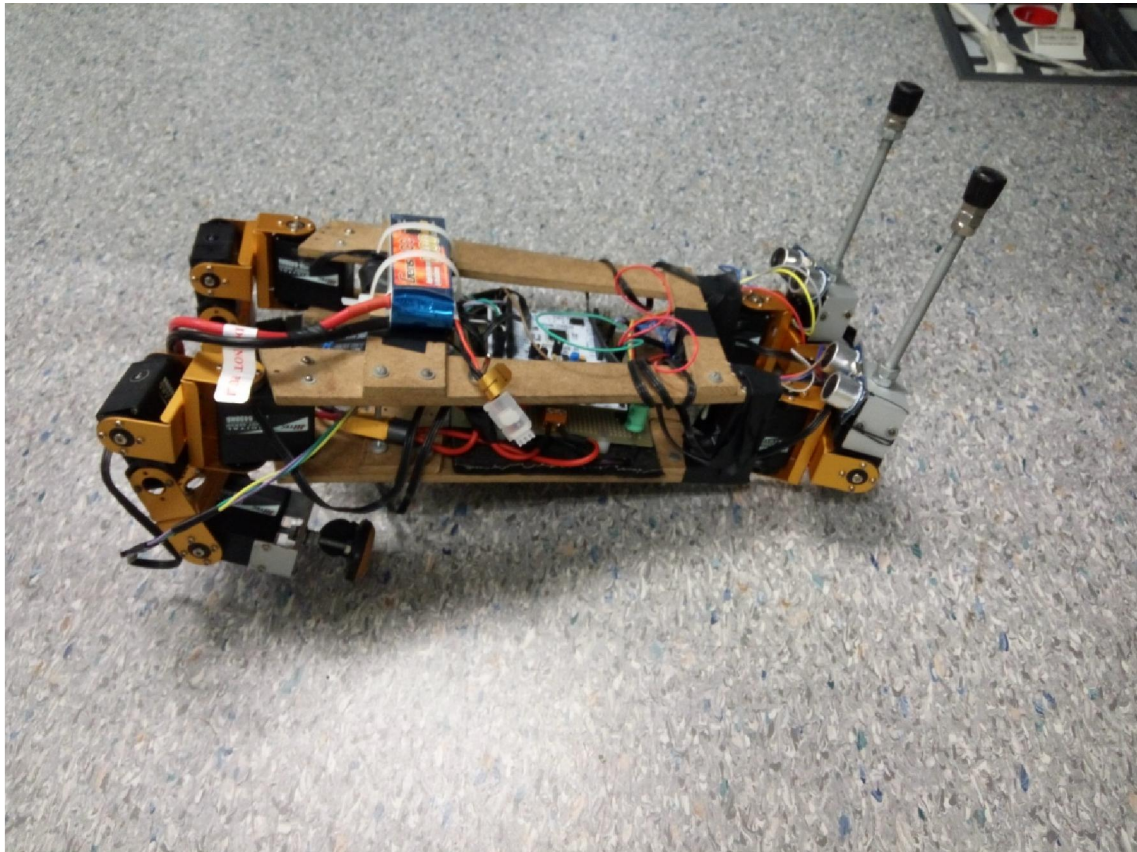
Las primeras ideas de movimiento para obtener la posición vertical estable del cuadrúpedo veían como punto de partida una posición contraída de las patas posteriores, casi simulando la posición de un perro sentado.



A partir de esta posición, se ha intentado lograr la posición vertical a través de varias secuencias posibles de movimiento de los servos; la idea de fondo era la de empujar con los servos del medio y de abajo, de una forma lo más sincronizada posible, de manera que la fuerza total generada y el momento correspondiente fuesen los más elevados posible.

La idea de fondo era la de hacer levantar la parte anterior del robot, mediante un empuje de los servo motores en contacto con el suelo.

Estas pruebas no han dado un resultado positivo, porque a pesar de que la fuerza de los servos ha resultado suficiente para este movimiento y de que se ha logrado una buena sincronización entre ellos, el movimiento obtenido ha sido, en todas las pruebas, opuesto al deseado, llevando el robot a caerse hacia su parte anterior.



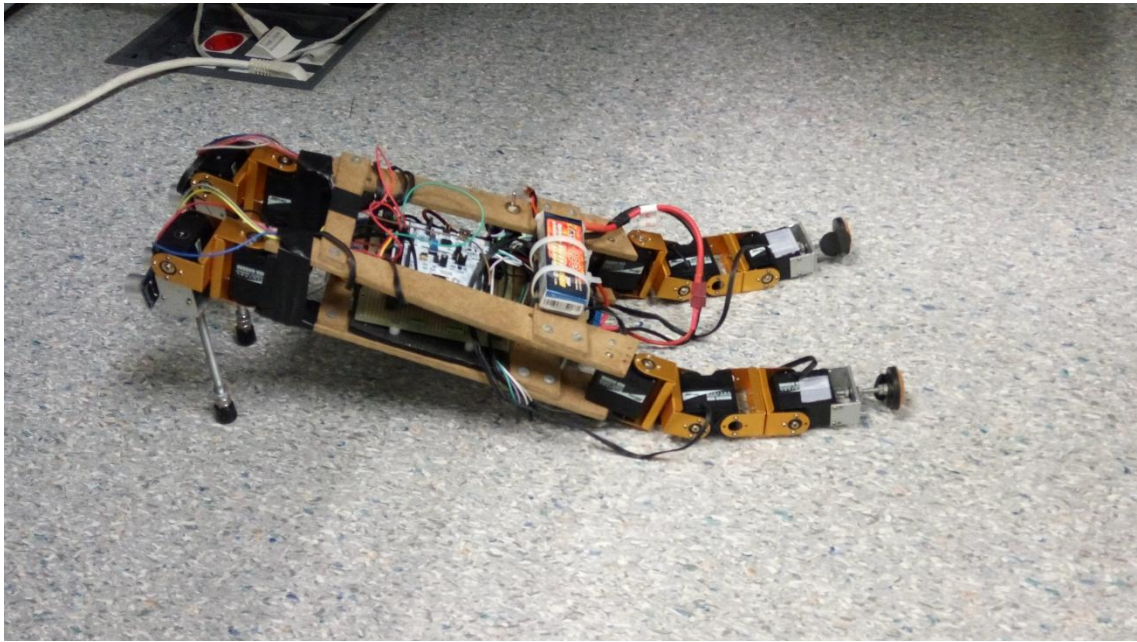
Claramente, esto es debido a que el baricentro del cuadrúpedo en esta posición está evidentemente desplazado hacia adelante con respecto al punto de contacto de los servos posteriores; este punto de contacto además no está muy bien definido porque teóricamente se extiende a toda la superficie de los servos de abajo y del medio de las patas posteriores.

La conclusión de este estudio ha sido que esta posición (parecida a la de un perro sentado) quizás no es la apropiada para este tipo de movimiento en este diseño concreto de cuadrúpedo.

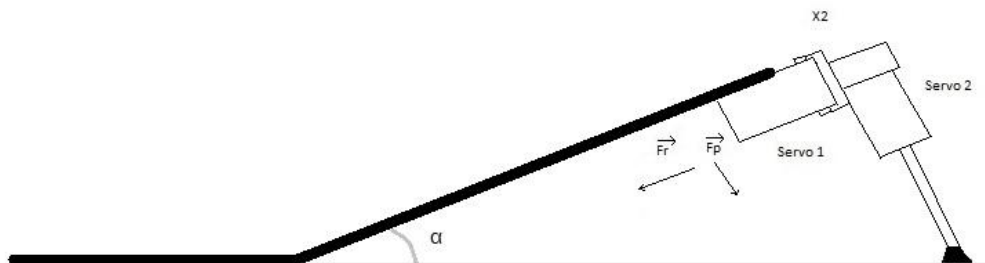
- *Posición vertical (posición extendida)*

Después de haber descartado la posición de perro sentado como posición de partida para el movimiento de levantamiento del cuadrúpedo, se ha intentado buscar una posición inicial que permitiese solucionar el problema de la posición del baricentro encontrado en las pruebas anteriores.

Para ello, se han finalizado estudios dinámicos sobre el robot en una multitud de posiciones y movimientos; entre estos se han obtenido resultados interesantes en una posición definida sucesivamente como posición extendida.



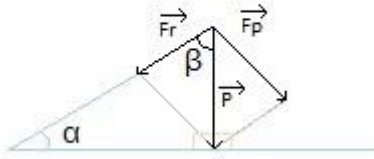
Sucesivamente se analiza el estudio dinámico realizado para esta posición:



El peso de cada servo, desde la hoja de características del fabricante es $P_s = 54,7g$.

En la extremidad delantera del perfil del robot se ha representado una pata (compuesta por los 2 servos), que genera la mayor parte del momento opuesto al de levantamiento a lo largo del perfil del robot.

En la imagen se ha representado con F_r la fuerza radial y con F_p la fuerza perpendicular al perfil inclinado del robot, que son la descomposición de la fuerza P .



Siendo la fuerza P la suma de la fuerza peso de la pata y del soporte por dos, ya que las patas delanteras son dos.

$$P = 4 \text{ servos} + \text{soportes} = 4 * 54,7 + \text{soportes} \approx 4 * 54,7 + 30 = 248,4 \approx 250g = 2,45N$$

Considerando el ángulo de inclinación del cuadrúpedo como $\alpha \approx 20^\circ$, se obtiene que:

$$\beta = 180^\circ - 90^\circ - \alpha \approx 70^\circ$$

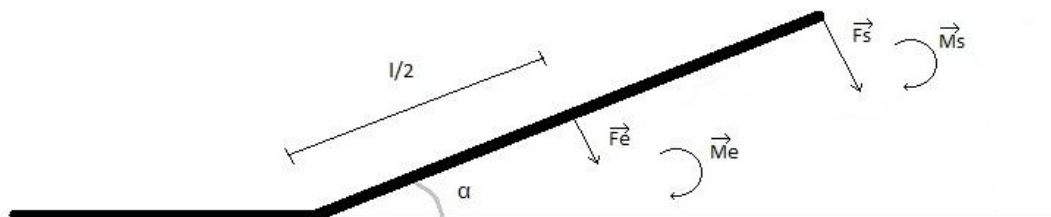
La componente radial es:

$$Fr = P * \cos \beta \approx 250 * \cos 70^\circ = 85,51g \approx 85,5g$$

La fuerza perpendicular a la longitud de la estructura del robot resultante de los servos y los soportes será:

$$Fp = P - Fr = 250 - 85,5 = 164,5g \approx 1,6N$$

Teniendo en cuenta también la fuerza peso generada por la misma estructura, representada con F_e y considerándola aplicada mediamente en el centro de la longitud de la parte levantada del suelo, se puede simplificar el diagrama de fuerzas de la siguiente forma:



Donde M_s y M_e son los momentos generados por las fuerzas F_s y F_e , siendo F_s (fuerza peso servos) la fuerza perpendicular total calculada anteriormente (F_p).

Considerando que la masa de la estructura, calculada por el software SolidWorks ha resultado ser de 283,59g, que la de la batería es de 111,9 g y que la de la suma de los otros componentes se ha estimado aproximadamente alrededor del valor 70g, incluyendo la tarjeta de control, todo el cableado, el soporte de la tarjeta y los sensores, se alcanza un peso total de 465,49g.

La fuerza F_e será entonces:

$$F_e = 465,49 * \cos \alpha = 465,49 * \cos 20^\circ \approx 437,42g$$

Siendo la longitud de la parte levantada de $\approx 29,5$ cm, aproximando esta longitud a 30cm para facilitar los cálculos, ya que con un valor ligeramente superior el caso es ligeramente más desfavorable (se prefiere un caso ligeramente más desfavorable a uno más favorable del de la realidad), el momento total generado por los servos delanteros con sus soportes será:

$$M_s = F_s * l = F_p * l = 1,6 * 0,3 = 0,48Nm$$

El momento resultante de los dos momentos generados por las dos fuerzas concentradas en esos dos puntos será:

$$M_{tot} = M_s + M_e = 0,48 + F_e * 0,3/2$$

$$\text{Siendo } F_e = 437,42g \approx 4,28N \Rightarrow M_{tot} = 0,48 + 4,28 * 0,3/2 = 0,48 + 0,642 = 1,122Nm$$

Que se puede aproximar por exceso a 1,13Nm.

El momento necesario, pues, para vencer el momento total generado por la parte delantera del robot tiene que ser $> M_{tot} = 1,13Nm$.

Recordando las características de los servos utilizados:

Par generado: 6,7Kg*cm con 6 Voltios de alimentación, 8,1Kg*cm con 7,4 Voltios

Lo cual significa que cada servo es capaz de generar:

- 6,7 Kg*cm $\approx$ 0,65 Nm por 6 voltios de alimentación
- 8,1 Kg*cm $\approx$ 0,79 Nm por 7,4 Voltios de alimentación

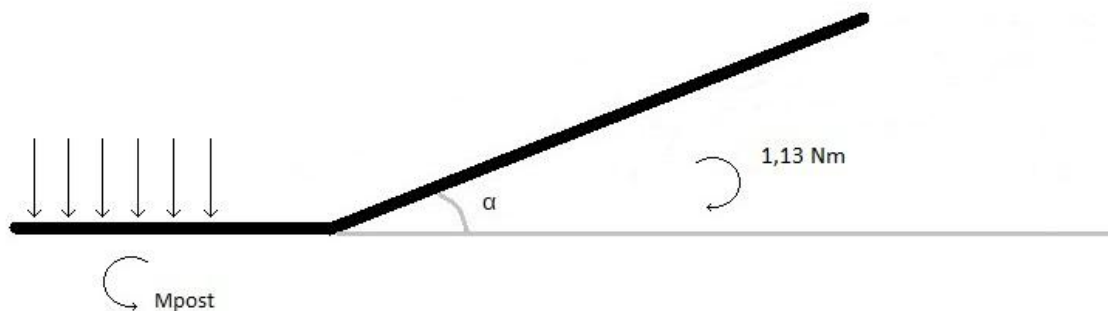
Considerando que al principio, en las primeras pruebas, en esta posición, se estaba utilizando una fuente de alimentación de 5 voltios y que solo en las pruebas sucesivas se ha utilizado una batería LIPO de dos celdas, capaz de alimentar los servos con una tensión comprendida entre 8,4 voltios y 6 voltios, se puede entender claramente que el par generado por los servos resulta insuficiente para generar un momento capaz de contrarrestar los 1,13Nm.

Se evidencia también que por mucho que el movimiento de los servos parezca sincrónico, en realidad esto no puede ser realmente así, es decir, la fuerza total generada por los 4 servos anteriores no es la fuerza generada por uno de ellos multiplicada por cuatro, ni mucho menos.

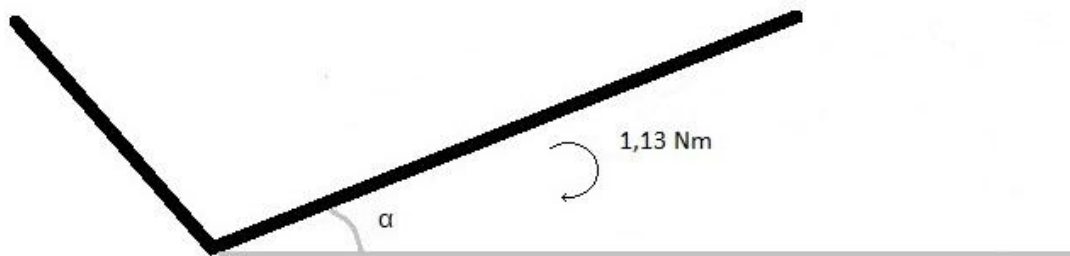
Esto se ha confirmado durante las pruebas, donde poniendo unas masas encima de las patas posteriores se ha conseguido levantar el robot con dificultad.

