



UNIVERSIDAD DE VALLADOLID



**ESCUELA TÉCNICA SUPERIOR
DE INGENIEROS DE TELECOMUNICACIÓN**

TRABAJO FIN DE GRADO

INGENIERÍA DE TECNOLOGÍAS ESPECÍFICAS DE LA TELECOMUNICACIÓN

DESARROLLO DE SISTEMA PARA EL ESTUDIO DE EXOESQUELETOS DE EXTREMIDADES INFERIORES CONTROLADOS MEDIANTE SENSORES

AUTOR: Pablo Riesco Gil

TUTOR: Alonso Alonso Alonso

2018

Índice de contenidos

| | |
|---|------------|
| 1. Introducción | 4 |
| Ámbito de trabajo | 4 |
| Introducción a los exoesqueletos..... | 4 |
| Historia de los exoesqueletos | 6 |
| Estado del arte | 9 |
| Etapas de la deambulación y la bipedestación | 20 |
| Ámbito local | 23 |
| Objetivos..... | 27 |
| Estructura del documento | 27 |
| 2. Hardware..... | 29 |
| Sensores | 29 |
| Anatomía y esfuerzo muscular | 30 |
| Clasificación de los sensores según la magnitud medida..... | 31 |
| Sensores piezoeléctricos..... | 33 |
| Galgas extensiométricas | 39 |
| Sensores en la deambulación..... | 57 |
| Microcontrolador Arduino..... | 58 |
| Actuadores | 62 |
| Robot bípedo | 62 |
| Servomotores..... | 66 |
| Caja de circuitos | 67 |
| Visión global del sistema..... | 73 |
| 3. Software..... | 87 |
| Fases de la bipedestación..... | 87 |
| Diagrama de estados del sistema | 90 |
| Diagrama de flujo del sistema | 91 |
| Análisis del código fuente | 93 |
| 4. Pruebas y ensayos | 108 |
| Puesta a punto de los sensores | 108 |
| Comprobación del software | 112 |
| Implementación del hardware..... | 113 |
| Comprobación del sistema completo..... | 118 |

| | |
|----------------------------------|------------|
| 5. Conclusiones..... | 122 |
| 6. Líneas futuras | 125 |
| Futuro de los exoesqueletos..... | 125 |
| Mejoras del sistema actual..... | 127 |
| 7. Anexos..... | 129 |
| 8. Referencias | 149 |
| 9. Glosario | 151 |

1. INTRODUCCIÓN

1.1. ÁMBITO DE TRABAJO

1.1.1. Introducción a los exoesqueletos

Existen numerosos impedimentos asociados a la pérdida de fuerza y de control del tren inferior en el ser humano. Ciertos problemas como accidentes, enfermedades, o simplemente una edad avanzada evita que los pacientes puedan realizar tareas tan cotidianas como caminar o levantarse. Un gran colectivo de población que a menudo se ve afectado es el grupo de la tercera edad que, hayan sufrido alguna dolencia reseñable o no, los propios músculos ya no funcionan lo suficiente como para moverse con soltura sin ayudas externas.

La solución global que más se estudia actualmente es el uso de exoesqueletos accionados por el propio paciente. Un exoesqueleto es, *grosso modo*, una estructura anclada a las piernas del paciente que realiza los movimientos por él. Para ello, el sistema debe obtener cierta información del usuario, tales como la intención de movimiento y la propia posición de las piernas que marca la etapa de deambulación. Hay diferentes modos de captar esta información, dependiendo de las posibilidades del paciente.

Se utiliza el término “paciente” para designar a aquellos usuarios que utilizan el exoesqueleto para suplir alguna carencia motora asociada al tren inferior. Sin embargo, el exoesqueleto tiene otras utilidades más allá de la ayuda a personas con movilidad reducida o rehabilitación. El proyecto DARPA, por ejemplo, comenzó a financiarse en el año 2000 con el propósito de desarrollar un exoesqueleto que ayudase a los soldados del ejército estadounidense a soportar grandes cargas mientras corrían. De este trabajo se crearon dos prototipos: el BLEEX (*Berkeley Lower Extremity Exoskeleton's*) y el Sarcos, representados en la Figura 1.1. Este último es capaz de aumentar tanto la fuerza en las piernas como en los brazos, si bien apenas tiene autonomía (Zoss, A., Kazerooni, H. y Chu, A., 2006).



Figura 1.1. Ejemplos de exoesqueletos militares: Sarcos (izquierda) y BLEEX (derecha)

Otro ejemplo de exoesqueleto pensado para usuarios completamente sanos es el X1, desarrollado conjuntamente por la NASA y el IHMC (*Institute for Human & Machine Cognition*), en Florida. El X1 está ideado para prevenir la fatiga de los astronautas, así como ofrecer una herramienta para ejercitarse en gravedad nula (Rea, R. et al., 2013).

Un exoesqueleto no puede manejarse de la misma manera para alguien que no tiene ningún control de sus piernas que para un anciano que apenas tiene fuerza, pero aún mantiene sanas las neuromotoras de las piernas. Sensorizar la intención del paciente puede resultar una tarea muy complicada, y es el tema principal que abordar en este trabajo.

El problema que supone la instalación correcta de sensores no puede tratarse de forma aislada. Depende principalmente de la capacidad del paciente, así como de la propia ergonomía del exoesqueleto. Del mismo modo, no es suficiente con barajar las posibilidades que hay disponibles solo para la deambulación, sino que se nos plantea el mismo dilema a la hora de sensorizar una ayuda para levantarse y sentarse.

Así pues, se deben tener en cuenta las condiciones del paciente a la hora de elegir una adecuada forma de sensorizar. Se han discutido las distintas opciones disponibles para obtener la señal de acción del exoesqueleto:

- **Sensores de presión neumáticos.** Baratos y ergonómicos. Miden la fuerza de presión, ya sea en la contracción de los músculos o la presión que ejerce el pie sobre el suelo.
- **Sensores de presión de aire.** Siguen la misma filosofía que los neumáticos, pero al tratarse de cavidades con aire no son tan ergonómicos.
- **Sensores de presión sobre el suelo.** Adheridos al zapato, registran la cantidad de fuerza que ejerce el cuerpo en una zona concreta del pie.
- **Sensores de EMG.** Miden las ondas de electromiografía emitidas por las neuromotoras. Es necesario que el paciente aún conserve funcionales este tipo de neuronas en la pierna.

De forma equivalente, resulta de interés plantearse la posición en la que esos sensores irán instalados sobre el usuario. Es de esencial importancia conocer las zonas exactas del cuerpo en las que colocarlos, de modo que no sean un impedimento para la persona, y al mismo tiempo pueda proporcionar información suficiente.

Para todos los tipos de sensores, existen ciertos lugares típicos donde se pueden colocar. Generalmente, se suelen ubicar en la zona central del cuádriceps y de los gemelos, si bien para algunos exoesqueletos con sensores de presión de aire vienen situados en el bíceps femoral (Jung, P. et al., 2015). De forma convencional, se complementa con tres sensores de fuerza en el zapato para indicar la etapa de deambulación en la que se encuentra.

Sin embargo, no existe ningún estándar que aconseje el lugar donde ubicar los sensores. Precisamente ese va a ser uno de los objetivos primeros de investigación.

En definitiva y con todo lo explicado, deben satisfacerse dos soluciones para dos problemas bien distintos: la ayuda a la bipedestación y la ayuda a la deambulación.

1.1.2. Historia de los exoesqueletos

Antes de introducir la historia del desarrollo de los exoesqueletos hasta la actualidad, es esencial definir un concepto muy utilizado en el sector: el grado de libertad (DOF, del inglés, *Degrees of Freedom*). En términos de ingeniería, el número de grados de libertad

proporciona una información muy útil a la hora de describir un sistema. Tanto la posición como la rotación son los parámetros más comunes que otorgan grados de libertad.

A modo de ejemplo, un sistema capaz de moverse en los tres ejes cartesianos como de girar sobre sí mismo en todas las direcciones tiene seis grados de libertad en total. Sin embargo, un cuerpo que no pueda rotar y solo pudiera moverse en línea recta tendría un solo grado de libertad, tal y como ocurriría con un coche avanzando por una carretera de un solo carril.

Hoy en día, los exoesqueletos poseen alrededor de sesenta grados de libertad y tienen una respuesta del orden de los microsegundos. Como dato aclaratorio, es interesante saber que el cuerpo humano cuenta con unos seiscientos músculos, lo que se traduciría en más de trescientos grados de libertad posibles. No obstante, hay algunos músculos del cuerpo que son más importantes que otros en términos de eficiencia para el movimiento, al igual que es innecesario desarrollar un sistema que ayude a mover todo el cuerpo si solo se necesita asistir en el tren inferior.

Con todo, es importante diferenciar la propia biología del cuerpo humano de un sistema implementado electrónicamente. Aspectos como el funcionamiento en su conjunto como la ergonomía del exoesqueleto han sido esenciales en su desarrollo desde los inicios.

El diseño del exoesqueleto viene precedido por las tecnologías más aplicadas a la robótica. Para ello, es imprescindible conocer la anatomía humana y los movimientos del mismo de una forma realista. Un amplio estudio sobre esto fue llevado a cabo en el proyecto SANTOS Virtual Soldier, liderado por el Dr. Karim Abdel-Malek y bajo protección de la Agencia de Proyectos de Investigación Avanzados de Defensa (del inglés, DARPA) de los Estados Unidos.

El soldado virtual desarrollado es capaz de realizar todo tipo de movimientos y tareas con un realismo en sus posturas bastante plausible. Este soldado está diseñado de modo que cuenta con todo tipo de articulaciones a nivel óseo, tanto como los músculos que se mueven con ellas.

El primer exoesqueleto funcional coincidió con el primer robot humanoide, desarrollado por Miomir Vukobratovic en el Instituto Mihailo Pupin, en 1969 (Figura

1.2.). Nótese que se desarrolló antes un sistema de locomoción para las piernas que un robot humanoide autónomo. En cierto modo, los exoesqueletos pueden verse como los predecesores de los robots humanoides actuales (Vukobratovic, M. et al., 1974).

Con el tiempo, el interés por los exoesqueletos fue recuperado por intereses militares, y después como ayuda a personas con capacidades reducidas.

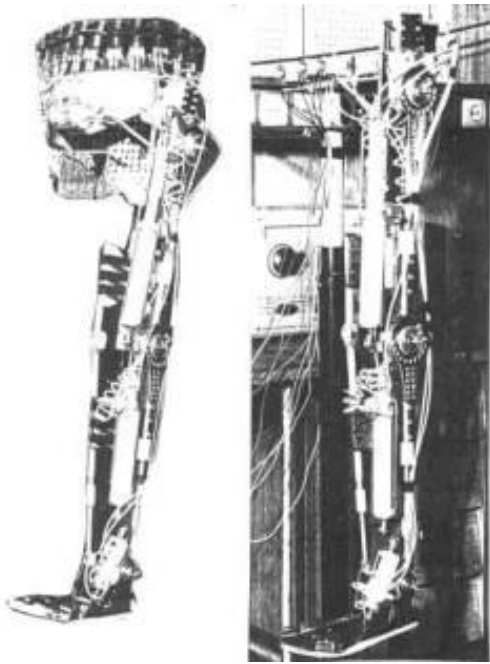


Figura 1.2. Ilustración del primer exoesqueleto, desarrollado en el Instituto Mihailo Pupin

Un diseño posterior fue desarrollado en la Clínica Ortopédica de Belgrado, con motores neumáticos y programado de forma electrónica. Este exoesqueleto tuvo un mayor éxito para la rehabilitación de personas parapléjicas o discapacidades similares.

Con el paso del tiempo, numerosos avances en el ámbito de la robótica se han utilizado para el desarrollo de exoesqueletos más potentes y precisos. Uno de los estudios más destacados es el del método ZMP o *Zero-Moment Point*, en el cual se analizan los movimientos que realiza un robot bípedo, de modo que la suma de sus fuerzas sea nula y se consiga la estabilidad (Vukobratovic, M., 2007).

Otros tipos de investigaciones sobre robótica humanoide pueden aplicarse al diseño de exoesqueletos, tales como el estudio de la trayectoria de las piernas durante la marcha, modelos de mecanismos sobre actuadores, compensación de fuerzas externas, cooperación entre partes de una articulación concreta, etc.

Una buena parte de los desarrollos propuestos en el naciente campo de la robótica fue llevada a cabo por el ya mencionado profesor Vukobratovic, el cual es reconocido por la mayoría de los expertos del sector. El profesor G. Hirzinger, actual director del Instituto de Robótica y Sistemas Dinámicos en Wessling, Alemania, declaró:

“El profesor Vukobratovic es para mí uno de los grandes pioneros de la robótica. Ha sido una las primeras personas en el mundo en darse cuenta de la importancia de las

dinámicas, incluso en una época en la que nadie que trabajara en la industria de la robótica pudiera pensar que los robots podrían llegar a ser algo más que mecanismos controlados por su posición. Estamos en un momento donde para los robots industriales se comienzan a tener cuenta las dinámicas que el profesor Vukobratovic predijo años atrás.”

En definitiva, los exoesqueletos han ido aumentando su complejidad y su sofisticación con el paso de los años. Las investigaciones en robótica y en anatomía humana han podido sustentar una buena base para comenzar a diseñar sistemas que proporcionan cada vez un mayor número de grados de libertad, así como de naturalidad en su movimiento.

Es evidente que otros factores, tales como el abaratamiento de los dispositivos electrónicos, los circuitos integrados y los materiales en los que se construye el exoesqueleto consiguen otorgar una mayor oportunidad para fabricar exoesqueletos comerciales a un precio cada vez más asequible.

1.1.3. Estado del arte

Exoesqueletos para la rehabilitación

Es importante aclarar que definir el estado actual de desarrollo para los exoesqueletos es complicado debido a la rapidez a la que avanzan las investigaciones más punteras. Además, las publicaciones al respecto no son demasiado accesibles, por lo que es probable que el estudio aquí realizado quede obsoleto al poco tiempo de salir a la luz.



Figura 1.3. Ejemplo de exoesqueleto ideado para la rehabilitación: Lokomat

Ya se han proporcionado algunos ejemplos de proyectos para desarrollar dispositivos de aumento de fuerza para usuarios con plenas capacidades, como los del proyecto DARPA o el X1 de la NASA. No obstante, en este trabajo se dará prioridad a exoesqueletos cuyo objetivo sea mejorar la vida de personas con movilidad reducida. Por otro lado, existen otros exoesqueletos que se utilizan como ayuda a la rehabilitación de personas que han sufrido lesiones, como el Lokomat o el ALEX. Estos son altamente útiles para mejorar las condiciones a largo plazo de los pacientes (Banala, K. et al., 2009).

Exoesqueletos comerciales

Se han nombrado algunos de los dispositivos más importantes enfocados en usuarios sanos de plenas capacidades. En este trabajo, sin embargo, es de especial interés mencionar aquellos exoesqueletos que estén destinados a aumentar la movilidad de personas que están limitadas por causas diversas.

Sistemas como el ALEX, ya citado con anterioridad, principalmente se centran en la rehabilitación de un paciente. El propio exoesqueleto cuenta con múltiples medidas de control y sensorización que proporcionan la información suficiente para poder monitorizar el progreso del paciente a tiempo real. Este tipo de dispositivos son muy

aparatosos y tienen una posición fija dentro del hospital donde se lleva a cabo la rehabilitación.

Es esencial que el exoesqueleto sea capaz de mantener cierta autonomía. Esta independencia permitiría al usuario utilizar el exoesqueleto como sustitutivo de la actual silla de ruedas en un futuro no muy lejano (Young, J. y Ferris, D., 2015). Son estos los dispositivos que se estudiarán como base de la investigación que se realiza en este trabajo.

- **Rewalk**



Figura 1.4. Exoesqueleto ReWalk

Existen numerosos laboratorios centrados en el diseño de nuevos dispositivos enfocados en la deambulaci3n de personas que no pueden hacerlo sin ayudas. Uno de los m1s importantes es el ReWalk, el cual actúa por medio de las rodillas y la cadera para permitir caminar y levantarse.

El ReWalk cuenta con sensores de intenci3n en la parte superior del cuerpo que se acciona por medio de peque1os cambios en el centro de gravedad del mismo, de modo que el dispositivo comienza a funcionar cuando el usuario pretenda moverse.

Este exoesqueleto es tambi3n utilizado en hospitales para mejorar la rehabilitaci3n de los pacientes, si bien su principal funci3n es actuar como un reemplazo de la silla de ruedas para personas que hayan sufrido lesiones en la m3dula espinal.

La gran limitaci3n de los exoesqueletos, las baterías, se transportan en la espalda del paciente. Sin embargo, termina siendo un aparato demasiado grande y pesado con una dificultad de utilizaci3n relativamente elevada. Adicionalmente, requiere que el usuario realice una gran aportaci3n al propio equilibrio, convirtiendo a las muletas en un

complemento casi obligatorio para aquellos que no puedan mantenerse erguidos por sí mismos (Zeilig, G. et al., 2012).

- **Ekso**

La empresa Bionics ha diseñado dos exoesqueletos distintos: el HULC y el Ekso.

El HULC (*Human Universal Load Carrier*) se basa en un dispositivo ligero que sirve para transportar cargas pesadas a una velocidad de marcha elevada. Funciona a partir de motores hidráulicos y está ideado únicamente para fines militares.

El exoesqueleto Ekso, sin embargo, está destinado para personas con capacidades de movilidad reducida. Al igual que el ReWalk, también actúa por medio de motores en las rodillas y en la cadera. En cambio, este dispositivo es más utilizado en la rehabilitación de pacientes con daños en la médula espinal, con una efectividad ya demostrada.

El Ekso, también llamado eLEGS, que proviene de *Exoskeleton Lower Extremity Gait System*, cuenta con un peso de 20 kilogramos y sus baterías tienen una duración de hasta 6 horas antes de ser recargadas.

En 2016, la compañía lanzó el exoesqueleto Ekso GT, el cual está más dirigido a la rehabilitación que al uso diario para el desplazamiento.

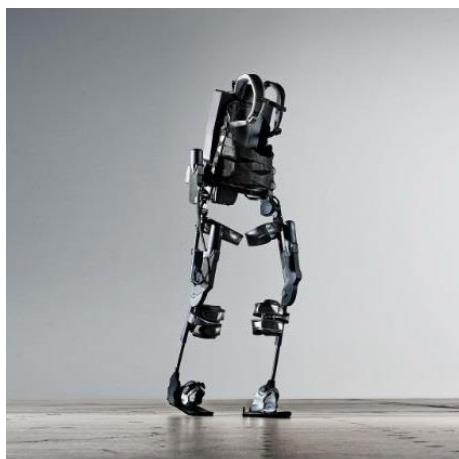


Figura 1.5. Exoesqueleto Ekso

- **HAL**

El HAL es otro modelo destacable, desarrollado por la empresa japonesa Cyberdyne. Cuenta con dos versiones de exoesqueleto: el HAL 3, un dispositivo de ayuda para el tren inferior; y el HAL 5, que implementa tanto piernas como brazos y torso.

Este exoesqueleto es especialmente interesante por su diseño ergonómico y más aún por su funcionamiento a través de las señales motoras EMG del propio usuario. La activación del exoesqueleto utiliza las propias señales que genera el cuerpo del usuario; de modo que es necesario que el sistema nervioso aún funcione correctamente, aunque después no sea capaz de transmitir las órdenes a los músculos, o bien no tenga la fuerza suficiente para moverlos por sí mismos.

El cuerpo humano genera una señal nerviosa que parte del cerebro cuando se piensa una orden concreta. Esa información en forma de señal bioeléctrica llega después al músculo que se pretende mover. El exoesqueleto HAL capta entonces esa orden por medio de sensores EMG y los reconoce como la intención de movimiento de un músculo concreto, activando así su mecanismo de forma artificial (Tsukahara, A. et al., 2010).

La empresa Cyberdyne es la única que implementa este sistema de señales EMG como principio de funcionamiento para exoesqueletos. A pesar de ello, la poca demanda fuera de los límites de Japón lo hace demasiado caro para la mayoría de potenciales usuarios y no goza de toda la popularidad que desearía (Kawamoto, H. et al., 2003).

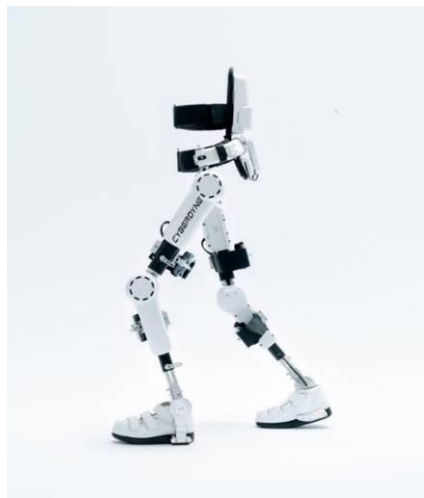


Figura 1.6. Exoesqueleto HAL 3

- **REX**

La empresa Rex Bionics también ha realizado su propio diseño realizado en fibra de carbono: el REX. Este dispositivo está ideado tanto para caminar como para ejercitarse de forma autónoma.

En este caso, el usuario tiene cada pie subido a una plataforma y apenas tiene que interactuar con la máquina para comenzar a caminar. El exoesqueleto se activa por medio de un mando controlador accionado por el propio usuario en el momento de iniciar la marcha.

Comparado con otros exoesqueletos, el REX tiene un tamaño considerable. No obstante, uno de los objetivos que buscaba la compañía era desarrollar un dispositivo que pudiera ser utilizado sin necesidad de andador o muletas complementarias como ocurre con el ReWalk o el HAL, entre otros. El hecho de poder tener las manos libres permite al usuario transportar pequeñas cargas o realizar ejercicios con el REX incorporado.

Como aspectos negativos más reseñables, se tiene el ya mencionado diseño abultado del exoesqueleto. Esto también deriva en una velocidad a veces demasiado lenta para una marcha fluida. Adicionalmente, el problema típico de la autonomía de estos dispositivos está muy latente en el REX, con unas baterías que apenas llegan a los sesenta minutos de duración.



Figura 1.7. Exoesqueleto REX

- **Honda**

La compañía Honda ha desarrollado dos prototipos destinados a la ayuda a la deambulación. El primero de ellos es el llamado *Stride Management Assist*, el cual pretende diferenciarse por la simplicidad y ligereza de su mecanismo, pues solo cuenta con dos motores que actúan sobre la cadera.

El segundo diseño se llama *Bodyweight Support Assist*. El principio de ayuda se basa en reducir el peso que deben soportar las piernas del usuario en el momento de utilizar escaleras o estando en cuclillas. Esto es especialmente útil para trabajadores que necesitan cargar su peso sobre las rodillas durante lapsos de tiempos prolongados.

Ninguno de estos dos exoesqueletos está diseñado para sustituir a la silla de ruedas, sino que su principal objetivo son personas con cierta falta de fuerza en las piernas, pero que aún pueden caminar sin necesidad de ayuda externa.

- **Indego**



Este exoesqueleto está siendo comercializado para personas con daños en la médula espinal. El Indego actúa como la mayoría de los dispositivos de ayuda completa: cuenta con dos motores rotatorios en la cadera y otro en cada una de las rodillas. La batería del mismo se soporta a la altura de la cintura.

El hecho de que este exoesqueleto fue desarrollado como un estudio de una universidad y no en un entorno industrial permite encontrar más detalles acerca del funcionamiento de sus actuadores. Estos pueden ser accionados tanto por medio de impedancias, a través del cual emula virtualmente un sistema de amortiguamiento de muelles; como por un sistema de altas ganancias,

Figura 1.8. Exoesqueleto Indego

donde se trata de forzar un determinado ángulo y trayectoria (Contreras-Vidal et al., 2016).

Las ventajas de este exoesqueleto es que permite al usuario intercambiar entre diferentes modos, tales como caminar, sentarse o mantenerse erguido. Además, el Indego presenta la característica de ser modular, de modo que permite mantener en funcionamiento ciertas partes del dispositivo si otras han dejado de hacerlo. También es posible dividir ciertas partes del mismo para poder transportarlo con mayor facilidad.

- **MB-ActiveKnee y ATLAS 2020/2030**

Actualmente, la empresa española Marsi-Bionics cuenta con un dispositivo e/n fase de evaluación clínica pensado para la rehabilitación de ciertas dolencias de rodilla, así como para la asistencia en la marcha en afectados de enfermedades como la esclerosis múltiple, entre otras.

El sistema MB-ActiveKnee es un exoesqueleto que ayuda a la marcha únicamente en la rodilla. Se trata de un mecanismo relativamente sencillo, con un motor de disco a la altura de la rodilla y sensores en una plantilla para los pies. Pesa 3 kg y cuenta con baterías propias, llegando a durar hasta 2.5 horas de uso continuo. Sin embargo, para poder utilizar el dispositivo el usuario debe poder mantener el equilibrio por su propia cuenta, además de tener una fuerza suficiente en los músculos como para moverse por sí mismo, aunque sea con dificultad.



Figura 1.9. Exoesqueleto MB-ActiveKnee

Aunque la empresa también cuenta con un exoesqueleto para personas mayores, el dispositivo que más ha sido utilizado es el ATLAS 2020/2030, dirigido únicamente para niños.

Este dispositivo cuenta pesa 14 kg y cuenta con un marco auxiliar para facilitar el equilibrio lateral del paciente. Este exoesqueleto suministra el movimiento mediante seis

motores en total: dos en las caderas, dos en las rodillas y otros dos en los tobillos para mover los pies; otorgando en total hasta 12 grados de libertad al usuario.

El ATLAS 2020/2030 tiene una duración de hasta 2.5 horas de uso continuado mediante baterías independientes aprovechando que, al ser pediátrico, los motores necesitan mover menos carga.

Este exoesqueleto ha sido utilizado como rehabilitación en el hospital de Barcelona Sant Joan de Déu para niños afectados de Atrofia Muscular Espinal, o AME, la cual les provoca una continua degeneración de sus músculos.



Figura 1.10. Exoesqueleto ATLAS 2020/2030

Componentes del diseño

- **Actuadores**

Los actuadores son aquellos que hacen que el exoesqueleto pueda moverse. Por ello, es importante tener en cuenta ciertos aspectos como el tamaño o el peso del mismo, así como la propia eficiencia de los motores.

Por lo general, la mayoría de diseños cuentan con motores eléctricos. Esto se debe a la facilidad con la que se pueden alcanzar los grados de rotación necesaria utilizando servomotores, algo ciertamente más complicado por medio de motores neumáticos o hidráulicos. Además, su actuación es más fiable y precisa que sus análogos no eléctricos (Young, J. y Ferris, P., 2015).

El gran problema para conseguir la autonomía de los exoesqueletos es sin duda la energía. Es necesario el uso de grandes cargas de batería, y transportarlas junto al usuario genera numerosas complicaciones en el diseño. El constante desarrollo de las baterías de litio y la reducción del gasto de energía de los componentes se convierte en algo prioritario a la hora de mejorar la autonomía.

Una solución sencilla, pero que no puede extenderse a todos los casos, es la de conectar el exoesqueleto mediante cables a una fuente de alimentación fija. Esta idea es especialmente útil en recintos cerrados, tales como un hospital o un domicilio. El problema de la energía, no obstante, seguiría presente para zonas exteriores. Así pues, debe mejorarse aún lo suficiente como para poder mantener la autonomía y la independencia del exoesqueleto durante varias horas seguidas, sin necesidad de recargar las baterías.

- **Sensores**

Es imprescindible el uso de sensores que capturen la información deseada, ya sea para controlar la posición, la fuerza o la propia rotación de cada parte del exoesqueleto. Las zonas más típicas de control son la cadera y las rodillas, es decir, las uniones de las extremidades del usuario (Jang, E. et al., 2010). Aquí se revisan las condiciones de

seguridad con respecto a la torsión de las piernas, y que a veces se complementa con un sensor de posición.

Es muy popular el uso de sensores en la planta del pie, capaces de controlar la fuerza que ejerce el usuario sobre el suelo y así obtener información sobre la posición y la etapa de deambulación en la que se encuentra. La forma más básica de este tipo de control sería un sensor de todo o nada (on/off) sobre el talón y sobre el primer dedo del pie (Kong, K. y Tomizuya, M., 2008).

Aunque no gozan de tanta popularidad, los acelerómetros y los giroscopios también se han dado cierto uso como medida complementaria de control.

Dentro del mundo de los sensores, es interesante mencionar de nuevo el exoesqueleto HAL, que es el único que cuenta con sensores de electromiografía, o EMG. Se han propuesto además sensores de electroencefalograma, o EEG, pero aún no han sido comercialmente utilizados para ningún diseño de exoesqueleto.

- **Materiales**

Es esencial definir el estado del arte acerca de los materiales con los que se fabrican los exoesqueletos hoy en día. En la elección del material se debe atender a dos requisitos: que sea ligero y que sea barato. Además, debe ser capaz de resistir la fuerza que se pueda ejercer durante su uso. Es por esta razón que se suelen utilizar metales para construir la estructura principal.

Normalmente se suele optar por alguna aleación de aluminio para construir las juntas del exoesqueleto. El material ideal es el titanio, el cual puede soportar bastante más fuerza sin perder su ligereza. Sin embargo, es mucho más caro que el aluminio.

Otra opción viable es utilizar plásticos reforzados con fibras tales como la fibra de carbono. Es menos pesado que el aluminio o el titanio, pero no es tan robusto. Es por esto por lo que la fibra de carbono solo se utiliza en exoesqueletos ligeros que ofrecen menos prestaciones.

1.1.4. Etapas de la deambulaci3n y la bipedestaci3n

Deambulaci3n

Para conocer el funcionamiento real de un exoesqueleto, es necesario conocer c3mo se mueve el propio cuerpo humano. Un exoesqueleto debe imitar lo m3ximo el movimiento de una persona cuando anda, cuando se sienta o cuando permanece de pie.

A la hora de caminar, una persona sana lleva a cabo una serie de flexiones y estiramientos de m3sculos que le permiten avanzar en la direcci3n que desea. Este movimiento se fundamenta en una serie de etapas representadas en la Figura 1.11.

La deambulaci3n se puede resumir como una secuencia de etapas continuas que se repiten indefinidamente formando ciclos. Estos ciclos pueden verse como una secuencia desde que el tal3n de pie contacta con el suelo hasta el siguiente contacto de este mismo tal3n.

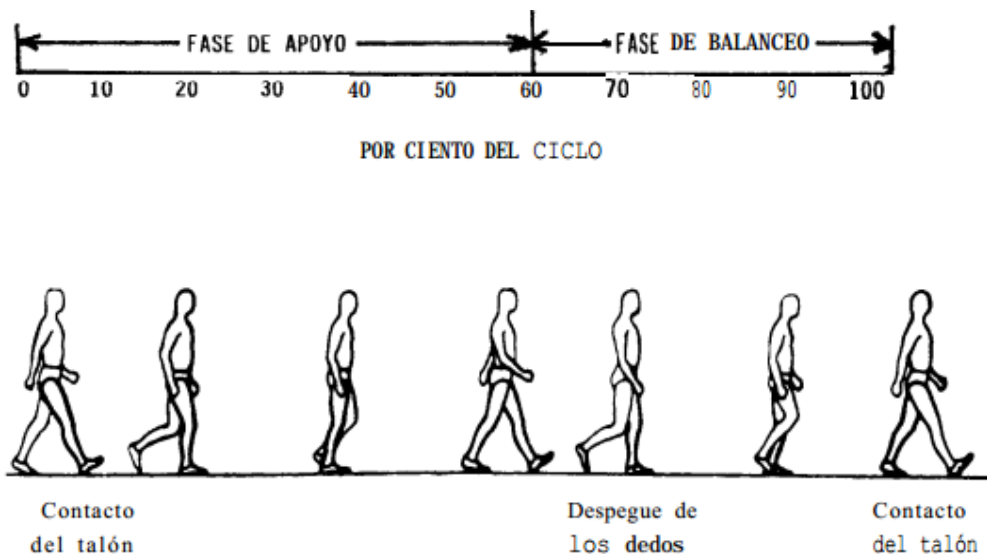


Figura 1.11. Etapas de deambulaci3n de una persona sana [Imagen cortes3a del blog Fisioterapia]

La marcha, del mismo modo, puede dividirse en dos etapas bien diferenciadas: la fase de apoyo y la fase de oscilaci3n. La fase de apoyo consta de alrededor del 60% del ciclo, mientras que la fase de oscilaci3n ocupa el 40% restante. Estas etapas suelen fraccionar en otras subfases, normalmente denominadas como inicial, media y final (Marco, C., 2001).

Sin embargo, es más recomendable repartir el ciclo de marcha en cuatro etapas, en función de qué pies están apoyados y cuáles están oscilando. Se ha de tener en cuenta que existen ciertos momentos al caminar en los que ambos pies están pisando el suelo.

De esta forma, el ciclo se puede dividir en cuatro fases o periodos diferenciados:

- **Primer periodo de doble apoyo:** da comienzo al ciclo de la marcha. Este comienza cuando el pie tomado como referencia hace contacto con el suelo con el talón, tal y como muestra la primera silueta de la Figura 1.11.
- **Primer apoyo unipodal o periodo portante:** el miembro tomado como referencia está apoyado en el suelo, mientras que el otro miembro (llamado contralateral) está oscilando.
- **Segundo periodo de doble apoyo:** los dos pies hacen contacto con el suelo. Es la etapa simétrica al primer periodo de doble apoyo.
- **Segundo apoyo unipodal o periodo oscilante:** el miembro tomado como referencia está oscilando y el contralateral está apoyado en el suelo. Es la etapa simétrica al periodo portante.

Otros factores como la velocidad de la marcha, el ángulo del paso o la longitud del paso son también importantes para definir la marcha de una persona.

Este tema será objeto de mayor estudio cuando se propongan soluciones para la sensorización de exoesqueletos para la deambulación en el apartado de Hardware.

Bipedestación

El término bipedestación en este documento alude a la ayuda que se proporciona al usuario del exoesqueleto para levantarse de un asiento o silla de ruedas, así como a volver a sentarse.