Además de esto, el problema principal ha sido otro:

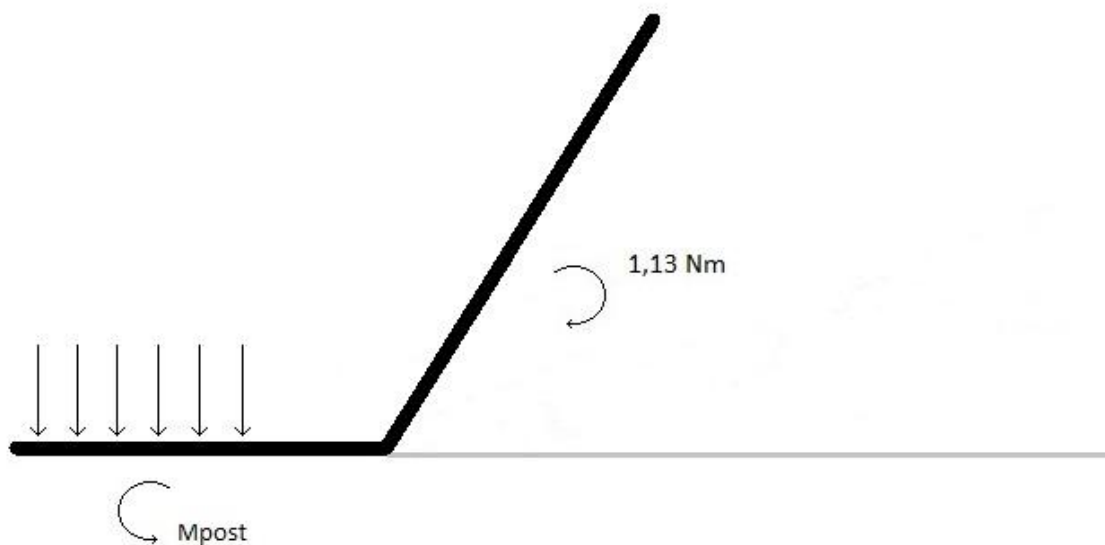
Para generar un momento tan grande es necesario tener una masa equivalente en la parte posterior apoyada al suelo capaz de generar un momento opuesto, de manera que los servos levanten la parte anterior y no la posterior.



Si el momento M_{post} generado en la parte posterior es $< 1,13 \text{ Nm}$:



Sin embargo, si el momento M_{post} , generado en la parte posterior es $> 1,13 \text{ Nm}$:



El problema principal de todo esto es que el peso de los 6 servos posteriores, más los soportes es:

$$P = 6 \text{ servos} + \text{soportes} = 6 \cdot 54,7 + \text{soportes} \approx 6 \cdot 54,7 + 50 = 378,2 \approx 378\text{g} \approx 3,71\text{N}$$

Considerando este peso concentrado en el medio de la base de apoyo posterior formada por los 6 servos, es decir a 8 centímetros de distancia de la parte levantada, el momento generado por esta fuerza es:

$$M_{\text{post}} = 3,71 * 0,08 \approx 0,3\text{Nm}$$

Lo cual significa que, además, se necesitan más de otros 0,8Nm para vencer el momento anterior y hacer levantar la parte anterior en lugar de la posterior del robot. Para conseguir este momento en la parte posterior sería necesario añadir unas masas en el punto central (a 8 centímetros del perno) cuyo peso tiene que ser superior a:

$$P_{\text{masas}} > 0,8/0,08 = 10\text{N} \approx 1020\text{g} = 1,02\text{Kg}$$

Este peso es demasiado elevado y añadir 510 gramos de masa en cada pata posterior comprometería el normal funcionamiento del robot en la condición de cuadrúpedo, además de la de bípedo, es más, estas masas ocuparían un volumen demasiado grande para el espacio disponible.

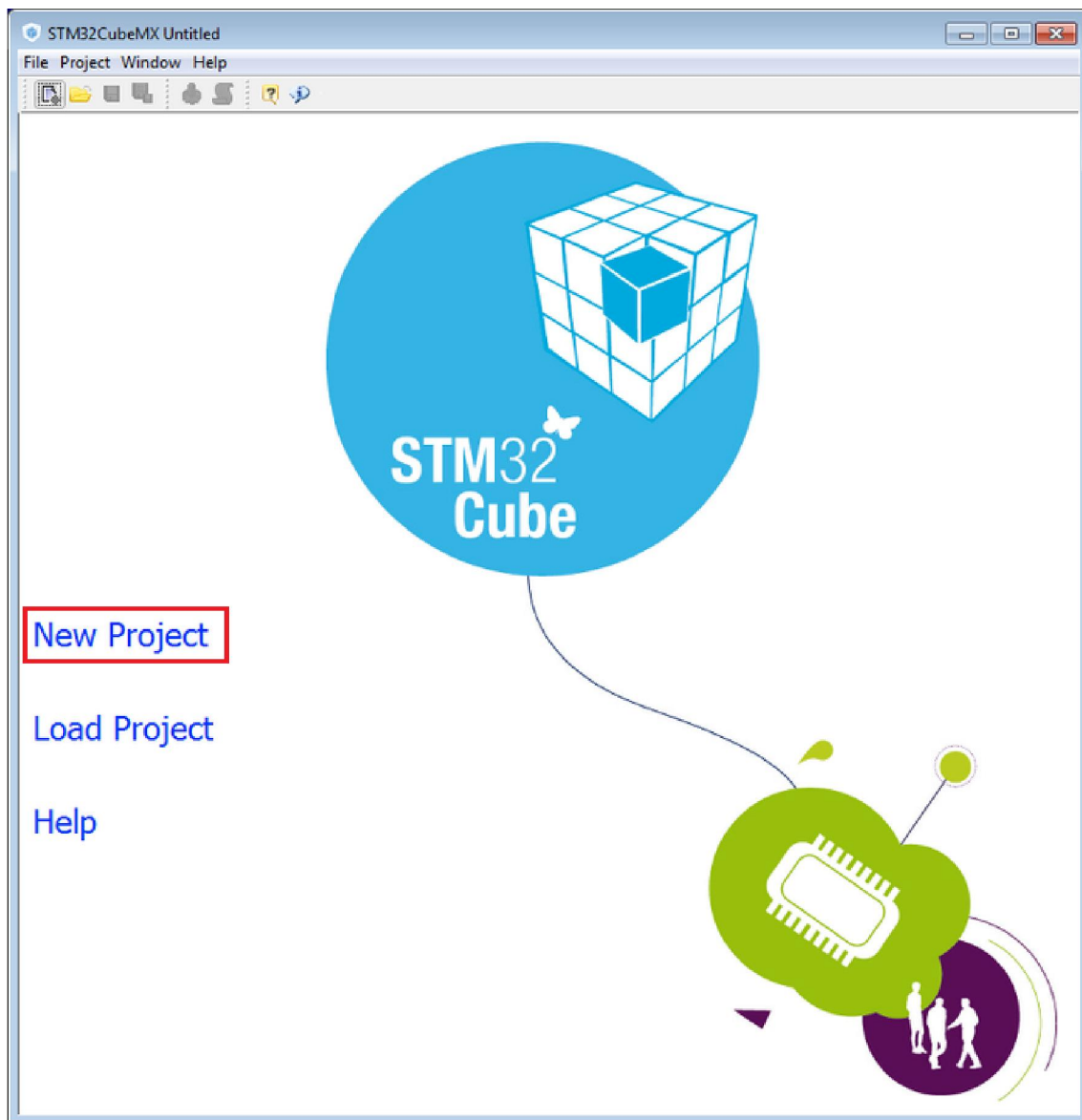
En conclusión, para el prototipo final se necesita encontrar una solución distinta a las estudiadas, que permitiese reducir de alguna forma el momento generado por la parte anterior del robot y de valorizar más la fuerza generada por los servos de las patas posteriores, ya que la parte anterior resulta mucho más larga que la posterior, lo cual implica que para generar los mismos momentos se necesitan fuerzas mucho más grandes.

PROGRAMACIÓN

- *Configuración de la tarjeta*

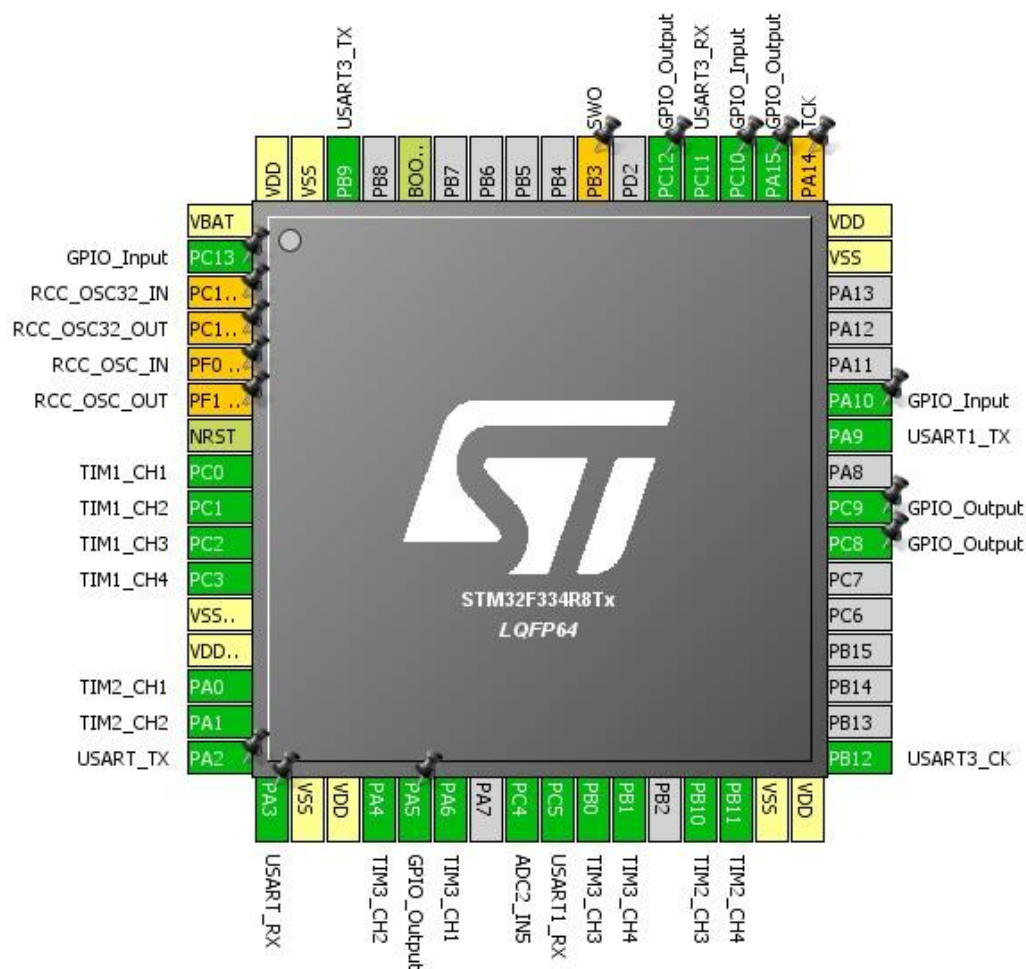
La parte de programación se puede considerar como una de las partes más importantes de todo el proyecto, ya que es necesaria para la configuración de la tarjeta, la creación de funciones que permitiesen elevar el nivel de programación a un nivel más abstracto y lejano al más básico lenguaje máquina (cada vez más importante conforme crece la complejidad del proyecto) y para la creación de una máquina de estados encargada de hacer trabajar el robot de manera cíclica y en distintos modos de funcionamiento.

En la primera parte, la configuración de la tarjeta, se ha utilizado un software proporcionado por STM: STM32CubeMX



Con este programa ha sido posible configurar la tarjeta STM32F334R8 Núcleo de principio a fin, sin escribir ni una sola línea de código, de manera muy profesional y ordenada, parte por parte; en particular se han configurado todas las entradas-salidas digitales y analógicas de la tarjeta, para los timers necesarios para enviar las señales de control a los servos y para los puertos seriales USART necesarios para la comunicación con los sensores.

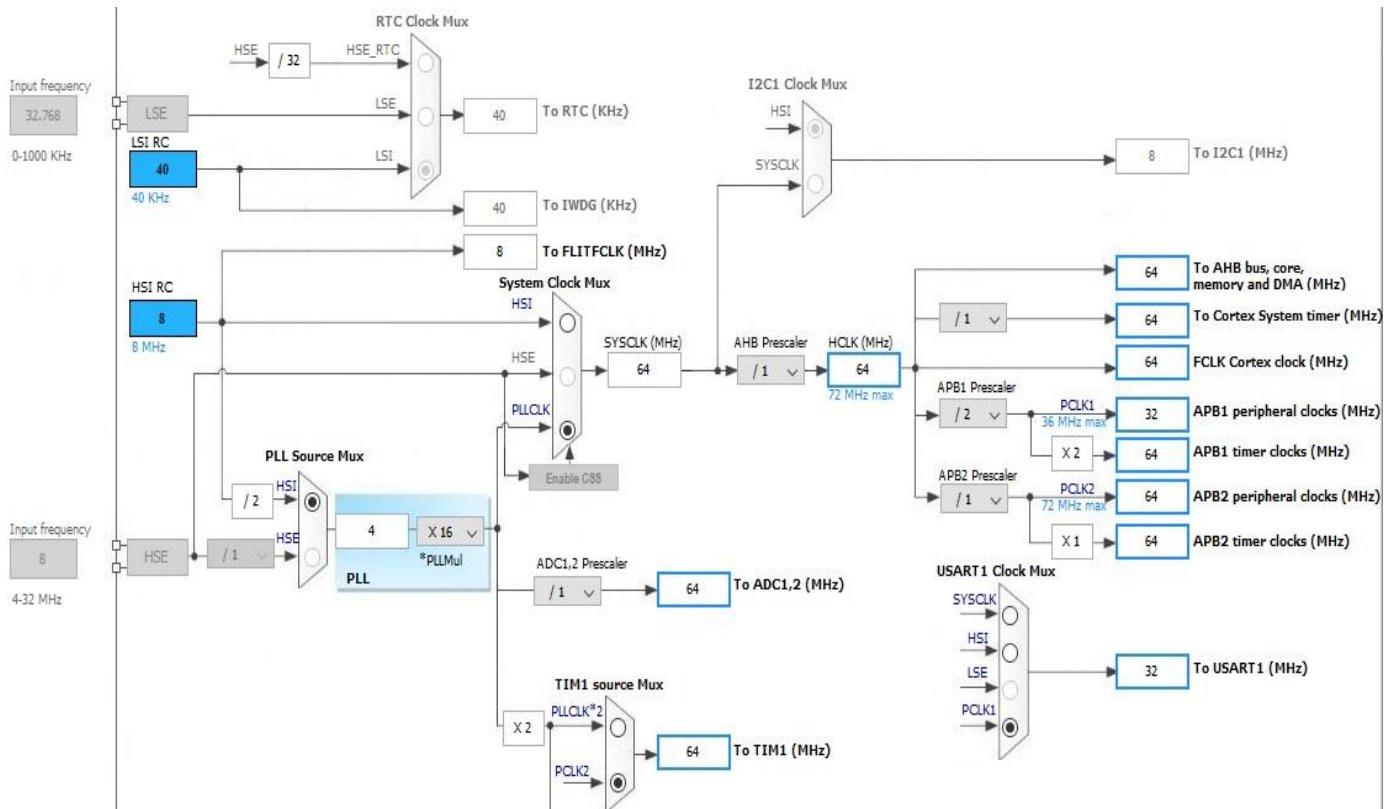
En la imagen siguiente se observa la configuración de los pines del micro-controlador:



El criterio principal utilizado en la configuración de los pines ha sido, en primer lugar, el de la posición física de los pines en la tarjeta; de hecho se ha intentado, en la fase de diseño, acercar lo más posible las salidas de los sensores a la posición en el robot de estos.

Con respecto a las salidas PWM, simplemente se han configurado los timers necesarios para el control de los servo motores, reservando un par más para posibles usos futuros, por ejemplo en otros servos adicionales, dejando en el momento de la configuración los pines considerados estándar por el programa. Todo esto porque, sucesivamente, en la fase de programación de las funciones de gestión de los timers, se ha asociado cada uno de estos timers con el servo-motor considerado más adecuado, haciendo innecesaria una elección preventiva de los pines.