Del mismo modo que para la deambulación, es aconsejable dividir todo el proceso en distintas etapas en función de la flexión y la relajación de ciertos músculos. Se presentan aquí las fases para levantarse, sabiendo que el proceso de sentarse es el equivalente inverso. Una representación de esto viene indicada en la Figura 1.12. (Schenkman, M. et al., 1990).

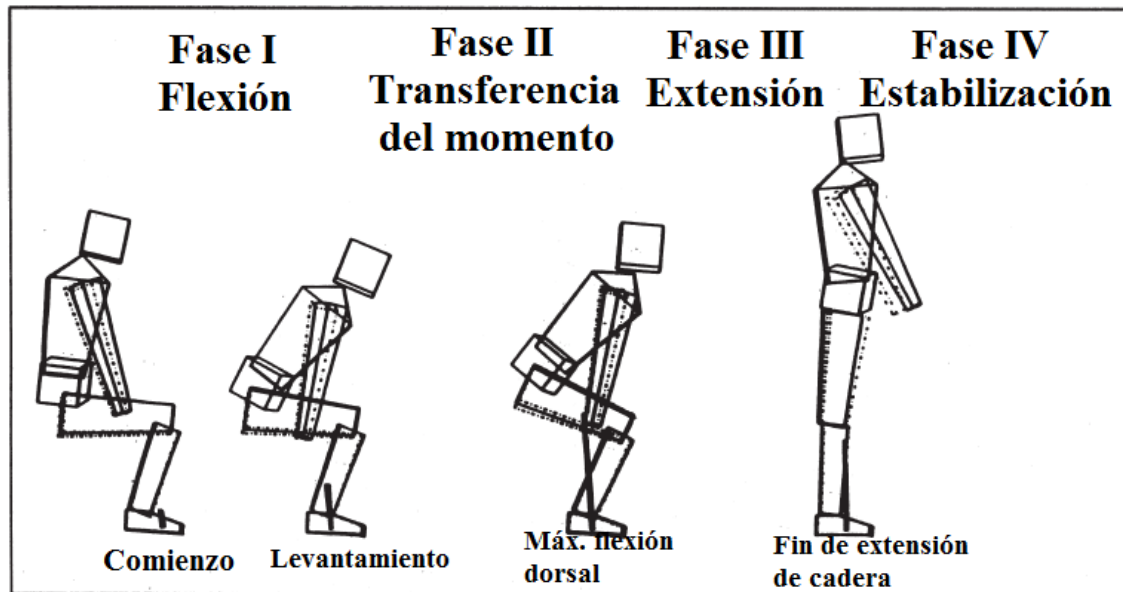


Figura 1.12. Etapas de la bipedestación

- **Fase de flexión.** Comienza con la iniciación del movimiento y termina cuando las nalgas se separan del asiento. Tanto el tronco como la pelvis, ya rotados, generan el momento que permite levantarse.
- **Fase de transferencia del momento.** Comienza desde que las nalgas se separan del asiento hasta que los tobillos están en su mayor estado de flexión. Es aquí cuando el momento generado en la fase anterior se transfiere al resto del cuerpo.
- **Fase de extensión.** Abarca desde el mayor estado de flexión de los tobillos hasta que la cadera deja de extenderse. Es en esta fase donde la persona pasa de un estado de flexión general a estar completamente erguida.
- **Fase de estabilización.** La cuarta y última fase comienza cuando la cadera ha terminado de extenderse y termina cuando toda acción relativa a la estabilización del cuerpo se haya completado. Este periodo marca el final del proceso de bipedestación, si bien no es fácil definir cuándo acaba, pues a menudo se producen pequeños balanceos sucesivos que suponen una pérdida momentánea del equilibrio.

Muchos de estos conceptos serán desarrollados en el apartado de Software con mayor detalle a la hora de establecer la sensorización y la propia implementación del movimiento del exoesqueleto.

1.2. Ámbito local

Para diseñar todo el sistema que se propone, se cuenta con numeroso material y recursos para documentarse. Sin embargo, a la hora de realizar la propia construcción del mismo, prácticamente todas las pruebas se han realizado en el laboratorio de Electrónica y Bioingeniería de la ETSIT de Valladolid.

En uso de este laboratorio ha dado acceso a una gran cantidad de componentes, la mayoría electrónicos, necesarios para la construcción del proyecto.



Figura 1.13. Armario de componentes del laboratorio

En la Figura 1.13 se muestra el armario principal en el que se guardan los componentes más comunes: resistencias, condensadores, diodos, circuitos integrados... Adicionalmente, se cuenta con un conjunto de cajones con otros útiles, tales como cables y conectores. Todo esto se apoya con otro armario con herramientas para serrar, soldar y cortar lo que se necesite.

Por último, el laboratorio cuenta con un puesto de trabajo compuesto por una fuente de alimentación configurable, un multímetro y un osciloscopio.



Figura 1.14. Fuente de alimentación ISO-TECH disponible en el laboratorio

La fuente de alimentación, de la marca ISO-TECH, cuenta con tres tomas distintas. La primera, el canal 1, es el denominado “Maestro”. Este será el que se utilizará para controlar la alimentación positiva con la que alimentemos los integrados durante los experimentos. El canal 2 es el “Esclavo”. Este depende del voltaje que hayamos configurado en el primer canal, y se utilizará en las pruebas para suministrar la alimentación negativa. Esto se hace mediante un puente entre la borna negativa del canal 1 y la borna positiva del canal 2.

La fuente también cuenta con una toma GND por defecto. Además, también suministra alimentación por una tercera toma, que no es configurable y siempre devolverá 5 V.

Este dispositivo, a diferencia del resto de los del laboratorio, estará presente en el prototipo final que se construirá. No obstante, todos son especialmente útiles para realizar las mediciones pertinentes y estudiar la parte electrónica del trabajo.



Figura 1.15. Multímetro disponible en el laboratorio

El multímetro de marca ISO-TECH es el dispositivo de color negro fotografiado en la Figura 1.15. Este equipo es especialmente útil cuando se desean realizar mediciones en un circuito, pues es capaz de obtener los valores de resistencia óhmica, voltaje o corriente existentes entre dos puntos cualesquiera de un circuito, sin tener que recurrir a operaciones manuales. El multímetro es una herramienta imprescindible en las tareas de diseño de la parte electrónica, si bien ya no será necesario una vez el prototipo esté en funcionamiento.



Figura 1.16. Osciloscopio Hameg disponible en el laboratorio

El último de los equipos con los que cuenta el puesto de trabajo es un osciloscopio. Este osciloscopio, de la marca Hameg, permite visualizar una señal en dominio del tiempo a la escala que se desee. Con la configuración adecuada, también es visible la transformada de Fourier en la misma pantalla, si bien no será necesario para el trabajo a realizar.

El osciloscopio cuenta con dos entradas independientes para sondas. En la pantalla se podrán visualizar las dos señales de forma simultánea, e incluso el resultado de ciertas operaciones básicas, tales como su suma o su diferencia.

Aunque el osciloscopio será aprovechado para el estudio de los sensores piezoeléctricos, su labor podrá ser reemplazado simplemente por el nivel de voltaje total recogido en el multímetro, aunque se pierda información al no visualizar la forma de la señal obtenida.

1.3. Objetivos

Con todo lo ya expuesto anteriormente, se plantean a continuación una serie de objetivos a realizar dentro del ámbito de trabajo presentado. Deben ser separados dentro de las dos ramas o problemas ya mencionados en esta introducción: la ayuda a levantarse y la ayuda a la deambulacion aplicada con el uso de un exoesqueleto de pierna.

En lo relativo a la ayuda a bipedestación:

- Realización de un estudio sobre las alternativas posibles.
- Diseñar una sensorización apropiada sobre el usuario, de modo que pueda transmitir la señal de forma inequívoca cuando desee levantarse o sentarse.
- Construir una maqueta que simule el funcionamiento de un exoesqueleto de pierna a tamaño real, accionado por los sensores anteriormente diseñados.
- Programar el software asociado por medio de Arduino que permita recrear el escenario completo.

Tal y como está planteado este problema y la solución que finalmente se aporta, el diseño está pensado para aquellas personas que no tengan capacidad para levantarse, pero aún

mantengan cierta motricidad en los músculos del tren inferior. Esto se debe a que es necesario transmitir una pequeña señal de intención para iniciar el sistema.

En lo relativo a la ayuda a la deambulación:

- Realización de un estudio sobre los lugares óptimos donde situar los sensores del exoesqueleto.

1.4. Estructura del documento

El trabajo está estructurado de la siguiente manera:

1. **Introducción.** Es el apartado ya expuesto. El principal objetivo es dar una idea clara del ámbito en el que se está trabajando, así como enseñar conceptos básicos necesarios para entender el problema que se pretende solucionar. Adicionalmente, se resumen una serie de datos relativos a los exoesqueletos, tales como su historia, su situación en el mercado y los componentes básicos de todo diseño. También se han explicado los medios de los que se han dispuesto y los objetivos principales de este estudio.
2. **Hardware.** En este apartado se reúnen todos los elementos del sistema denominados como *hardware*, es decir, que pueden tocarse y manipularse directamente. Así, al igual que en el apartado de *software*, se da una primera explicación sobre anatomía que ayude a entender las decisiones que se han tomado, y después se explican los componentes que forman todo el sistema. Se le da una especial atención a los sensores, los cuales son uno de los principales temas de este trabajo. De ellos se presenta una larga explicación, tanto de los sensores finalmente elegido como de las optativas presentes que se han barajado. El resto de elementos se explican después, con multitud de fotografías ilustrativas y características de diseño.
3. **Software.** Todo aquello que se programa a través de un ordenador se recopila en el apartado de *software*. Tras una breve explicación anatómica, se muestran los diagramas lógicos que imperan en el sistema diseñado. Finalmente, se comenta el código implementado en Arduino de una manera exhaustiva.

4. **Pruebas y ensayos.** Una vez construido el sistema, se procede a probarlo y a demostrar su correcto funcionamiento dividiendo el prototipo en varias partes. También se incluyen algunas pruebas no implementadas en la versión final del proyecto.
5. **Conclusiones.** Este apartado presenta una breve disertación acerca de la utilidad del sistema desarrollado, así como de enseñanzas que puedan haber aportado los resultados.
6. **Líneas futuras.** El contenido de este punto debe dividirse en dos temas. Primero, las líneas futuras de los propios exoesqueletos y hasta donde llegará el estado del arte en los años próximos. Segundo, las posibles líneas futuras de este trabajo y de cómo podría ampliarse y mejorarse el sistema actual.
7. **Anexos.** Conjunto de anexos de elementos y manuales utilizados durante el desarrollo del trabajo.
8. **Referencias.** Conjunto de artículos y trabajos científicos ajenos que han servido como referencia para la realización de este documento.

2. HARDWARE

2.1. Sensores

Para garantizar un pleno funcionamiento del sistema de control, es imprescindible capturar la intención del usuario de que quiere levantarse o sentarse. Esta señal puede ser recogida de múltiples formas en función de la manera en que esté implementado el sensor, así como el tratamiento que le demos a su respuesta.

Un sensor es un dispositivo que se ocupa de detectar cambios físicos o químicos y transformarlos en magnitudes eléctricas. La variable que recoge puede ser de diferentes características: temperatura, distancia, humedad, intensidad lumínica, movimiento... Los sensores no deben confundirse con los transductores, los cuales convierten una forma de energía a otra. Es decir, sensores necesitan de una alimentación de energía externa y no varían energía, sino magnitudes. El transductor, por tanto, suele incluir al sensor.

En definitiva, lo primero que se debe definir es qué tipo de sensor debe incorporar el sistema.

2.1.1. Anatomía y esfuerzo muscular

Tal como se ha propuesto en los objetivos, el fin último es el de transmitir una señal de intención de una persona que quiera levantarse o sentarse y simular con una maqueta el movimiento de un exoesqueleto.

Hay infinidad de maneras posibles de hacer transmitir una señal que indique el comienzo del proceso. La más sencilla de todas sería con la instalación de un pulsador que la propia persona accionara en el momento oportuno. Sin embargo, ese sistema se convierte en un proceso totalmente lineal y con un mínimo control del usuario. Además, podría no representar un movimiento natural acompasado con el propio esfuerzo de esa persona, sino que el exoesqueleto realizaría todo el esfuerzo sin atender al estado o la posición de quien lo porta.

El objetivo, por tanto, es dividir el proceso en distintas etapas, en cada una de las cuales se activase una especie de pulsador, o similar, que accionase una parte del proceso.

Esto sigue sin otorgar al usuario un completo control del movimiento del exoesqueleto, pero sí ofrece una sensación de decisión real sobre el mismo.

Una vez se descartan los pulsadores como método de control principal, se han de buscar soluciones que ofrezcan una experiencia más natural al usuario. Esto es, la intención del usuario deberá obtenerse del fruto del esfuerzo que hacen sus extremidades cuando quiere levantarse, aunque no sea capaz de conseguirlo sin ayuda de un exoesqueleto.

La instalación de sensores en el cuerpo se hace indispensable para obtener una señal fiable de control. Los exoesqueletos de pierna comerciales suelen ofrecer varias alternativas que, si bien son diferentes entre sí, guardan multitud de similitudes.

Es muy común realizar una sensorización mediante la presión ejercida en ciertos puntos del cuerpo, especialmente en las piernas y en los pies. Se realizará un estudio de los tipos de sensores con posibilidad de implementarse en función de la magnitud que miden. No obstante, para todos ellos se tratarán de implementar en el tren inferior del usuario. Entonces, se hace indispensable conocer bien cómo se constituyen los músculos de la zona de interés.

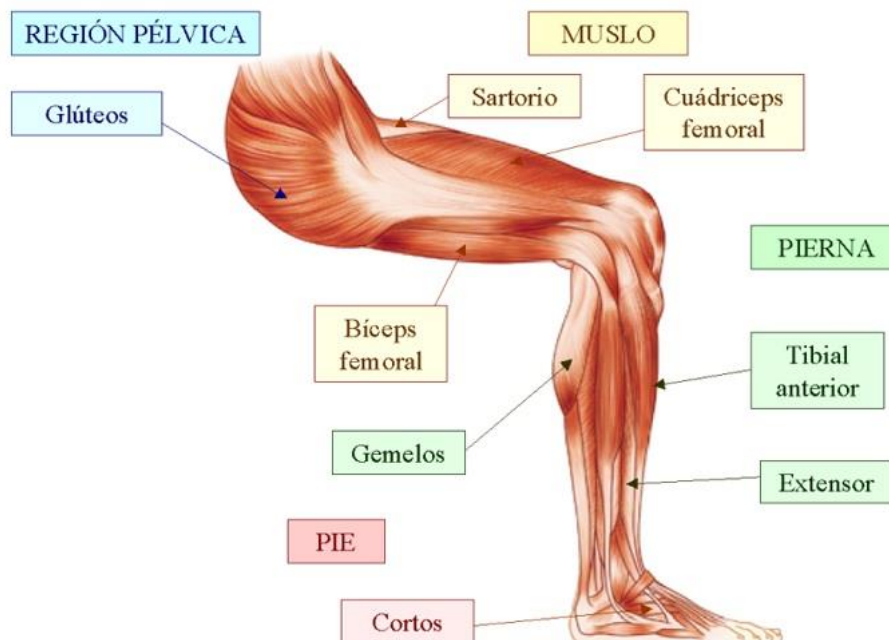


Figura 2.1. Músculos principales del tren inferior

En la Figura 2.1. se muestran los músculos más importantes de la pierna. Cabe destacar que existen otros músculos además de los expuestos, pero no son esenciales para el estudio actual.

Al iniciar el acto de bipedestación, el músculo que antes se contrae son los glúteos, seguido por el bíceps femoral. El tibial anterior se contrae para soportar parte del peso, hasta que el cuádriceps femoral realiza el esfuerzo principal para levantarse.

Una idea interesante a la hora de saber qué sensorizar es la contracción de estos músculos en cada etapa del proceso. Algunos de ellos aumentan su volumen en gran medida cuando deben realizar un movimiento que implica cierta fuerza. Además, en personas incapaces de levantarse sin ayuda, estos músculos se esfuerzan aún más que los de una persona que no necesita de asistencia externa.

Por tanto, un lugar significativo para la solución del problema serían los cuádriceps y el bíceps femoral, pues son los músculos que hacen la mayor parte del esfuerzo en la totalidad del proceso, y la instalación de dispositivos en ellos sería relativamente sencilla.

Del mismo modo, la medida de la presión ejercida en cada parte de la planta de los pies podría ser de especial interés. Se discutirá esta idea en apartados posteriores.

2.1.2. Clasificación de sensores según la magnitud medida

A continuación, se expone una clasificación de los sensores más importantes según la magnitud medida, junto con una breve explicación de su funcionamiento, algún ejemplo de transductor y los posibles usos para la implementación sobre un exoesqueleto.

- **Posición lineal y angular.** Mide la posición del sensor y devuelve un potencial. Un ejemplo típico es el potenciómetro analógico. Dado que se pretende obtener la intención del usuario sin pulsadores ni joysticks, este tipo de sensores no es el adecuado. Podría ser útil, sin embargo, para el control de la posición de los motores del mecanismo.
- **Velocidad lineal y angular.** Mide el ángulo de giro y la velocidad del sensor. Es el funcionamiento básico de los giroscopios. De nuevo, se tiene el mismo

caso que para los potenciómetros. No es adecuado para el trabajo a tratar, si bien resulta de interés medir esta magnitud para el control del equilibrio en los exoesqueletos.

- **Caudal.** Son dispositivos destinados a medir el caudal de un fluido a través de un conducto, tal y como ocurre con las turbinas en las centrales hidroeléctricas. En los exoesqueletos no hay fluidos circulando, por lo que un sensor de caudal se torna inútil en este sistema.
- **Temperatura.** Captan las variaciones de la temperatura, ya sea ambiente o de un aparato. Hay numerosos tipos de sensores de temperatura. Los más conocidos son los termistores (variación de la resistividad en función de la temperatura), el termopar o el termostato. A priori, el usuario no va a ser capaz de realizar variaciones de la temperatura para indicar su intención de movimiento, por lo que no sirve en para un exoesqueleto. Podría funcionar, no obstante, como sensor de control para dispositivos que no pueden operar a partir de una cierta temperatura.
- **Presencia.** Detectan variaciones en el movimiento, ya sea mediante infrarrojos, ultrasonidos, etc. Es posible captar la intención con un sensor de presencia activable por el propio usuario, pero el mecanismo no resultaría natural. Por lo tanto, los sensores de presencia quedarían descartados.
- **Sonido.** Los sensores acústicos detectan cambios en el sonido para producir una respuesta, tal y como sucede con un micrófono. Es posible implementar un sistema con un sensor acústico y tratar la señal para que reaccionara a un cierto patrón relacionado con la voz. Con todo, no sería una intención coherente con el movimiento natural del cuerpo.
- **Luz.** Del mismo modo que es posible captar cambios en el sonido, también se puede detectar la variación en la intensidad lumínica recibida por el sensor. Es el mecanismo utilizado por los fotodiodos o las fotorresistencias. Sin embargo, también se descartan del estudio por no reflejar un cambio de magnitud natural en la bipedestación.
- **Presión.** Basado en el efecto piezoeléctrico, determinados cristales adquieren polarización eléctrica cuando se ejerce una cierta tensión mecánica sobre él. El movimiento que se ejerce mediante la bipedestación conlleva cierta tensión y relajación de los músculos en momentos determinados. Esos cambios de

presión en ciertos puntos podrían ser de utilidad a la hora de recoger una intención fiable del usuario. Se discutirán los sensores piezoeléctricos más adelante con un caso práctico.

- **Deformación.** Los sensores de deformación, típicamente las galgas extensiométricas, basan su funcionamiento en el efecto piezorresistivo, en el cual se aprovecha la propiedad de ciertos materiales de cambiar el valor nominal de su resistencia al ejercer una cierta deformación sobre él. Un ejemplo muy común es el de las básculas convencionales, las cuales utilizan una, dos o incluso cuatro galgas extensiométricas.

2.1.3. Sensores piezoeléctricos

Funcionamiento e implementación

Un sensor piezoeléctrico es capaz de convertir magnitudes de presión en magnitudes eléctricas. En este trabajo se utilizará el sensor RS Pro BM15015-06HC, ilustrado en la Figura 2.2., el cual es capaz de cumplir con los requisitos básicos para la investigación.

El sensor RS Pro BM15015-06HC se caracteriza por su reducido tamaño y su facilidad de colocación. Sin embargo, es un dispositivo bastante frágil y su alta sensibilidad puede causar problemas para captar la intención del usuario al levantarse. Además, este sensor funciona como transductor, esto es, no necesita de una alimentación externa para convertir a magnitudes eléctricas.

A pesar de todo esto, el sensor debe llevar implementado un circuito de acondicionamiento, pues la señal que devuelve es demasiado pequeña como para introducirla directamente en el microcontrolador Arduino. Se utilizarán dos sensores distintos simultáneamente, uno para cada pierna.

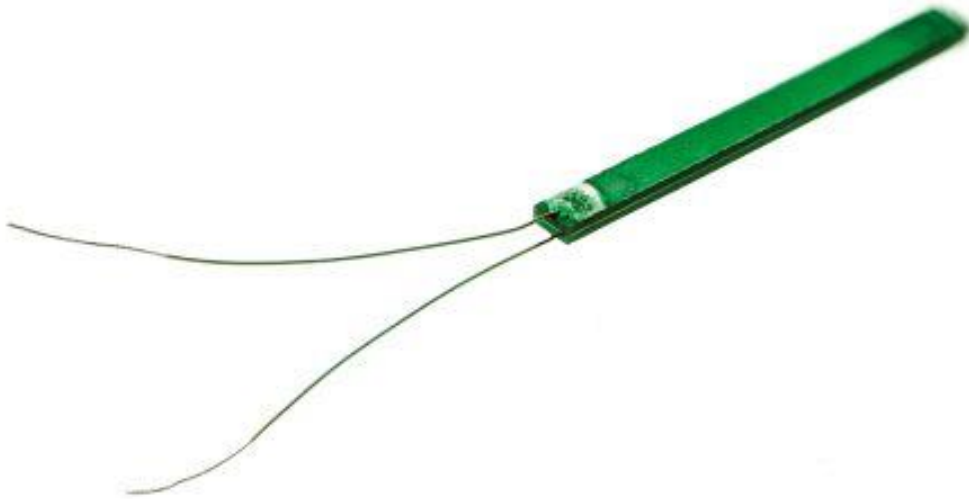


Figura 2.2. Sensor piezoeléctrico RS Pro BM15015-06HC

Una fotografía real de uno de los piezoeléctricos a utilizar se muestra en la Figura 2.3. Nótese que se le ha revestido con plástico para protección y mejor instalación.

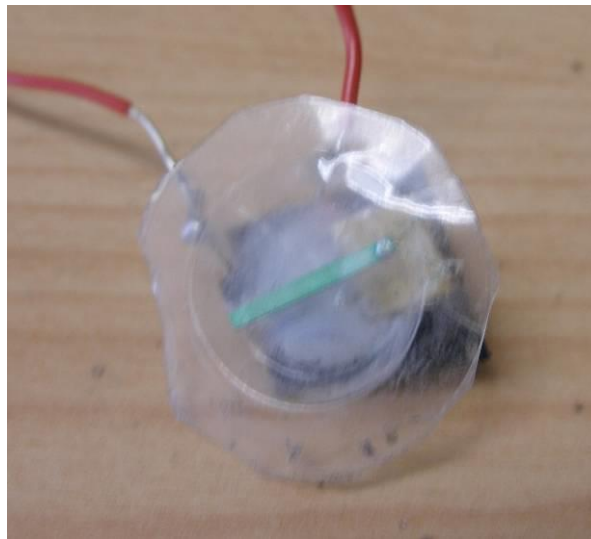


Figura 2.3. Fotografía del sensor piezoeléctrico RS Pro BM15015-06HC

Dado que el circuito de acondicionamiento de este sensor ya fue estudiado en otro trabajo, se realizará a continuación un breve resumen del tratamiento necesario que se le ha de proporcionar a la señal de entrada.

- **Amplificación.** La señal que proporciona el sensor es de una amplitud demasiado pequeña como para ser tratada directamente y debe ser amplificada con un integrado TL082CN mostrado en la Figura 2.4. Este integrado cuenta con dos

amplificadores operacionales que se alimentarán a ± 5 V, de modo que es posible realizar la amplificación de las señales de entrada de ambos sensores.

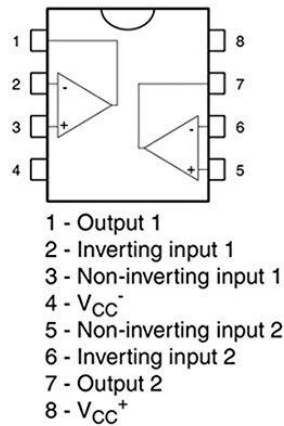


Figura 2.4. Diagrama del integrado TL082CN

Para la amplificación se utilizará una configuración representada en la Figura 2.5. de modo que la ganancia de la etapa será $1 + \frac{R_2}{R_1}$. La señal de entrada es muy pequeña, por lo que la ganancia introducida deberá ser muy elevada. En este caso, $v_{out} = \left(1 + \frac{R_2}{R_1}\right) v_{in} = 101v_{in}$.

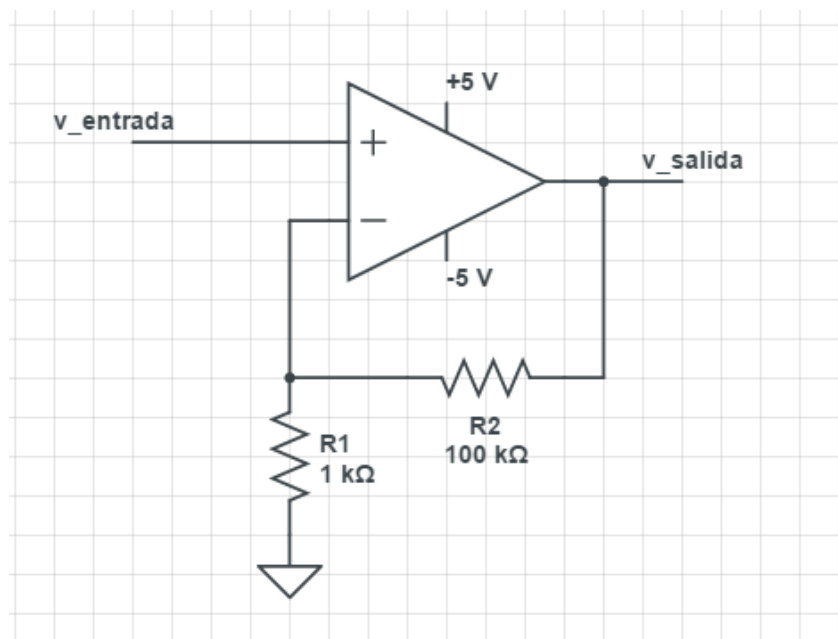


Figura 2.5. Diagrama del amplificador del sensor piezoeléctrico

- **Detector de envolvente.** La señal amplificada que se obtiene es una sucesión de pulsos que oscilan muy rápidamente. Es altamente recomendable obtener un valor más constante cuando el sensor se active. Al atravesar la señal por un detector de envolvente, eliminará en gran medida esas rápidas oscilaciones y mantendrá la propia envolvente de la señal.

En la Figura 2.6. se muestra el diagrama del detector de envolvente, compuesto por un diodo 1N4148, una resistencia de $15\text{ k}\Omega$ y un condensador de 100 nF .

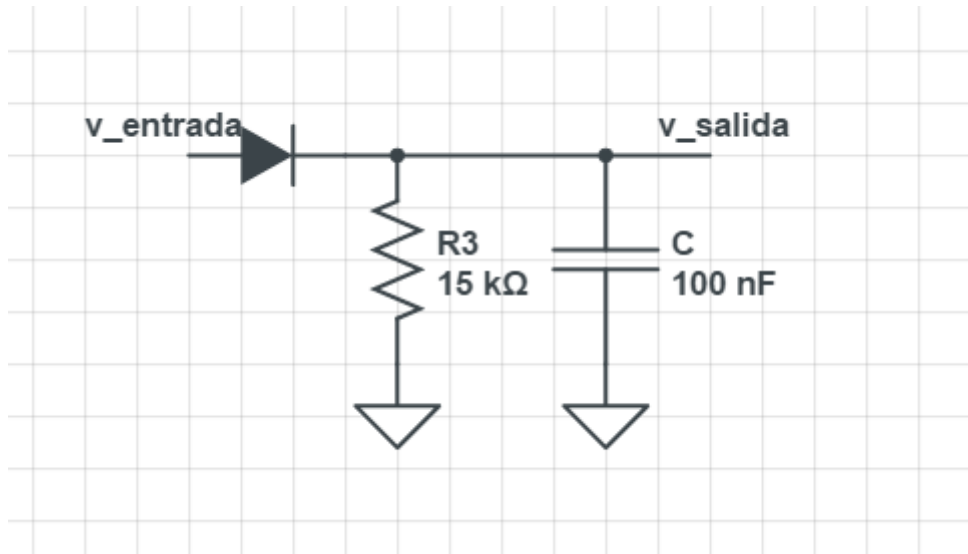


Figura 2.6. Esquema del detector de envolvente

- **Filtro paso bajo.** Se trata de un filtro pasivo que se encarga de eliminar tonos no deseados de altas frecuencias. En la Figura 2.7. se indica el diagrama esquemático del filtro RC, con una resistencia de $100\ \Omega$ y un condensador de 10 nF . Con estos datos resulta una frecuencia de corte de unos 160 kHz .

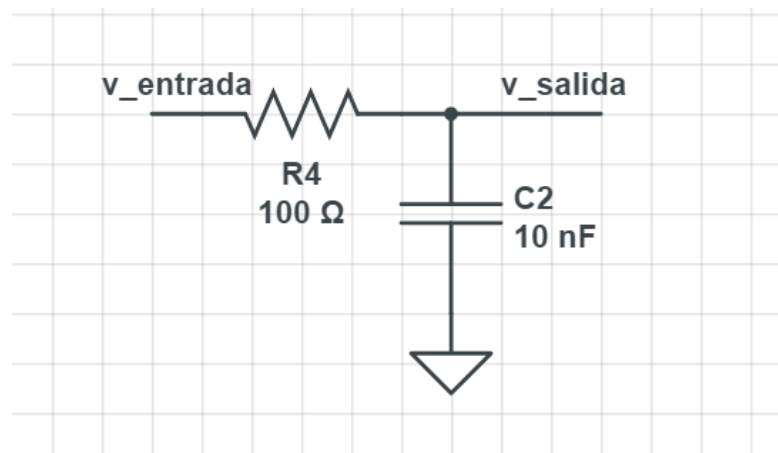


Figura 2.7. Esquema del filtro paso bajo

- **Etapa inversora.** Finalmente, la señal se hace atravesar por una etapa inversora que invierte las amplitudes finales, de modo que la señal está en alta cuando se presiona (5 V) y en baja cuando no se presiona (0 V).

Esta etapa se realiza con un integrado CD4011BCN, el cual cuenta con cuatro puertas NAND. Solo se utiliza una puerta para cada sensor. En la Figura 2.8. se muestra el esquema de la etapa inversora.

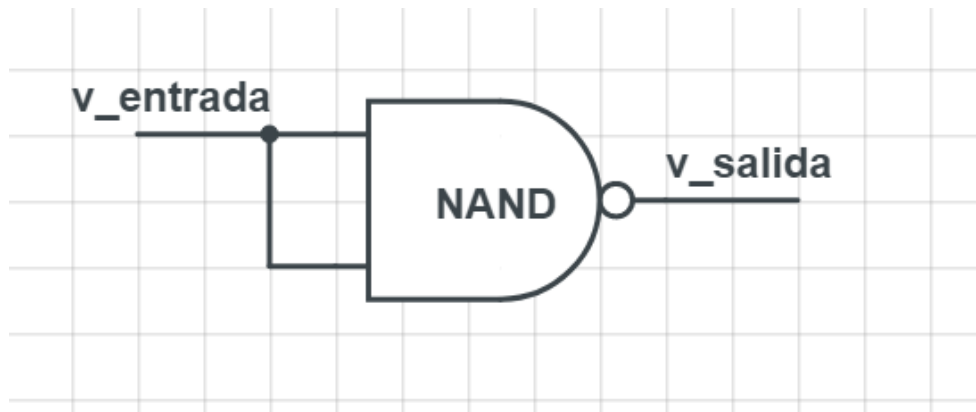


Figura 2.8. Esquema de la etapa no inversora

La señal, una vez ha atravesado por todas las etapas explicadas, se puede introducir como entrada digital en el microcontrolador Arduino. También se conecta una sonda del osciloscopio para monitorizarla a tiempo real.

En la Figura 2.9. está representado el esquema completo de acondicionamiento del sensor piezoeléctrico.

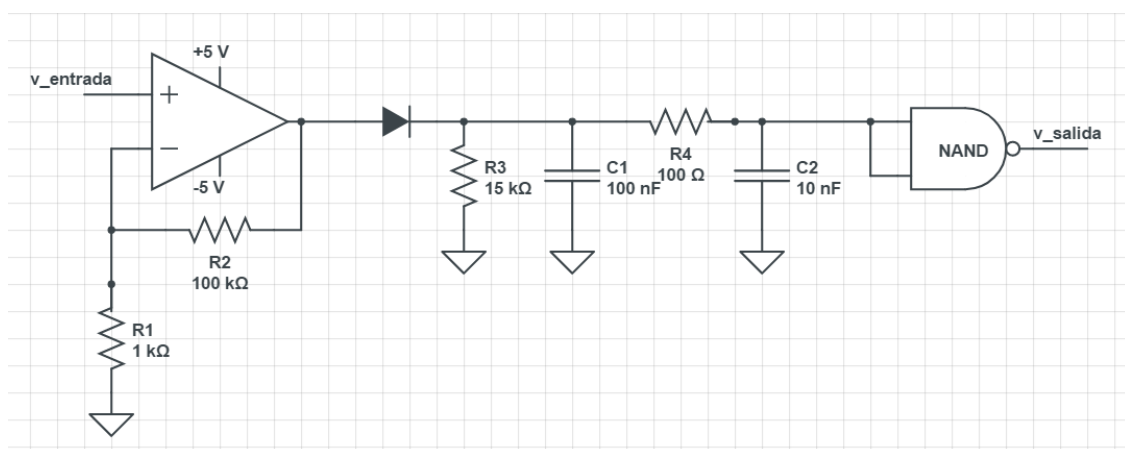


Figura 2.9. Esquema completo del acondicionamiento del sensor piezoeléctrico

Se muestra en la Figura 2.10. una fotografía del circuito completo para ambos sensores piezoeléctricos implementado en el laboratorio. Los dos cables rojos salientes por la izquierda corresponden a las entradas positivas de cada sensor, mientras que los cables rojos salientes por la derecha indica la señal ya acondicionada.

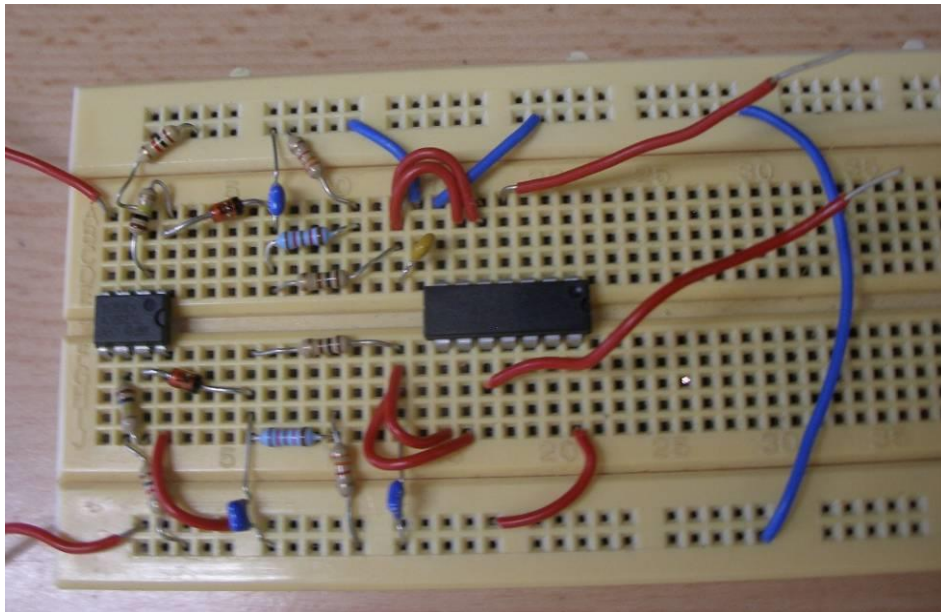


Figura 2.10. Fotografía del circuito de acondicionamiento de ambos sensores piezoeléctricos

El objetivo de utilizar un sensor piezoeléctrico es el de comprobar la viabilidad de este tipo de sensores a la hora de implementar un sistema de control para un exoesqueleto. Para ello, es esencial comprender que el sensor se activa al rozar o presionar levemente su superficie, por lo que es altamente sensible y no puede soportar un gran esfuerzo, a diferencia de los sensores de deformación.

Resultados del experimento

La primera idea de la instalación de los sensores piezoeléctricos estaba dirigida hacia la parte delantera de la pierna, en el cuádriceps. Mediante una cinta elástica alrededor de la pierna se puede sujetar el sensor en la posición que se desee. Los resultados de este experimento, sin embargo, no fueron los más alentadores.

Situando el sensor en cualquier parte del cuádriceps, desde el punto más cercano a la rodilla hasta casi llegando a la ingle, el sensor piezoeléctrico devuelve siempre una señal en alta, pues siempre está en contacto con la piel y se hace imposible distinguir cuándo se quiere iniciar la bipedestación.

Fijándose con mayor atención a cómo se contraen los músculos al comenzar a levantarse, se notó un gran cambio de volumen en el bíceps femoral, en la parte trasera de la pierna, justo encima de la articulación de la rodilla. Instalando el sensor en ese punto sí se podía recibir un cambio de señal en la primera etapa de la bipedestación. No obstante, cuando la persona finalmente se levantaba, el sensor se caía de la cinta al no estar completamente en contacto con la piel. Si se apretaba más para evitar este movimiento, surgía de nuevo el mismo problema que para el cuádriceps: el sensor siempre devolvía un voltaje de 5 V.

Así, se llega a la conclusión de que un sensor tan sensible como el piezoeléctrico utilizado no es de utilidad para cumplir con el objetivo de captar una señal fiable de que el usuario quiere levantarse o sentarse. Con todo, existe una posibilidad de dar uso a estos sensores como complemento del sistema principal.

2.1.4. Galgas extensiométricas

Funcionamiento y características

Las galgas extensiométricas son sensores que responden a la deformación del mismo, basándose en el efecto piezoeléctrico. A modo de ejemplo, las básculas convencionales se aprovechan de este efecto para calcular la masa de un objeto. Una vez conocidas las limitaciones relativas a la fragilidad del sensor piezoeléctrico RS Pro BM15015-06HC, es necesario introducir un nuevo concepto para realizar una comparativa entre ambos sensores.

Considerando el sensor como un hilo metálico de resistividad ρ , longitud l y sección transversal A , se puede expresar su resistencia eléctrica R como:

$$R = \rho \frac{l}{A}$$

El cambio de alguna de estas magnitudes hará variar la resistencia eléctrica R . Aplicando una fuerza F , y en función del material a deformar, se puede cambiar el valor de la longitud l siempre que no se supere su límite elástico (Figura 2.11.).

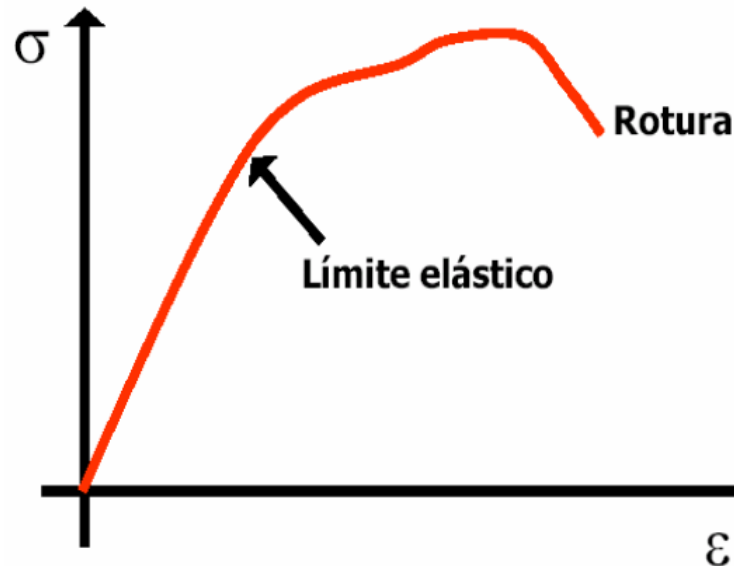


Figura 2.11. Relación entre el esfuerzo ejercido σ y la deformación ϵ de un material

Este cambio viene dado por la llamada ley de Hooke, donde E es el módulo de elasticidad del material (o módulo de Young), σ es el esfuerzo (medido como tensión, en N/m^2) y ϵ la deformación unitaria (como la diferencia de longitud entre la longitud inicial):

$$\sigma = \frac{F}{A} = E\epsilon = E \frac{dl}{l}$$

Es decir, el módulo de Young representa la deformación del material en función del esfuerzo ejercido. Si el módulo E es grande, el material se resistirá a deformarse y resultará un material frágil.

La instalación de la galga viene representada en la Figura 2.12., vista desde un lateral y desde la parte superior. El montaje más simplificado se compondría de una base muy delgada de un material no conductor y flexible. Sobre esta base va adherido un hilo metálico conductor, con los terminales en sus extremos. Este hilo actuará como un resistor variable en función de la fuerza vertical ejercida. El método habitual para acondicionar una galga extensiométrica suele ser mediante un puente Wheatstone (Alzate, E. et al., 2007).

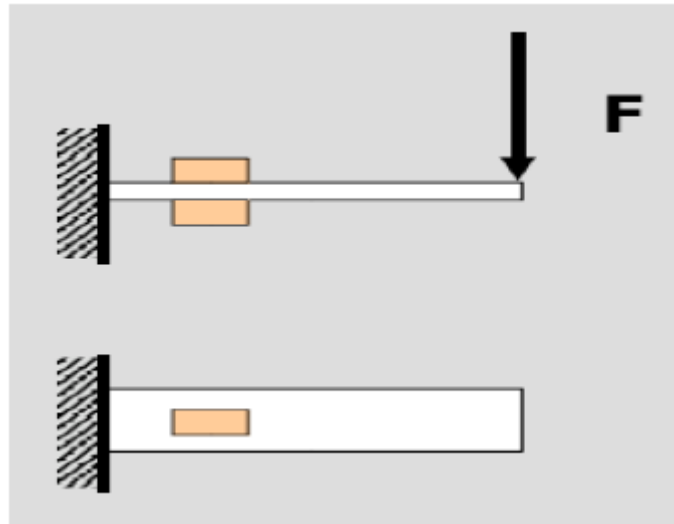


Figura 2.12. Montaje básico de una galga extensiométrica

En este trabajo se reutilizarán cuatro galgas extensiométricas extraídas de una báscula convencional. En la Figura 2.13. se muestra una fotografía de una de las galgas junto con la carcasa de protección de plástico.



Figura 2.13. Fotografía de la galga extensiométrica a utilizar, con la protección de plástico

Tanto en esta fotografía como en las mostradas en la Figura 2.14. es posible ver que el sensor dispone de tres salidas: el cable negro, el cable blanco y el cable rojo. Habitualmente las galgas extensiométricas cuentan con solo dos terminales, por lo que se deduce que uno de ellos actuaría como una salida intermedia para la circuitería de la báscula original.

Midiendo las resistencias entre estos tres cables, tenemos los siguientes resultados:

- Cables blanco-negro: 2000 Ω
- Cables blanco-rojo: 1000 Ω
- Cables rojo-negro: 1000 Ω

Es decir, que el cable rojo está en un punto medio entre los otros dos terminales. Para nuestro caso, una mayor resistencia aporta una sensibilidad del sensor más elevada, por lo que es preferible utilizar los terminales blanco-negro ($R_0 = 2000 \Omega$).



Figura 2.14. Fotografías de la galga extensiométrica, sin la protección de plástico

Estas galgas extensiométricas no pueden ser utilizadas tal cual se presentan en la fotografía anterior. Si se pretende medir la presión que ejerce un usuario sobre la parte delantera o trasera de sus pies será necesario monitorizarlo de la manera más efectiva y, al mismo tiempo, ergonómica.

Por ello, se ha procedido a instalar cada uno de los sensores en unas zapatillas que deberá ser puestas para activar el sistema. Se han elegido unas zapatillas ligeras principalmente por razones económicas, pero se deseaba también un calzado con una suela fina para que no amortiguara el peso sobre los sensores. Sin embargo, se verá más adelante que esta cualidad no es demasiado importante para el sistema, pues se ha comprobado con otro calzado más grueso que la presión puede transmitirse casi sin pérdidas hasta el microcontrolador. Por tanto, se podría afirmar que la instalación podría funcionar sin problemas en otro tipo de calzados.



Figura 2.15. Zapatillas con los sensores instalados a las suelas

En la Figura 2.15 se aprecia cómo han sido adheridos a las suelas las galgas extensiométricas. La altura a la que se han instalado es especialmente importante, pues es en estos puntos donde la presión es mayor al cambiar de postura. Véase la Figura 2.16. Aunque se marquen los puntos de mayor presión durante la marcha, son igualmente válidos para la bipedestación. De hecho, las galgas delanteras se han colocado en el punto 3 y las galgas traseras en el punto intermedio entre el 5 y el 6.

También se han reforzado con un trozo alargado de corcho para que el usuario no se desestabilice al levantarse, por la altura adicional que otorgan los sensores.



Figura 2.16. Puntos de mayor presión durante la deambulación (Wu, H. et al., 2016)

Acondicionamiento

Para estudiar de manera óptima el acondicionamiento de las galgas extensiométricas será necesario definir el valor de la resistencia que ofrece en función de lo que se deforme. Se define la resistencia de la galga como (García Casado, FJ., 2010):

$$R_g = R_0(1 + K\varepsilon)$$

Donde K es el cociente de la variación de la resistencia dR/R entre la variación longitudinal dL/L , y que se conoce como coeficiente de galga. El valor ε viene dado por la deformación dL/L . Todo ello va multiplicado por un factor R_0 , que indica el valor de la resistencia de la galga cuando no hay deformación alguna o, dicho de otro modo, cuando no se ejerce presión sobre ella. Recordemos que, en este caso, $R_0 = 2 \text{ k}\Omega$.

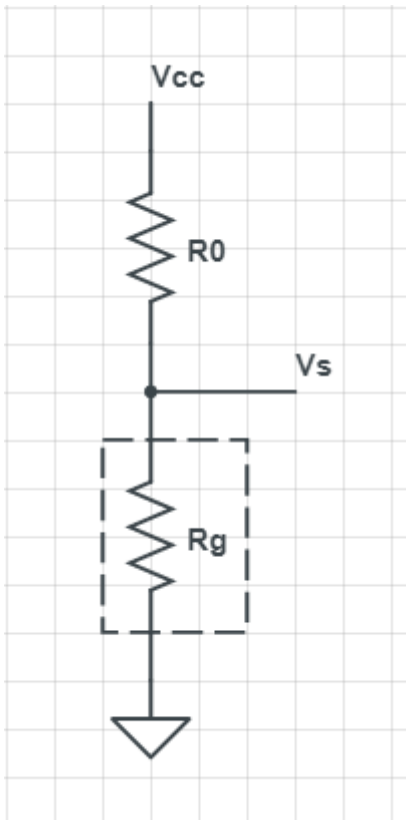


Figura 2.17. Divisor de impedancias para una galga
la señal que se obtiene es V_s , la cual después deberá ser amplificada para poder ser tratada de forma digital.

A la hora de acondicionar una galga extensiométrica se deben tener en cuenta distintos aspectos a controlar. Se ilustrará a continuación con un divisor de impedancias sencillo en el que se incluye una galga sustituyendo a una de las resistencias.

En la Figura 2.17. se muestra el funcionamiento del divisor de impedancias, donde se coloca una resistencia de valor R_0 junto con la galga de valor $R_g = R_0(1 + K\varepsilon)$. La

Resolviendo el circuito, se obtiene que:

$$V_s = \frac{1 + K\varepsilon}{2 + K\varepsilon} V_{cc}$$

Uno de los factores deseables en este acondicionamiento es que la señal obtenida tenga un valor constante, un offset, lo más pequeño posible. De esta forma, cuando posteriormente se quiera amplificar el Arduino pueda tratar la señal sin saturarse. En este

caso, cuando la deformación es nula ($R_g = R_0$), V_s es distinto de cero. Por tanto, el divisor de impedancias tiene offset, lo cual no es algo deseable.

Otro factor importante es la sensibilidad, definida como $S = dV_s/d\varepsilon$. Lo ideal es que esta sensibilidad no dependa de la deformación ε , pues esto implicaría que los cambios en la resistencia de la galga no serían constantes. Para el divisor de impedancias la sensibilidad es la siguiente:

$$S = \frac{dV_s}{d\varepsilon} = \frac{K}{2 + k\varepsilon^2} V_{cc}$$

Depende de ε , por lo que no es lineal. De forma general es aconsejable que, además de constante, el valor de la sensibilidad sea lo más grande posible.

El último factor a tener en cuenta es la influencia de la temperatura en la medición. De esta variable es importante saber que siempre va a estar presente a menos que se utilicen varias galgas simultáneamente para un mismo acondicionamiento.

Una vez conocidos los parámetros de calidad de un circuito de acondicionamiento para una galga extensiométrica, se estudia a continuación la configuración más habitual para este tipo de sensores: el puente de Wheatstone.

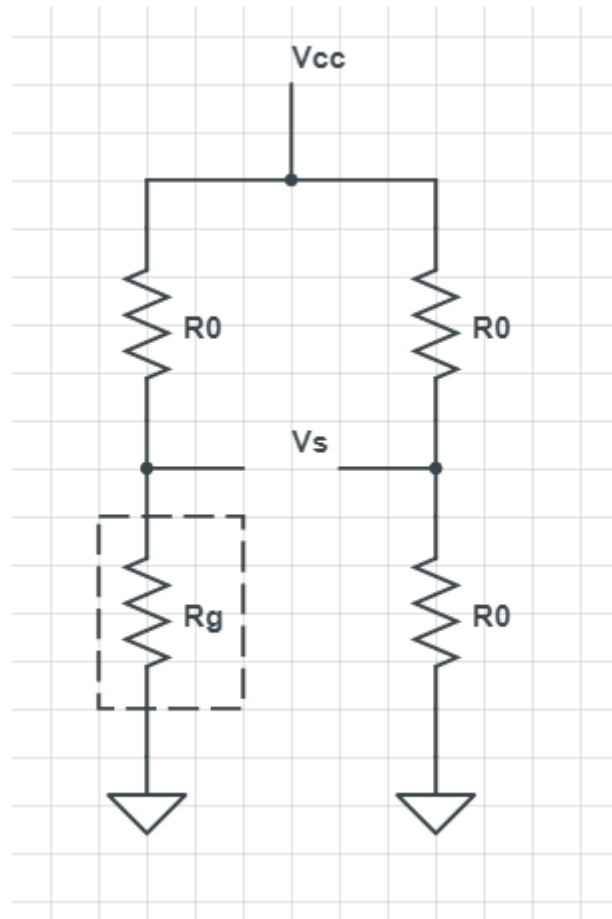


Figura 2.18. Puente de Wheatstone para una galga extensiométrica

En la Figura 2.18. se muestra la configuración puente para una sola galga extensiométrica. En este caso, resolviendo el circuito se tiene que:

$$V_s = \frac{K\varepsilon}{2(2 + K\varepsilon)} V_{cc}$$

Siendo ahora V_s la diferencia de potencial entre los dos puntos intermedios de ambas ramas. En este caso, cuando la deformación ε es nula, el voltaje de salida es también nula. Esto quiere decir que el offset es cero, lo cual es un factor deseable a la hora de amplificar la señal.

En cuanto a la sensibilidad, se obtiene lo siguiente:

$$S = \frac{dV_s}{d\varepsilon} = \frac{K}{(2 + K\varepsilon)^2} V_{cc}$$

Aparece la deformación ε en la expresión de la sensibilidad, por lo que los valores a obtener no serán lineales.

El puente de Wheatstone es una solución eficaz para el prototipo que se desea construir. Sin embargo, que la salida V_s no sea lineal puede acarrear ciertos problemas para ciertos niveles de tensión, especialmente tras haber amplificado la señal. Por tanto, se propone a continuación una configuración puente lineal por medio de un amplificador operacional dentro del circuito acondicionador.

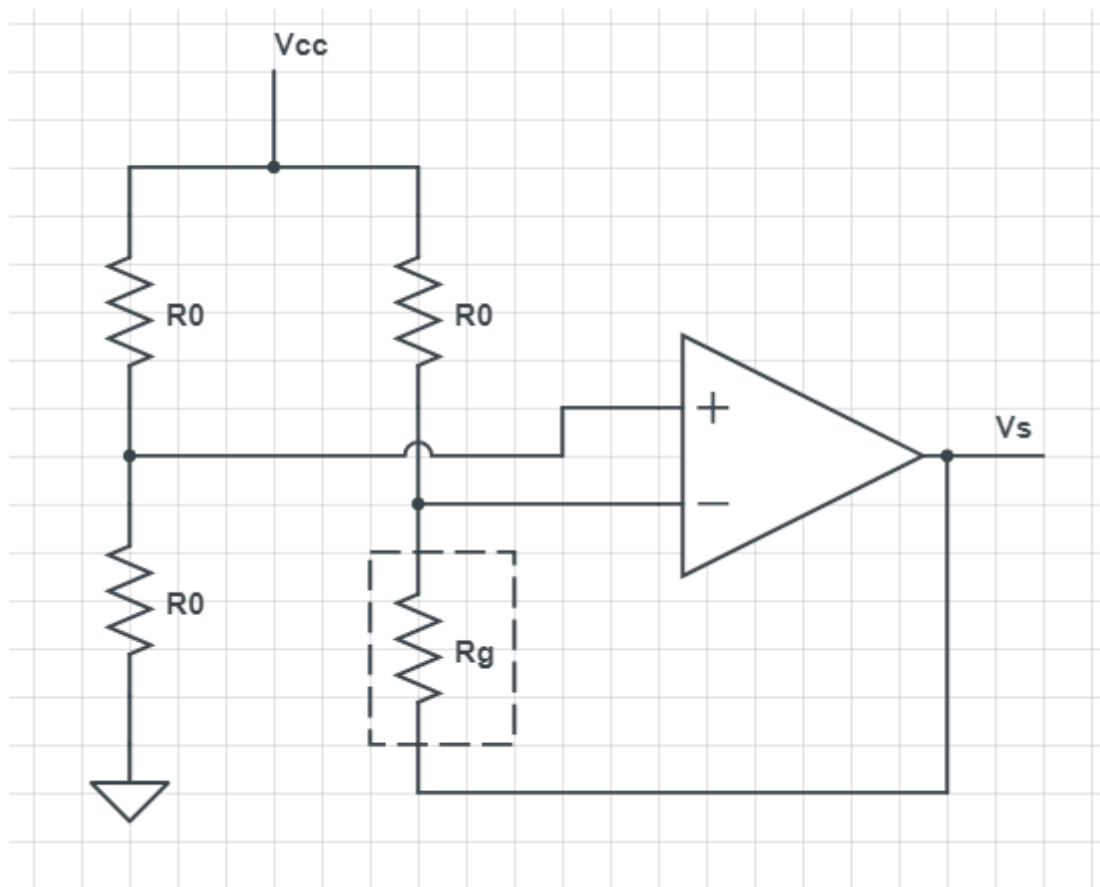


Figura 2.19. Puente linealizado con amplificador operacional para una galga extensiométrica

Se recuerda que el valor de $R_g = R_0(1 + K\varepsilon)$, por lo que la galga extensiométrica y las resistencias fijas están relacionadas por el valor R_0 .

En este caso, la señal de salida es de valor negativo:

$$V_s = -\frac{K\varepsilon}{2}V_{cc}$$

Para una deformación nula, el valor de salida V_s es igual a cero. Es decir, el circuito elimina todo offset. En cuanto a la sensibilidad se tiene:

$$S = \frac{dV_s}{d\varepsilon} = -\frac{K}{2}V_{cc}$$

Por tanto, es un circuito que ofrece linealidad en la salida. Sin embargo, el circuito también está influenciado por la temperatura. Se necesitaría una galga extensiométrica adicional para poder paliar esos efectos. En este trabajo, no obstante, no se tendrán en cuenta a la hora de realizar mediciones, pues el prototipo trabajará en condiciones de temperatura relativamente constantes y sin llegar a extremos.

Una vez construido el puente linealizado con un amplificador operacional, es necesario amplificar la débil señal que obtenemos en V_s . Por esta razón era tan importante obtener una señal con un offset nulo. Así, se realiza la amplificación del mismo modo que se hizo con el sensor piezoeléctrico.

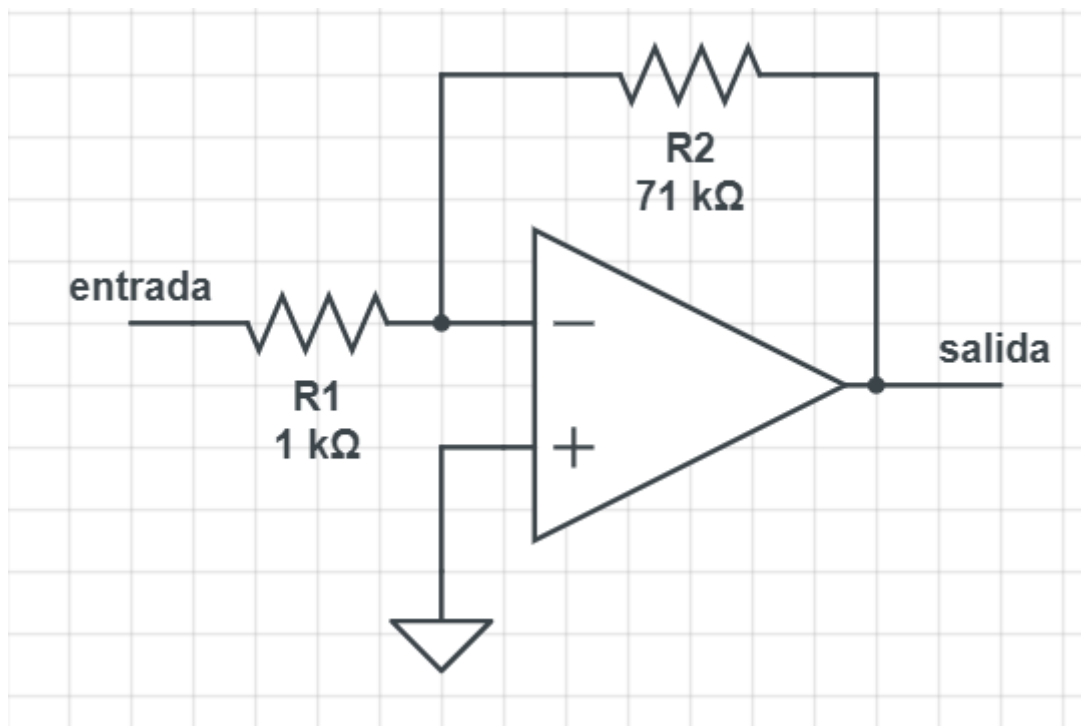


Figura 2.20. Configuración inversora para la amplificación de la señal

Se ha medido experimentalmente que la señal de salida V_s del puente linealizado con el amplificador operacional es de, aproximadamente, -0.05 V cuando se alimenta a 9 V. Se

ha elegido ese voltaje porque es el que ofrecen las pilas alcalinas que se utilizan en el laboratorio.

Esta entrada de -0.05 V cuando no se ejerce presión sobre la galga extensiométrica debe ser amplificada mediante el circuito representado en la Figura 2.20. El microcontrolador Arduino es capaz de soportar tensiones positivas de hasta 5 V , por lo que la señal deberá amplificarse todo lo posible hasta llegar a ese límite. Debe tenerse en cuenta el valor máximo, que se corresponde con el obtenido cuando no se presiona el sensor.

$$G = \frac{5\text{ V}}{-0.05\text{ V}} \approx -100$$

Es necesaria una ganancia máxima de -100 . Nótese que se ha optado por una configuración de amplificación inversora para cambiar de signo el voltaje final, pues Arduino no acepta señales negativas. Resolviendo el circuito anterior, se tiene que la expresión en función de las resistencias es la siguiente:

$$v_{salida} = \left(-\frac{R_2}{R_1}\right) v_{entrada}$$

$$G = -\frac{R_2}{R_1}$$

En definitiva, el cociente entre R_2 y R_1 debe ser de, como máximo, 100 . Se debe buscar una combinación que se acerque lo máximo a ese número, pero sin sobrepasarlo. Teniendo en cuenta que las galgas extensiométricas son recicladas, es de esperar que los niveles de salida de cada uno de los sensores sean levemente distintos. Si se ajusta demasiado la ganancia al límite calculado, es muy probable que en algún momento se sobrepasen los 5 V , impidiendo un posterior procesado de la señal.

Tras mediciones experimentales en el laboratorio, se ha optado por utilizar $R_1 = 1\text{ k}\Omega$ y $R_2 = 71\text{ k}\Omega$, como resultado de la suma de dos resistencias, una de 33 y otra de $58\text{ k}\Omega$. De este modo, se obtiene una ganancia de -71 .

Realizando mediciones con el circuito implementado, una vez acondicionados los sensores, se tienen distintos voltajes dependiendo de la galga que se utilice. Para la zapatilla izquierda, por ejemplo, se tienen unos 2.78 V sin ejercer fuerza sobre el sensor

delantero (2.56 V cuando se presiona), y 3.33 V en el trasero (2.95 V al presionar). Para la zapatilla derecha se obtienen valores próximos a los del sensor trasero derecho. Es preferible perder resolución en el Arduino a que este voltaje sobrepase los 5 V de límite. Esta diferencia cuando se presiona y cuando no es la que se recogerá y se procesará para activar el actuador correspondiente.

Implementación y componentes

En la Figura 2.21. se presenta el circuito completo de acondicionamiento de cada una de las galgas extensiométricas. Para el sistema completo serán necesarios cuatro circuitos como los expuestos, cada uno con un sensor diferente conectado en R_g . Así pues, por circuito serán necesarias 3 resistencias R_0 de $2\text{ k}\Omega$, una galga extensiométrica R_g con una impedancia de $2\text{ k}\Omega$ en reposo, una resistencia R_1 de $1\text{ k}\Omega$, una resistencia R_2 de $71\text{ k}\Omega$, dos pilas de 9 V para alimentar el circuito, y dos amplificadores operacionales integrados en el TL082CN.

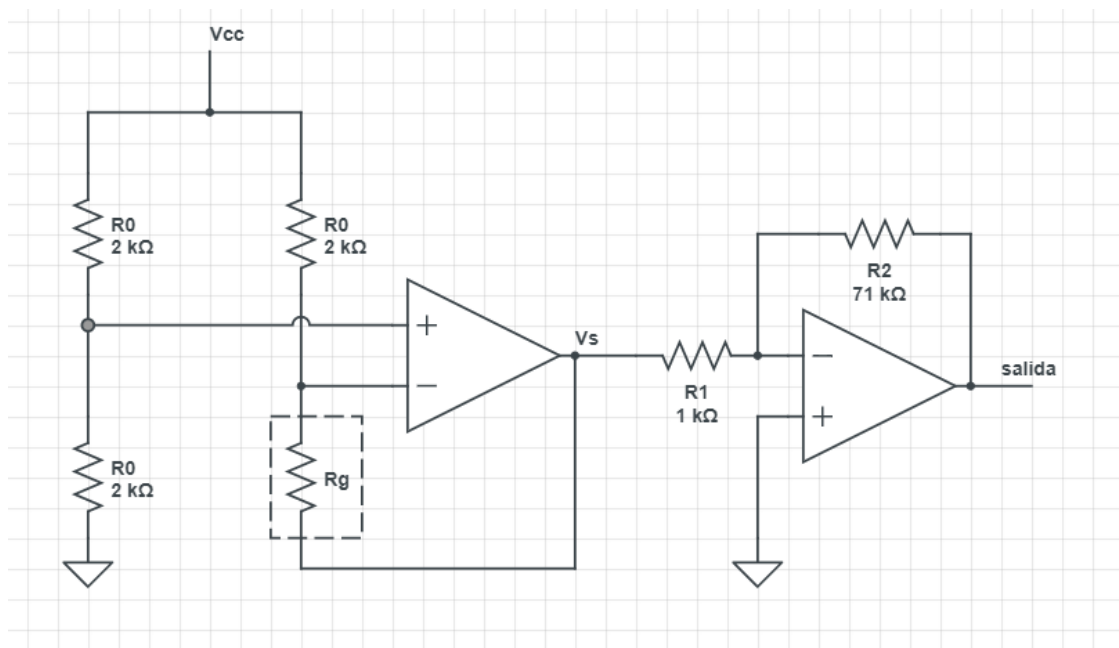


Figura 2.21. Circuito de acondicionamiento completo para cada galga extensiométrica

Ante la incapacidad de utilizar esos valores de resistencias en un solo componente, se ha optado por dividir esas resistencias en grupos de dos resistores en serie. Para las de $2\text{ k}\Omega$ se han utilizado dos resistencias de $1\text{ k}\Omega$ (marrón-negra-roja) y las de $71\text{ k}\Omega$ se han

obtenido como suma de resistencias de 33 k Ω (naranja-naranja-naranja) con las de 58 k Ω (verde-gris-negra).

El integrado a utilizar en este circuito es el TL084CN. Este componente es especialmente interesante para nuestro trabajo, puesto que se necesitan 2 amplificadores operacionales por cada sensor; esto es, 8 en total. Con el TL084CN se pueden aprovechar todos los amplificadores operacionales, recurriendo a dos integrados de estos.

Recuérdese que la alimentación de estos integrados, V_{cc}^+ y V_{cc}^- son +9 y -9 V, respectivamente.

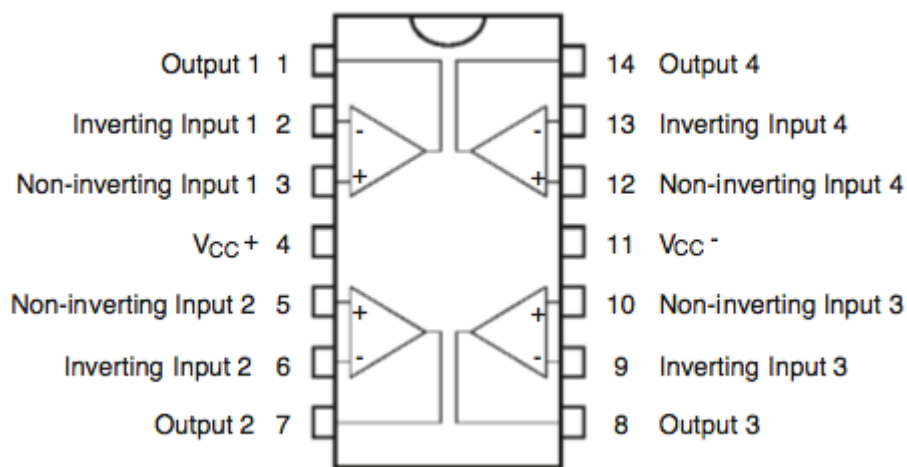


Figura 2.22. Esquema de conexiones de un integrado TL084CN

Construcción del circuito



Figura 2.23. Zócalo de 14 pines para el integrado TL084CN

Una vez conocidas las especificaciones del circuito, se debe proceder a soldarlo a una placa de circuito impreso, o PCB. Se utiliza una placa con conexiones de cobre cuyas pistas están conectadas de forma horizontal. A esta se le sueldan los componentes oportunos con estaño. Estos componentes son los ya mencionados anteriormente: resistencias y el integrado TL084CN. También deben añadirse ciertas conexiones con cables auxiliares y entradas y salidas de la placa, las cuales unirán este circuito de acondicionamiento con el Arduino o con las galgas extensiométricas de las zapatillas.