En la imagen siguiente se observan las frecuencias ajustadas para el microcontrolador:



A continuación se visualizan los parámetros definidos para los timers:

Parameter Settings

User Constants

NVIC Settings

GPIO Settings

DMA Settings

Configure the below parameters :

Search :

Search (Ctrl+F)

Counter Settings

Prescaler (PSC - 16 bits value)

PRESCALER

Counter Mode

Up

Counter Period (AutoReload Register - 16 bits valu...

PWM_PERIOD - 1

Internal Clock Division (CKD)

No Division

Repetition Counter (RCR - 16 bits value)

0

Trigger Output (TRGO) Parameters

Master/Slave Mode

Disable (no sync between this TIM (Master) and its Slaves)

Trigger Event Selection TRGO

Reset (UG bit from TIMx_EGR)

Trigger Event Selection TRGO2

Reset (UG bit from TIMx_EGR)

Break And Dead Time management - BRK Configuration

BRK State

Disable

BRK Polarity

High

BRK Filter (4 bits value)

0

Break And Dead Time management - BRK2 Configuration

BRK2 State

Disable

BRK2 Polarity

High

BRK2 Filter (4 bits value)

0

Break And Dead Time management - Output Configuration

Automatic Output State

Disable

Off State Selection for Run Mode (OSSR)

Disable

Off State Selection for Idle Mode (OSSI)

Disable

Lock Configuration

Off

Clear Input

Clear Input Source

Disable

PWM Generation Channel 1

Mode

PWM mode 1

Pulse (16 bits value)

POS_INIZIALE

Fast Mode

Disable

CH Polarity

High

CH Idle State

Reset

PWM Generation Channel 2

Mode

PWM mode 1

Pulse (16 bits value)

POS_INIZIALE

Fast Mode

Disable

CH Polarity

High

CH Idle State

Reset

PWM Generation Channel 3

Mode

PWM mode 1

Pulse (16 bits value)

POS_INIZIALE

Fast Mode

Disable

CH Polarity

High

CH Idle State

Reset

PWM Generation Channel 4

Mode

PWM mode 1

Pulse (16 bits value)

POS_INIZIALE

Fast Mode

Disable

CH Polarity

High

53

Para las USART, de las cuales la USART1 y la USART2 han sido utilizadas en realidad como UART, mientras que la USART3 se ha configurado, efectivamente, como USART y ha sido utilizada para conectar la tarjeta a un ordenador con un monitor serie, para testear el correcto funcionamiento de los sensores de posición.

☒ Parameter Settings
 ☒ User Constants
 ☒ NVIC Settings
 ☒ GPIO Settings
 ☒ DMA Settings

Configure the below parameters :

Search :  

[-] Basic Parameters	
Baud Rate	9600 Bits/s
Word Length	7 Bits (including Parity)
Parity	None
Stop Bits	1
[-] Advanced Parameters	
Data Direction	Receive and Transmit
Over Sampling	16 Samples
Single Sample	Disable
[-] Advanced Features	
TX Pin Active Level Inversion	Disable
RX Pin Active Level Inversion	Disable
Data Inversion	Disable
TX and RX Pins Swapping	Disable
Overrun	Enable
DMA on RX Error	Enable
MSB First	Disable

- Programación mediante el software uVisión 5

Una vez configurada correctamente la tarjeta, se ha pasado a la fase de programación verdadera, utilizando el software Keil uVision 5.



Sucesivamente se expresarán los puntos clave del programa creado para el robot, dejando de lado las partes más básicas comunes a la mayoría de ellos.

El primer punto clave, tal como se ha mencionado anteriormente, ha sido la creación de unas funciones que permitiesen elevar el nivel de programación de los servos a un nivel más elevado, de manera que el uso de dichas funciones simplificase notablemente la programación en sí y permitiese alejarse de lo que son los aspectos más físicos de los servos.

```
//funcion basica para mover los servos
void set_pwm_width(TIM_HandleTypeDef *htim, uint32_t Channel, int pulsos) // pulsos de 1 a 399
{
    switch (Channel)
    {
        case TIM_CHANNEL_1: htim->Instance->CCR1 = pulsos; break;
        case TIM_CHANNEL_2: htim->Instance->CCR2 = pulsos; break;
        case TIM_CHANNEL_3: htim->Instance->CCR3 = pulsos; break;
        case TIM_CHANNEL_4: htim->Instance->CCR4 = pulsos; break;
    }
}
```

Función utilizada para asociar directamente un valor numérico a uno de los cuatro canales de cada timer.

```

//funciones para mover servos (movimiento en porcentaje)
void ServoTIM1_Ch1(uint8_t movimiento){

    set_pwm_width(&htim1, TIM_CHANNEL_1, ((movimiento/10.64)+2.8)*4);

}

void ServoTIM1_Ch2(uint8_t movimiento){

    set_pwm_width(&htim1, TIM_CHANNEL_2, ((movimiento/10.64)+2.8)*4);

}

void ServoTIM1_Ch3(uint8_t movimiento){

    set_pwm_width(&htim1, TIM_CHANNEL_3, ((movimiento/10.64)+2.8)*4);

}

void ServoTIM1_Ch4(uint8_t movimiento){

    set_pwm_width(&htim1, TIM_CHANNEL_4, ((movimiento/10.64)+2.8)*4);

}

void ServoTIM2_Ch1(uint8_t movimiento){

    set_pwm_width(&htim2, TIM_CHANNEL_1, ((movimiento/10.64)+2.8)*4);

}

void ServoTIM2_Ch2(uint8_t movimiento){

    set_pwm_width(&htim2, TIM_CHANNEL_2, ((movimiento/10.64)+2.8)*4);

}

void ServoTIM2_Ch3(uint8_t movimiento){

    set_pwm_width(&htim2, TIM_CHANNEL_3, ((movimiento/10.64)+2.8)*4);

}

void ServoTIM2_Ch4(uint8_t movimiento){

    set_pwm_width(&htim2, TIM_CHANNEL_4, ((movimiento/10.64)+2.8)*4);

}

void ServoTIM3_Ch1(uint8_t movimiento){

    set_pwm_width(&htim3, TIM_CHANNEL_1, ((movimiento/10.64)+2.8)*4);

}

void ServoTIM3_Ch2(uint8_t movimiento){

    set_pwm_width(&htim3, TIM_CHANNEL_2, ((movimiento/10.64)+2.8)*4);

}

void ServoTIM3_Ch3(uint8_t movimiento){

    set_pwm_width(&htim3, TIM_CHANNEL_3, ((movimiento/10.64)+2.8)*4);

}

void ServoTIM3_Ch4(uint8_t movimiento){

    set_pwm_width(&htim3, TIM_CHANNEL_4, ((movimiento/10.64)+2.8)*4);

}

```

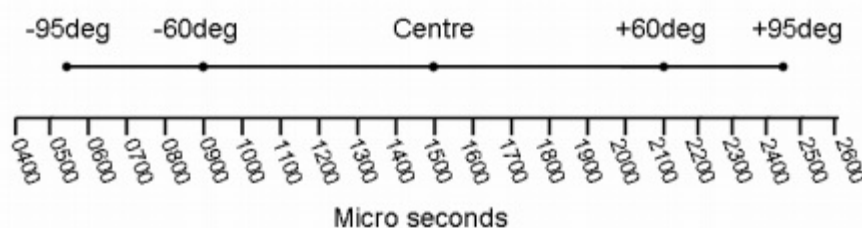
Estas funciones, como se puede observar, se apoyan directamente en la función anterior (*set_pwm_width()*) y su objetivo es el de permitir manejar un servo de tipo HSR-8498HB en todo su rango de funcionamiento, a través de un valor en porcentaje que se refiere a una posición específica entre sus extremos, donde el 0 corresponde a un extremo y el 100 al otro.

Estas funciones, entonces, permiten utilizar un servo HSR-8498HB en todo su rango, olvidando las especificaciones técnicas del servo y utilizando simplemente un valor entre 0 y 100, correspondientes a los dos límites en los cuales el servo es capaz de posicionarse.

El cálculo utilizado se basa en la siguiente lógica:

La frecuencia de funcionamiento de un servo de tipo HSR-8498HB es de 50 Hz, que corresponden a 20ms.

Con la función *set_pwm_width()* se envía al servo una señal de 50 Hz donde el duty cycle es nulo si se envía un 0 en el parámetro *int pulsos* y del 100% si se envía un 399 en el mismo parámetro; es decir, permite generar pulsos cuya anchura va de 0 a 20ms. En el caso específico de los servos HSR-8498HB, desde las especificaciones:



Los límites de funcionamiento definidos por el fabricante son 550 microsegundos y 2450 microsegundos.

Por razones de seguridad y de aproximación en los cálculos, se ha decidido utilizar como límites de referencia 560 microsegundos y 2440 microsegundos, obteniendo el siguiente rango de funcionamiento:

0,56ms -----> 2,44ms

Pudiendo generar con la función *set_pwm_width()* pulsos en el intervalo

0ms -----> 20ms

Mediante una regla de tres, se ha pasado del rango 0ms/20ms al rango 0,56ms/2,44ms de la siguiente forma:

$2,44 - 0,56 = 1,88\text{ms}$ (rango de trabajo del servo)

$$20/1,88 = 10,64$$

Llegados a este punto se ha desplazado el valor obtenido de 0,56ms, sumando 2,8.

Hay que precisar que el valor sumado ha tenido que ser 2,8 y no 0,56, porque en realidad el valor enviado a la función será en porcentaje, comprendido pues entre 0 y 100 y no entre 0 y 20, es decir, hay que desplazar el valor correspondiente en el rango 0/100, que es efectivamente 2,8 ($2,8 = 0,56 * 100/20$). Una vez convertido el rango de trabajo del servo desde los valores especificados por el fabricante por una frecuencia de 50Hz, correspondientes a 20ms de pulso máximo a un valor entre 0 y 100, se ha multiplicado por 4 obteniendo finalmente un valor entre 0 y 400, siendo estos los extremos requeridos por la función *set_pwm_width()*.

En conclusión, pasando un valor en porcentaje a las nuevas funciones *ServoTIMx_Chx()*, se hacen trabajar los servos HSR-8498HB en todo el rango de funcionamiento permitido, indicado por el fabricante.

Una vez creadas las funciones *ServoTIMx_Chx()*, se ha asociado cada pata del robot a la salida PWM correspondiente a uno de los canales de cada timer; se han buscado las salidas PWM más cómodas y cercanas para cada pata.

```
void PataTrasera_IzqAbajo(uint8_t movimiento){  
    ServoTIM3_Ch1(movimiento);  
}
```

Este paso es muy importante porque permite trabajar directamente con los nombres de las patas, según sus posiciones en el robot, olvidando los canales y los timers a los cuales están conectados los servos que se desean controlar.

Las funciones creadas, además de la del ejemplo, son: *PataTrasera_Izq/Der Abajo/Medio/Arriba* para las patas traseras, y *PataDelantera_Izq/Der Abajo/Arriba* para las patas delanteras.

Otro punto clave en la parte de la programación, llegados a un nivel de abstracción suficiente para tal propósito, ha sido la creación de un sistema encargado de regular y variar la velocidad de los servos con la ayuda de unas funciones de retardo aplicadas a las funciones de movimiento de los servos.

Esta parte de la programación se ha considerado clave porque ha sido implementada a partir de los resultados de los estudios iniciales sobre el movimiento de levantamiento. Una de las consideraciones hechas a partir de esos estudios ha sido que mover los servos a la máxima velocidad podía influir de forma negativa sobre buena parte de los movimientos de esa fase, condicionando irremediablemente el equilibrio y facilitando muchísimo las caídas.

Se ha buscado una solución para poder utilizar un tipo de movimiento en cualquier punto del programa, definido como suave a través de funciones de este tipo:

```
void PataTrasera_IzqAbajo_suave(uint8_t movimiento, uint8_t iniziale){  
  
    int i;  
  
    if (movimiento>iniziale){  
        for(i=iniziale;i<=movimiento;i++){  
            HAL_Delay(Ret_MovSuave+Ralentam);  
            PataTrasera_IzqAbajo(i);  
        }  
    }  
    if (movimiento<iniziale){  
        for(i=iniziale;i>=movimiento;i--){  
            HAL_Delay(Ret_MovSuave+Ralentam);  
            PataTrasera_IzqAbajo(i);  
        }  
    }  
}
```

Cuya velocidad depende de forma directa de una constante: `Ret_MovSuave` y de una variable: *Ralentam*

En la constante definida entre las constantes principales del programa se ha buscado un valor de retardo básico adecuado, de manera que utilizando las funciones de control de los servos con terminación “_suave” se obtenga un movimiento ligeramente más lento de lo normal; lo cual, como se ha indicado anteriormente, evita golpes y movimientos bruscos que puedan llevar a caídas.

Además, el uso de la variable *Ralentam* permite modificar la velocidad de los servos en cualquier punto del programa, ralentizándola de forma inversamente proporcional al valor de la variable. Esto es muy útil para conseguir distintas velocidades para cada movimiento, eligiendo la que se considera más oportuna en cada caso.

Para poder, efectivamente, utilizar funciones de tipo “_suave” se han creado variables que almacenan en cada momento la posición actual del servo, ya que el movimiento suave es entre dos puntos y se necesita entonces saber el punto de partida del movimiento (`uint8_t iniziale`).

Posteriormente, se ha decidido utilizar este tipo de funciones también para los modos de funcionamiento como cuadrúpedo, así como se ve en la imagen siguiente:

```

void arriba (uint8_t pata){

    switch (pata)
    {
        case TraseraIzq:
            PataTrasera_IzqAbajo_suave(48,PataTrasera_IzqAbajo_POS);
            PataTrasera_IzqAbajo_POS=48;
            PataTrasera_IzqMedio_suave(78,PataTrasera_IzqMedio_POS);
            PataTrasera_IzqMedio_POS =78;
            break;

        case TraseraDer:
            PataTrasera_DerAbajo_suave(50,PataTrasera_DerAbajo_POS);
            PataTrasera_DerAbajo_POS =50;
            PataTrasera_DerMedio_suave(83,PataTrasera_DerMedio_POS);
            PataTrasera_DerMedio_POS=83;
            break;

        case DelanteraIzq:
            PataDelantera_IzqAbajo_suave(18,PataDelantera_IzqAbajo_POS);
            PataDelantera_IzqAbajo_POS=18;
            break;

        case DelanteraDer:
            PataDelantera_DerAbajo_suave(14,PataDelantera_DerAbajo_POS);
            PataDelantera_DerAbajo_POS=14;
            break;

    }
}

```

En este trozo de código se aprecia la implementación de la función *void arriba(uint8_t pata)*, utilizada, junto con *void abajo(uint8_t pata)*, *void adelante(uint8_t pata)*, *void atras(uint8_t pata)*, para configurar todos los movimientos de los servos utilizados en el modo de funcionamiento como cuadrúpedo; como por ejemplo la función para el movimiento hacia adelante:

```

uint8_t Adelante(void){

    //paso1
    arriba(DelanteraIzq); //pata 2 arriba
    arriba(TraseraDer); //pata 3 arriba
    adelante (DelanteraIzq); //pata 2 adelante
    adelante(TraseraDer); //pata 3 adelante
    atras(DelanteraDer); //pata 4 atras
    atras(TraseraIzq); //pata 1 atras
    //posición inestable

    //paso2
    abajo(DelanteraIzq); //pata 2 abajo
    abajo(TraseraDer); //pata 3 abajo
    arriba(TraseraIzq); //pata 1 arriba
    //posición estable

    //paso 3
    arriba(DelanteraDer); //pata 4 arriba
    adelante(TraseraIzq); //pata 1 adelante
    adelante(DelanteraDer); //pata 4 adelante
    atras(DelanteraIzq); //pata 2 atras
    atras(TraseraDer); //pata 3 atras
    //posición inestable

    //paso 4
    abajo(DelanteraDer); //pata 4 abajo
    abajo(TraseraIzq); //pata 1 abajo
    arriba(TraseraDer); //pata 3 arriba
    //posición estable

}

```

Programadas de esta manera, las funciones de movimiento del robot son independientes también de los valores de posición de los servos, es decir, si se cambia un servo o la configuración de uno de ellos, estas funciones de movimiento se quedan igual, cambiarían simplemente los valores en las funciones utilizadas dentro de ellas (*arriba, abajo, adelante, atrás*).