El integrado no debe ser soldado directamente, pues su sustitución en caso de fallo será más complicada, sino que deberá ir adherido a un zócalo de 14 pines (también denominado DIP14), representado en la Figura 2.23.

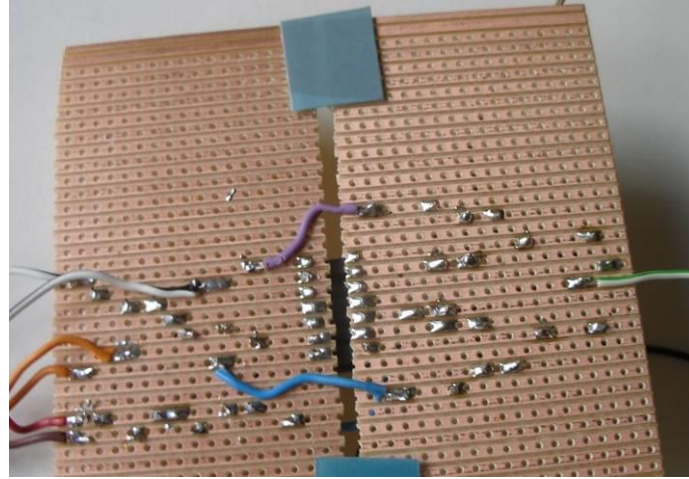


Figura 2.24. Vista inferior del circuito de acondicionamiento de dos sensores

En la Figura 2.24, se muestra una fotografía de la parte inferior de la placa sobre la cual se han soldado los componentes. Se presenta antes el lado inferior porque es importante que se aprecie la conexión de las pistas a lo largo de toda la placa en sentido horizontal. Es decir, por ejemplo, el cable naranja inferior de la fotografía está conectado con el cable de color azul.

En este caso, los cables naranjas y los de color blanco-negro de la izquierda son las entradas de los dos sensores que acondiciona el circuito. También a la izquierda, el cable rojo aporta la alimentación positiva y el marrón la tierra. A la derecha, el cable verde-blanco aporta la alimentación negativa. Las dos salidas de la placa son visibles en desde la parte superior, las cuales se recogen en los cables de Arduino.

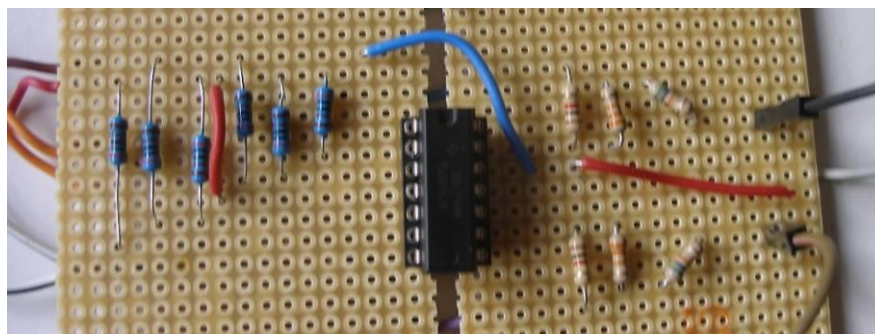


Figura 2.25. Vista superior del circuito de acondicionamiento de dos sensores

La parte superior de la placa contiene los componentes del circuito. Las resistencias azules de la izquierda son las de $2\text{ k}\Omega$, mientras que las de la derecha son las que componen la etapa de amplificación inversora. Recuérdese que el integrado utilizado es el TL084CN, el cual contiene 4 amplificadores inversores. Por tanto, cada placa de las fotografiadas es capaz de acondicionar señales de 2 sensores en total.

Placa anexa en configuración sumador no inversora

En el apartado de Pruebas y ensayos se explican más detalladamente los motivos por los que son necesarios añadir esta nueva placa al final del sensor trasero derecho. Más concretamente, el sumador se sitúa justo a la salida del amplificador inversor. Su propósito es añadir un nivel de continua estable, de modo que el voltaje esté comprendido entre 0 y 5 V para que Arduino pueda procesarlo.

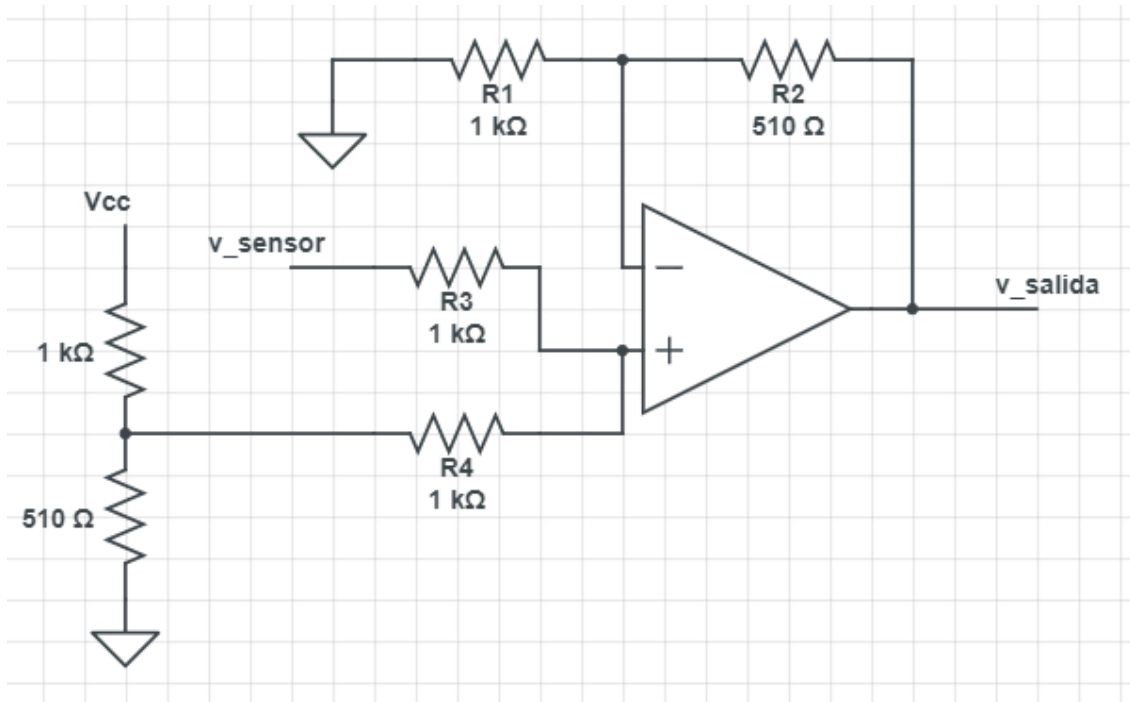


Figura 2.26. Esquema del circuito sumador en configuración no inversora

En la Figura 2.26 se muestra el esquema utilizado para construir una configuración no inversora. El principio básico del circuito se centra en que en la salida se obtiene la suma de las señales introducidas en la entrada positiva del operacional. Nótese que una de las

entradas de V_+ es un divisor de voltaje, por lo que hacia la rama de R_4 pasará aproximadamente una tercera parte del valor total de V_{cc} , que son 9 V.

La señal resultante se puede calcular mediante los pasos que vienen a continuación.

La tensión de salida, la cual se introducirá como entrada analógica en el Arduino, será:

$$v_{salida} = \left(1 + \frac{R_2}{R_1}\right) * V_+$$

Siendo V_+ , aplicando Millman:

$$V_+ = \frac{\frac{V_{sensor}}{R_3} + \frac{\frac{1}{3}V_{cc}}{R_4}}{\frac{1}{R_3} + \frac{1}{R_4}}$$

Como R_3 y R_4 son iguales ($R_3 = R_4 = 1k\Omega$), entonces:

$$V_+ = \frac{1}{2} \left(V_{sensor} + \frac{1}{3} V_{cc} \right)$$

Este es el motivo por el que R_1 es el doble que R_2 . Si no queremos que haya ganancia, y si $(1+R_1/R_2) = 2$, el factor de multiplicación se anula y en la señal final solamente queda el sumatorio de las señales de entrada. Es decir, la señal resultante es el valor inicial del sensor más un offset que depende de V_{cc} , que en este caso es 3.10 V.

De este modo se consigue solventar el problema que presentaba el funcionamiento del sensor trasero derecho. En la Figura 2.27 se muestra una fotografía del circuito en cuestión. Por simplicidad, se ha optado por un integrado TL081CN, el cual cuenta con un solo amplificador operacional. No debe olvidarse que se alimenta con ± 9 V como el resto de integrados.

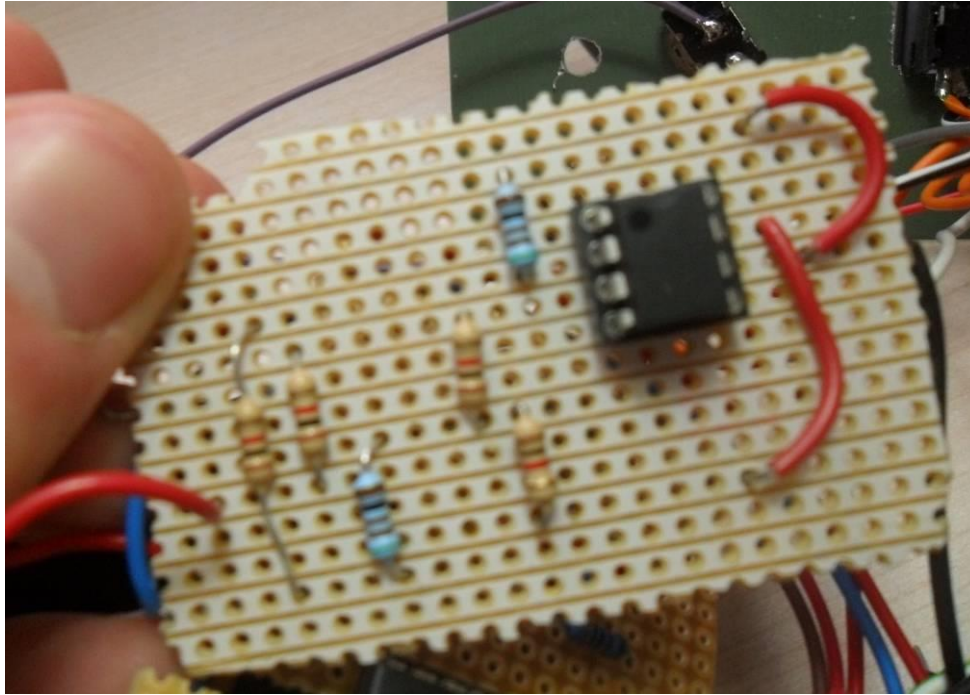


Figura 2.27. Fotografía de la placa en configuración no inversora

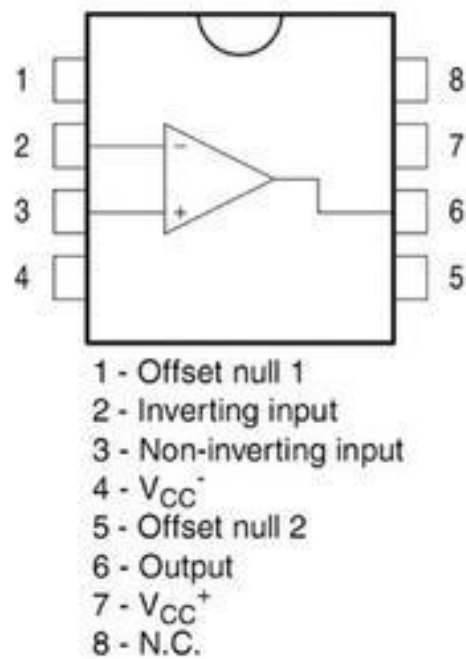


Figura 2.28. Esquema de conexiones del integrado TL081CN

Medición de los sensores

Las mediciones reales de las galgas extensiométricas, una vez han sido acondicionadas, se leerán en el microcontrolador Arduino para la posterior decisión del sistema. Los valores medios obtenidos para cada sensor son los siguientes:

| Sensor | Límite (V) |
|---------------------|------------|
| Derecho Delantero | 0.60 |
| Derecho Trasero | 1.25 |
| Izquierdo Delantero | 1.31 |
| Izquierdo Trasero | 0.36 |

Tabla 2.1. Valores medios de los voltajes de cada sensor, medidos en voltios

Estos valores servirán como límite para decidir si un sensor está activo o no, pues si se presiona lo suficiente, el voltaje del sensor determinado bajará unos 0.2 V de su valor inicial en reposo.

2.1.5. Sensores en la deambulaci3n

En la introducci3n ya se expusieron con cierto detalle las etapas de la deambulaci3n, vez presentada la forma b3sica de sensorizaci3n del proceso de levantarse con galgas extensiom3tricas, resulta de inter3s desarrollar un planteamiento te3rico similar para la deambulaci3n.

El control mediante sensores toma numerosas ideas de la forma de medici3n para levantarse. Ya se discuti3 la posibilidad de incluir ciertos tipos de sensores a lo largo de la pierna, llegando a la conclusi3n de que los sensores de presi3n son los m3s adecuados. En este planteamiento, los sensores piezoel3ctricos vuelven a ser una posibilidad a tener en cuenta, pues el sujeto realiza movimientos m3s marcados y a la vez repetitivos en cada momento de la marcha.

Sin embargo, el m3todo m3s fiable para la medici3n del movimiento es, de nuevo, la sensorizaci3n de los cambios de presi3n sobre cada punto del pie. No es de extrañar que la mayor3a de exoesqueletos comerciales partan de este supuesto, ya sea como sistema de medici3n principal o bien como sistema de refuerzo y control de equilibrio.

Ciertamente, los cambios de presi3n sobre los pies constituyen un modelo fiable de implementaci3n. No obstante, es necesario tener en cuenta que deber3n funcionar

estando en continuo movimiento, y en ciertos instantes hasta se encontrarán despegados del suelo. En este caso, realizar un sistema como este termina presentando grandes problemas ergonómicos, y a menudo el diseño del exoesqueleto se vuelve aparatoso e incómodo, como ya se indicó con el REX, de Rex Bionics.

Por motivos de fiabilidad, resulta conveniente acompañar los sensores de los pies con otros en los tobillos y en las caderas. En este caso, los piezoeléctricos no serían prácticos, pues el movimiento de una persona al caminar no es perfectamente estable, y mucho menos para alguien con dificultades. El margen para decidir si estos sensores están activos o no sería demasiado estrecho como para poder confiar en su valor y conocer en qué momento de la marcha se encuentra. Una solución sencilla y muy aplicada es la instalación de un control externo para que, usualmente con las manos, el usuario dirija en tiempo real el dispositivo.

Otra idea esencial que debe tenerse en cuenta es la intención del usuario a la hora de caminar, sentarse o mantenerse de pie. Exoesqueletos como el Indego implementan un marco global en el que la persona decide en qué modo se encuentra. Ya quiera caminar, sentarse o quedarse de pie inmóvil debería poder comunicárselo al sistema, pues los mismos sensores (y, con su respuesta, los actuadores) interpretarán la información de distinta manera según la configuración del modo.

En definitiva, la sensorización para la deambulaci3n puede tomar casi todas las ideas utilizadas en el control de la bipedestaci3n, pero deben tenerse en cuenta nuevos factores que dificultan la implementaci3n. Asimismo, aparecen nuevos requisitos a nivel de software que exigen la diferenciaci3n de si el usuario desea caminar, quedarse de pie sin moverse o levantarse y sentarse.

2.2. Microcontrolador Arduino

Las señales de entrada tomadas de cada sensor son recibidas posteriormente a una placa Arduino. Esta placa se compone de un microcontrolador y un entorno de desarrollo (IDE), para la programación de software. Este microcontrolador se encarga de hacer cumplir las órdenes y transmitir las por las salidas habilitadas, en este caso hacia los actuadores.

Para este trabajo se utilizará una placa Arduino Mega 2560, representado en la Figura 2.29.

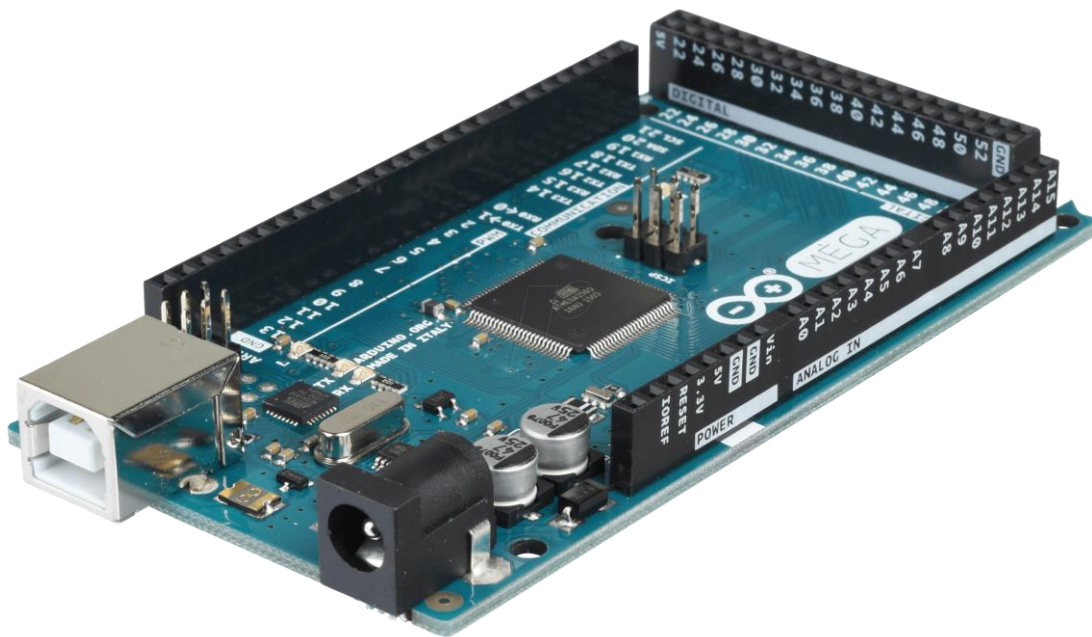


Figura 2.29. Placa integrada Arduino Mega 2560

Un esquema introductorio viene dado en la Figura 2.30., desde una vista de la planta del microcontrolador. Las dos entradas de la parte izquierda se utilizan para conectar la placa vía USB al ordenador para su programación y para alimentar la placa con baterías externas. La alimentación externa se hace imprescindible cuando se pretende mover objetos con servomotores, como es el caso.

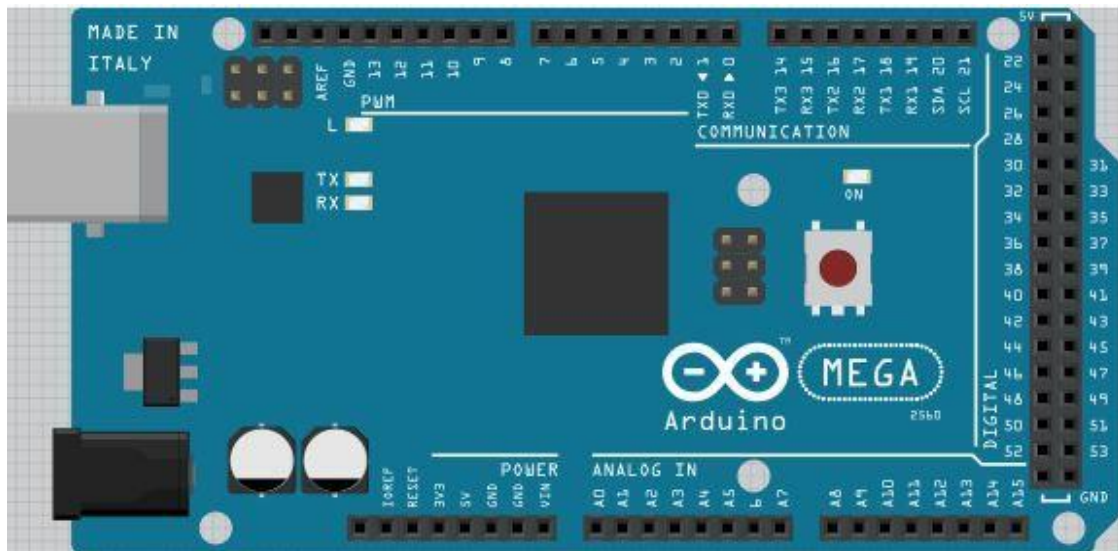


Figura 2.30. Esquema de la placa integrada Arduino Mega 2560

En la parte superior se tienen los pines PWM (2 al 13), a los cuales se conectarán los pines de control de los servomotores y se les dará la señal cuando deban realizar algún movimiento.

La placa Arduino Mega cuenta con multitud de entradas y salidas, pero para controlar el sistema solo se necesitarán unas pocas de ellas. La parte de *Communication*, así como los pines digitales de la derecha (22 al 53) no serán necesarios, más que para conectar alguna tierra GND. Los pines Analog In (A0 al A7) se reservarán para recibir la señal ya acondicionada de los sensores para su posterior tratamiento. Finalmente, los pines Power sirven para alimentar los componentes. El pin etiquetado como 5 V, por ejemplo, deberá estar conectado a todos los servomotores.

Se adjunta en la Figura 2.31. un esquema simple con las conexiones que deberá tener el microcontrolador Arduino, con los grupos de cables agrupados en colores: en rojo, la alimentación; en verde, las entradas; en azul, las salidas; y en negro, el circuito para el diodo LED.

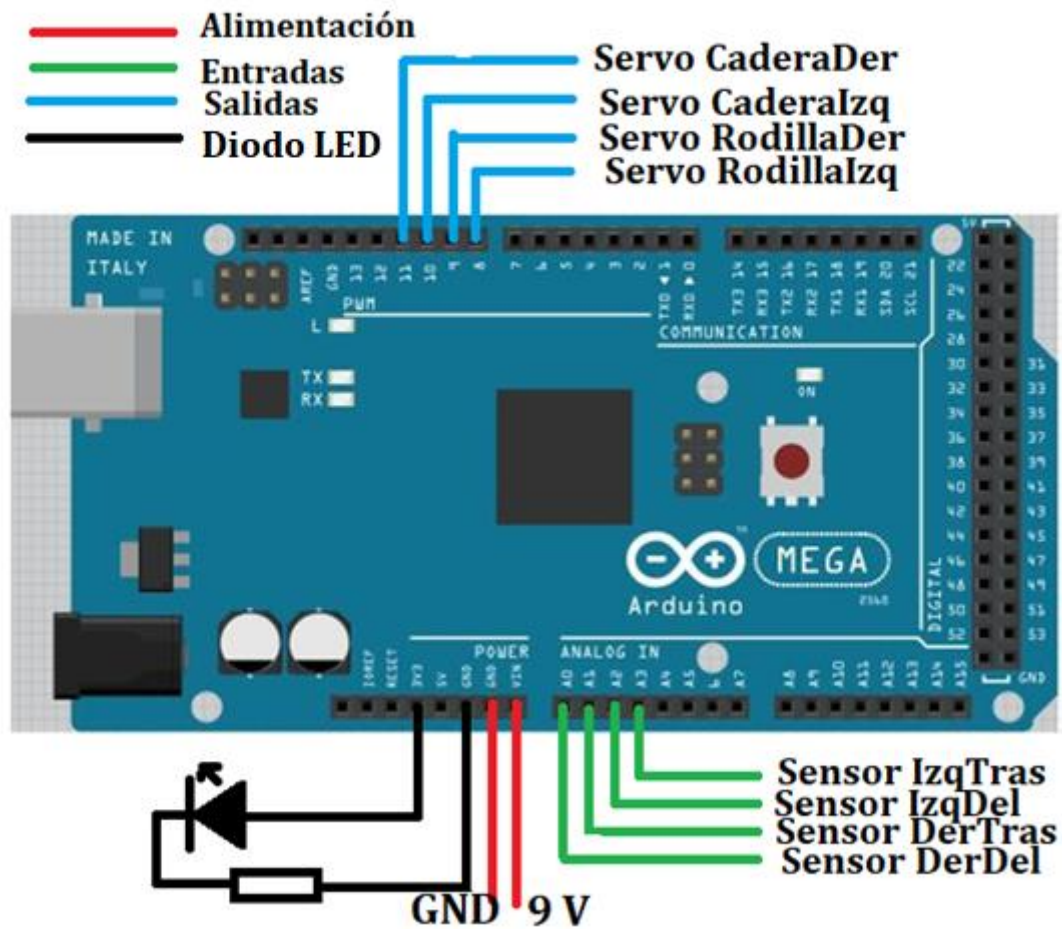


Figura 2.31. Esquema de conexiones del microcontrolador Arduino alojado en la caja de circuitos

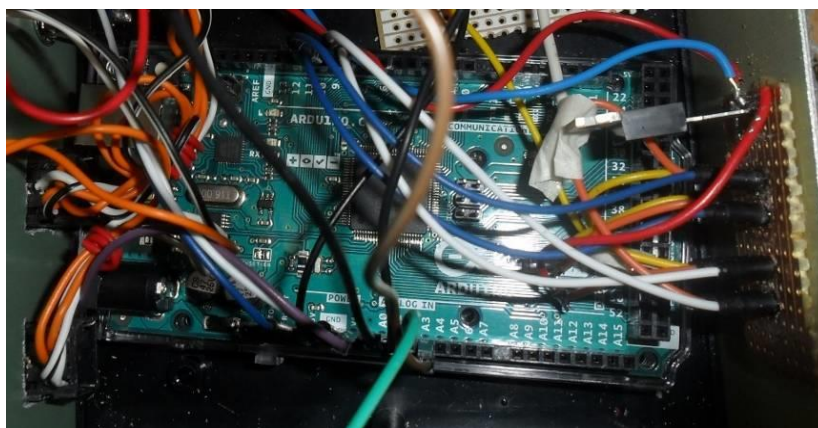


Figura 2.32. Fotografía de las conexiones de Arduino dentro de la caja. En la parte inferior izquierda, la alimentación seguido de los sensores, y a la parte superior, los servomotores

Dado que en la Figura 2.32. las conexiones de Arduino se entremezclan con el resto de cableado, se adjunta a continuación un esquema aclaratorio con los colores de cada cable en la realidad.

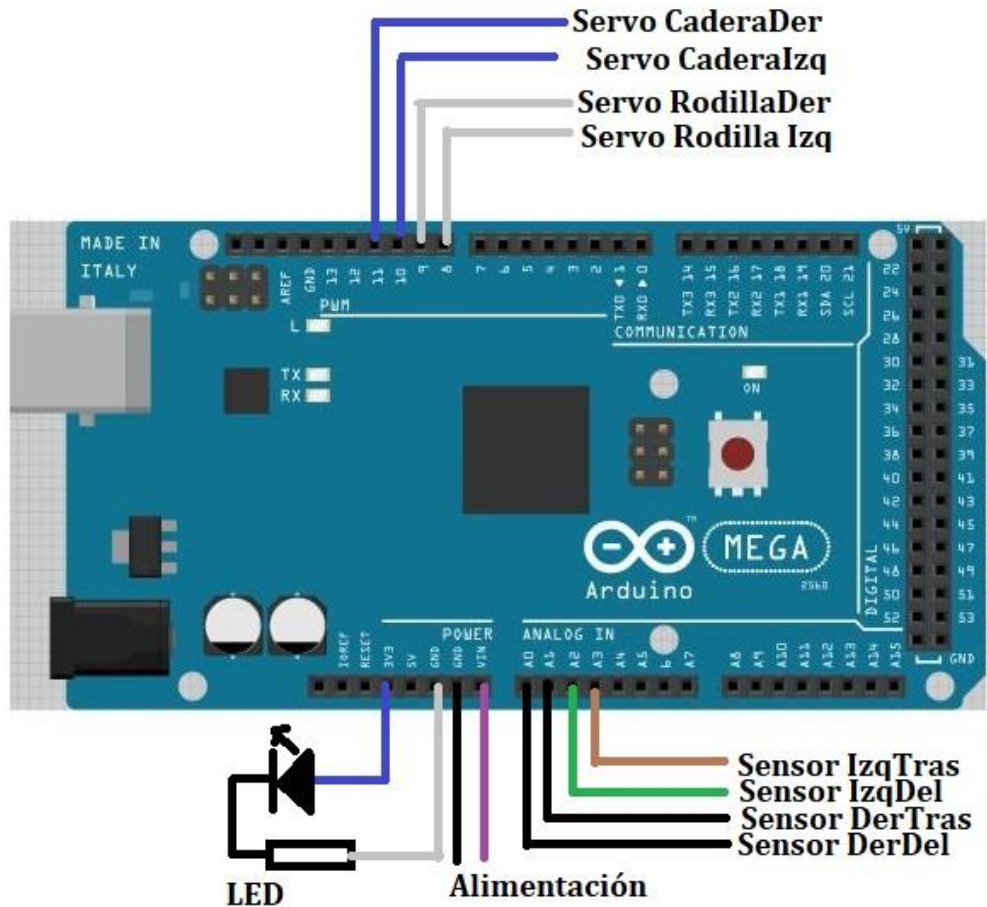


Figura 2.33. Esquema de conexiones del microcontrolador Arduino alojado en la caja de circuitos, con los colores reales de cada cable

2.3. Actuadores

2.3.1. Robot bípedo

Una vez extraída una señal de intención fiable de los sensores y el microcontrolador la ha procesado mediante el correspondiente código, es necesario que algún subsistema responda a ese algoritmo. En un sistema real, estos actuadores deberían poner en marcha el exoesqueleto de una manera precisa en función de la propia intención del usuario. En este proyecto, sin embargo, incluir un exoesqueleto completo aumentaría el coste en gran

medida. Por tanto, se opta por utilizar un robot accionado por servomotores que simularía el funcionamiento de ese exoesqueleto real, pero sin valorar aspectos económicos y ergonómicos, entre otros.

Se presentan a continuación una colección de fotografías descriptivas del robot bípedo.

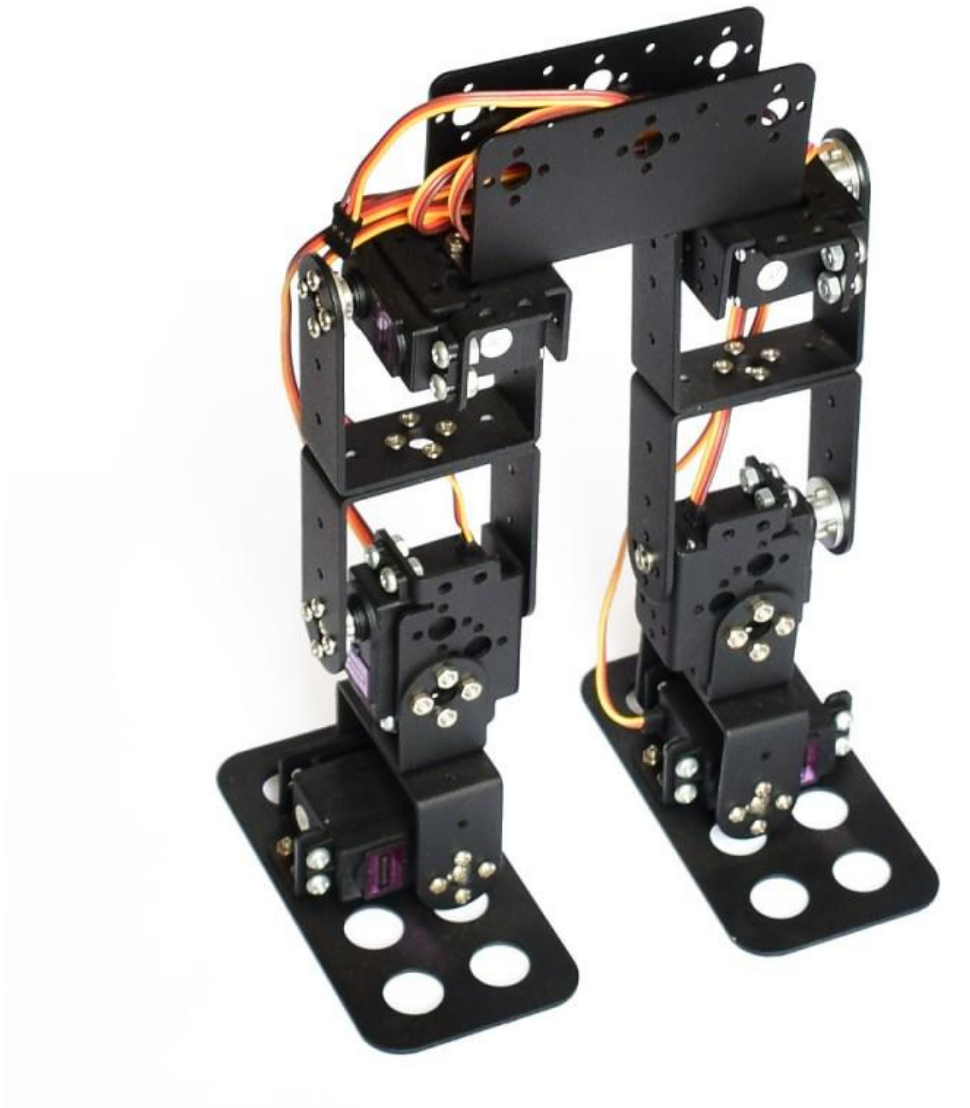


Figura 2.34. Fotografía del robot bípedo a utilizar

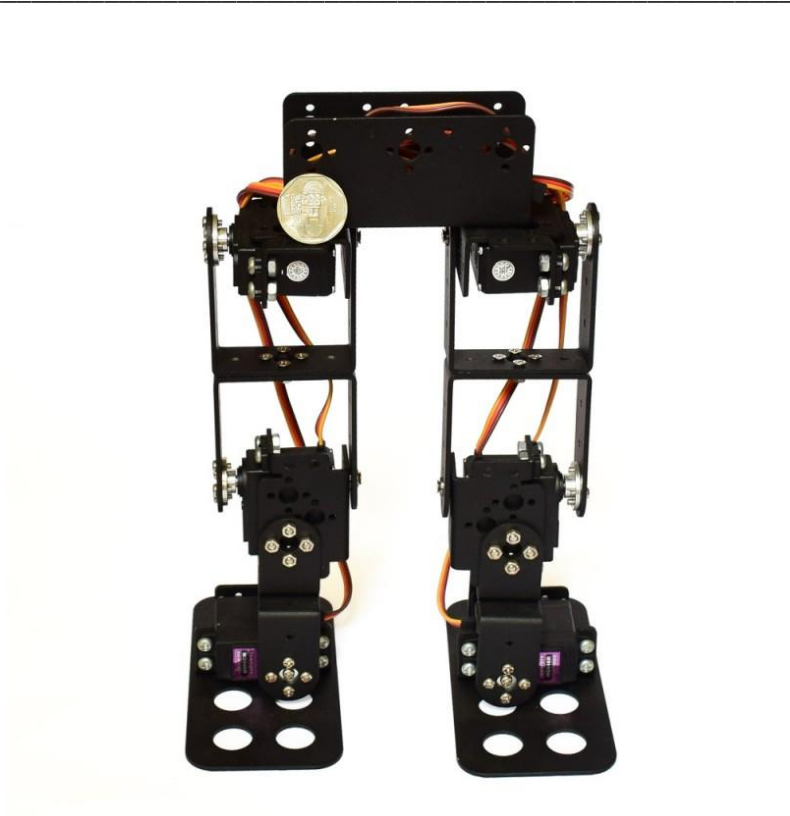


Figura 2.35. Fotografía del alzado del robot bípedo a utilizar

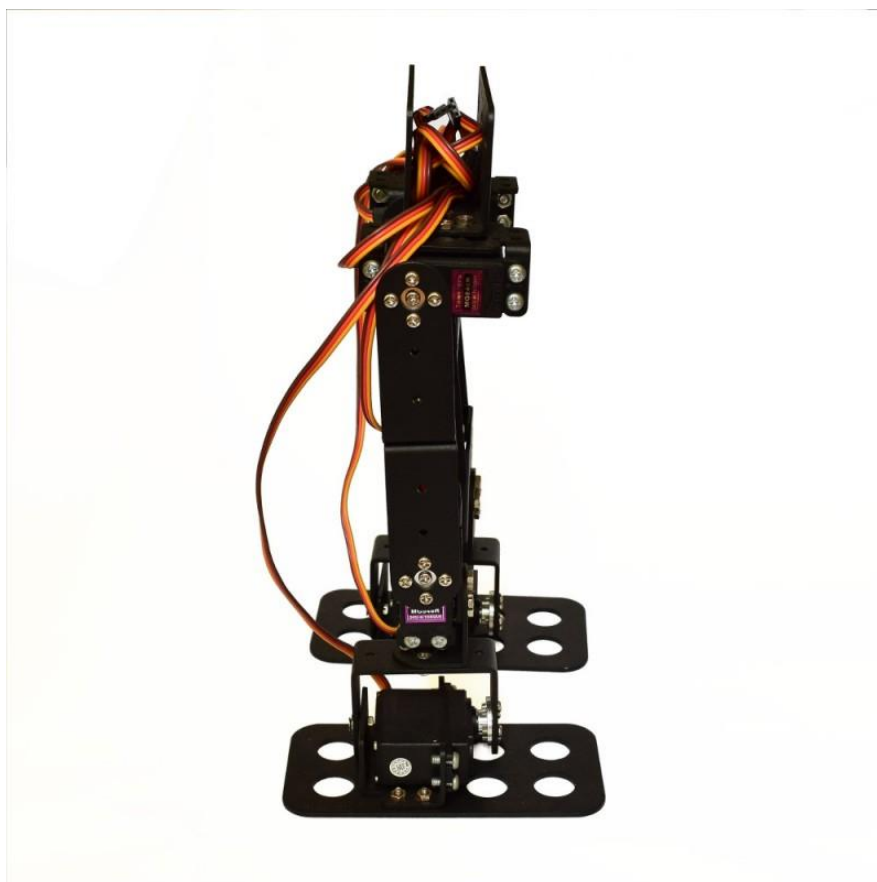


Figura 2.36. Fotografía del perfil del robot bípedo a utilizar



Figura 2.37. Piezas individuales del robot bípedo a utilizar

El robot es capaz de llegar a una altura de unos 25 cm cuando está completamente erguido. Sus piezas son de aluminio y sus uniones se componen únicamente de tornillos y tuercas, por lo que no necesita de pegamentos ni adhesivos.