El último punto clave de la programación es la creación de la función de levantamiento, con el soporte de todas las funciones de manejo de los servos observadas anteriormente. Esta función: *uint8_t Levanta (void)* en realidad se acompaña de otra función de levantamiento: *void Levantaapy (void)*, utilizada para estudiar el levantamiento del cuadrúpedo con el apoyo de una pared.

Las estructuras de estas funciones serán presentadas en el apartado relativo a los modos de funcionamiento.

- *Modos de funcionamiento*

En este apartado se estudian los modos de funcionamiento del robot en la máquina de estados creada:



Diagrama de funcionamiento

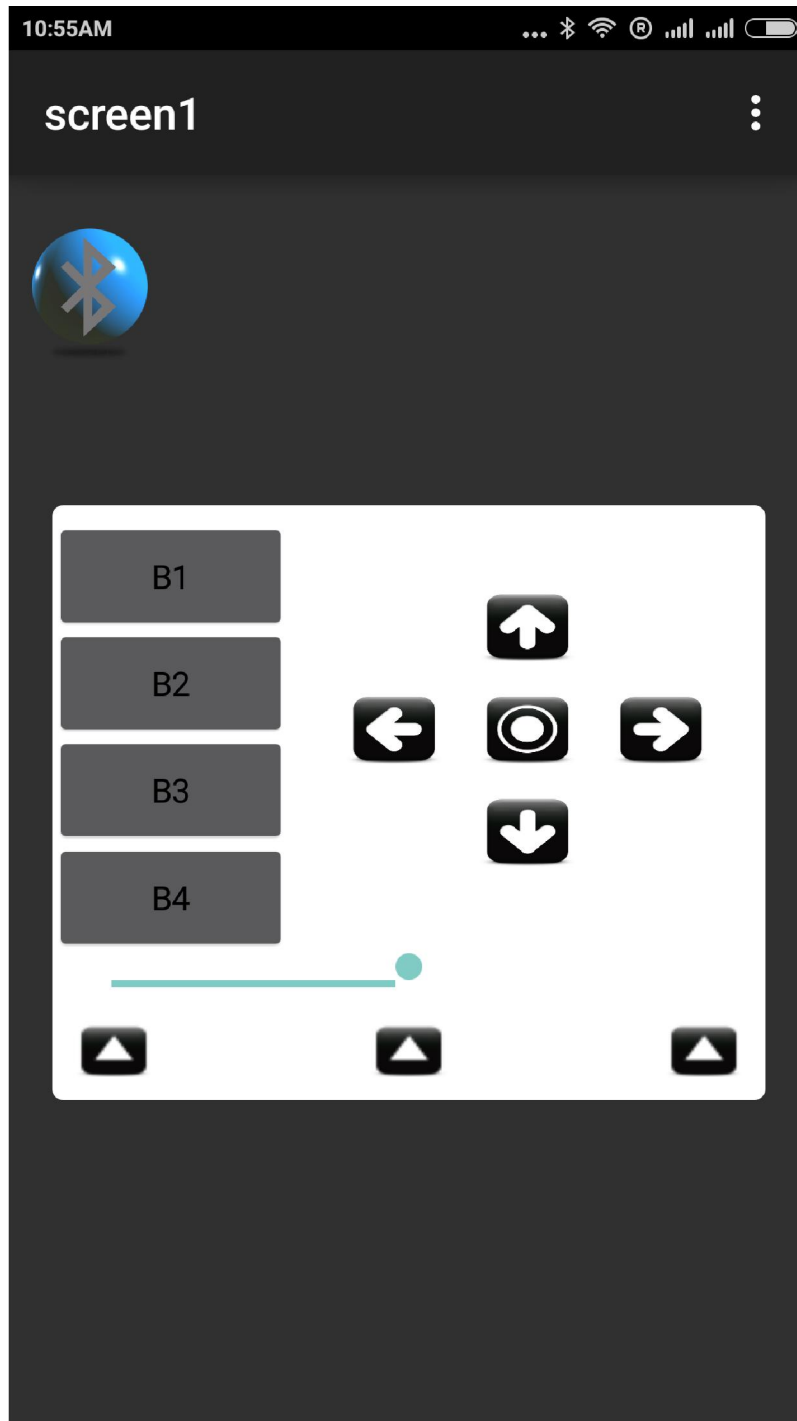
La máquina de estado que regula el funcionamiento del robot, como se aprecia en la imagen, se basa en una lectura cíclica del comando enviado por Bluetooth. Una vez leído el comando, si este es uno de los comandos reconocidos, el estado mutará a uno entre Adelante, Atrás, Derecha, Izquierda, Levanta, Parado o Adelapy; si el comando no es uno de los esperados o no se recibe ningún comando por Bluetooth, se vuelve a ejecutar el estado anterior.

Los estados típicos de un cuadrúpedo son claramente Adelante, Atrás, Derecha, Izquierda, y Parado, a los cuales se ha añadido el estado Levanta para pasar a la modalidad de Bípedo.

Otro modo de funcionamiento añadido sucesivamente es Adelapy (Adelante con apoyo); en esta modalidad el cuadrúpedo avanza paso a paso buscando una superficie vertical para poderse apoyar y levantarse. En cada ciclo de funcionamiento en esta

modalidad se mide la distancia a través de los dos sensores HC - SR04 y en el caso que esta distancia resulte inferior a un límite prefijado (6 cm en este caso) se pasa a ejecutar la función Levantaapy, encargada efectivamente de hacer que el robot se apoye con las patas anteriores al obstáculo o la superficie vertical encontrada y después que se levante, aprovechando también los servos anteriores.

El programa utilizado para la transmisión de las señales de comando Bluetooth es el siguiente:



Desde el cual se envían por las flechas los movimientos hacia adelante, hacia atrás, derecha e izquierda; con el botón del medio la posición Parado, mientras que con B1 y B2 los levantamientos: Levanta y Adelapy.

PROTOTIPO FINAL

El prototipo final es el resultado de todos los estudios y conclusiones obtenidas sobre los modelos anteriores, en todas las fases del proyecto: diseño, programación, pruebas, etc.

El objetivo principal del prototipo final es conseguir la posición vertical, buscando sucesivamente una secuencia de movimientos que permitiese hacer levantar el cuadrúpedo de forma completa, haciéndolo permanecer en posición erecta de forma estable.

Resumiendo los puntos clave del modelo final, en orden:

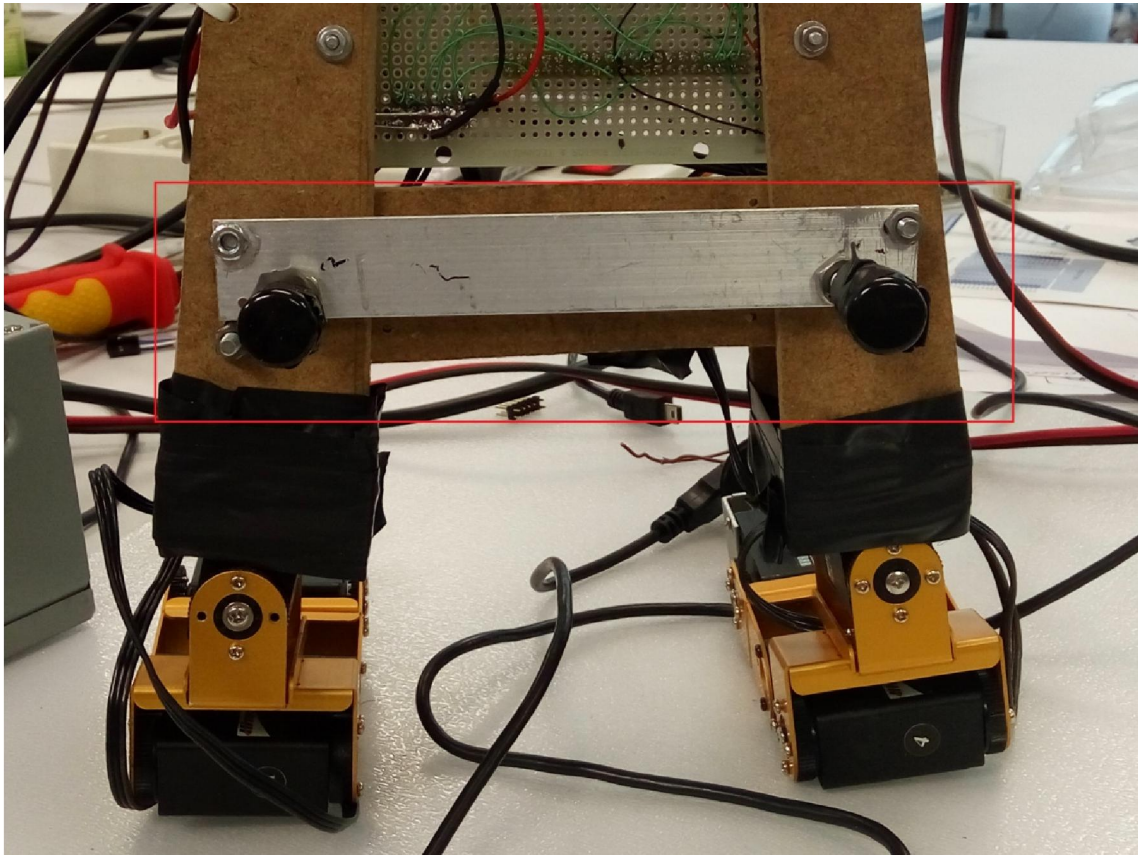
- Proporcionar al robot la posibilidad de alcanzar la posición vertical en una posición estable.
- Buscar una secuencia de movimientos para levantar el robot del todo.
- Encontrar una condición de equilibrio estable en posición de bípedo.
- Crear una función para andar.

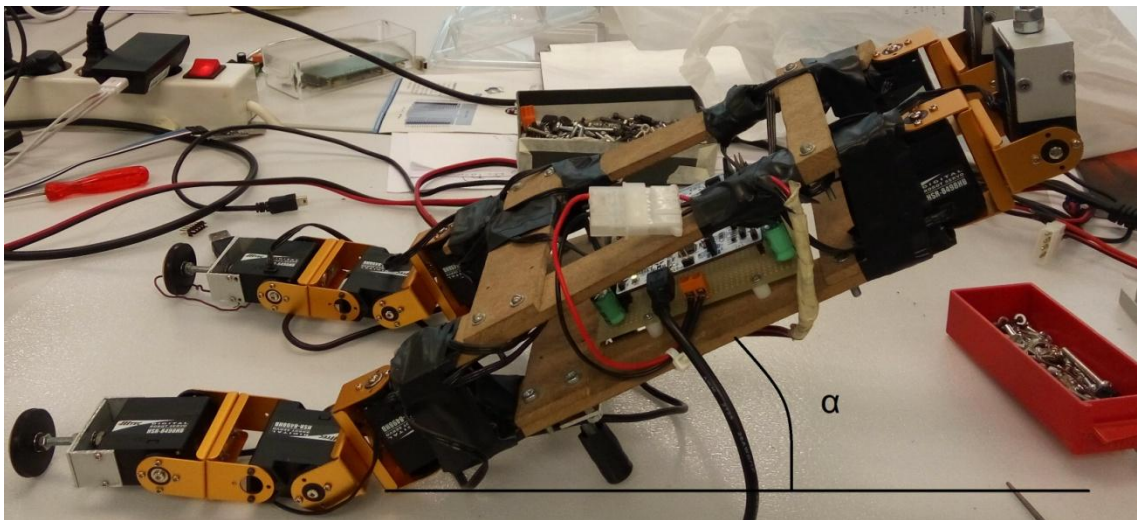
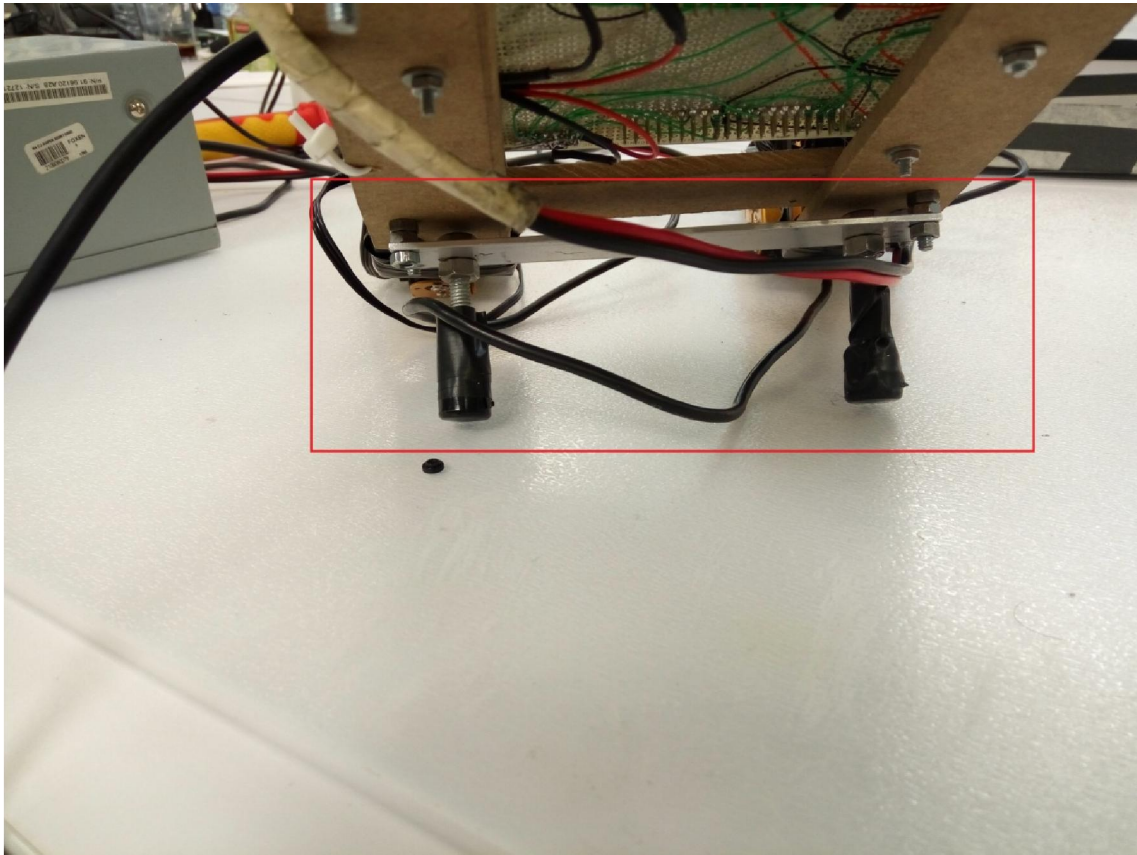
Estos puntos clave se han estudiado siguiendo el orden, empezando entonces por el primer punto, estudiado y probado ya anteriormente.

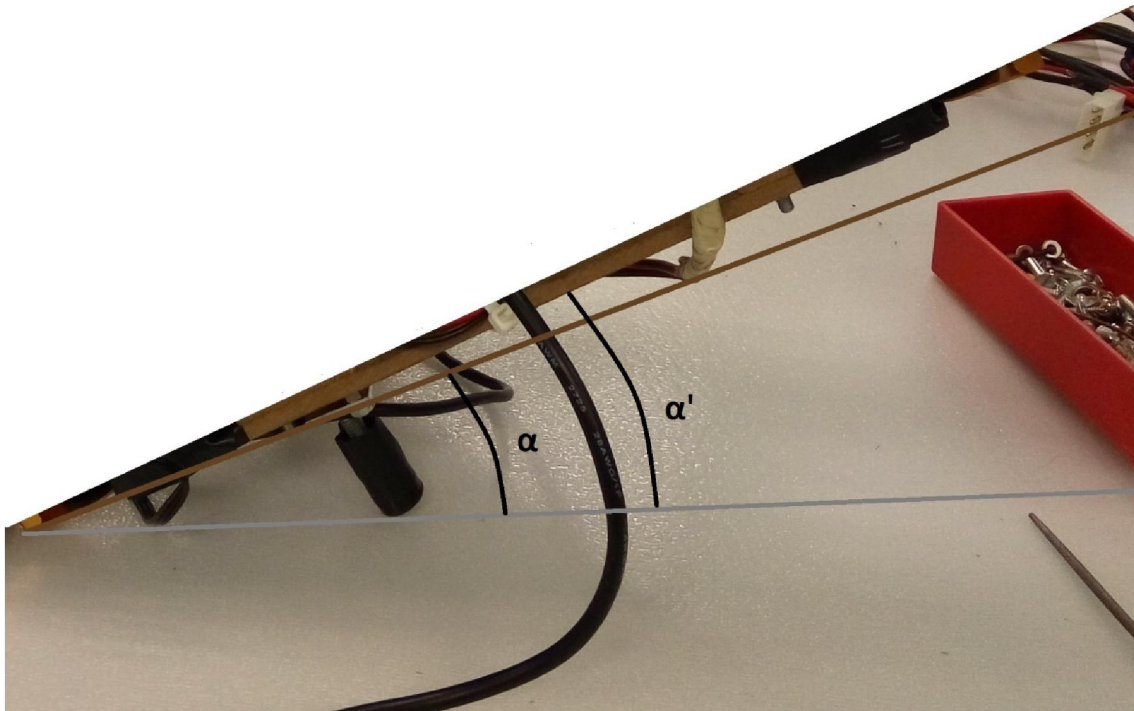
Sobre la base de todos los estudios y pruebas de levantamiento efectuadas se ha llegado a la conclusión de que, para este modelo en concreto, es necesaria una solución física distinta que permita aligerar la parte anterior del robot durante la acción de levantamiento, para evitar los inconvenientes de las distintas soluciones probadas anteriormente.

En particular, en el apartado relativo al estudio dinámico de la posición extendida, se han calculado los momentos generados a lo largo del robot y el momento necesario para obtener un equilibrio de fuerzas. Por los resultados del análisis se ha podido demostrar que los momentos generados por la fuerza peso de los servos delanteros son demasiado grandes para poder ser vencidos por los de los servos posteriores; el efecto de esto ha generado un movimiento opuesto al deseado.

Se ha buscado una solución para poder reducir el momento generado por los servos delanteros, sin modificar la disposición de los mismos de ninguna forma. La idea ha sido la de introducir unas patas secundarias fijas, de manera que el robot en posición extendida formase un ángulo mayor con respecto al suelo que el ángulo formado al estar sujeto solo por las patas delanteras apoyadas en el suelo.







En esta última imagen se ha comparado el ángulo formado por el robot en posición extendida con respecto al suelo, antes de añadir las patas fijas (ángulo α) y después (ángulo α') .

En realidad el tamaño de las patas fijas añadidas ha sido posteriormente cambiado por uno mucho más grande, como consecuencia de las primeras pruebas efectuadas y del estudio dinámico presentado a seguir.

- *Estudio dinámico*

El tamaño definitivo de las patas fijas ha sido calculado en base al ángulo formado con el suelo; precisamente se ha buscado un valor determinado de ángulo que redujese de forma sustancial el momento total generado por la parte levantada del suelo, hasta el punto de permitir a los servos levantar el cuadrúpedo sin problemas.

Para eso, el momento total de la parte anterior se tiene que acercar al momento capaz de generar un servo HSR-8498HB, que es de 0,79 Nm por 7,4 Voltios de alimentación.

Sin embargo, observando que en la dinámica del movimiento de levantamiento trabaja más de un servo y que la batería es capaz de alimentarlos con una tensión superior a 7,4 Voltios en una buena condición de carga, se puede llevar a cabo perfectamente el movimiento hasta valores cercanos a los 0,9 Nm del Momento total (M_{tot}) generado por la parte levantada delantera del robot.

Es más, para evitar que dichas patas fijas fuesen demasiado largas (lo cual no sería aceptable en el funcionamiento normal de cuadrúpedo), se ha considerado como resultado óptimo conseguir una reducción del momento total generado en la parte anterior del 25%.

Recordando los cálculos del estudio dinámico por la posición extendida:

- Peso de los servos anteriores más soportes:

$$P_{sa} \approx 2,45N$$

- Peso de los servos posteriores más soportes:

$$P_{sp} \approx 3,71N$$

Que genera medianamente el siguiente momento anti horario en la parte posterior:

$$M_{post} = 3,71 * 0,08 \approx 0,3Nm$$

Considerando esta vez el ángulo de inclinación del cuadrúpedo como incógnita, se busca un momento en la parte anterior que sea, como se ha mencionado, igual que el obtenido anteriormente con un ángulo $\alpha = 20^\circ$, reducido del 25%:

$$\text{Por } \alpha = 20^\circ: M_{tot} = 1,13Nm$$

$$M_{ant} = 75\% \text{ de } M_{tot} \Rightarrow M_{ant} = 1,13 * 75 / 100 \approx 0,85 \text{ Nm}$$

Donde con M_{ant} se indica el nuevo momento anterior buscado.

$$\alpha = ?$$

$$\beta = 180^\circ - 90^\circ - \alpha = 90^\circ - ?$$

por $M_{post} = 0,3Nm$, la fuerza perpendicular correspondiente, recordando que la longitud de la parte levantada es de 30 cm, será

$$F_p = M_{sant}/l = [M_{ant} - M_{estruc}]/l$$

Con M_{sant} momento generado por los servos anteriores más soportes y M_{estruc} momento generado por el resto de la estructura.

Siendo el peso de la estructura calculado anteriormente de 465,49g, que corresponde a $\approx 4,565\text{N}$

$$F_p = (0,85 - 4,565 \cdot \cos \alpha \cdot l/2)/0,3 = (0,85 - 4,565 \cdot \cos \alpha \cdot 0,3/2)/0,3 \approx 2,83 - 2,2825 \cdot \cos \alpha$$

Siendo F_p

$$F_p = P_{\text{sant}} - F_r = 2,45 - P_{\text{sant}} \cdot \cos \beta = 2,45 - 2,45 \cdot \cos(90^\circ - \alpha)$$

Donde F_r es la fuerza radial y P_{sant} el peso de los servos anteriores con soportes.

Igualando las dos expresiones

$$2,83 - 2,2825 \cdot \cos \alpha = 2,45 - 2,45 \cdot \cos(90^\circ - \alpha)$$

$$\text{Donde } \cos(90^\circ - \alpha) = \sin \alpha \Rightarrow$$

$$-2,2825 \cdot \cos \alpha + 2,45 \cdot \sin \alpha = -0,38 \Rightarrow$$

$$\cos \alpha = (-0,38 - 2,45 \cdot \sin \alpha) / -2,2825 \approx 0,17 + 1,073 \cdot \sin \alpha$$

Elevando todo al cuadrado

$$\cos^2 \alpha \approx 0,03 + 1,15 \cdot \sin^2 \alpha + 0,17 \cdot 2,146 \sin \alpha$$

De las relaciones fundamentales de la trigonometría

$$\sin^2 \alpha + \cos^2 \alpha = 1 \Rightarrow$$

$$\sin^2 \alpha + 0,03 + 1,15 \cdot \sin^2 \alpha + 0,17 \cdot 2,146 \sin \alpha = 1 \Rightarrow$$

$$2,15 \cdot \sin^2 \alpha + 0,03 + 0,17 \cdot 2,146 \sin \alpha = 1 \Rightarrow$$

$$2,15 \cdot \sin^2 \alpha + 0,36 \cdot \sin \alpha - 0,97 \approx 0$$

$$\sin \alpha \approx \frac{-0,36 \pm \sqrt{2,91}}{2 \cdot 2,15} \Rightarrow (1) \sin \alpha \approx 0,59 / (2) \sin \alpha \approx -0,76$$

La segunda solución de $\sin \alpha$ ha de descartarse, ya que originaría un ángulo negativo.

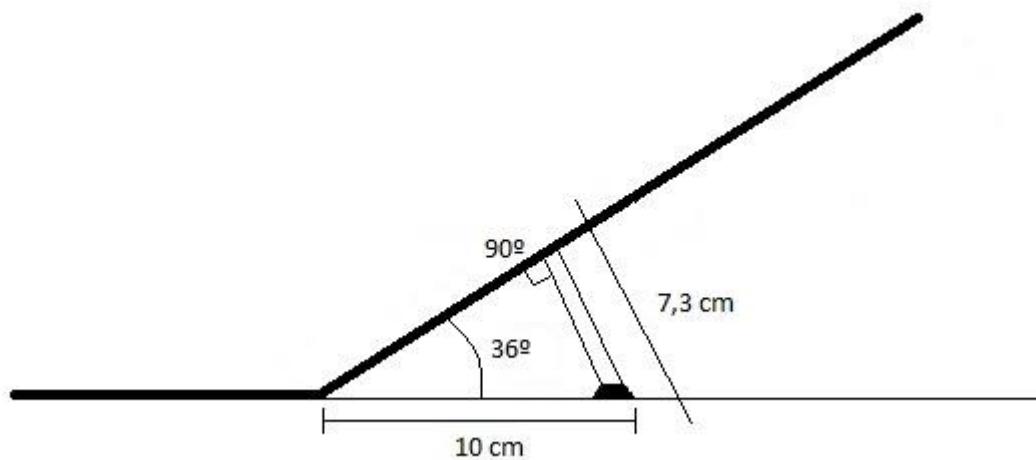
Considerando la primera solución, $\sin \alpha \approx 0,59$, el ángulo correspondiente es:

$$\text{Arc sen } (0,59) \approx 36^\circ$$

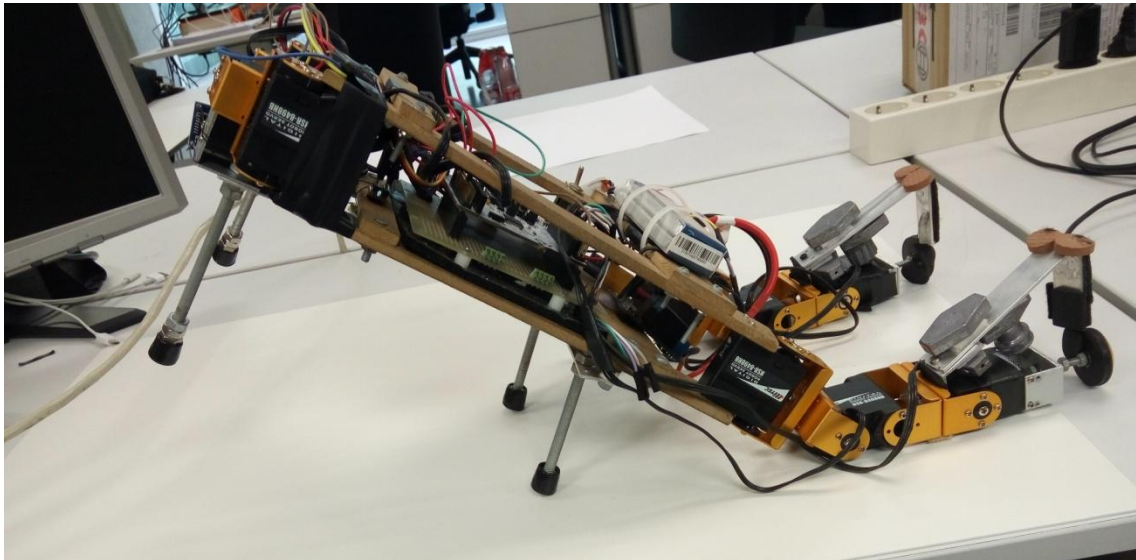
El ángulo α necesario para que el momento total generado por la parte anterior del robot se reduzca de un 25% con respecto al caso estudiado anteriormente, con $\alpha \approx 20^\circ$, ha resultado ser $\alpha \approx 36^\circ$.

Siendo la distancia entre el punto de apoyo de las patas fijas y la parte posterior calculada en 10 cm, y aproximando el ángulo formado entre la inclinación de la parte delantera del robot y las patas fijas de 90° , la longitud de las patas fijas se calcula como:

$$\text{Long} = 10 \cdot \tan(36^\circ) \approx 7,3 \text{ cm}$$



Sigue la imagen del robot con las patas fijas de 7,3 cm de longitud:



Una vez encontrado el tamaño ideal para las patas fijas, se ha buscado una secuencia de movimientos que simplificara el proceso de levantamiento y que lo hiciese más estable en la parte final, momento en el cual el robot se tiene que sujetar en posición vertical.

Además de esto, el momento total de la parte delantera (M_{tot}) sigue siendo de $0,85 \text{ Nm} > 0,3 \text{ Nm}$, que es el momento que mediante se genera en la parte posterior debido a la sola fuerza peso de los servos; esto significa que aun así se necesita un contrapeso capaz de generar más de $0,55 \text{ Nm}$, es decir:

$$P_{masas} > 0,55/0,08 \approx 6,87\text{N} \approx 700\text{g}$$

Aplicadas a una distancia de 8 cm de las patas fijas.

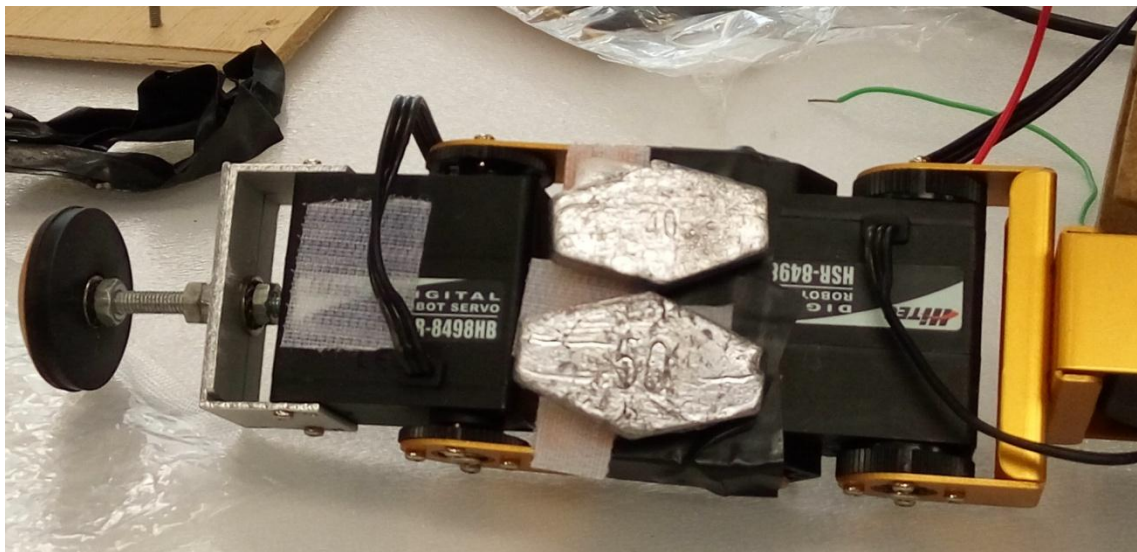
La solución perfecta para mejorar el nivel de estabilidad y para permitir reducir dicha masa es un movimiento de levantamiento distinto, que prevé la apertura de las patas traseras, hasta alcanzar el punto de contacto de las patas fijas con el suelo y entonces empujar el suelo hacia adelante y al mismo tiempo hacia abajo.

El punto de contacto de las extremidades en este caso siendo más adelantado que el del perno se lleva una parte del peso de la parte delantera hacia atrás, cuantificado después de varias pruebas experimentales a más de 100g por pata, es decir reduciendo la masa necesaria en la parte posterior a al menos

$$700 - (100 \cdot 2) \cdot 2 \approx 300\text{g} \text{ (se ha multiplicado cada pata por dos, ya que la masa que se lleva la parte posterior se ve quitada a la anterior).}$$

Resumiendo, con este nuevo movimiento de levantamiento se alcanza una posición vertical más estable y además se reduce la masa necesaria para balancear el momento de la parte anterior a unos 300g.

Se ha decidido pues aplicar a cada pata unas pequeñas masas de plomo de 50 y 40 gramos, entre el servo-motor de abajo y el del medio, ya que este es el punto de la pata que queda más atrás durante el movimiento que lleva al cuadrúpedo a alcanzar la posición vertical y que tiene pues más influencia sobre el momento generado:



En el apartado siguiente, llamado *levantamiento del prototipo final*, se analizan paso a paso todas las fases del movimiento que lleva al cuadrúpedo a la posición vertical.