Atendiendo a la estructura del robot, se puede dividir su estructura en diferentes partes en función de donde están colocados los motores. Cada motor permite una articulación, y consigo un nuevo grado de libertad. Una sola pierna dispone de un servomotor en la cadera, otro en la rodilla y otro en el tobillo. Por tanto, el robot entero tiene 6 grados de libertad en total.

El objeto de este estudio se centra en el proceso de bipedestación. Por consiguiente, los pies van a estar siempre quietos y apoyados en el suelo. Esto permite simplificar el modelo y suprimir los motores que hacen girar los pies de izquierda a derecha, manteniendo solamente los de las rodillas y los de las caderas. Así, aunque el robot permita 6 grados de libertad, solamente se aprovecharán 4 de ellos.

2.3.2. Servomotores

Para iniciar el movimiento del robot es necesario un actuador. Para esto se cuenta con cuatro servomotores instalados en las rodillas y en las caderas, tal y como se muestra en la Figura 2.35.

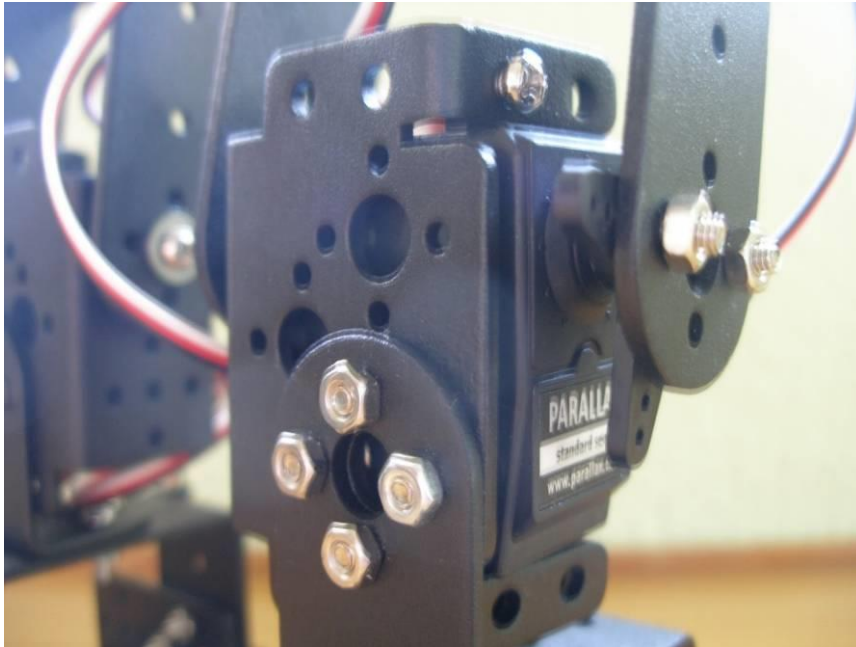


Figura 2.38. Instalación del servomotor para la rodilla izquierda del robot

Los motores que se han elegido son del tipo Parallax Standard Servo (#900-00005), que puede mantener la posición desde 0 hasta 180° y es activado por el microcontrolador Arduino.



Figura 2.39. Fotografía del tipo de servomotor instalado

Cuenta con tres terminales, cada uno representado con un color:

- Blanco: señal de control.
- Rojo: tensión de alimentación (5 V).
- Negro: tierra (0 V).

El resto de las especificaciones pueden encontrarse en la sección de Anexos.

2.4. Caja de circuitos

La construcción de una caja es una labor relativa al acabado del prototipo y que ofrece numerosas ventajas, tanto el sistema en sí como a la persona que deba utilizarlo. La implementación de una caja como elemento exterior supone agrupar todas las placas y componentes en un lugar reducido, pero al mismo tiempo la posibilidad de mantenerlos en un espacio fijo en el que no exista cabida a cortocircuitos indeseados o malos contactos.

El prototipo final debe poseer ciertas características que son deseables en un montaje electrónico. Antes que todo, debe ser fiable, por lo que no se debe temer que un cable o un componente quede desoldado de su lugar y el sistema deje de funcionar de repente. Otro factor que tener en cuenta es la estética. Si bien no se trata de un producto comercial que llevar a un grupo masivo de consumidores, el acabado final debe ser atractivo para sus usuarios. Finalmente, aunque no menos importante, es muy recomendable construir una caja para mejorar el sistema en términos de ergonomía y comodidad de uso. Así, aunque el usuario no conozca cómo está construido el sistema, siempre podrá encenderlo con un simple botón y usarlo sin ningún tipo de problema.

Para ello, las placas y demás componentes elementales del sistema deben estar dentro una caja cerrada. Esto lleva a pensar al primer problema que debe solucionarse: las señales que entran de los sensores de las zapatillas y las que salen hacia el robot. Es necesario instalar distintos conectores para cada voltaje (para alimentar las placas y los servomotores), ya sea los mencionados de control del sistema como aquellos que sirven de alimentación (las señales de control de los sensores y la de acción de los motores).

Adicionalmente, la caja debe ser taladrada y los conectores se instalarán en los paneles de aluminio laterales. De esta manera, se independizarán los elementos exteriores, tales como las zapatillas y el robot, de toda la circuitería de acondicionamiento del interior de la caja.

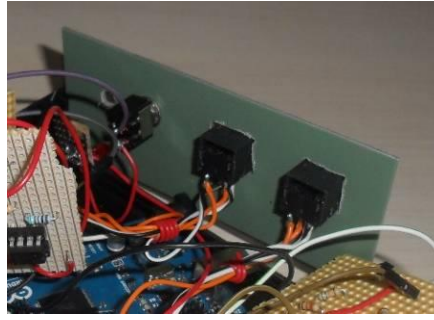


Figura 2.40. Fotografía de conexiones del lado interior de una lámina lateral de la caja (sensores de zapatillas, interruptor ON/OFF y hueco para diodo LED)

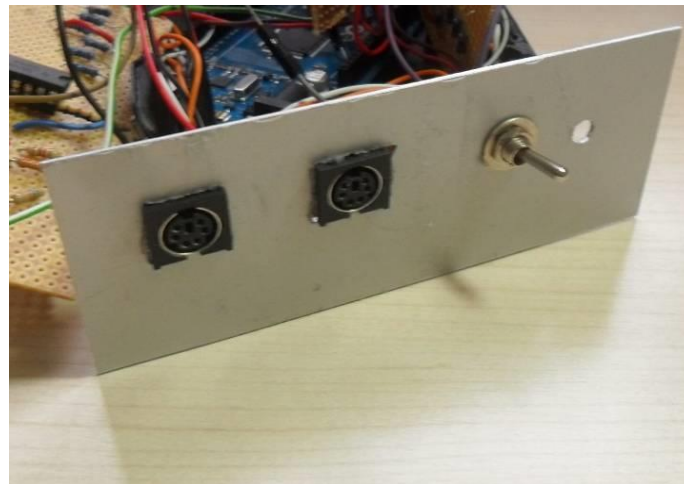


Figura 2.41. Fotografía de conexiones del lado exterior de una lámina lateral de la caja (sensores de zapatillas, interruptor ON/OFF y hueco para diodo LED)

Del otro lado deberán ir las conexiones del robot de Arduino y el cable de alimentación. El robot, en total, se compone de cuatro servomotores. Cada uno de estos motores tiene tres pines: la señal de control y dos de alimentación. La señal de control será independiente y saldrá directamente del microcontrolador, mientras que la alimentación de 0 a 5 V será compartida. Aun así, deberán instalarse los pines de todos los motores para conectarlos entre sí.

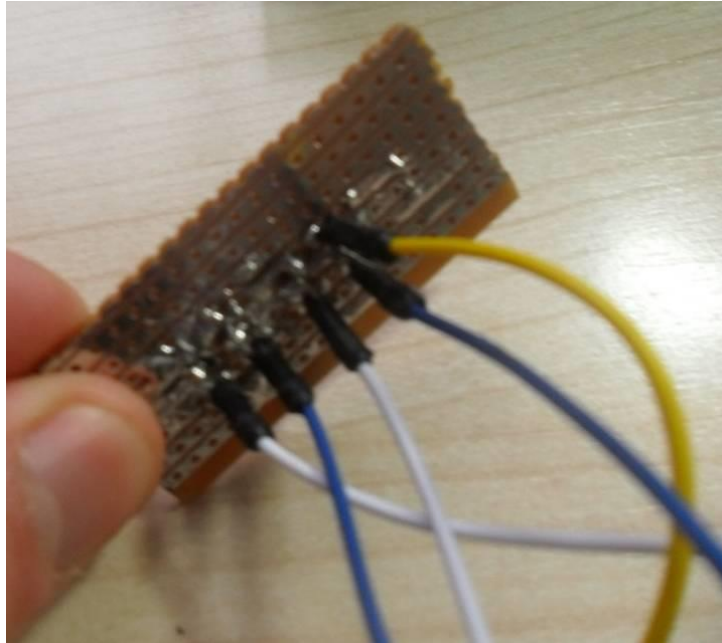


Figura 2.42. Fotografía de conexiones del lado interior de los servomotores del robot: señales de control (cables blancos y azules), tierra (amarillo) y alimentación (naranja, aún sin instalar)

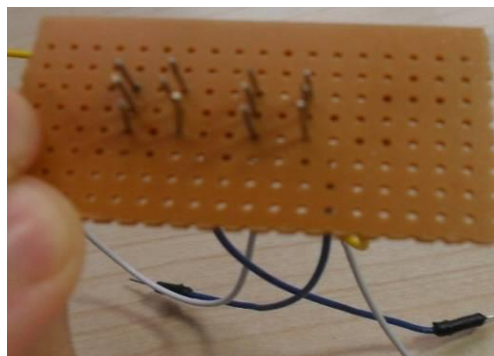


Figura 2.43. Fotografía de conexiones del lado exterior de los servomotores del robot agrupados en pines de tres

Tal como se aprecian en las fotografías anteriores, las placas quedan demasiado enredadas entre sí, y es muy posible que los cables se desuelden o hagan un mal contacto con otro componente. Por tanto, el siguiente paso será reconfigurar la distribución de los cables y anclar las placas a las paredes de la caja.

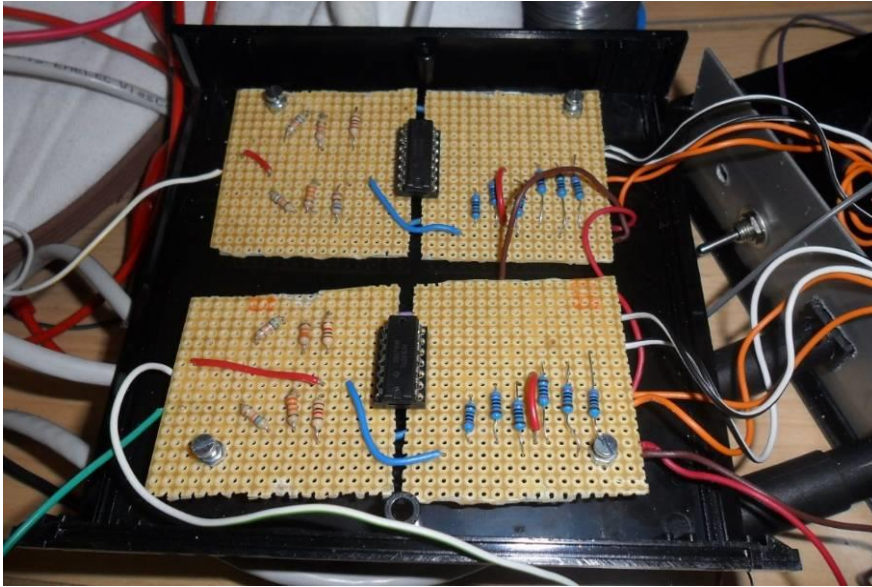


Figura 2.44. Fotografía demostrativa del atornillado de las placas de acondicionamiento de los cuatro sensores

Para recolocar los cables, es necesario realizar un par de modificaciones para asegurar el correcto funcionamiento.

La primera es agrupar las alimentaciones en una placa auxiliar, de la cual saldrán los cables hacia las placas de los sensores, el sumador y el microcontrolador Arduino. La segunda, agrupar los cables de modo que no haya grandes enredos ni tensiones dentro de la caja. Con el fin de poder abrir la caja y permitir posteriores modificaciones, los cables deberán ser más largos que la distancia que deben cubrir entre placas. Este cable adicional, o “de reserva”, se enrollará en los laterales cuando la caja se encuentre cerrada.

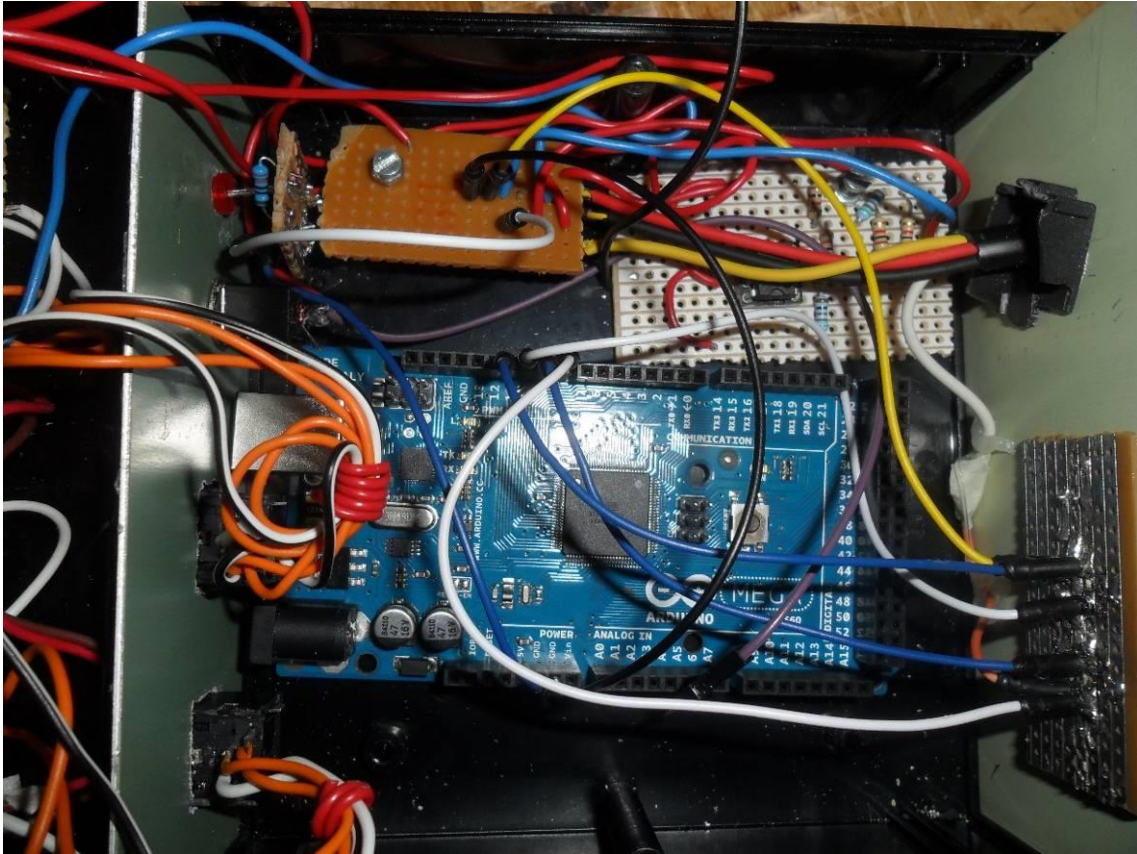


Figura 2.45. Fotografía de la parte inferior del interior de la caja, compuesto por el Arduino, el circuito sumador (placa blanca), la placa de alimentación (marrón) y el LED. En la pared derecha, los conectores del robot y los cables de alimentación. Nótese que las conexiones correspondientes a las entradas de los sensores no están conectadas todavía.

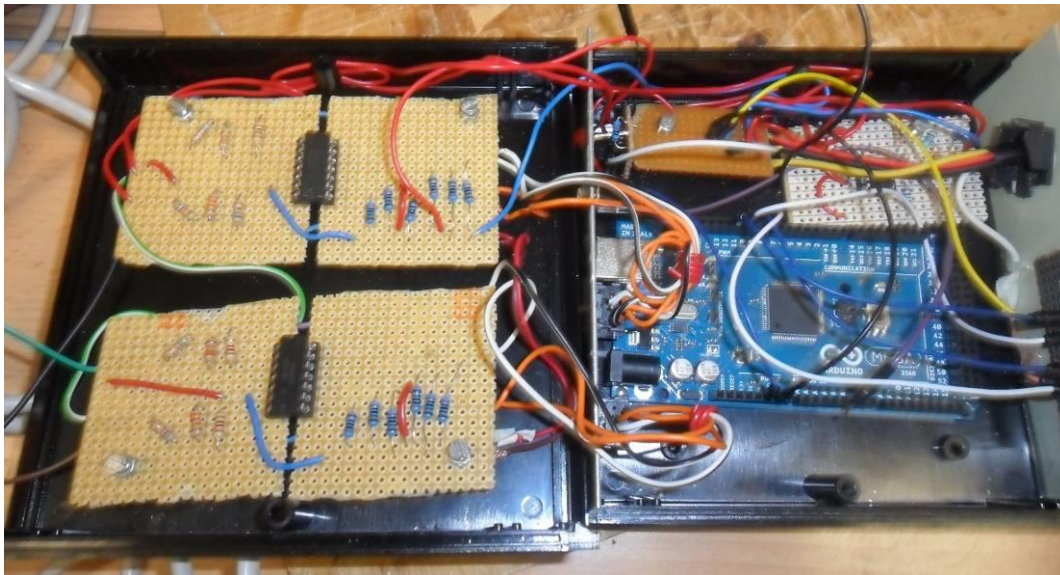


Figura 2.46. Fotografía de los elementos del interior de la caja

De forma adicional, se instalan los cables de alimentación mediante un solo conector. Este llevará cuatro cables: 9, -9, 0 y 5 V (rojo, amarillo, negro y blanco, respectivamente).

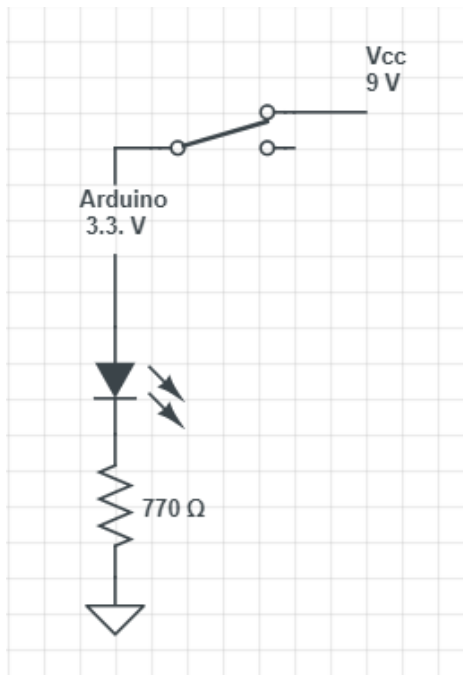


Figura 2.47. Esquema del interruptor y la bombilla del sistema

Finalmente, se añade una pequeña placa con una bombilla LED roja y una resistencia para protegerla. Esta placa es visible en las dos fotografías anteriores. Se alimenta mediante una salida de 3.3 V de Arduino, por lo que solamente se encenderá cuando el interruptor esté cerrado y proporcione los 9 V a todo el circuito. Si no, el microcontrolador Arduino estará apagado y no enviará órdenes ni corriente eléctrica.

Por último, para asegurar una potencia adecuada para los servomotores, se ha utilizado un regulador de tensión LM7805, representado en la Figura 2.48.

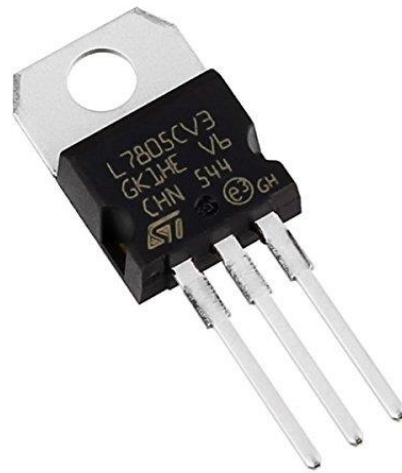


Figura 2.48. Fotografía del regulador de tensión LM7805

Dos servomotores se alimentarán directamente de la toma de 5 V de la fuente. Los otros dos, uno de la rodilla y otro de la cadera, ambos del lado izquierdo, serán alimentados por los 5 V que proporcione el regulador. Este voltaje lo obtendrá de los 9 V de la fuente. Así pues, la patilla izquierda del regulador se conectará a 9 V, la central a tierra y la derecha servirá de alimentación de los dos servos, con 5 V.

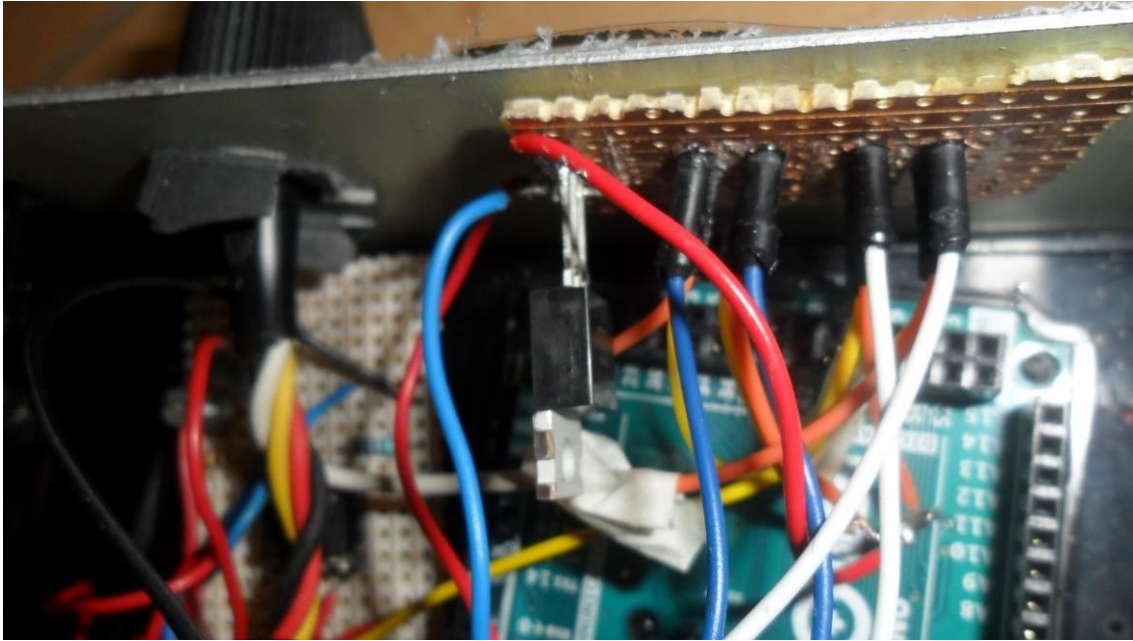


Figura 2.49. Fotografía del regulador de tensión 7805 instalado en la pared de los servos. La patilla inferior a 9 V (cable rojo), la central a tierra (cable azul) y la superior suministra 5 V a los servos (cable rojo)

2.5. Visión global del sistema

A lo largo de todo el documento se han ido presentado cada uno de los componentes que engloban el proyecto. Sin embargo, no se ha expuesto una visión en conjunto de todo el sistema y cómo están realizadas las conexiones entre ellos.

El sistema se divide en tres partes bien diferenciadas: las zapatillas con los sensores instalados, la caja con la circuitería y el robot que hace de actuador.

Zapatillas con sensores

Se ha optado por utilizar unas zapatillas cómodas adaptadas a un entorno familiar, tal como el hogar en el que vive el hipotético usuario, con una suela lo suficientemente fina como para que la fuerza que se hace sobre los sensores se traduzca bien en la señal de salida. Posteriormente, se ha comprobado experimentalmente que esta premisa no es tan importante.



Figura 2.50. Zapatillas con los sensores instalados a las suelas (parte superior)



Figura 2.51. Zapatillas con los sensores instalados a las suelas (parte inferior)

De cada una de estas zapatillas salen dos cables por sensor, esto es, cuatro cables en total. Estos cables se llevarán por separado, de modo que habrá dos cables gruesos, uno por zapatilla, que lleguen hasta la caja.

Como se tienen cuatro sensores con dos cables cada uno, a las placas se soldarán hasta ocho cables para acondicionar la señal.

Caja de circuitos

La caja de plástico sirve de contenedor y de fijador de toda la circuitería que necesita el proyecto. Como las conexiones se indicarán luego, se presenta ahora la vista exterior de la caja, la cual verá el usuario final.



Figura 2.52. Vista general de la caja de circuitos, ya conectados los cables de las zapatillas y la alimentación

En la versión final, la caja cuenta con un etiquetado simple indicando qué entradas o salidas pasan por dicho conector, así como instrucciones básicas en caso de que los conectores estén separados.



Figura 2.53. Fotografía del etiquetado exterior del lado frontal de la caja

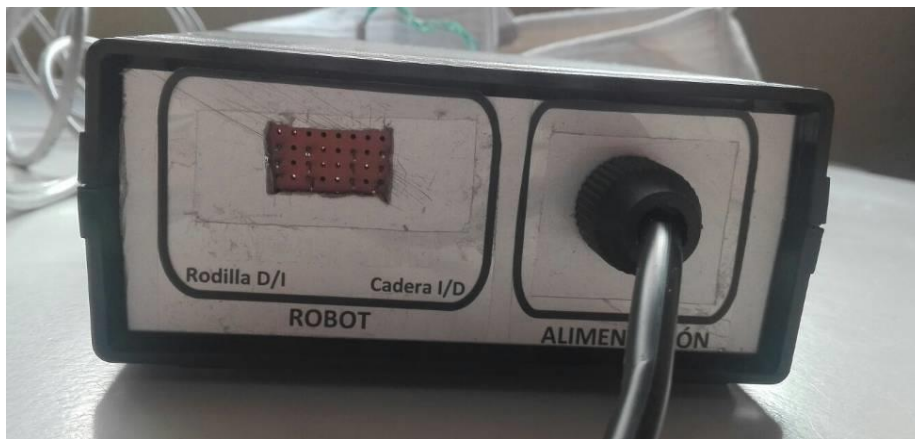


Figura 2.54. Fotografía del etiquetado exterior del lado trasero de la caja

Robot bípedo

Finalmente, como actuador que sustituye a un exoesqueleto real, se utiliza un robot bípedo negro de aluminio. Las salidas de la caja pertenecen a las de los cuatro servomotores que deben utilizarse para mover el robot: la señal de control y dos señales de alimentación. Cada servomotor tiene su propio cableado, por lo que deberán colocarse en los pines según el orden indicado en la caja, siempre con la señal de control (cable blanco) en el pin superior.

A este robot se le ha añadido un asiento hecho de madera a su medida.

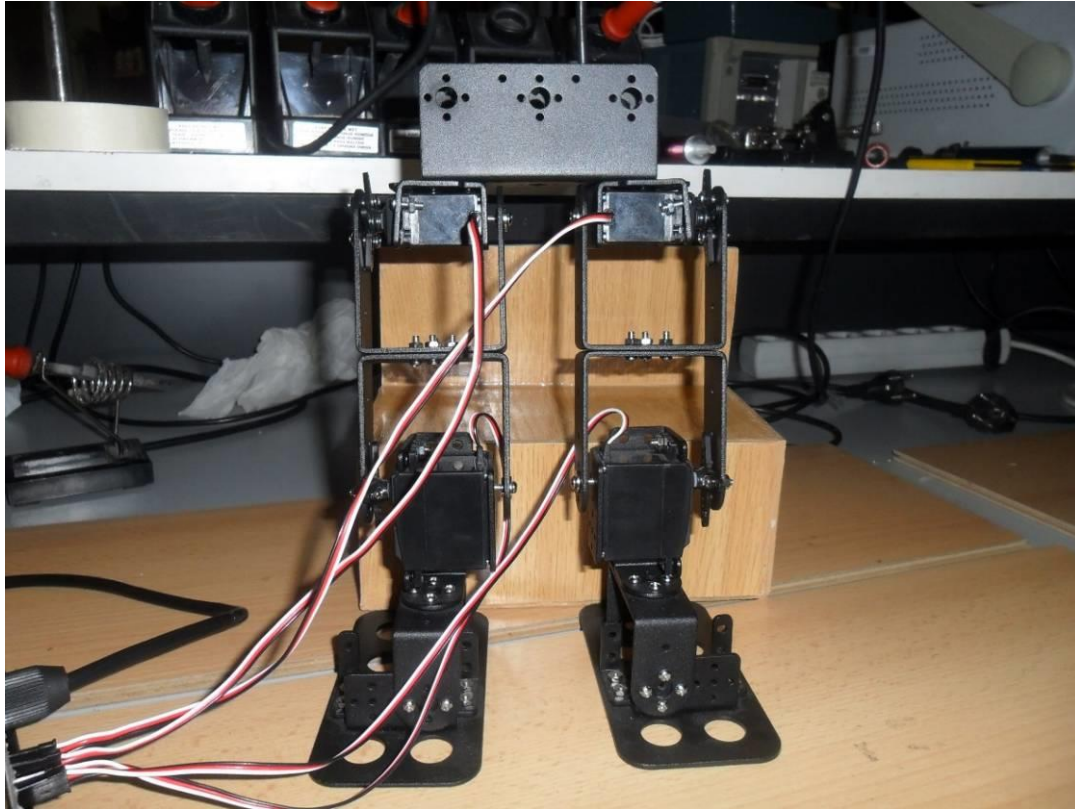


Figura 2.55. Fotografía del robot bípedo erguido con los cables conectados a la caja, y colocado delante del asiento de madera



Figura 2.56. Fotografía del asiento de madera utilizado para el robot

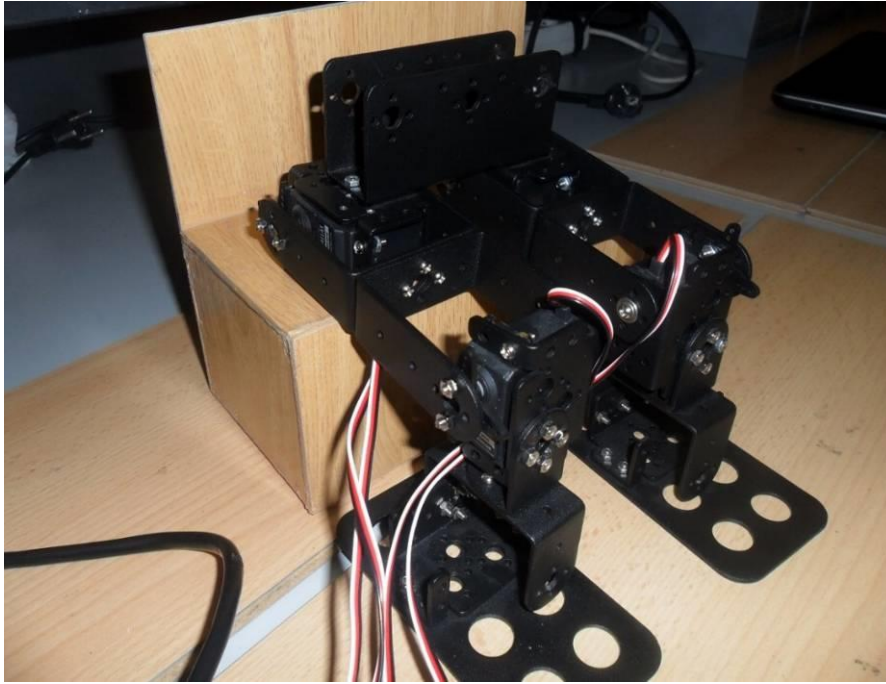


Figura 2.57. Fotografía del robot apoyado sobre el asiento de madera

Esquema global

En su conjunto, las zapatillas, la caja de circuitos y el robot componen la totalidad del prototipo. Para hacerlo funcionar, adicionalmente sería necesario conectar una fuente de alimentación externa.