Una vez alcanzada la posición vertical estable, se ha buscado una secuencia de movimientos para levantar el cuadrúpedo a la posición erecta más bien típica de un robot bípedo.

Después de que el cuadrúpedo se levante del todo, debe permanecer en la posición erecta de forma estable; para esto se han introducido unos soportes secundarios fijos para las patas traseras:





Y como se observa en la imagen, las masas de plomo se han distribuido en la parte interior y exterior del soporte, aprovechando este nuevo espacio disponible.

Obteniendo más espacio disponible, se ha preferido aumentar las masas de plomo con otra de 40 gramos, ya que el cambio de disposición de ellas a lo largo del soporte ha extendido la concentración de las masas a un área de influencia más grande, que genera un momento ligeramente inferior.

- Levantamiento del prototipo final

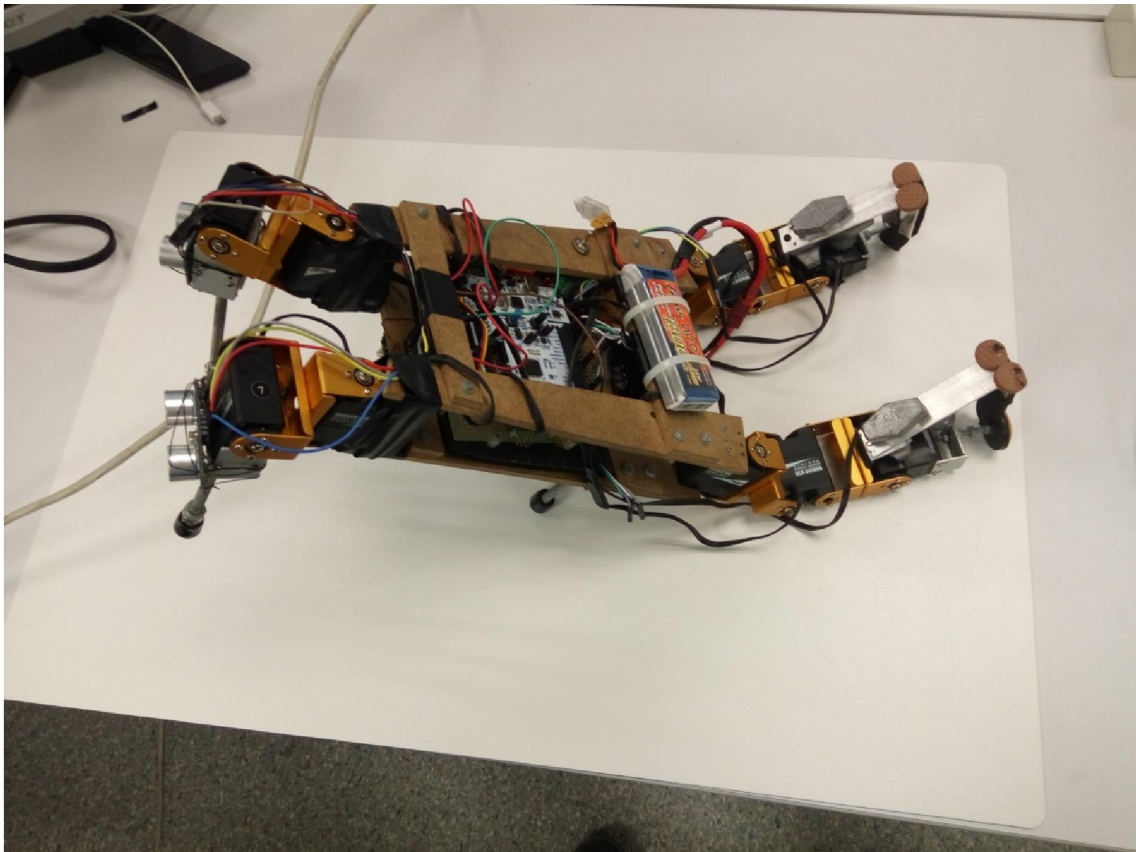
El movimiento de levantamiento con los soportes del prototipo final se divide en las siguientes fases:

- Robot en la posición extendida de partida
- Preparación de la fase de empuje
- Empuje de las patas traseras

- Estacionamiento en posición vertical

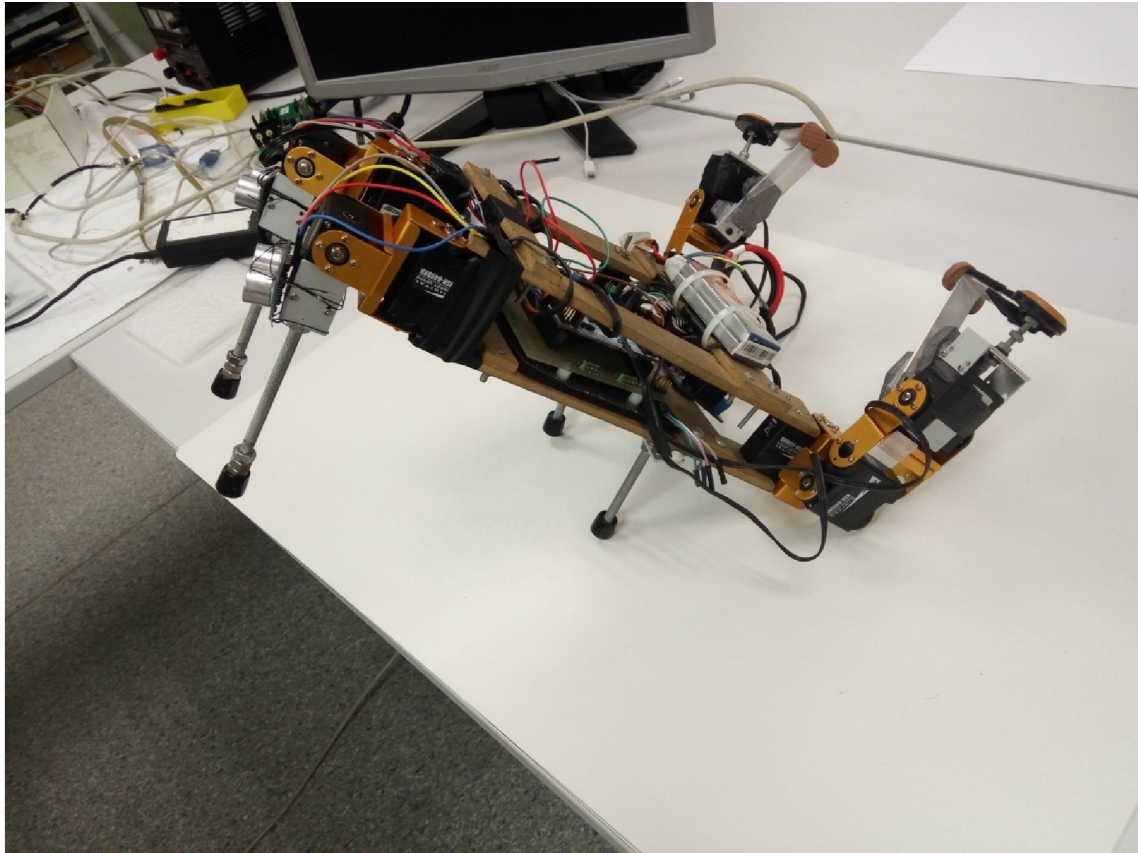
La posición de partida es una posición muy cercana a la posición denominada anteriormente como posición extendida.

Se trata de una posición muy estable que permite obtener un ángulo aproximativo de 36° del cuadrúpedo con respecto al suelo:



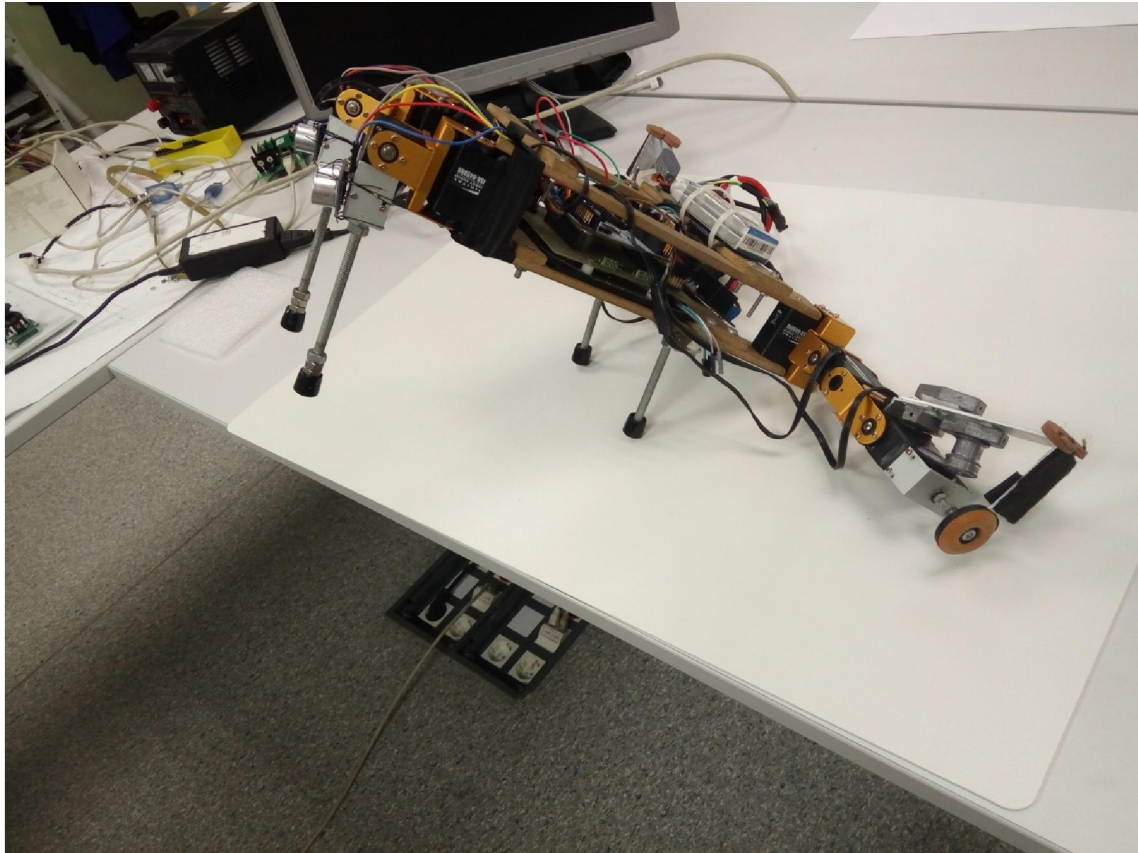
La segunda posición es necesaria para preparar la fase de empuje y consiste en mover hacia fuera los servos de arriba de las patas posteriores, de manera que el movimiento de los servos del medio y de abajo pase de ser perpendicular al suelo a ser paralelo a él.

En realidad, el ángulo de giro no llega a ser de 90° hacia fuera y entonces el movimiento de los servos del medio y de abajo no llega a ser perfectamente paralelo al suelo sino ligeramente inclinado hacia abajo; es por eso que en la fase sucesiva las dos extremidades de las patas posteriores llegan al contacto con el suelo, permitiendo a los servos empujar hacia adelante.

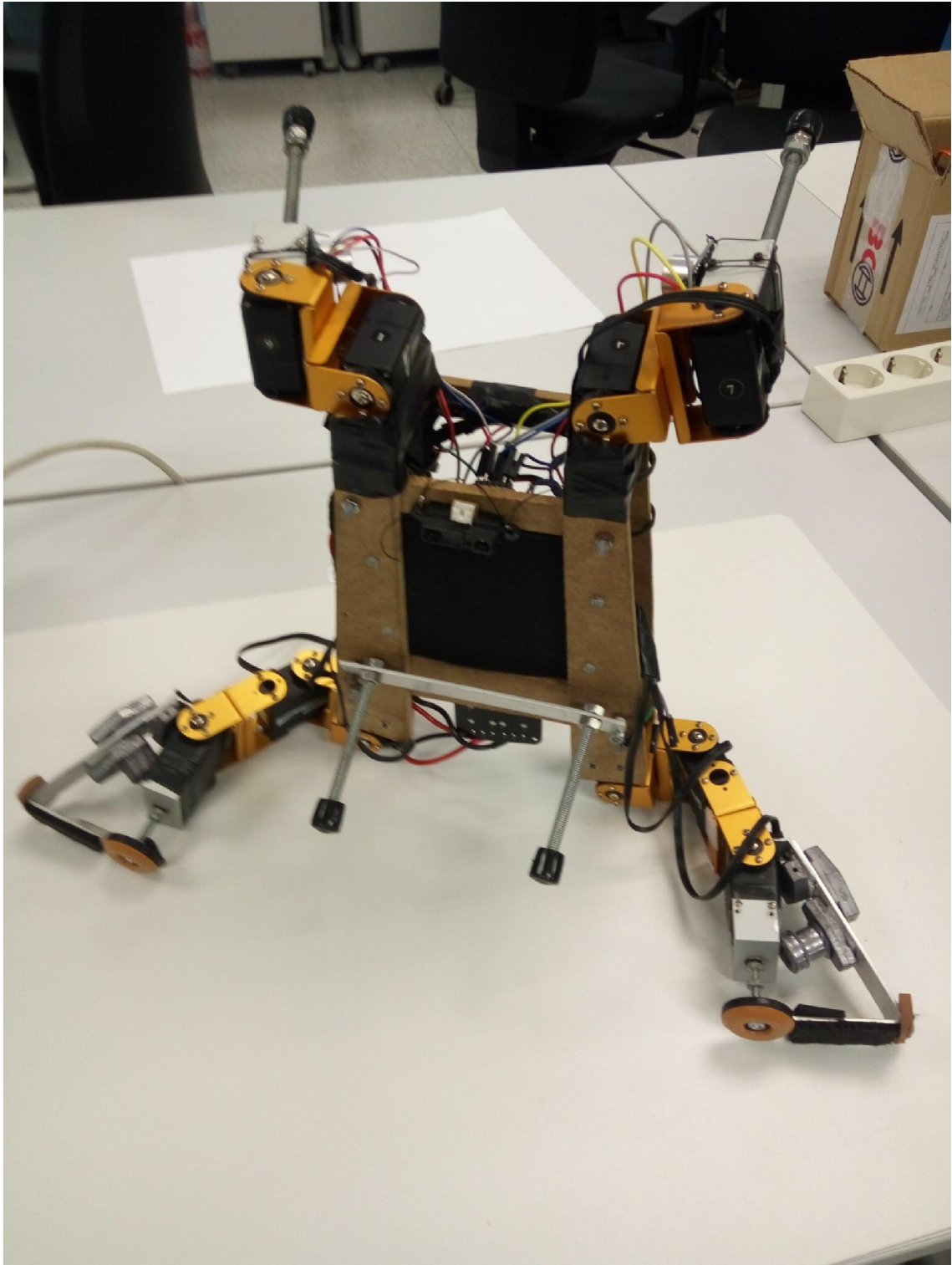


En la fase de empuje los servos del medio de las patas posteriores se cierran hacia adelante del robot, hasta que las extremidades entren en contacto con el suelo. A partir de ese momento las dos extremidades se encuentran en una posición más adelantada con respecto al punto de contacto de los servos de arriba y esto permite hacer rotar el cuadrúpedo hacia atrás, descargando en el suelo casi todo el peso por este punto de apoyo y no por las extremidades; por esta razón la parte delantera rueda hacia atrás sin levantarse del suelo.

Los servos delanteros se mueven hacia arriba y hacia atrás para llevar el baricentro lo más atrás posible y ayudar el movimiento de levantamiento:



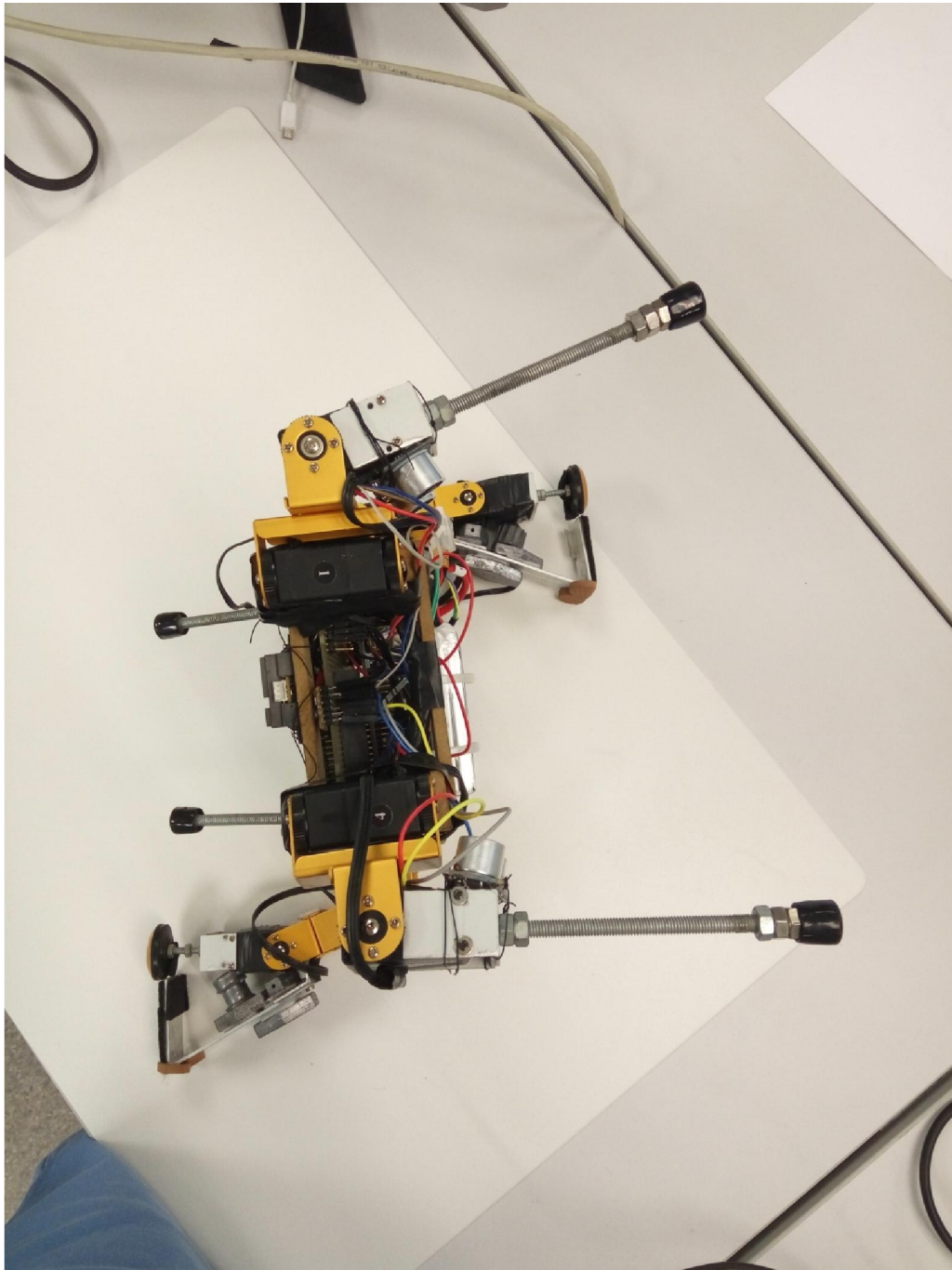
En la fase final la parte delantera del robot se encuentra ya en la posición vertical y gracias al volumen de apoyo ocupado por las patas traseras se queda de forma estable en esta posición:

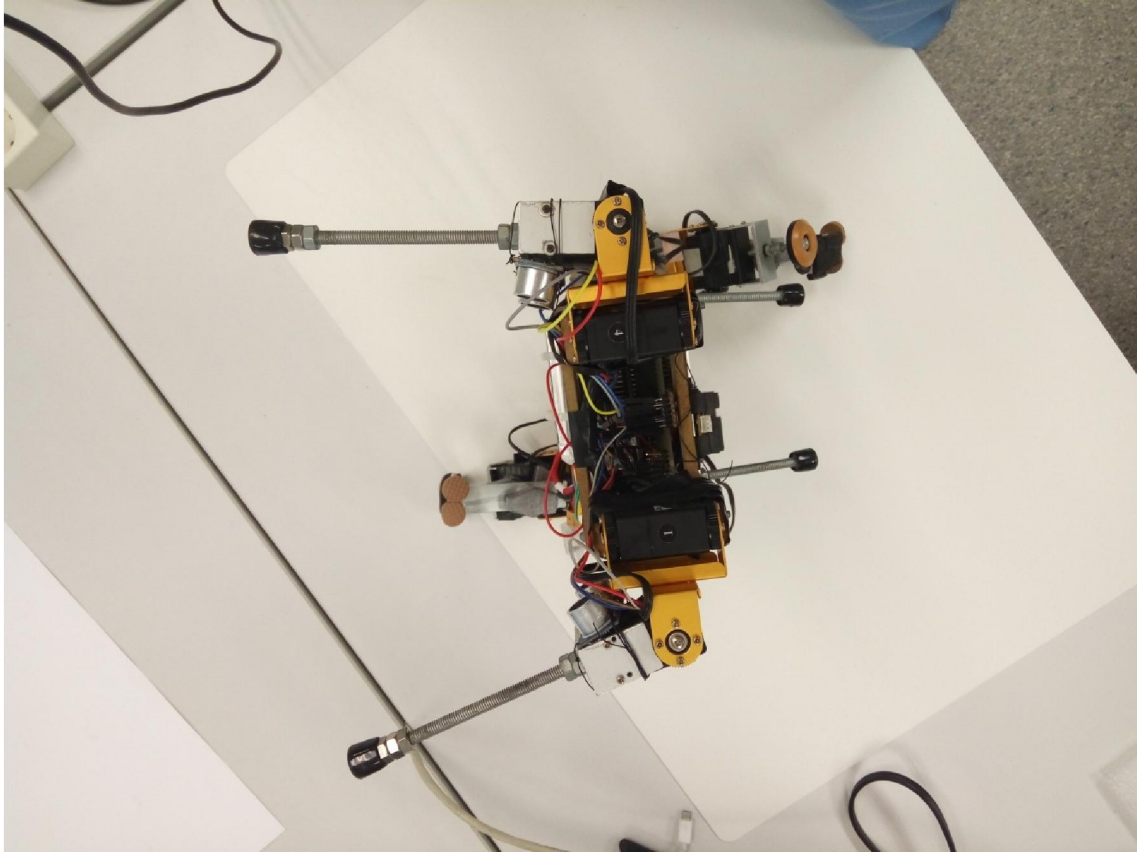


- *Posición erecta*

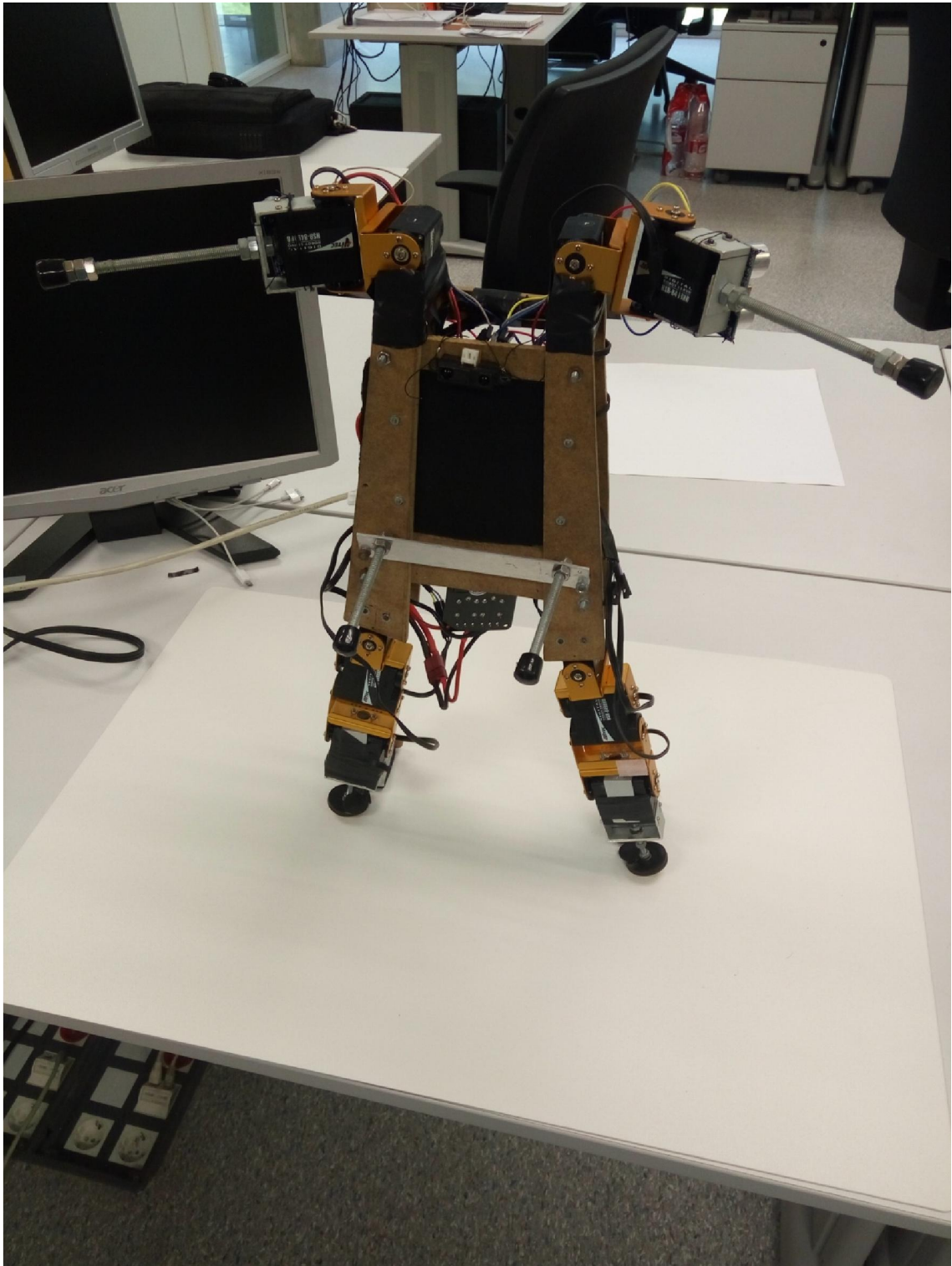
La fase sucesiva ha sido la de llevar el cuadrúpedo a la posición erecta.

Para esto se ha llevado una pata atrás y se han girado las dos de 90° para disponerlas en posición de empuje, reportando otra vez las patas anteriores hacia adelante para balancear el peso durante el movimiento:





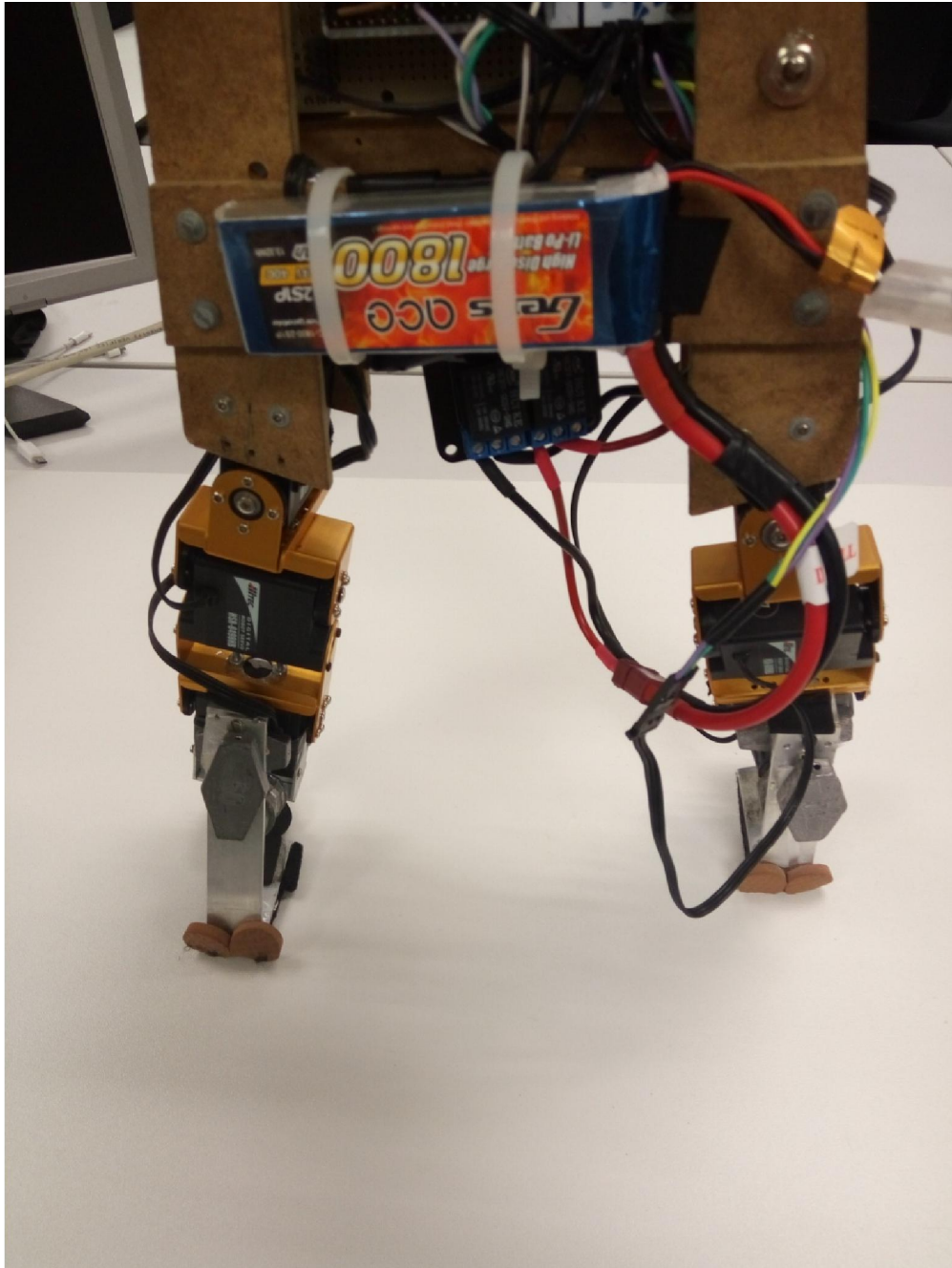
Llegados a esta posición, se han utilizado los dos servos de abajo y del medio de las patas posteriores para empujar hacia abajo y levantar el robot hasta la posición erecta de forma estable, aprovechando también las patas anteriores para ir balanceando el peso hacia adelante o hacia atrás:



- *Función para andar*

Se ha creado también una simple función para andar, a partir de la posición erecta alcanzada en la fase anterior.

Esta función desplaza ligeramente una pata hacia el exterior y doblando la rodilla la mueve hacia delante unos pocos centímetros (6-7 cm), volviéndola a cerrar hacia el interior; después, se vuelve a extender la pata del todo y el movimiento empieza otra vez con la otra pata.



CONCLUSIONES Y ESTUDIOS FUTUROS

Uno de los objetivos de todo el trabajo desarrollado en este proyecto ha sido el de presentar el estudio de un prototipo de robot cuya finalidad principal es la de constituir la base de referencia para la creación de modelos mejorados en proyectos futuros.

De hecho, en este proyecto se han analizado los pros y los contras de todas las decisiones tomadas en la fase de diseño y de realización del prototipo final y los resultados de las pruebas experimentales.

Como se ha mencionado en la introducción, la organización del trabajo no ha sido secuencial en las varias fases porque, siendo este un proyecto multidisciplinar y sin estudios previos, se ha preferido trabajar en paralelo en las distintas tareas para tener la posibilidad de modificar cada una de ellas, aprovechando los resultados de los estudios de las otras.

Después de una primera parte de pruebas de levantamiento con servos HDKJ S3090D, se ha empezado con la elección de los componentes y del diseño de la estructura, ya que están estrictamente relacionadas; sucesivamente, una vez que las ideas sobre el aspecto y las funcionalidades del robot tenían más cuerpo, se ha empezado con la parte del montaje y de la programación del microcontrolador, sin dejar de lado, sin embargo, la del diseño, que se ha ido mejorando paso por paso.