Figura 2.58. Fotografía del proyecto completo: la caja de circuitos, el robot y las zapatillas

Esquema de conexiones

Finalmente, se expone un esquema completo con todas las conexiones realizadas al microcontrolador Arduino dentro de la caja. Si no se conoce la disposición de las conexiones entre placas y entre el microcontrolador, es demasiado complicado tratar de deducirlo viendo el montaje real. A continuación, se expone una explicación aclaratoria de esto de una forma esquemática.

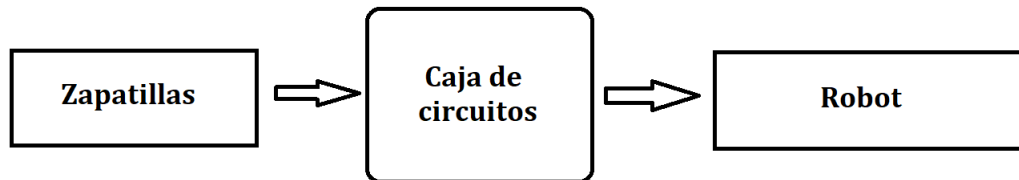


Figura 2.59. Esquema simplificado del sistema completo

En el esquema de la Figura 2.57., el elemento que carga la mayor complejidad es la caja de circuitos, por lo que el análisis se centrará en él. Para relacionarlo con los extremos, tomaremos ocho cables de entrada de las zapatillas (cuatro sensores con dos cables cada uno) y seis de salida para el robot: cuatro señales de control y dos de alimentación.

Las entradas de los sensores entran a la caja mediante dos conectores DIN de seis pines cada uno. Cada zapatilla cuenta con un conector distinto, y de estos siguen cuatro cables, dos naranjas y dos rayados blanco-negro. Los colores iguales marcan los extremos de las resistencias variables que suponen las galgas extensiométricas. Estos cables de colores son conectados a la galga, que es representada como R_g en el circuito de acondicionamiento de la Figura 2.58. Dos placas son necesarias para cuadruplicar este esquema, ancladas en la pared superior de la caja.

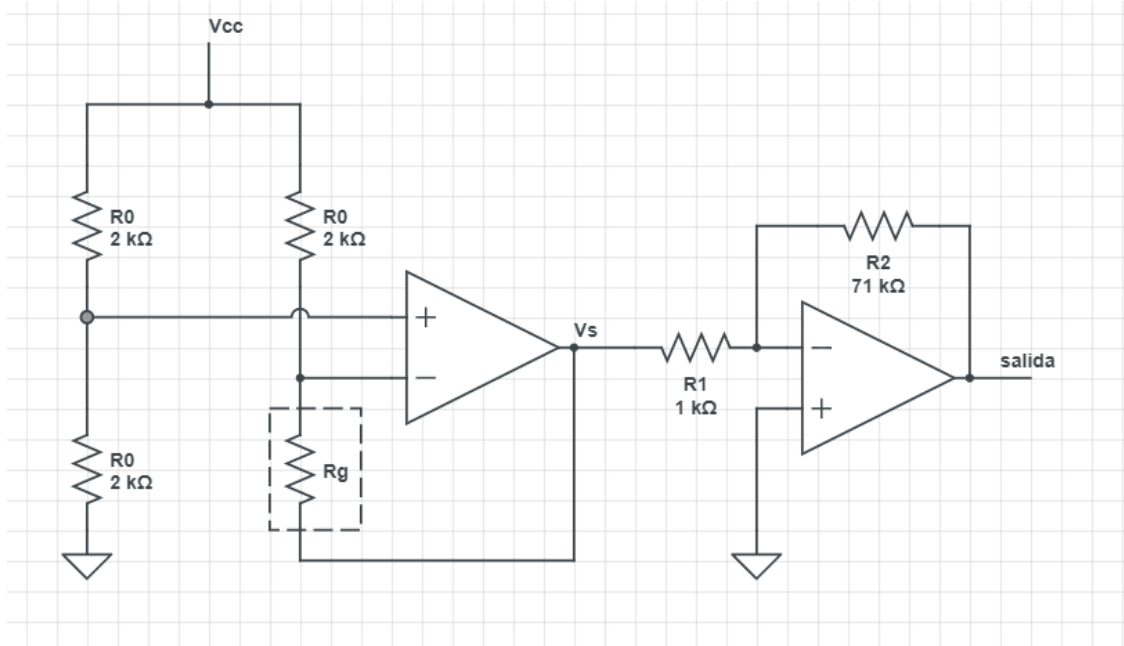


Figura 2.60. Circuito de acondicionamiento completo para cada galga extensiométrica

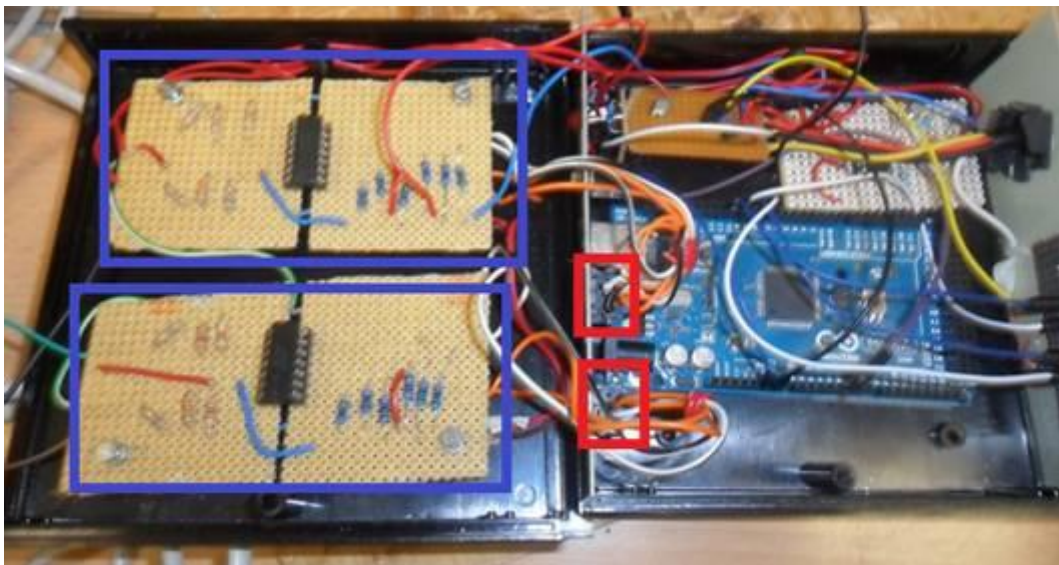


Figura 2.61. Fotografía de la caja de circuitos. En recuadros rojos, los conectores de cada zapatilla con cuatro cables cada uno (naranjas y rayados en blanco y negro). Estos cables se conectan a las placas de acondicionamiento, recuadradas en azul.

De estas placas de acondicionamiento salen los cables con el voltaje amplificado de cada sensor. Estos irán directamente al microcontrolador Arduino, salvo el sensor trasero derecho. Según se vio en el apartado de Pruebas y ensayos, este sensor necesita una placa auxiliar que sirve de sumador no inversor.

Por tanto, la placa con el recuadro azul superior en la Figura 2.59. tendrá dos cables negros de salida. Uno de ellos, el correspondiente al sensor delantero derecho, irá a Arduino, y el otro deberá ser tratado antes. De la otra placa de acondicionamiento

saldrán también otros dos cables de colores verde (delantero izquierdo) y dorado (delantero trasero).

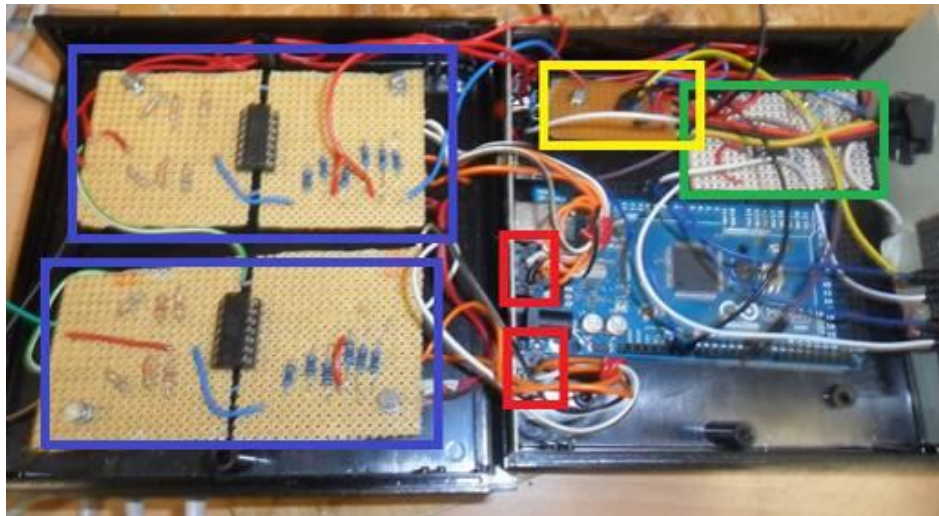


Figura 2.62. Fotografía de la caja de circuitos. En un recuadro verde se marca la placa auxiliar sumadora para el sensor trasero derecho. En un recuadro amarillo se marca placa de alimentación

El circuito sumador está marcado dentro del recuadro verde. Un cable negro une la placa de acondicionamiento (recuadro azul superior) con este circuito auxiliar. En la Figura 2.63. se recuerda el esquema del circuito implementado en la placa.

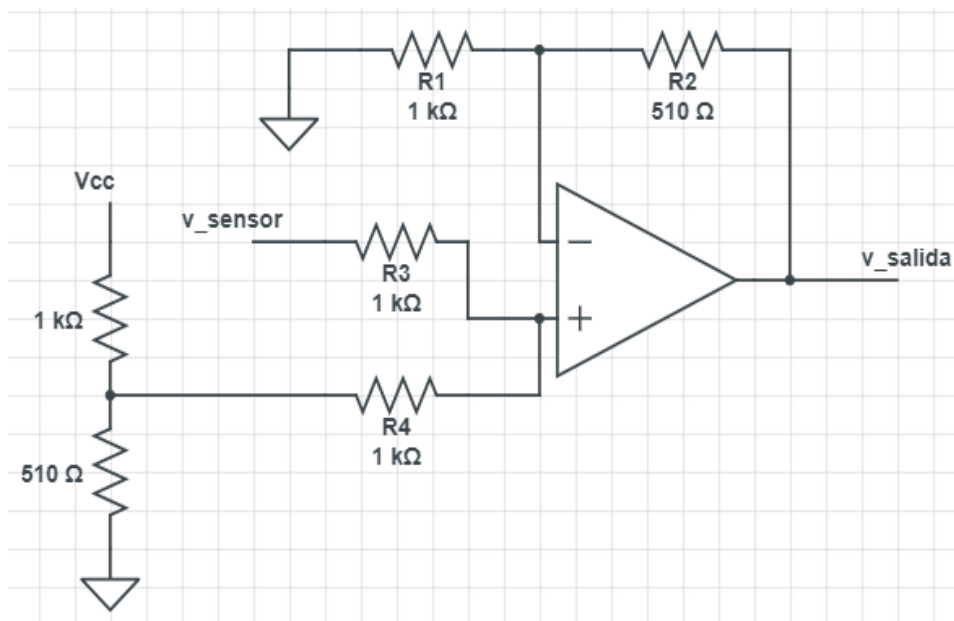


Figura 2.63. Esquema del circuito sumador en configuración no inversora

Adicionalmente, se marca en un recuadro de color amarillo la placa de alimentación. Esta placa no tiene ningún componente; simplemente sirve para unir de forma ordenada las alimentaciones de 9, 0 y -9 V. Por encima de la placa sumadora pasan los cables de alimentación. Los cables rojo, negro y amarillo se conectan a la placa de alimentación, suministrando 9, 0 y -9 V respectivamente. El cable blanco se une directamente a los pines del robot, situados en la misma pared.

De esta placa de alimentación saldrán todos los cables necesarios para suministrar corriente a las placas. Los de 9 y -9 V con cables de color rojo, y la tierra con los de color azul. Estos cables siguen las paredes de la caja manteniendo un orden en ella.

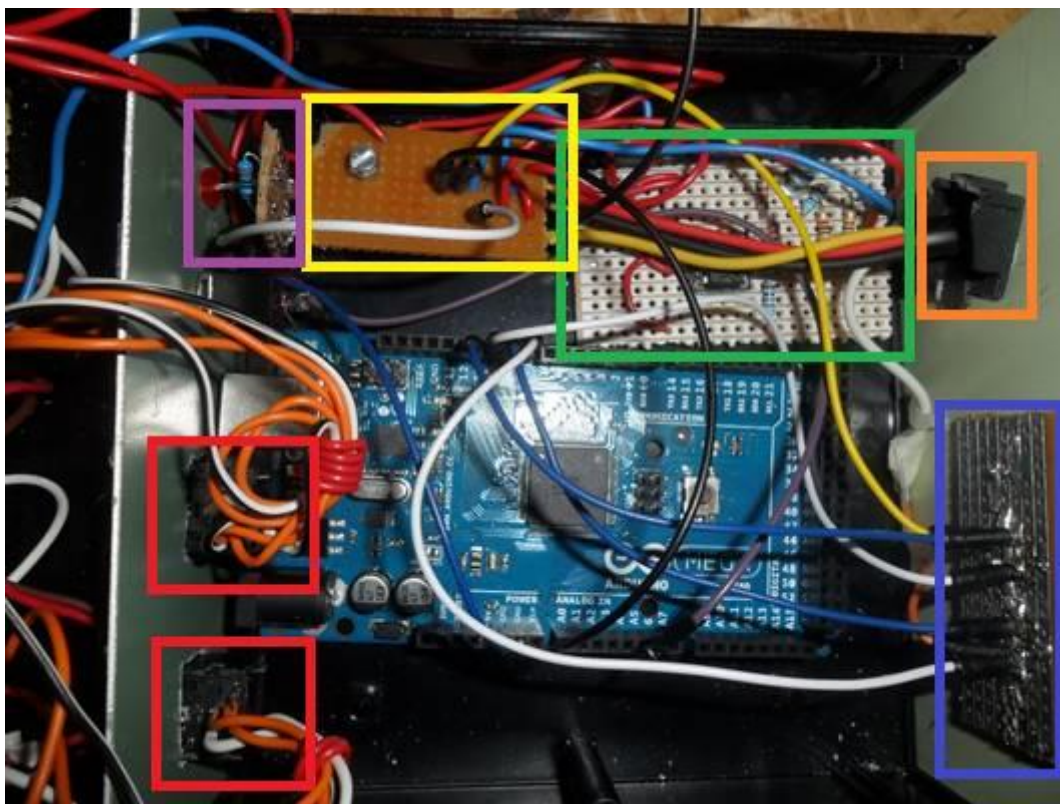


Figura 2.64. Fotografía de la caja de circuitos con recuadros marcando los elementos más importantes de esta sección

En la Figura 2.64. se muestra la mitad inferior de la caja de circuitos. En recuadros rojos, se muestran con mayor detalle los conectores de los sensores de las zapatillas ya mencionados anteriormente. En el recuadro amarillo está la placa de alimentación, y en el verde la placa auxiliar sumadora de offset. A la derecha, en la pared vertical de aluminio, en naranja se marca el conector de alimentación y en azul los pines de salida del robot. Justo en el centro descansa el microcontrolador Arduino. Finalmente, a la

izquierda, en un recuadro morado, una bombilla con una resistencia de protección para indicar cuándo el sistema está encendido.

Conociendo ya todos los componentes de la caja, basta con explicar cómo están dispuestas las conexiones al microcontrolador Arduino. Para esto, se deben diferenciar distintos tipos de conexiones. El modelo utilizado es el Mega 2560. Es un dispositivo grande, por lo que la mayoría de pines quedarán “al aire”.

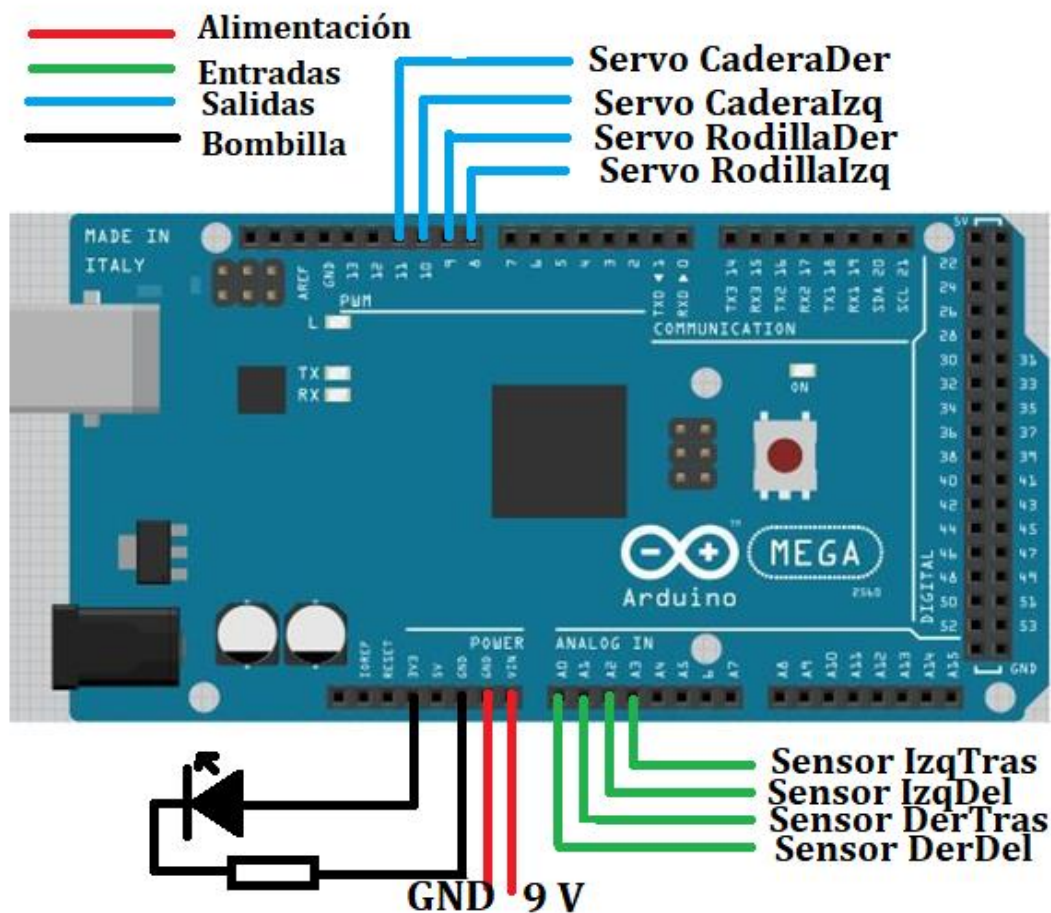


Figura 2.65. Esquema de conexiones del microcontrolador Arduino alojado en la caja de circuitos

Se pueden distinguir cuatro tipos de conexiones para el microcontrolador. En la Figura 2.65. están divididos por colores. El primero y más importante es el de alimentación, marcados en color rojo. Dos cables son necesarios para suministrar corriente. A diferencia de las placas de acondicionamiento, no es necesario conectar los -9 V desde la placa de alimentación a Arduino.

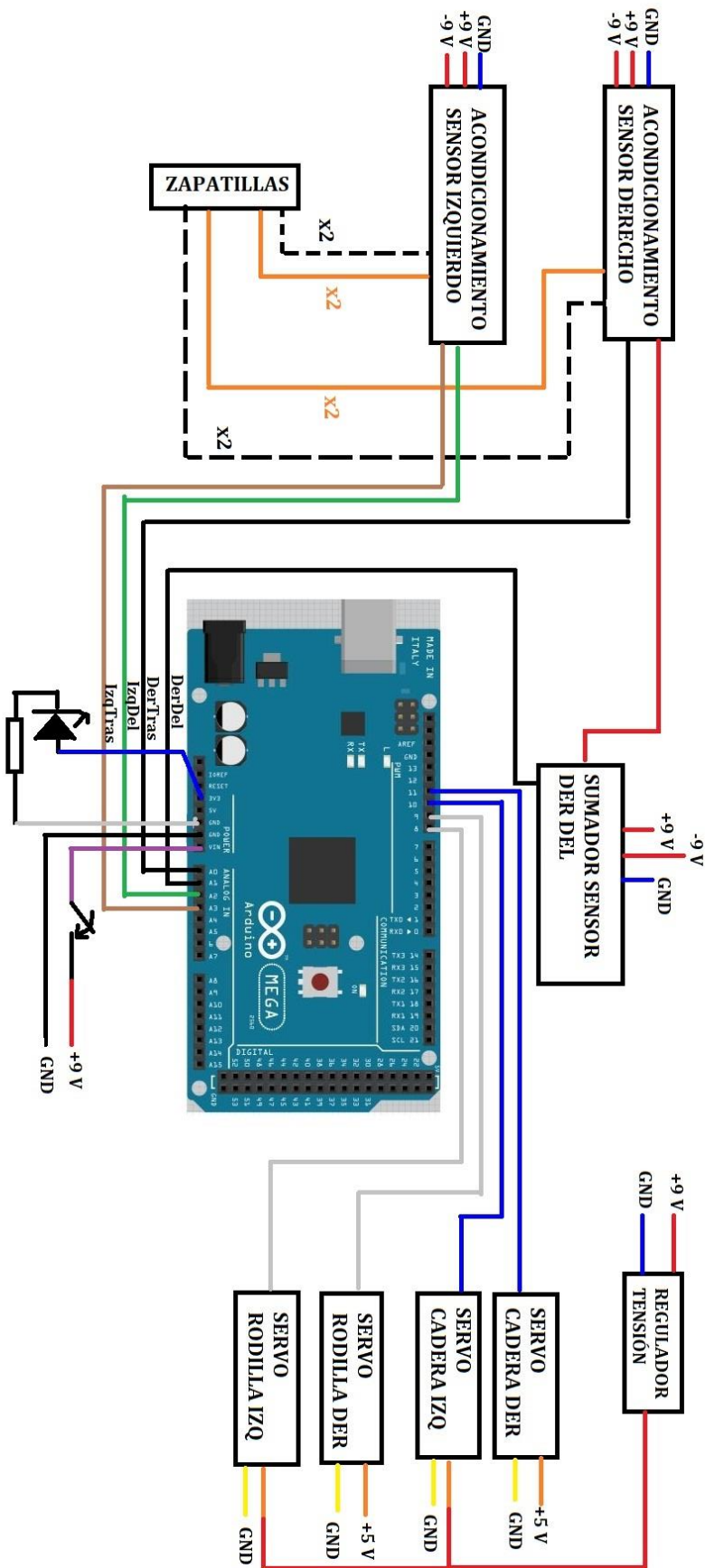
Las conexiones representadas en verde suponen el final del hardware de la primera etapa del proyecto. Estos cables vienen directamente de las placas de acondicionamiento de los sensores, y la mayoría tienen diferentes colores según el sensor: sensor derecho delantero (negro), sensor derecho trasero (también negro, pero llega desde la placa auxiliar sumadora), sensor izquierdo delantero (verde) y sensor izquierdo trasero (dorado).

Una vez realizado todo el procesamiento necesario de las señales provenientes de las conexiones verdes, los cables representados en azul llegarán hasta los pines que unen a los servomotores del robot. Estos cables, en la realidad, son dos de color azul y dos de color blanco. Nótese que la alimentación de los motores se suministra directamente desde la fuente, sin que pase por el microcontrolador.

Por último, se utiliza un circuito independiente para encender un LED de color rojo cuando el microcontrolador está encendido. Se prefiere alimentar desde la salida de 3,3 V propia de Arduino en vez de suministrarlo con la de 9 V para añadir mayor protección a la bombilla.

También es interesante apreciar que todas las tierras están interconectadas, por lo que es indiferente tomar una de tantas tomas GND del microcontrolador como utilizar una conexión directa desde los 0 V de la placa de alimentación.

En la página siguiente se adjunta un diagrama con las conexiones de todo el sistema completo. Nótese que, por simplicidad, se ha suprimido del grafo la placa auxiliar de alimentación. Además, los cables blancos se han representado como grises, y los mixtos blanco-negro como líneas rayadas.



3. SOFTWARE

3.1. Fases de la bipedestación

Antes de poder describir en qué consiste el programa implementado en Arduino es necesario conocer al detalle cómo funcionan las fases de la bipedestación. Este tema ya se trató de forma somera en la introducción, pero sin aportar algunos detalles ciertamente importantes a la hora de implementar el software.

Se recuerdan de nuevo las etapas en el proceso de levantarse:

- **Fase de flexión.** Comienza con la iniciación del movimiento y termina cuando las nalgas se separan del asiento. Tanto el tronco como la pelvis, ya rotados, generan el momento que permite levantarse.
- **Fase de transferencia del momento.** Comienza desde que las nalgas se separan del asiento hasta que los tobillos están en su mayor estado de flexión. Es aquí cuando el momento generado en la fase anterior se transfiere al resto del cuerpo.
- **Fase de extensión.** Abarca desde el mayor estado de flexión de los tobillos hasta que la cadera deja de extenderse. Es en esta fase donde la persona pasa de un estado de flexión general a estar completamente erguida.
- **Fase de estabilización.** La cuarta y última fase comienza cuando la cadera ha terminado de extenderse y termina cuando toda acción relativa a la estabilización del cuerpo se haya completado. Este periodo marca el final del proceso de bipedestación, si bien no es fácil definir cuándo acaba, pues a menudo se producen pequeños balanceos sucesivos que suponen una pérdida momentánea del equilibrio.

Así pues, estos movimientos y esfuerzos deben tenerse en cuenta a la hora de implementar un sistema basado en la rotación de motores acoplados a las rodillas y a las caderas.

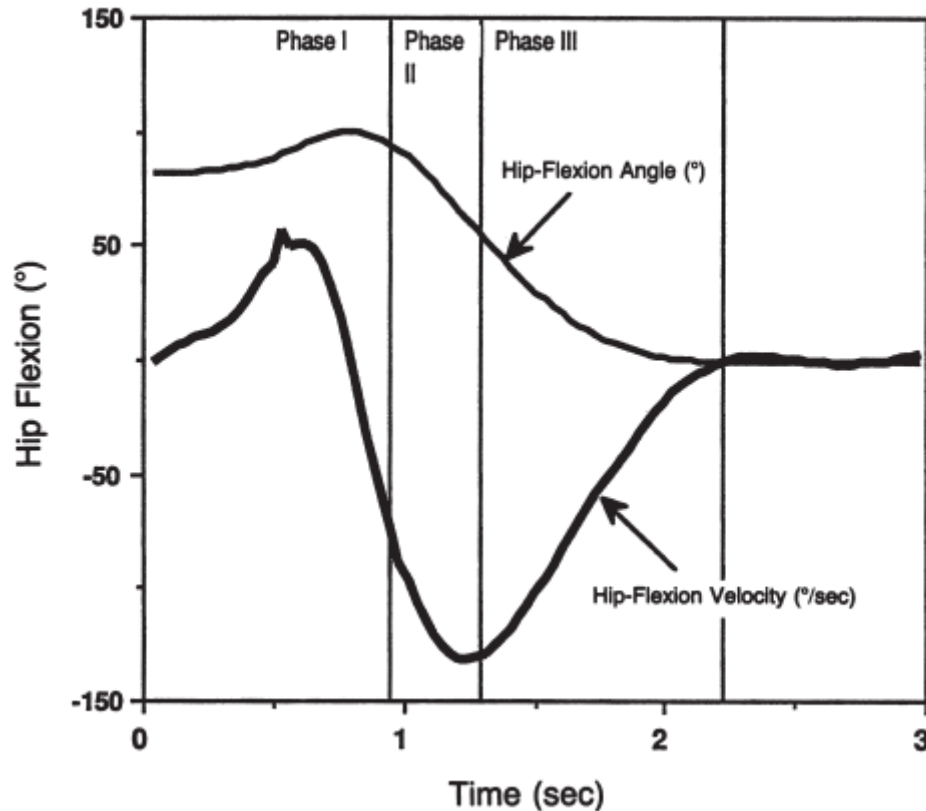


Figura 3.1. Flexión de caderas (ángulo y velocidad) en cada una de las fases expuestas para levantarse

En la Figura 3.1. (Schenkman, M. et al., 1990) se muestra cómo varía el ángulo de torsión de las caderas, así como la velocidad de flexión en cada una de las fases ya expuestas. En la gráfica, es especialmente notable una gran velocidad de giro en la primera fase, donde la persona pretende iniciar el movimiento realizando la transferencia del momento.

Debe notarse también cómo, en la tercera fase, las caderas se enderezan una vez el cuerpo ya está levantado, si bien lo hacen a una menor velocidad que en la primera fase.

En definitiva, las caderas se flexionan casi del todo durante la primera fase, y se terminan enderezando en la tercera fase.

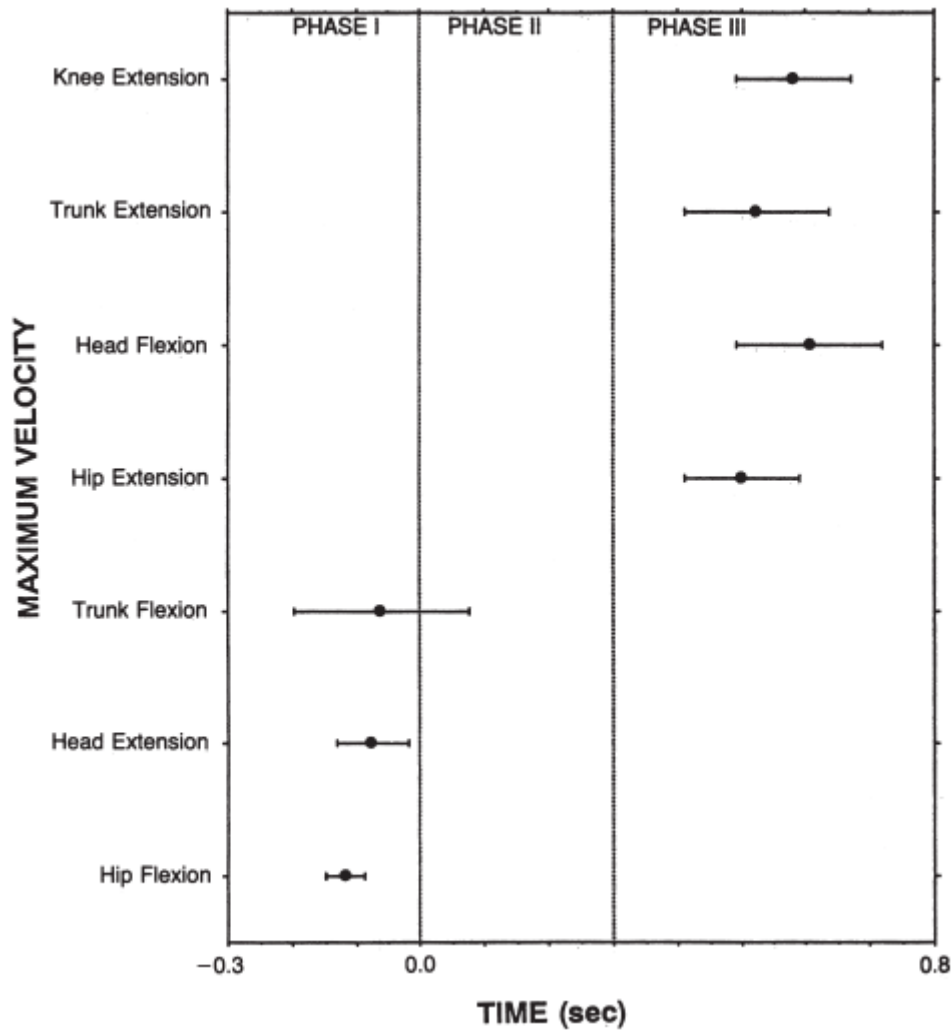


Figura 3.2. Comparación de velocidades máximas alcanzadas en cada una de las fases

Algo similar ocurriría en el caso de las rodillas. Esta cuestión, al igual que otras no tan relevantes para este trabajo, se ilustran en la Figura 3.2. Schenkman utilizó este grafo para comparar cuándo se alcanzaban las velocidades máximas de cada uno de los elementos y en función de las etapas, en orden: extensión de rodilla, extensión de tronco, flexión de cabeza, extensión de cadera, flexión de tronco, extensión de cabeza y flexión de cadera. Nótese que la cuarta etapa es la de estabilización, por lo que no se produce movimiento alguno reseñable.

De esta manera se concluyó que las máximas velocidades se alcanzaban durante las etapas primera (flexión) y tercera (extensión), y no durante la segunda fase (transferencia del momento). Esto puede ofrecer una visión más clara de lo que se pretende realizar con el exoesqueleto. Para un sistema automatizado como el propuesto, la transferencia del momento se torna algo innecesario a la hora de accionar los

servomotores, pues no se realiza ningún tipo de control sobre el equilibrio de la máquina. Por tanto, se deduce que para la implementación solo serán imprescindibles las etapas en las que hay movimiento, es decir, las fases de flexión y de extensión.

De la Figura 3.2. también puede obtenerse información del momento en el que se realiza el mayor movimiento de las caderas y de las rodillas. Es apreciable que, para las caderas, se flexionan rápidamente en la primera fase y se extienden en la tercera fase. Para el caso de las rodillas, como ya están inicialmente flexionadas, solamente en la tercera fase se enderezan.

Sin embargo, este estudio fue realizado para personas sanas sin ninguna dificultad para levantarse del asiento. Para una persona que necesita de cierta ayuda, este proceso se torna más lento y menos fluido, y las etapas se mezclan entre sí. En muchos casos, las caderas no terminan de extenderse hasta que buena parte de las rodillas han terminado de enderezarse, con el fin de poder mantener el equilibrio durante el proceso. Dado que el exoesqueleto no cuenta con un sistema de equilibrio autónomo, el sistema se verá inspirado en el modelo de bipedestación propuesto.

3.2. Diagrama de estados del sistema

Según se ha visto en el apartado anterior, son necesarios dos procesos de transición a la hora de levantarse completamente. Durante todo el estudio, se dará por hecho que la operación de sentarse cuando se está levantado funciona de forma inversa a lo ya descrito.

Estos dos procesos de transición darán lugar, por tanto, a tres estados permanentes o estables: sentado (1), intermedio (2) y levantado (3). Así, desde el estado 1 solo se puede pasar al 2, desde el 2 se puede volver al 1 o avanzar al 3, y desde el 3 solo se puede volver al 2. De esta forma, se puede crear un sistema funcional con una cierta sensación de libertad por parte del usuario, así como de naturalidad en sus movimientos.

El paso de un estado a otro viene condicionado por las señales obtenidas en los sensores de presión instalados en los pies. Atendiendo a los cambios de presión que se producen a la hora de realizar estos movimientos, se pueden dar las órdenes adecuadas a los motores. Para comenzar a levantarse (estado 1 a estado 2), por ejemplo, se suele cargar

un mayor peso sobre la parte delantera de los pies. Por tanto, si el sistema se encuentra en el estado 1 y se presionan ambos sensores delanteros, se procederá a avanzar al estado 2 de transición accionando los motores de la forma correspondiente.

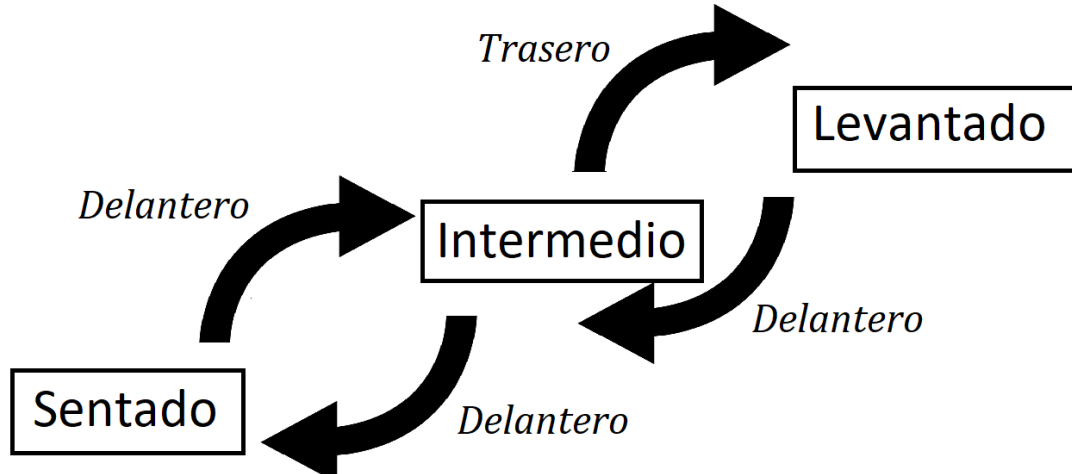


Figura 3.3. Diagrama de estados del sistema

Se define así el diagrama de estados representado en la Figura 3.3. Comenzando inicialmente en el estado “Levantado”, deben seguirse las flechas para cambiar a otro estado. Para activar el proceso, se deben apretar los dos sensores que figuran junto a cada flecha. Por ejemplo, si se quiere pasar del estado Levantado al Intermedio, se deben presionar los dos sensores delanteros. Si las dos zapatillas no captan la presión, o si además se presionan las galgas traseras, el sistema no cambiará.

3.3. Diagrama de flujo del sistema

No debe confundirse el diagrama de estados con el diagrama de flujo del sistema. Si bien los dos diagramas se refieren al mismo proceso de cambio del esquema principal, se realizan dos enfoques muy distintos para cada uno de ellos. Mientras que el diagrama de estados alude a un cambio más conceptual de la posición del robot, el diagrama de flujo se centra más en las funciones que se invocan dentro de Arduino. Ambos esquemas se refieren al mismo cuadro, aunque el diagrama de flujo se centra más en el software y cuenta con mayor detalle.

Para entender el diagrama de flujo, es necesario asociar los estados y los cambios entre ellos como funciones y números, de modo que sea posible trasladar el diagrama de estados a código fuente.

Por ello, a cada estado se le asignará un número alojado en la variable “estado”:

- **Sentado:** 1
- **Intermedio:** 2
- **Levantado:** 3

De igual modo, se traducirán los cambios de estado (las flechas en el diagrama de estados) a funciones cuando se activen los sensores correspondientes:

- **Levantado → Intermedio:** Sentarse – Fase 1 – `sentarse1()`
- **Intermedio → Sentado:** Sentarse – Fase 2 – `sentarse2()`
- **Sentado → Intermedio:** Levantarse – Fase 1 – `levantarse1()`
- **Intermedio → Levantado:** Levantarse – Fase 2 – `levantarse2()`

Nótese que los nombres de las funciones, como `sentarse1()`, no tiene argumentos de entrada ni argumentos de salida.

Dentro de cada función se define el nuevo ángulo y la velocidad a la que cambian los motores para alcanzar una nueva postura. Los ángulos que varían dependen de la fase: 60° las rodillas y 40° las caderas entre los estados 2 y 3, y 20° las rodillas y 70° las caderas entre los estados 1 y 2. Hay que tener en cuenta que las dos etapas no suman 90° porque la superficie sobre la que se apoya el robot no es totalmente horizontal.

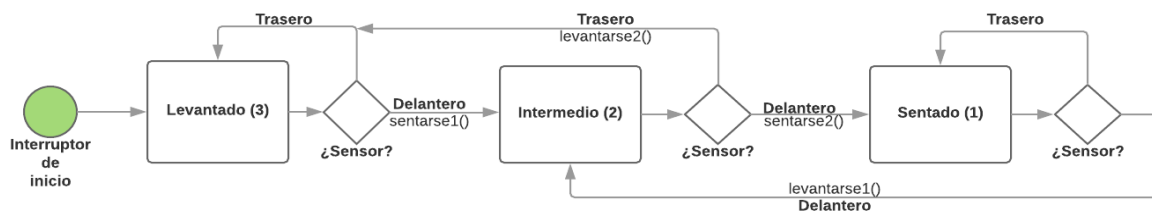


Figura 3.4. Diagrama de flujo del sistema

El diagrama de flujo se centra en los cambios de estados según cómo se presionan los sensores y las funciones a las que invoca. Debe tenerse en cuenta que, para cada lectura

de sensores y, por tanto, para cada cambio de estado, deben pasar 2.5 segundos (siendo adaptable a conveniencia) para incrementar la fiabilidad.

Además, el tiempo que tarda en rotar el sensor se añadirá a ese tiempo de espera para la siguiente lectura. Como apunte, ese tiempo adicional serán 20 ms por cada grado que deba moverse el motor que más rote, siendo este tiempo también adaptable a conveniencia. Por ejemplo, para pasar del estado 3 a 2 (de levantado al intermedio), en la función `sentarse1()`, los motores que más rotan son los de las rodillas. Esos motores rotarán hasta 60°, y además se considera que los motores tardan un tiempo despreciable en girar un grado. Por tanto, el tiempo de espera para hacer la siguiente lectura será de:

$$2.5 \text{ segundos} + 20\text{ms} * 60 = 3.7 \text{ segundos}$$

3.4. Análisis del código fuente

A continuación, se expone el código fuente utilizado para procesar los datos recibidos de los cuatro sensores de las zapatillas y enviarlos después al robot por medio de Arduino. Se presenta ahora el código completo, y acto seguido se irá analizando por fragmentos para seguirlo más fácilmente.

```
#include <Servo.h>

void sentarse1();
void sentarse2();
void levantarse1();
void levantarse2();

//Declaracion de servos
Servo servoRodillaIzq;
Servo servoRodillaDer;
Servo servoCaderaIzq;
Servo servoCaderaDer;

//Entradas analogicas para cada sensor
int sensorDerDel = 0;
int sensorDerTras = 1;
int sensorIzqDel = 2;
int sensorIzqTras = 3;

//Angulos de cada servo para posiciones erguidas
int anguloRodillaIzq = 140; //pin 8
int anguloRodillaDer = 50; //pin 9
```

```
int anguloCaderaIzq = 135; //pin 10
int anguloCaderaDer = 90; //pin 11

//Limite de voltaje para decidir si el sensor esta activo
float limiteDerDel = 0.50;
float limiteDerTras = 1.15;
float limiteIzqDel = 1.21;
float limiteIzqTras = 0.26;

//Estado de los sensores (0 == inactivo, 1 == activo)
int sensorDel = 0;
int sensorTras = 0;

//variable auxiliar
int i = 0;

int estado = 3;
//La variable estado indica la etapa del robot
//1: sentado
//2: intermedio
//3: de pie (comienza aqui)

void setup() {
  servoRodillaIzq.attach(8);
  servoRodillaDer.attach(9);
  servoCaderaIzq.attach(10);
  servoCaderaDer.attach(11);

  Serial.begin(9600);

  servoRodillaDer.write(anguloRodillaDer);
  servoRodillaIzq.write(anguloRodillaIzq);
  servoCaderaDer.write(anguloCaderaDer);
  servoCaderaIzq.write(anguloCaderaIzq);

  delay(2500);
}

void loop() {

  //Lectura de los sensores y convertido a voltaje (0-5 V):
  float voltajeDerDel = analogRead(sensorDerDel) * (5.00/1024.00);
  float voltajeDerTras = analogRead(sensorDerTras) * (5.00/1024.00);
  float voltajeIzqDel = analogRead(sensorIzqDel) * (5.00/1024.00);
  float voltajeIzqTras = analogRead(sensorIzqTras) * (5.00/1024.00);

  //Decisor por zapatilla
  if(voltajeDerDel < limiteDerDel && voltajeIzqDel < limiteIzqDel)
    sensorDel = 1;
  else
    sensorDel = 0;

  if(voltajeIzqTras < limiteIzqTras && voltajeDerTras <
limiteDerTras)
    sensorTras = 1;
  else
    sensorTras = 0;

  //Decision de la funcion a llamar segun el estado
  if(estado == 1){
```

```
        if(sensorDel == 1 && sensorTras == 0){
            levantarse1();
        }
    }
    else if(estado == 2){
        if(sensorDel == 1 && sensorTras == 0){
            sentarse2();
        }
        else if(sensorDel == 0 && sensorTras == 1){
            levantarse2();
        }
    }
    else if(estado == 3){
        if(sensorDel == 1 && sensorTras == 0){
            sentarse1();
        }
    }
}
delay(2500);
}

void sentarse1(){ //estado 3 -> 2
    Serial.println("Sentarse - Fase 1");

    for(i = 1; i <= 60; i++){
        servoRodillaDer.write(anguloRodillaDer + i);
        servoRodillaIzq.write(anguloRodillaIzq - i);
        if(i <= 40){
            servoCaderaDer.write(anguloCaderaDer + i);
            servoCaderaIzq.write(anguloCaderaIzq - i);
        }
        delay(20);

        if(i == 40){
            anguloCaderaDer = anguloCaderaDer + i;
            anguloCaderaIzq = anguloCaderaIzq - i;
        }
    }

    anguloRodillaDer = anguloRodillaDer + i;
    anguloRodillaIzq = anguloRodillaIzq - i;

    estado = 2;
}

void sentarse2(){ //estado 2 -> 1
    Serial.println("Sentarse - Fase 2");

    for(i = 1; i <= 70; i++){
        servoCaderaDer.write(anguloCaderaDer + i);
        servoCaderaIzq.write(anguloCaderaIzq - i);
        if(i <= 20){
            servoRodillaDer.write(anguloRodillaDer + i);
            servoRodillaIzq.write(anguloRodillaIzq - i);
        }
        delay(20);

        if(i == 20){
            anguloRodillaDer = anguloRodillaDer + i;
            anguloRodillaIzq = anguloRodillaIzq - i;
        }
    }
}
```

```
}

anguloCaderaDer = anguloCaderaDer + i;
anguloCaderaIzq = anguloCaderaIzq - i;

estado = 1;
}

void levantarse1(){ //estado 1 -> 2
  Serial.println("Levantarse - Fase 1");

  for(i = 1; i <= 70; i++){
    servoCaderaDer.write(anguloCaderaDer - i);
    servoCaderaIzq.write(anguloCaderaIzq + i);
    if(i <= 20){
      servoRodillaDer.write(anguloRodillaDer - i);
      servoRodillaIzq.write(anguloRodillaIzq + i);
    }
    delay(20);

    if(i == 20){
      anguloRodillaDer = anguloRodillaDer - i;
      anguloRodillaIzq = anguloRodillaIzq + i;
    }
  }

  anguloCaderaDer = anguloCaderaDer - i;
  anguloCaderaIzq = anguloCaderaIzq + i;

  estado = 2;
}

void levantarse2(){ //estado 2 -> 3
  Serial.println("Levantarse - Fase 2");

  for(i = 1; i <= 60; i++){
    servoRodillaDer.write(anguloRodillaDer - i);
    servoRodillaIzq.write(anguloRodillaIzq + i);
    if(i <= 40){
      servoCaderaDer.write(anguloCaderaDer - i);
      servoCaderaIzq.write(anguloCaderaIzq + i);
    }
    delay(20);

    if(i == 40){
      anguloCaderaDer = anguloCaderaDer - i;
      anguloCaderaIzq = anguloCaderaIzq + i;
    }
  }

  anguloRodillaDer = anguloRodillaDer - i;
  anguloRodillaIzq = anguloRodillaIzq + i;

  estado = 3;
}
```

El lenguaje utilizado en Arduino es C++. Se puede, además, dividir el programa en cuatro partes diferenciadas. Primero, la declaración de variables globales que se irán actualizando eventualmente; segundo, una función llamada `setup()` que se invocará una vez al principio del programa; tercero, una función `loop()` que se repetirá indefinidamente hasta que se corte la corriente; y cuarto, las funciones que hacen mover los servomotores. Las funciones `loop()` y `setup()` se repiten en todos los programas de Arduino.

Además del código incluido en este documento, se han añadido líneas auxiliares en las que mostraban por pantalla los valores de cada uno de los sensores, así como si estaban activos o no los pares de sensores delanteros o traseros. Estas se han utilizado para realizar pruebas del sistema, pero no son necesarios para el funcionamiento final del código, por lo que no se han incluido aquí.

Variables globales

Se analizarán primero las variables globales a utilizar.

```
#include <Servo.h>

void sentarse1();
void sentarse2();
void levantarse1();
void levantarse2();

//Declaracion de servos
Servo servoRodillaIzq;
Servo servoRodillaDer;
Servo servoCaderaIzq;
Servo servoCaderaDer;

//Entradas analogicas para cada sensor
int sensorDerDel = 0;
int sensorDerTras = 1;
int sensorIzqDel = 2;
int sensorIzqTras = 3;

//Angulos de cada servo para posiciones erguidas
int anguloRodillaIzq = 140; //pin 8
int anguloRodillaDer = 50; //pin 9
int anguloCaderaIzq = 135; //pin 10
int anguloCaderaDer = 90; //pin 11

//Limite de voltaje para decidir si el sensor esta activo
float limiteDerDel = 0.50;
float limiteDerTras = 1.15;
```

```
float limiteIzqDel = 1.21;
float limiteIzqTras = 0.26;

//Estado de los sensores (0 == inactivo, 1 == activo)
int sensorDel = 0;
int sensorTras = 0;

//variable auxiliar
int i = 0;

int estado = 3;
//La variable estado indica la etapa del robot
//1: sentado
//2: intermedio
//3: de pie (comienza aqui)
```

```
#include <Servo.h>
```

En la primera línea se incluye la librería Servo, la cual contiene todas las funciones relativas a los servomotores. Con ella se podrán controlar de una forma sencilla y sin tener que recurrir a complicados algoritmos en C++. Las funciones heredadas de esta librería se utilizarán más adelante.

```
void sentarse1 ();
void sentarse2 ();
void levantarse1 ();
void levantarse2 ();
```

Esta es la declaración de las funciones que llamarán a los servomotores. Cada una de estas ha sido ya explicada con el diagrama de flujo. Debe recordarse que son funciones que no necesitan de argumentos de entrada, ni tiene de salida.

```
//Declaracion de servos
Servo servoRodillaIzq;
Servo servoRodillaDer;
Servo servoCaderaIzq;
Servo servoCaderaDer;
```

Se declaran los nombres de cada uno de los servomotores. En todo el programa se utilizarán las abreviaturas Der para derecha, Izq para izquierda, Del para delantero y Tras para trasero. En estas líneas se aprecia que las palabras rodilla y cadera no tienen abreviatura. Esto se debe a que solo serán llamadas en las funciones declaradas anteriormente.


```
//Entradas analogicas para cada sensor
int sensorDerDel = 0;
int sensorDerTras = 1;
int sensorIzqDel = 2;
int sensorIzqTras = 3;
```

Ha de tenerse en cuenta que los cuatro sensores tendrán una entrada analógica cada uno en el Arduino, y los motores una salida de control cada uno. A cada sensor se le asigna entonces una entrada, que en Arduino se les añade una A (de analógica) al principio. Así, al sensor derecho trasero, o `sensorDerTras`, le corresponderá la entrada A1.

```
//Angulos de cada servo para posiciones erguidas
int anguloRodillaIzq = 140; //pin 8
int anguloRodillaDer = 50; //pin 9
int anguloCaderaIzq = 135; //pin 10
int anguloCaderaDer = 90; //pin 11
```

Es imprescindible conocer las posiciones de cada servomotor al principio del programa, esto es, cuando el robot está levantado. Los ángulos iniciales dependen de su disposición dentro del mismo robot, así como del ángulo que tenían cuando se hizo la instalación. Aquí se indican estos ángulos, junto con los pines de salida en un comentario (que se asignarán más adelante). Hay que tener en cuenta que los ángulos del lado derecho rotarán hacia un ángulo mayor que el inicial, mientras que los del lado izquierdo rotarán a uno menor. No debe olvidarse que los servomotores pueden marcar un ángulo de hasta 180°.

```
//Limite de voltaje para decidir si el sensor esta activo
float limiteDerDel = 0.50;
float limiteDerTras = 1.15;
float limiteIzqDel = 1.21;
float limiteIzqTras = 0.26;
```

Al iniciar el programa, en reposo, cada sensor marcará un voltaje relativamente estable. Este voltaje será previsiblemente el más alto que indicará Arduino. Cuando un sensor es presionado de forma intencional, la resistencia de la galga extensiométrica será mayor y el voltaje descenderá. Si ese voltaje es menor que el indicado en estas líneas, se considerará que ese sensor está activo. Debe tenerse en cuenta que estos valores se han obtenido de forma experimental.

```
//Estado de los sensores (0 == inactivo, 1 == activo)
int sensorDel = 0;
int sensorTras = 0;
```

Cada sensor puede considerarse activo si su valor es inferior a su límite. Sin embargo, el sistema solo responderá en el caso de que ambos sensores, delantero o traseros, estén activos al mismo tiempo. Así, las variables `sensorDel` y `sensorTras` solo se activarán cuando los dos sensores delantero o trasero estén activos, respectivamente. Estas variables marcarán un 0 en reposo, y 1 si están activos.

```
//variable auxiliar
int i = 0;
```

Esta es una variable auxiliar que se utilizará en los bucles de las distintas funciones.

```
int estado = 3;
//La variable estado indica la etapa del robot
//1: sentado
//2: intermedio
//3: de pie (comienza aqui)
```

La variable `estado` marca en qué posición está el robot dentro del diagrama de estados ya explicado. Se recuerda que comienza en el estado 3, o levantado, tal y como figura en el comentario.

Función `setup()`

```
void setup() {
  servoRodillaIzq.attach(8);
  servoRodillaDer.attach(9);
  servoCaderaIzq.attach(10);
  servoCaderaDer.attach(11);

  Serial.begin(9600);

  servoRodillaDer.write(anguloRodillaDer);
  servoRodillaIzq.write(anguloRodillaIzq);
  servoCaderaDer.write(anguloCaderaDer);
  servoCaderaIzq.write(anguloCaderaIzq);

  delay(2500);
}
```

Esta función se invocará de forma automática cuando arranque el programa. De esta forma, Arduino puede realizar algunas sentencias de inicialización que solo deben ser invocadas una vez al principio de la ejecución.

```
servoRodillaIzq.attach(8);  
servoRodillaDer.attach(9);  
servoCaderaIzq.attach(10);  
servoCaderaDer.attach(11);
```

Una tarea típica que se suele realizar en la función `setup()` es la de asignar los pines de ciertos elementos, como los servos. La manera para conseguir esto es con la sentencia `nombreDeServo.attach(numeroDePin);` sacada de la librería `Servo`. Por tanto, en la línea `servoRodillaDer.attach(9);` se está asignando la señal de control del servomotor de la rodilla derecha al pin número 9. Si se quiere enviar esa señal al servo, se deberá conectar el pin 9 al control del mismo.

```
Serial.begin(9600);
```

A la hora de programar y realizar pruebas, el microcontrolador deberá ser conectado al ordenador vía USB. El número que se indica en esta sentencia marca la velocidad a la que se comunicarán Arduino y el ordenador, en baudios o bits por segundo. Para este caso, no es necesario contar con una velocidad demasiado alta, así que se ha optado por una intermedia.

```
servoRodillaDer.write(anguloRodillaDer);  
servoRodillaIzq.write(anguloRodillaIzq);  
servoCaderaDer.write(anguloCaderaDer);  
servoCaderaIzq.write(anguloCaderaIzq);
```

Aquí se utilizan los valores globales definidos de los ángulos iniciales de cada servomotor en el estado levantado. No sirve de nada definir estos ángulos si no se transmiten al servo, por lo que estas líneas son imprescindibles. La operación se realiza con `nombreDeServo.write(anguloDeServo);` donde el nombre del servo ya debía estar previamente declarado en las variables globales. Esta función forzará al motor a colocarse en el ángulo indicado lo más rápidamente posible.

```
delay(2500);
```

Sentencia muy común que sirve para retrasar el programa 2500 ms, o 2.5 s. Durante este tiempo, el programa no continuará con las siguientes líneas de código. Esta primera espera sirve para que los sensores se estabilicen en su valor correcto y no se activen de manera imprevista.

Función loop()

```
void loop() {  
  
    //Lectura de los sensores y convertido a voltaje (0-5 V):  
    float voltajeDerDel = analogRead(sensorDerDel) * (5.00/1024.00);  
    float voltajeDerTras = analogRead(sensorDerTras) * (5.00/1024.00);  
    float voltajeIzqDel = analogRead(sensorIzqDel) * (5.00/1024.00);  
    float voltajeIzqTras = analogRead(sensorIzqTras) * (5.00/1024.00);  
  
    //Decisor por zapatilla  
    if(voltajeDerDel < limiteDerDel && voltajeIzqDel < limiteIzqDel)  
        sensorDel = 1;  
    else  
        sensorDel = 0;  
  
    if(voltajeIzqTras < limiteIzqTras && voltajeDerTras <  
limiteDerTras)  
        sensorTras = 1;  
    else  
        sensorTras = 0;  
  
    //Decision de la funcion a llamar segun el estado  
    if(estado == 1){  
        if(sensorDel == 1 && sensorTras == 0){  
            levantarse1();  
        }  
    }  
    else if(estado == 2){  
        if(sensorDel == 1 && sensorTras == 0){  
            sentarse2();  
        }  
        else if(sensorDel == 0 && sensorTras == 1){  
            levantarse2();  
        }  
    }  
    else if(estado == 3){  
        if(sensorDel == 1 && sensorTras == 0){  
            sentarse1();  
        }  
    }  
    delay(2500);  
}
```

La función `loop()` se repetirá automáticamente cada vez que llegue al final, sin volver a pasar por la función `setup()`. En este caso, este bucle se encargará de realizar la lectura

de los sensores y decidir si es oportuno realizar la llamada a alguna función que mueva a los servomotores a una nueva posición.

```
//Lectura de los sensores y convertido a voltaje (0-5 V):  
float voltajeDerDel = analogRead(sensorDerDel) * (5.00/1024.00);  
float voltajeDerTras = analogRead(sensorDerTras) *  
(5.00/1024.00);  
float voltajeIzqDel = analogRead(sensorIzqDel) * (5.00/1024.00);  
float voltajeIzqTras = analogRead(sensorIzqTras) *  
(5.00/1024.00);
```

Estas líneas suponen el núcleo de la lectura de los sensores. De igual manera que se ha hecho antes, cada línea se centra en un sensor en concreto. Para ello se recurre a la función `analogRead(numeroDePin)`; la cual lee el voltaje del pin introducido, que contiene la salida del acondicionamiento de la galga correspondiente. Sin embargo, el valor que asigna Arduino no son directamente voltios, sino que utiliza una escala de 0 a 1024 para ordenar los valores que puede leer, de 0 a 5 V. Es decir, si en el pin indicado hay 2.5 V, la función `analogRead()` devolverá 512. Si en ese pin hay más de 5 V, la función devolverá 1024. Si hay menos de 0 V, devolverá 0. Como nos interesa que ese valor esté indicado en voltios, basta con multiplicar el resultado por 5/1024.

```
//Decisor por zapatilla  
if(voltajeDerDel < limiteDerDel && voltajeIzqDel < limiteIzqDel)  
  sensorDel = 1;  
else  
  sensorDel = 0;  
  
if(voltajeIzqTras < limiteDerTras && voltajeDerTras <  
limiteIzqTras)  
  sensorTras = 1;  
else  
  sensorTras = 0;
```

Cada conjunto de líneas se refiere al grupo de sensores delanteros o traseros. En él se decide si la pareja de sensores está activa o no. Esto se realiza comparando el valor real obtenido con el límite establecido. Si los dos sensores están por debajo del límite, entonces el sensor (`sensorDel` o `sensorTras`) indicará un 1. Si no, devolverá un 0.

```
//Decision de la funcion a llamar segun el estado  
if(estado == 1){  
  if(sensorDel == 1 && sensorTras == 0){
```

```

    levantarse1 ();
  }
}
else if(estado == 2){
  if(sensorDel == 1 && sensorTras == 0){
    sentarse2 ();
  }
  else if(sensorDel == 0 && sensorTras == 1){
    levantarse2 ();
  }
}
else if(estado == 3){
  if(sensorDel == 1 && sensorTras == 0){
    sentarse1 ();
  }
}
}

```

Esta sección se ocupa de controlar el acceso de una función u otra para mover los motores correspondientes. Para esto es necesario conocer el estado en el que se encuentra el sistema, pues algunas funciones están prohibidas para ciertos estados. En el estado 3 (levantado), por ejemplo, está prohibido llamar a la función `sentarse2()`. Entonces, se comprueba primero el estado actual y después los sensores activos. Si cumple las condiciones, se invocarán las funciones para mover el robot. Nótese que no se llamará a ninguna función cuando tanto `sensorDel` como `sensorTras` están activos.

```
delay(2500);
```

Inclusión de un retardo periódico. En este caso, este retardo sirve para evitar los cambios bruscos de presión en los sensores que suceden de improviso.

Funciones de cambios de estado

```

void sentarse1() { //estado 3 -> 2
  Serial.println("Sentarse - Fase 1");

  for(i = 1; i <= 60; i++){
    servoRodillaDer.write(anguloRodillaDer + i);
    servoRodillaIzq.write(anguloRodillaIzq - i);
    if(i <= 40){
      servoCaderaDer.write(anguloCaderaDer + i);
      servoCaderaIzq.write(anguloCaderaIzq - i);
    }
    delay(20);

    if(i == 40){
      anguloCaderaDer = anguloCaderaDer + i;
      anguloCaderaIzq = anguloCaderaIzq - i;
    }
  }
}

```

```
    anguloRodillaDer = anguloRodillaDer + i;
    anguloRodillaIzq = anguloRodillaIzq - i;

    estado = 2;
}

void sentarse2(){ //estado 2 -> 1
    Serial.println("Sentarse - Fase 2");

    for(i = 1; i <= 70; i++){
        servoCaderaDer.write(anguloCaderaDer + i);
        servoCaderaIzq.write(anguloCaderaIzq - i);
        if(i <= 20){
            servoRodillaDer.write(anguloRodillaDer + i);
            servoRodillaIzq.write(anguloRodillaIzq - i);
        }
        delay(20);

        if(i == 20){
            anguloRodillaDer = anguloRodillaDer + i;
            anguloRodillaIzq = anguloRodillaIzq - i;
        }
    }

    anguloCaderaDer = anguloCaderaDer + i;
    anguloCaderaIzq = anguloCaderaIzq - i;

    estado = 1;
}

void levantarse1(){ //estado 1 -> 2
    Serial.println("Levantarse - Fase 1");

    for(i = 1; i <= 70; i++){
        servoCaderaDer.write(anguloCaderaDer - i);
        servoCaderaIzq.write(anguloCaderaIzq + i);
        if(i <= 20){
            servoRodillaDer.write(anguloRodillaDer - i);
            servoRodillaIzq.write(anguloRodillaIzq + i);
        }
        delay(20);

        if(i == 20){
            anguloRodillaDer = anguloRodillaDer - i;
            anguloRodillaIzq = anguloRodillaIzq + i;
        }
    }

    anguloCaderaDer = anguloCaderaDer - i;
    anguloCaderaIzq = anguloCaderaIzq + i;

    estado = 2;
}

void levantarse2(){ //estado 2 -> 3
    Serial.println("Levantarse - Fase 2");

    for(i = 1; i <= 60; i++){
        servoRodillaDer.write(anguloRodillaDer - i);
        servoRodillaIzq.write(anguloRodillaIzq + i);
```

```

if(i <= 40){
  servoCaderaDer.write(anguloCaderaDer - i);
  servoCaderaIzq.write(anguloCaderaIzq + i);
}
delay(20);

if(i == 40){
  anguloCaderaDer = anguloCaderaDer - i;
  anguloCaderaIzq = anguloCaderaIzq + i;
}
}

anguloRodillaDer = anguloRodillaDer - i;
anguloRodillaIzq = anguloRodillaIzq + i;

estado = 3;
}

```

Esta parte del programa se ocupa de hacer rotar los motores y cambiar de estado. Las cuatro funciones existentes no realizan ningún tipo de control, pues ya se ha hecho antes de llamarlas. Por tanto, se limita a mover al robot a la velocidad adecuada y establecer un nuevo estado. Como las cuatro funciones son similares, se procederá a analizar solo la primera de ellas.

```

void sentarse1(){ //estado 3 -> 2
  Serial.println("Sentarse - Fase 1");

  for(i = 1; i <= 60; i++){
    servoRodillaDer.write(anguloRodillaDer + i);
    servoRodillaIzq.write(anguloRodillaIzq - i);
    if(i <= 40){
      servoCaderaDer.write(anguloCaderaDer + i);
      servoCaderaIzq.write(anguloCaderaIzq - i);
    }
    delay(20);

    if(i == 40){
      anguloCaderaDer = anguloCaderaDer + i;
      anguloCaderaIzq = anguloCaderaIzq - i;
    }
  }

  anguloRodillaDer = anguloRodillaDer + i;
  anguloRodillaIzq = anguloRodillaIzq - i;

  estado = 2;
}

```

La función `sentarse1()` se ocupa de pasar del estado 3, levantado, al estado 2, intermedio. Siempre habrá un comentario clarificando el cambio de estados. Como ya se ha

mencionado antes, estas funciones no necesitan argumentos de entrada, ni tampoco devuelven ninguno.

La función `Serial.println()` se utiliza para mostrar un texto en pantalla. No es estrictamente necesaria para el funcionamiento, pero sí sirve a la hora de hacer pruebas y ensayos.

Acto seguido, el programa entra en un bucle de 60 iteraciones, que es el ángulo mayor de los motores a mover. La sentencia `nombreDeServo.write(anguloDeServo + i);` se limita a mover el motor un grado más cada vez. Como el ángulo inicial del servo es constante en esta función, solo varía `i` en uno cada vez que entra al bucle. No se debe olvidar que, para bajar, los motores del lado derecho deben incrementar el ángulo, y los motores del lado izquierdo deben decrementarlo.

A posteriori, dentro de cada iteración, los motores que deban girar menos lo harán grado a grado hasta llegar a su ángulo. En este caso, los motores de la cadera rotarán hasta 40°, girando un solo grado en cada iteración del bucle.

Para reducir la velocidad a la que se mueven los motores, se introduce un retardo de 20 ms en cada grado que recorren. Esto se realiza para que el movimiento sea más natural y evite desestabilizar el equilibrio del robot al moverse tan rápido.