En la parte final del trabajo se han implementado las mejores soluciones estudiadas y probadas para el prototipo final, hasta llevarlo a la posición vertical de forma estable, con una secuencia de movimientos muy precisa.

Llegados a este punto del proyecto, no pudiendo modificar más el diseño, se han analizado los resultados de todo el trabajo y se han definido los varios aspectos mejorables, debidos, por una parte, a los problemas físicos relacionados con el tipo de estructura utilizada, y por otra, a los componentes y a su disposición.

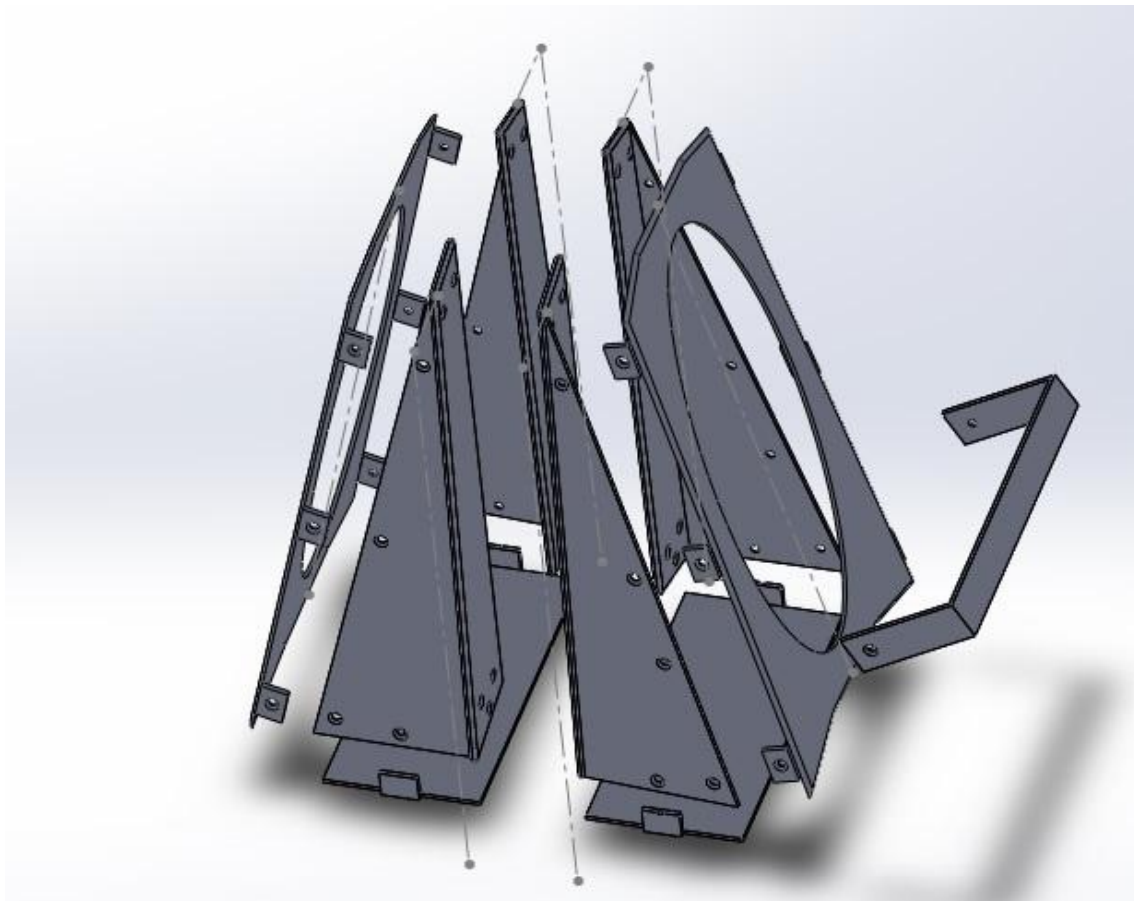
Es evidente que una vez llegados a la fase del prototipo final, es decir, a un punto muy avanzado del proyecto, ha resultado imposible modificar todo el sistema para solucionar los problemas encontrados en esta etapa y para superar algunas de las limitaciones encontradas en el modelo; además siendo este un prototipo básico para estudios futuros, la idea ha sido la de explicar detalladamente la razón de todas las decisiones tomadas, de documentar todos los resultados obtenidos y también expresar unas sugerencias para la mejora del diseño en los prototipos siguientes.

Una de las mejoras estructurales muy importantes, casi indispensable para el diseño, es la reducción del tamaño del cuerpo del cuadrúpedo, ya que teniendo ahora una idea muy clara de los que van a ser todos los componentes utilizados, con dimensiones y pesos, se puede encontrar una disposición óptima también en una estructura de dimensiones bastante reducidas.

Claramente, esto, como salta a la vista, depende sobre todo de la cuestión del peso y del momento generado por el peso de los servos anteriores a una distancia tan elevada del punto de levantamiento.

Una correcta reducción de la longitud de la estructura acompañada por otros pequeños detalles estructurales haría disminuir el momento que se contrapone al movimiento de levantamiento del robot hasta un 30%.

En la imagen siguiente se puede observar la idea de proyecto para un modelo futuro mejorado, donde el material utilizado es de la familia de los materiales plásticos, pensado para un diseño en impresora 3D. En concreto se ha utilizado el Nylon 101, ampliamente utilizado en el mercado de los velívolos radio controlados.



En este nuevo modelo además, como es fácil observar, se ha creado una base más alta además de más ancha (al igual que en el modelo anterior), lo suficiente alta para permitir posicionar la tarjeta Nucleo STM32 en vertical en una de las dos bases que se ven en la figura y en la otra base la batería Lipo; la geometría ideada, además, permite posicionar todo el cableado y parte de los sensores dentro de las tapaderas laterales de forma triangular, de manera que quedan más protegidos y al mismo tiempo se da un aspecto más profesional al robot.

Las tapas superior e inferior, que tienen función de tener compacta la estructura, se han diseñado huecas para reducir sensiblemente el peso; además de esta solución, se han evaluado otras donde en lugar de un solo hueco central, hay muchos huecos pequeños distribuidos por la parte central (solución más adoptada en la robótica) u otra donde el hueco está cubierto por una lámina finísima de plástico de baja densidad, ya que no tendría ninguna función de absorbimiento de las fuerzas a lo largo de la estructura. Un último detalle relevante es la pata fija, en este caso del mismo material que la estructura (Nylon 101), más espeso que la estructura y que descarga el peso en la parte lateral; si el peso total del robot es muy elevado en comparación al de la estructura, se pueden crear más puntos de contacto atornillados, que repartan el peso en distintas zonas.

En la imagen siguiente se observan las características físicas del nuevo diseño:

Propiedades de masa de Ensamblaje_Nylon

Configuración: Predeterminado

Sistema de coordenadas: -- predeterminado --

Masa = 201.92 gramos

Volumen = 175582.77 milímetros cúbicos

Área de superficie = 197014.14 milímetros cuadrados

Centro de masa: (milímetros)

X = 14.52

Y = -26.35

Z = 236.23

Ejes principales de inercia y momentos principales de inercia: (gramos * milímetros cuadrados)

Medido desde el centro de masa.

Ix = (0.01, 0.99, 0.12) Px = 1061172.10

Iy = (-0.04, -0.12, 0.99) Py = 1375376.67

Iz = (1.00, -0.02, 0.04) Pz = 1594198.49

Momentos de inercia: (gramos * milímetros cuadrados)

Obtenidos en el centro de masa y alineados con el sistema de coordenadas de resultados.

Lxx = 1593754.37 Lxy = 7147.93 Lxz = -8247.58

Lyx = 7147.93 Lyy = 1065665.97 Lyz = 37036.01

Lzx = -8247.58 Lzy = 37036.01 Lzz = 1371326.92

Momentos de inercia: (gramos * milímetros cuadrados)

Medido desde el sistema de coordenadas de salida.

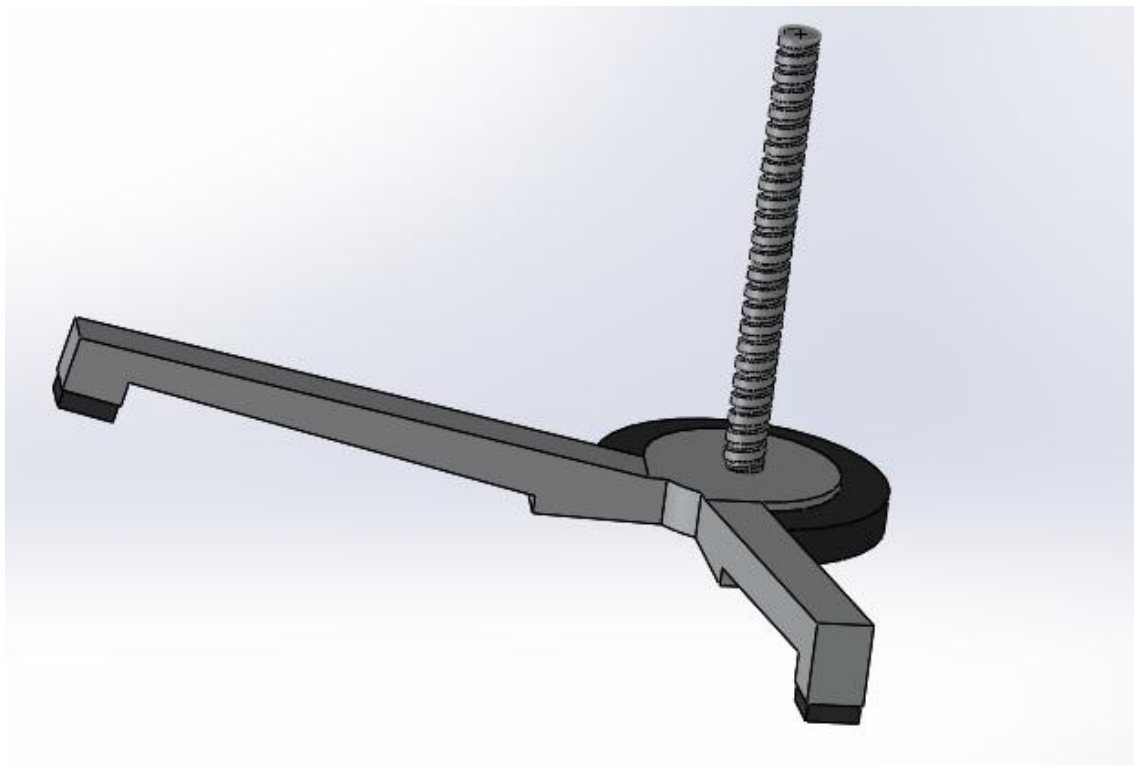
Ixx = 13002375.61 Ixy = -70116.00 Ixz = 684561.25

Iyx = -70116.00 Iyy = 12376732.97 Iyz = -1219654.63

Izx = 684561.25 Izy = -1219654.63 Izz = 1554071.79

Se evidencia que en este caso la masa total resulta de 201,92 gramos, frente a los 283,59 gramos del prototipo final realizado en este proyecto, aunque esta vez incluyendo también el peso de la pata fija.

En las pruebas sobre el prototipo final los problemas principales se han encontrado en la fase de posición erecta, sobre todo en la estabilidad durante los pasos; además, en condición de parado, aunque el robot conseguía quedarse levantado sin caerse, lo hacía de forma no muy estable, es decir, con un pequeño empuje perdía estabilidad y se caía al suelo. Esto claramente no es aceptable y es necesario buscar una solución que mejore la estabilidad del robot en el modo de funcionamiento de bípedo. Las causas principales de estas faltas de estabilidad son el baricentro demasiado alto, que se ha conseguido reducir de posición en el nuevo modelo presentado anteriormente y las extremidades de las patas muy poco anchas y rígidas. Este último factor tiene amplios márgenes de mejoras, ya que no ha sido uno de los puntos donde se ha centrado más esta investigación; se propone pues una simple solución que se puede implementar en los nuevos modelos:



En este nuevo modelo se han posicionado dos extremidades abiertas y ligeramente levantadas del suelo; esto permite al robot utilizar solo la parte circular durante el normal funcionamiento como cuadrúpedo y aprovechar también de las otras dos extremidades a partir del momento en que se eleva a la posición vertical. De esta manera los puntos de apoyo pasan a ser tres con los dos laterales bien abiertos y equidistantes del punto central, que sigue siendo el punto de apoyo más extendido. Con una solución de este tipo el robot ganaría mucha estabilidad en dirección frontal y también lateral debido como dicho al ángulo de apertura de los dos nuevos apoyos.

En un modelo más avanzado se podría utilizar un pequeño sistema de amortiguación en las dos barritas posteriores para estresar menos el material en los puntos más solicitados y para adquirir más flexibilidad y naturaleza en los movimientos.

Queriendo hacer un presupuesto global de este nuevo modelo aplicado al proyecto, es decir, con todos los componentes utilizados para el modelo anterior y modificando el diseño del cuerpo y de las extremidades posteriores de las patas con los nuevos presentados en este apartado final, se pueden hacer, por ejemplo, las siguientes suposiciones:

- Se producen 1000 ejemplares
- El diseño del cuerpo se hace con una impresora 3D
- El montaje es manual debido a la complejidad y al mercado restringido

Estimación del coste de la realización de la estructura:



Nombre del ensamblaje: Ensamblaje_Nylon

Peso total:	3.03 lb
Peso total del material:	8.34 lb

Cantidad para producir

Número total de ensamblajes:	1000
Tamaño del lote:	1

Coste estimado por ensamblaje: 192.99 USD

Plantilla de Costing principal: multibodytemplate_default(englishstandard).sld.ctc

Comparación:  Actual 192.99 USD Anterior 192.99 USD

Desglose de costes

Piezas calculadas:	192.99 USD	100%
Piezas adquiridas:	0.00 USD	0%
Piezas de Toolbox:	0.00 USD	0%
Marca	0.00 USD	0%

- Como se puede ver el coste ha resultado de 193 USD $\approx$ 173 euros
- El segundo coste principal es el de los servos, en este caso los HSR-8498HB, cuya calidad es más elevada que la de los HDKJ S3090D. Para un lote de 10.000 servos, siendo 10 los necesarios por cada robot, el mejor precio por cada servo encontrado en internet se acerca a los 35,5 euros.

- El precio de cada tarjeta Núcleo STM32F334R8 para un lote de 1000 sería inferior a 10 euros.
- El precio de una batería Lipo de 2 celdas (7,4V) saldría de ~ 14 euros por pieza.
- El coste de los sensores más el cableado y otros componentes de montaje se estima de ~ 10 euros por ejemplar.

Coste componentes total estimado: 562 euros.

Calculando dos horas de trabajo de mano de obra por cada pieza, cuyo coste sería cuantificable en 40 euros,

Coste total: 602 euros.

Si se desea tener un útil de 50 euros por cada pieza y considerando el IVA (21%), el precio por cada pieza estimado es de: $(602 + 50) * 121/100 \sim 789$ euros.

Bibliografía

- A Controller for the LittleDog Quadruped Walking on Rough Terrain
(John R. Rebulq, Peter D. Neuhaus, Brian V. Bonnlander, Matthew J. Johnson, Jerry E. Pratt)
- Autonomous Evolution of Dynamic Gaits with Two Quadruped Robots
(Gregory S. Hornby, Seichi Takamura, Takashi Yamamoto, and Masahiro Fujita)
- BigDog, the Rough-Terrain Quadruped Robot
(Marc Raibert, Kevin Blankespoor, Gabriel Nelson, Rob Playter and the BigDog Team)
- Biologically inspired Quadruped Robot Biosbot: Modeling, simulation and experiment
(Xu Guan, Haojun Zheng, Xiuli Zhang)
- Dynamic Compliant Walking of a Quadruped Robot: Preliminary Experiments
(Martin de Lasa and Martin Buehler)
- "Cheap" Rapid Locomotion of a Quadruped Robot: Self-Stabilization of Bounding Gait
(Fumiya Iida and Rolf Pfeifer)