Una vez los motores terminen de girar, cada servomotor apunta su nuevo ángulo y el nuevo estado, saliendo después de la función. Después de esto, el programa volverá a la primera línea de la función `loop()`, continuando el ciclo del proceso.

4. PRUEBAS Y ENSAYOS

Este apartado se centra en los procedimientos que se han seguido a lo largo del trabajo para comprobar el correcto funcionamiento del sistema, primero de forma parcial y después de forma global.

En esta sección se detallan todos los procesos empleados hasta la finalización del proyecto. Algunas ideas han sido modificadas posteriormente y otras, incluso, se han descartado por la poca efectividad o fiabilidad de estas. Todos los conceptos abordados en el transcurso del trabajo se presentan en este punto como adición a los sistemas finalmente propuestos, con el fin de ofrecer más ideas al lector o explicar ciertos conceptos no tratados con anterioridad.

4.1. Puesta a punto de los sensores

Reajuste de voltajes

Tal como se lleva repitiendo durante todo el documento, el sistema completo se divide en tres partes diferenciadas: los sensores, la programación software y el robot actuador. Para comenzar el proyecto, los sensores conforman una pieza clave, pues son los encargados de transmitir la intención del usuario al decisor del sistema.

Es imprescindible tener en mente que la señal recibida del sensor una vez acondicionado debe estar comprendido en el rango de voltaje que Arduino acepta en el microcontrolador, esto es, entre 0 y 5 V. Es especialmente importante no olvidar este planteamiento, pues obliga a la señal de salida de los sensores a ser positiva, y al mismo tiempo no ser demasiado amplia. Este es el motivo por el que la amplificación realizada en el acondicionamiento es inversora, si bien esto ya fue explicado al detalle en la sección de Hardware.

Sin embargo, antes de comenzar a soldar los componentes a una placa, es altamente recomendable montar el circuito en una placa de pruebas protoboard. En este primer intento, el voltaje a la salida rondaba los 3 V en reposo para todos los sensores, salvo el trasero derecho, que bajaba hasta los 2 V. Estos valores se obtuvieron con la

misma configuración e idéntica ganancia que en el montaje final. La diferencia cuando se presionaba rondaba los 0.2 V en todos los sensores.

Lo ideal hubiera sido amplificar la señal con una ganancia superior a la utilizada, de modo que llegase casi hasta los 5 V máximos que acepta Arduino. Este concepto, no obstante, no es demasiado efectivo en un escenario real, pues el voltaje puede cambiar por fenómenos inesperados en ciertos momentos.

Una vez realizado el montaje con los componentes soldados, el voltaje recibido había perdido un offset notable. De media, las señales de los cuatro sensores habían perdido 2 V de continua, en reposo o presionando.

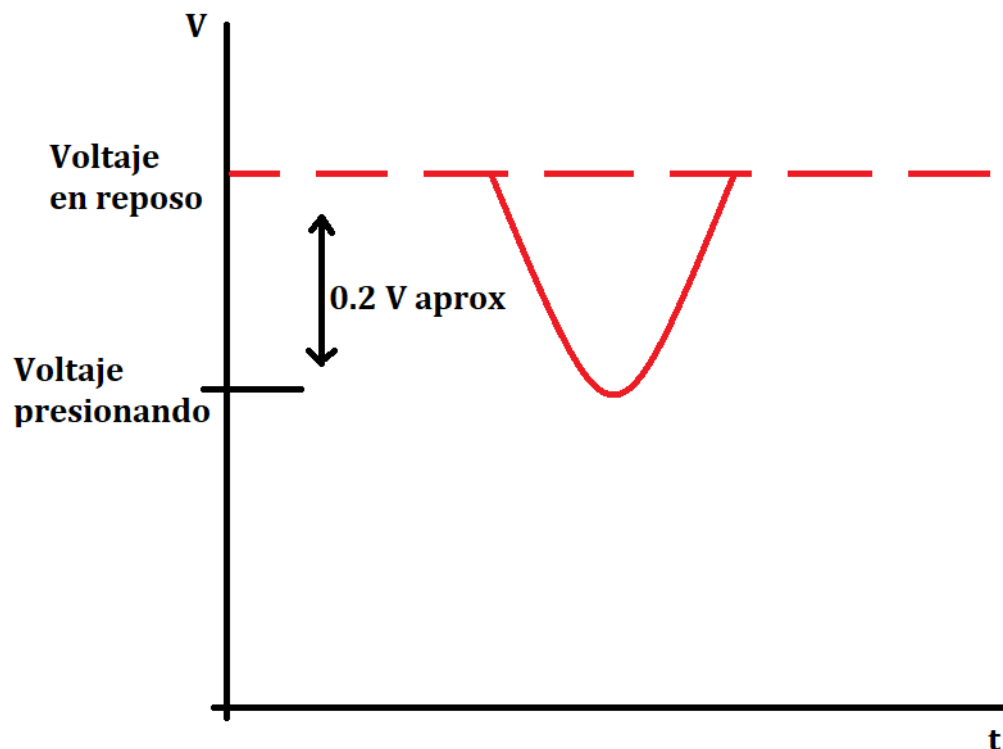


Figura 4.1. Representación básica del voltaje de los sensores

La Figura 4.1 sirve para comprender el problema que ha supuesto la pérdida de este voltaje de 2 V en cada sensor. En el prototipo final, cada sensor se sitúa en torno a un nivel de voltaje cercano a 0.6 V en reposo, si bien cambia según cada sensor. A la hora de comprobarlos en tiempo real es visible que, aunque el offset se haya visto resentido, la

diferencia entre el voltaje en reposo y cuando se presiona la zapatilla no cambia, situándose en torno a los 0.2 V (de nuevo, cambia según el sensor).

Por tanto, aunque baje esta pérdida del nivel de continua suponga una pérdida de eficiencia para todos los sensores, no hay mayor problema para decidir después si un sensor determinado está activo o no.

Esto ocurre para tres de los sensores, pero para el trasero derecho esa pérdida supone descender a un voltaje negativo cercano a cero, y que baja a un nivel aún menor cuando se presiona. Asimismo, hay que aclarar que la diferencia de 0.2 V al presionar la zapatilla todavía se mantiene intacta.

Se procedió a un extenso estudio de cuál era el motivo por el que los sensores perdían en offset. Se consideró una pérdida por atenuación de cable o interferencias, pues la señal recorría una distancia notablemente mayor que en la placa de pruebas. Esto no tenía sentido, sin embargo, pues la diferencia de voltaje al presionar también sería menguada. Finalmente, la propuesta más probable fue los pequeños cambios teóricos y prácticos entre las resistencias que conforman los puentes de Wheatstone. Esto explica por qué no todos los offset caen de igual manera, pero sí de forma cercana. Con todo, este problema se ha decidido resolver de otra forma por meros motivos académicos, por lo que se procedió a implementar una placa auxiliar que sumara un offset al sensor trasero derecho, introduciéndolo de nuevo en el rango aceptado por Arduino. Adicionalmente, este método podría ser también útil para compensar el envejecimiento de los componentes a lo largo del tiempo

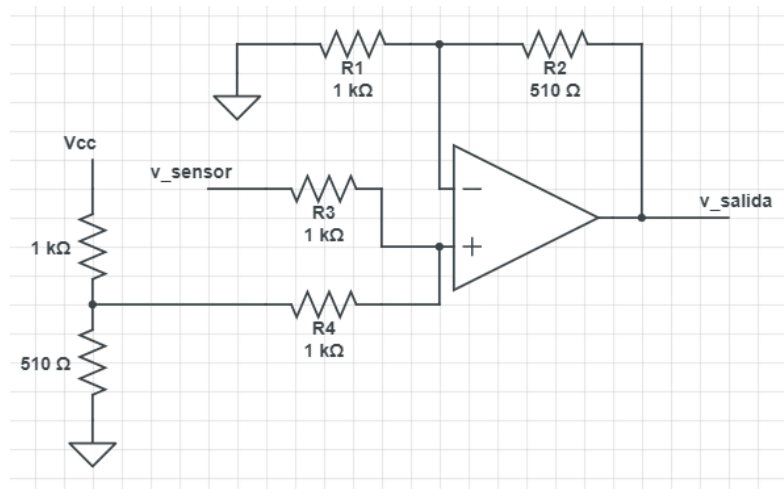


Figura 4.2. Esquema del circuito sumador en configuración no inversora

Las propiedades y características de esta placa son ampliamente expuestas en el apartado de Hardware, junto con fotografías del montaje real.

Así, los límites establecidos para cada sensor se representan en la Tabla 4.1., si bien es posible que deban ser reajustados ligeramente si se produce algún cambio en el sistema:

| Sensor | Límite (V) |
|---------------------|------------|
| Derecho Delantero | 0.50 |
| Derecho Trasero | 1.15 |
| Izquierdo Delantero | 1.21 |
| Izquierdo Trasero | 0.26 |

Tabla 4.1. Valores de los límites de los voltajes, medidos en voltios

Comprobación del funcionamiento de los sensores

A la hora de comprobar si los sensores se están comportando de la manera esperada, es necesario considerar que las señales finales de las zapatillas están bien acondicionadas.

Para conocer si los sensores devuelven los resultados correctos, se debe recurrir al uso de un ordenador donde conectar Arduino vía USB. Acto seguido, se recurrirá usarán las zapatillas para cambiar el nivel de la señal recibida. Esta señal será representada en la pantalla del ordenador mediante un comando especial llamado `serial.println()`, el cual muestra los valores introducidos entre los paréntesis del mismo.

Es importante aclarar que no se indicarán los límites de voltaje de cada sensor, pues son constantes durante toda la ejecución del programa y harían confusa la lectura de los datos en pantalla. Por eso, se ha utilizado un papel de forma auxiliar para revisar de forma continua estos límites en cada iteración del programa, sin depender de la pantalla de comandos de Arduino.

Estos datos deben ser familiares para el experimentador. Así, si un sensor devuelve un valor inusual se podrá averiguar con ligereza el motivo de la anomalía. Si, por ejemplo, los sensores devuelven un valor nulo, significa que el sistema no tiene la alimentación conectada. En cambio, si devuelve un valor por encima de los 5 V permitidos, es muy probable que algún cable relativo a las galgas extensiométricas no haga un buen contacto con las placas.

Como resultado final, en adición, se incluye si los sensores se encuentran activos para esa iteración. Solo se muestra si los sensores delanteros o traseros están siendo presionados en su conjunto, no de forma individual. De este modo, con tener presente el diagrama de estados, se puede comprobar si el sistema funciona correctamente o debe ser corregido.

En caso de cambiar de estado, la función encargada de rotar los motores imprimirá por pantalla qué fase de movimiento está ejecutando. Con todos estos datos ya es posible comprobar el sistema sin siquiera haber conectado los motores del robot.

Esta porción del proyecto ha sido comprobado y registrado en el vídeo titulado *Comprobación del funcionamiento de los sensores*, anexo a este documento. En él se muestran diferentes cambios de estados con el propósito de llegar del estado levantado al estado sentado y retroceder después al primero. También se ha demostrado que funcionan otras combinaciones intermedias.

No debe olvidarse que las iteraciones se repiten cada 2.5 segundos. El sistema no lee ningún dato entre este periodo, pues afectaría a la fiabilidad del sistema, como ya ha sido explicado en puntos anteriores.

4.2. Comprobación del software

La programación Arduino supone el cerebro de todo el sistema. El software es el que se encarga de procesar los datos de las zapatillas y actuar sobre el robot en consecuencia. Con esto en mente, no resulta incoherente plantearse la división propia del software para la comprobación de ambas funciones: la recogida de datos, y la transmisión de los mismos.

Sin embargo, dado que ya se hizo un análisis exhaustivo de cada línea de código implementada, no tiene sentido reiterar las conclusiones extraídas del apartado de Software.

Simplemente, se puede decir que el programa se divide en estas dos partes de una manera muy sencilla. Toda la integridad del programa se basa en la lectura de datos y su procesamiento. Solamente las cuatro funciones auxiliares (`sentarse1()`,

`sentarse2()`, `levantarse1()` y `levantarse2()`) se ocupan de la transmisión de la información a los servomotores. Por tanto, basta con dejar comentadas esas líneas de código e incluir un mensaje que indique cuándo se invocan esas funciones.

4.3. Implementación del hardware

Instalación de la alimentación

Una disyuntiva especialmente importante que plantea el proyecto es el modo de alimentar todos los circuitos. Inicialmente, y yendo a términos ideales, se prefiere que el prototipo se pueda mover y sea independiente de alimentaciones externas. Es por esto por lo que el primer planteamiento diseñado cuenta con varias pilas de alimentación.

En total, el circuito necesita de 3 voltajes distintos de alimentación: -9, 0 y 9 V. Esto se puede conseguir utilizando dos pilas de 9 V, con sus bornas apropiadamente unidas. La forma de hacer esto es unir mediante un cable la borna positiva de una con la borna negativa de la otra, quedando tres salidas en total con los voltajes deseados.

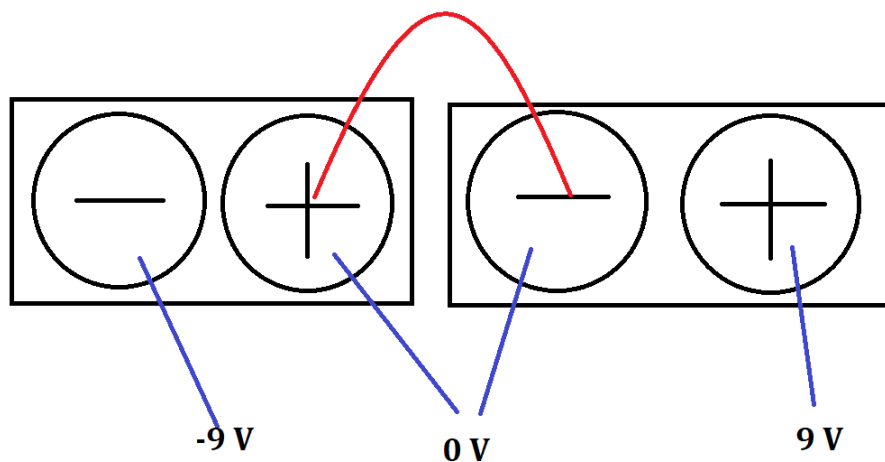


Figura 4.3. Esquema de unión de las pilas para obtener los tres voltajes necesarios (en rojo el cable, en azul los indicadores de voltaje)

En el circuito real se aprovecha la placa de alimentación común para realizar las conexiones. Para conectar las pilas se utiliza un contacto especial mostrado en la Figura 4.4.

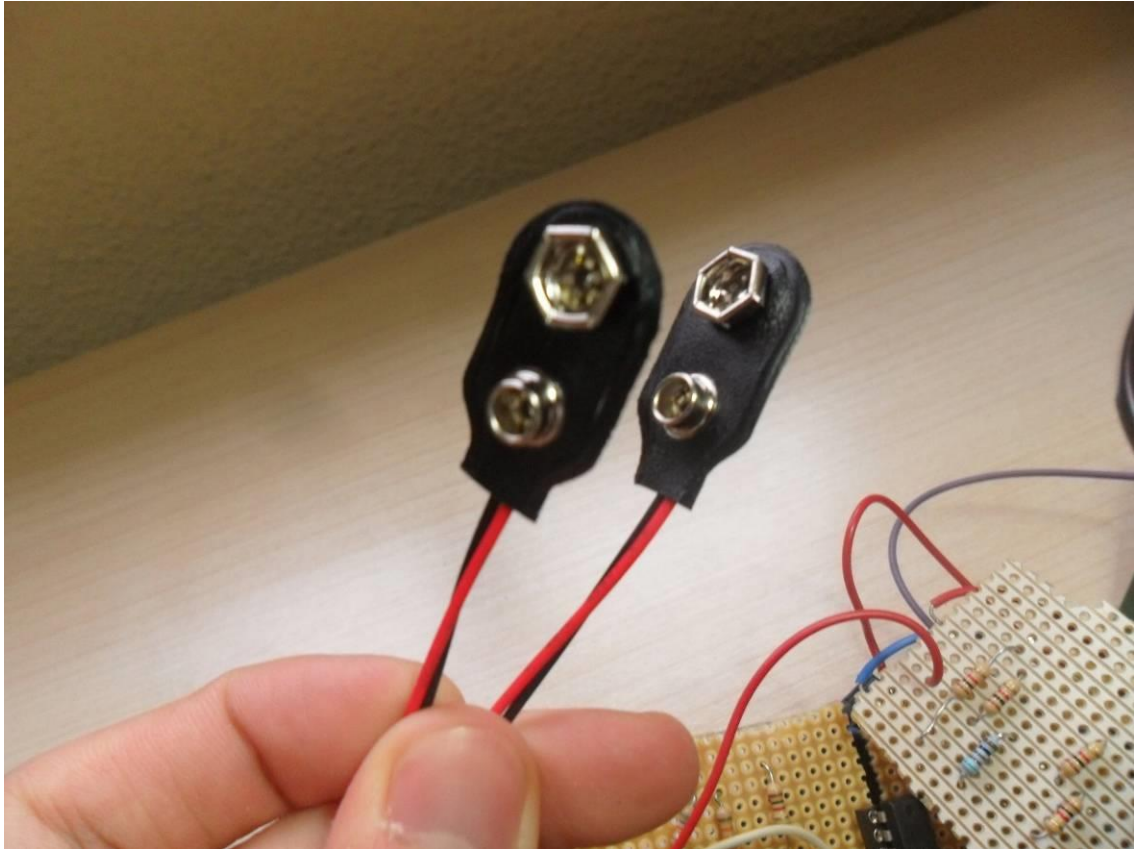


Figura 4.4. Fotografía de los contactos para unir las pilas de 9 V al circuito

De esta manera, las placas de acondicionamiento de los sensores quedarían bien alimentados. Sin embargo, hay que tener especial cuidado de que ambas pilas estén bien cargadas, pues una gran diferencia de voltaje suministrado por ambas podría provocar una mala lectura de datos de los sensores.

Una vez se tiene acondicionados los sensores, se necesita otra alimentación única para el microcontrolador y los servomotores. El voltaje mínimo es de 9 V, que es el que le suministrará. Arduino, no obstante, solo necesita 5 V para trabajar. El voltaje restante no solo se desperdiciará, sino que terminará por calentar la placa.

Así pues, se implementa el circuito con 3 pilas distintas para alimentarlo. Esto produce dos inconvenientes importantes que no se tenían con una alimentación fija.

Alimentación externa y alimentación interna

La primera complicación surge de la poca estabilidad de las pilas. A pesar de ser recargables, la duración de estas no es demasiado larga y se van desgastando rápidamente a medida que transcurre el tiempo. La pila que alimenta Arduino, por ejemplo, dura alrededor de 15 minutos una vez ha sido cargada. Esto dificulta en gran medida el proceso de pruebas y pone en duda la autonomía que se pretendía conseguir con el uso de baterías portátiles.

Asimismo, las pilas que suministran corriente al acondicionamiento de los sensores no se desgastan a la misma velocidad. Como tanto estas como la pila que alimenta el microcontrolador se van gastando, el nivel de voltaje que recibe Arduino va descendiendo progresivamente. Esto supone un gran desafío lógico, dado que en este caso el uso de límites de decisión estáticos no sirve.

En las siguientes tablas se muestran los valores obtenidos en función varios parámetros: el tipo de alimentación utilizada, y si se han presionado o no los sensores con las zapatillas puestas o con las manos.

| Sensor | Sin presionar | Presionando (manos) | Presionando (pies) |
|---------------------|----------------------|----------------------------|---------------------------|
| Derecho Delantero | 0.69 | 0.49 | 0.42 |
| Derecho Trasero | 1.46 | 1.39 | 1.18 |
| Izquierdo Delantero | 1.36 | 1.13 | 1.21 |
| Izquierdo Trasero | 0.72 | 0.54 | 0.34 |

Tabla 4.2. Comparación de valores obtenidos de los sensores utilizando alimentación externa, medida en voltios, y diferenciando el modo de presionar las zapatillas

De la Tabla 4.2 podemos extraer una conclusión evidente, y es que se ejerce más fuerza presionando cuando se tienen las zapatillas puestas, que es el modo diseñado para utilizarlas. Por tanto, no tiene sentido seguir midiendo los voltajes cuando se presionan con las manos.

A continuación, se compararán estos valores con los obtenidos cuando se utilizan pilas dentro de la caja, en diferentes estados.

| Sensor | Sin presionar | Presionando (recién cargadas) | Presionando (tras 3h, sin usar) |
|---------------------|---------------|-------------------------------|---------------------------------|
| Derecho Delantero | 0.42 | 0.31 | 0.29 |
| Derecho Trasero | 0.74 | 0.40 | 0.36 |
| Izquierdo Delantero | 0.63 | 0.59 | 0.48 |
| Izquierdo Trasero | 0.26 | 0.06 | 0.03 |

Tabla 4.3. Comparación de valores obtenidos de los sensores utilizando pilas, en voltios, con las pilas recién cargadas sin presionar y presionando, y los valores cuando se presionan tras haber transcurrido 3 horas de su carga, pero sin haberlas utilizado

En la Tabla 4.3. es fácilmente apreciable que todos los valores, incluidos los obtenidos sin presionar, ahora son más pequeños. Hay que tener especial cuidado con la carga, pues Arduino marcará un valor nulo si llega un voltaje negativo.

Experimentalmente, se concluye que la carga de las pilas efectivamente afecta a los voltajes medidos, de modo que los límites establecidos no pueden previamente fijados. Además, se ha concluido que la pila que más influye en la caída de voltaje es la que alimenta el microcontrolador, de modo que no recibirá ningún dato útil si la pila se ha desgastado lo suficiente.

Si con límites estáticos no se puede decidir cuándo un sensor está o no activo, se debe recurrir a límites dinámicos que vayan variando con el tiempo o teniendo memoria de las medidas obtenidas con anterioridad. Para esto se proponen dos modelos.

El modelo más simple se basa en la suposición de que los valores de los voltajes recibidos decrecerán de una manera más o menos constante. Por tanto, basta con ir actualizando el límite de decisión en función los valores actuales. Si se toman como referencia límite unos valores leídos cuando no se presionan las zapatillas, basta con deducir que los sensores estarán activos cuando bajen un cierto voltaje fijo de ese límite de referencia, digamos 0.05 V. Este sistema, sin embargo, debe ser actualizado de forma periódica, por lo que es necesario incluir temporizadores en el programa que se disparen en un momento en que el propio usuario no esté presionando las zapatillas. Esto es, debe saber cuándo se actualizan esos límites.

Otro modelo más fiable, pero más complejo, se basa en el uso de la memoria de datos. Si se guarda en una matriz, digamos 4x10, los diez últimos valores de cada uno de los cuatro sensores (y desechando los que sean especialmente distintos), podemos calcular una media que serviría como referencia límite y que se iría actualizando en cada lectura de datos.

La segunda complicación es más específica y tiene una solución poco eficiente. El microcontrolador debe ocuparse también de alimentar a los cuatro servomotores, y estos deben mover una carga asociada al tamaño del robot. El alto consumo que esto conlleva requiere que los servos deban ser alimentados de forma independiente, añadiendo una cuarta alimentación que esté comprendida entre 4 y 6 V (que es lo que aceptan los servomotores estándar).

Estas dos complicaciones, sumadas a que este prototipo está más orientado al uso del sistema en un entorno controlado y con una alimentación estable, decidieron que lo más rentable en términos de eficiencia era utilizar una fuente de alimentación fija que proporcionara por cable -9, 0, 9 y 5 V.

Esta fuente de alimentación conectará de la manera que se aprecia en la Figura 4.5., utilizando todas las salidas de las que dispone el dispositivo. Deberán hacerse dos puentes para unir el voltaje de referencia, o tierra, de modo que queden cuatro conectores salientes, que se engancharán a los propios conectores que tiene la caja.

La fuente deberá estar encendida y con los dos botones centrales sin pulsar. Los voltajes de maestro y esclavo deberán marcar los dos 9.0 V, mientras que el amperaje máximo deberá ser lo suficientemente alto como para que no cancele el paso de corriente.

En la fotografía, de izquierda a derecha, se conectarán los cables de -9 V (amarillo), 0 V (negro), +9 V (rojo) y +5 V (blanco).



Figura 4.5. Fotografía de la distribución de cables en la fuente de alimentación (de izquierda a derecha: -9, 0, 9 y 5 V)

4.4. Comprobación del sistema completo

El sistema en su totalidad se compone de unas zapatillas provistas de sensores de presión, una caja que incluye toda la electrónica necesaria, y un robot bípedo que simula un exoesqueleto real.

La comprobación de su funcionamiento se ha realizado con la recreación de un ciclo completo de sentarse y levantarse, es decir, pasar por el estado intermedio dos veces, primero hacia abajo y después hacia arriba.

El objetivo principal es realizar los movimientos siguiendo al robot a la vez que se presionan los sensores con las zapatillas puestas. Se considerará que el sistema funciona si es capaz de realizar las cuatro etapas satisfactoriamente.

Se ha incluido un vídeo demostrativo llamado *Comprobación del funcionamiento del sistema completo*, en el cual se visiona tanto al usuario accionar los sensores como al robot moverse junto a él. No se ha visto necesario incluir un plano directo hacia las

zapatillas, pues estas se quedan quietas durante todo el proceso y no son visibles los cambios los sensores desde fuera.

Si bien ya se ha realizado una explicación en el mismo vídeo, se procede a continuación a exponer un análisis más exhaustivo del proceso de comprobación, paso a paso. Es imprescindible conocer el diagrama de estados, ya explicado previamente. Se incluye a continuación una imagen recordatoria de su esquema. También se recomienda seguir el vídeo junto con la explicación que viene ahora y conocer todo el sistema en su totalidad, pues se incluirán conceptos relacionados con el código.

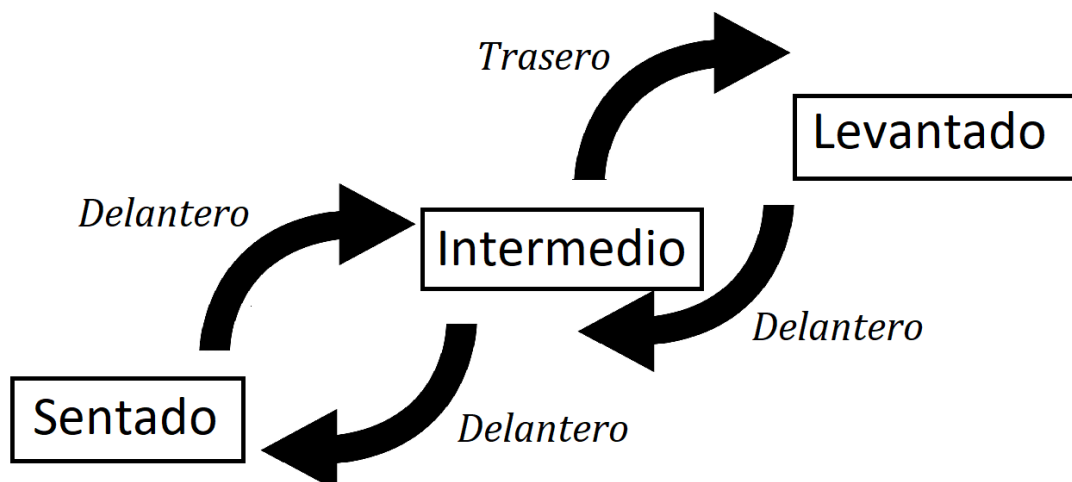


Figura 4.6. Recordatorio del diagrama de estados del sistema (deben presionarse los dos sensores, delanteros o traseros, para cada cambio de estado)

- El sistema no comienza a funcionar hasta que no se activa el interruptor alojado en la caja. Deben dejarse pasar 2.5 segundos antes de comenzar el funcionamiento, y dejar que el robot quede totalmente erguido en el estado Levantado.
- No se debe olvidar que siempre se empieza en el estado Levantado, si bien podría configurarse para comenzar en cualquier otro estado. Del estado levantado solo podemos avanzar al estado Intermedio mediante la presión de los sensores delanteros. Al accionarlos, el voltaje leído en el microcontrolador será menor que el límite y se llamará a la función `sentarse1()`, que moverá los servomotores del robot hasta llegar a un estado inclinado a caballo entre los dos estados extremos. La cadera se flexionará 40° y las rodillas 60°. La explicación de por qué giran esos ángulos viene dada al principio de la sección de Software, en las Fases de la bipedestación.

- La siguiente lectura no se produce hasta que el robot se haya estabilizado y después hayan transcurrido 2.5 segundos. Es posible que los sensores no capten la siguiente transición de estado en la primera lectura, por lo que se deberán esperar otros 2.5 segundos hasta la siguiente. Esta premisa estará presente en todo momento en que se quiera cambiar de estado.
- Se pretende ahora pasar del estado Intermedio al estado Sentado. Si presionamos los sensores traseros volveremos al estado inicial, y si presionamos tanto delanteros como traseros no sucederá nada. Para sentarnos completamente será necesario presionar los sensores delanteros de nuevo. El presionar los sensores delanteros o traseros para cada estado viene dado por la carga natural que hacemos las personas al levantarnos o sentarnos. Es por esto por lo que es complicado mantenerse en el estado Intermedio sin accionar los siguientes sensores. Al presionar los sensores delanteros, el robot se sentará completamente al llamar a la función `sentarse2()`, que rotará las rodillas 20° y las caderas 70° . Nótese que en total las rodillas han girado 80° (porque la superficie sobre la que se sienta no es completamente horizontal) y las caderas 100° , de modo que se ajusta a un modelo más realista que si giraran 90° cada uno.
- Ya se ha conseguido la mitad del proceso. Por defecto, las personas cargan el mayor peso sobre las punteras cuando se quieren levantar. Por tanto, para iniciar `levantarse1()` y volver al estado Intermedio, se tendrán que presionar de nuevo los sensores delanteros. No debe olvidarse que deben presionarse los dos sensores delanteros. El sistema no cambiará si solo se presiona uno de ellos. A continuación, se realiza el proceso inverso: las caderas rotan 70° y las rodillas 20° , ahora en sentido contrario al que lo hicieron en `sentarse2()`.
- Pasando por el estado Intermedio, es necesario cambiar la carga de peso a los sensores traseros para terminar de levantarse. El robot volverá a erguirse cuando se presionen los sensores traseros, llamando a la función `levantarse2()`. Esta función hará rotar los servomotores 60° las rodillas y 40° las caderas, terminando el proceso.
- Transcurridos los 2.5 segundos marcados después de que el robot se haya parado, se podría continuar el proceso de maneras distintas, como bajar al estado Intermedio y volver al de Levantado, o sentarnos completamente otra vez.

Adicionalmente, se ha incluido otro vídeo llamado *Comprobación del funcionamiento del robot*, en el cual se presenta un primer plano del robot siguiendo el mismo proceso explicado, pero sin ninguna explicación añadida ni la visión del usuario. En este vídeo también se ha accionado el robot mediante las zapatillas fuera de cámara, pero sin que el usuario cambie de posición a la vez que lo hacía el robot.

5. CONCLUSIONES

Este proyecto presenta una aproximación a lo que podría ser un exoesqueleto de pierna totalmente funcional con unos componentes muy escasos en número. Con tan solo cuatro sensores instalados en el calzado y unos motores se ha demostrado la posibilidad de construir un sistema capaz de traducir la intención del usuario a un movimiento real.

Del trabajo aquí presentado, sin embargo, se extraen conclusiones e ideas totalmente variadas. Además del desarrollo y la investigación relativa al estado del arte de los exoesqueletos, así como su método de funcionamiento, el ejercicio práctico aquí realizado no sirve solo para consolidar los conocimientos presentados teóricamente, sino para demostrar que con pocos recursos es posible construir un modelo a pequeña escala de lo que sería un exoesqueleto real.

A la hora de definir el propio sistema de funcionamiento del dispositivo, se ha hecho una comparación exhaustiva de todos los exoesqueletos existentes en la actualidad, tanto aquellos desarrollados para uso habitual como para aquellos con fines de rehabilitación de su usuario. Se han probado algunos métodos y lógicas para diseñar el sistema, y finalmente se ha llegado a la conclusión de que el uso de sensores de presión en los pies conforma una base casi imprescindible en todo exoesqueleto.

Por lo tanto, se deduce que el uso de sensores de presión es lo más adecuado desde un punto de vista lógico, llegando a la misma conclusión que tantas empresas desarrolladoras de este tipo de productos. De modo similar ocurre con la posición de los motores. Es interesante apreciar que el lugar de instalación de los motores que mueven el exoesqueleto es exactamente el mismo en todos los exoesqueletos.

A primera vista, estas ideas parecen demasiado obvias como para que merezcan la pena mencionarlas siquiera en este apartado. La cuestión, no obstante, radica en que las configuraciones posibles para diseñar un exoesqueleto de pierna no son tan numerosas como a priori pueda parecer. Todos los dispositivos cuentan con una cantidad enorme de parámetros idénticos que si se modificaran supondría una pérdida de efectividad del exoesqueleto, ya sea en términos de potencia, de eficiencia o de grados de libertad.

Es por esto por lo que el sistema construido en este proyecto presenta la forma más primitiva de diseño de un exoesqueleto de pierna, con pocas vías de modificación eficientes. Las versiones más mejoradas de este sistema vendrían de ampliaciones y mejoras construidas sobre esta base, como la implementación de giroscopios o la inclusión de más motores para incrementar los grados de libertad.

El sistema construido consta de unas zapatillas plenamente utilizables por el paciente, pero adolece de un exoesqueleto real que ayude al usuario a levantarse o a sentarse. A efectos de construcción de un prototipo, se ha optado por la utilización de una pequeña maqueta de un robot que simula el movimiento de las piernas del hipotético usuario.

Entonces, ¿cuál es el motivo por el que no se ha incluido un exoesqueleto a tamaño real? Esto se ha mencionado varias veces a lo largo del documento, pero es una aclaración tan importante que debe ser repetida en este apartado. En realidad, el único impedimento que surge para la no construcción de un exoesqueleto a tamaño real viene dado por razones económicas.

Implementar un exoesqueleto del tamaño de una persona resulta demasiado caro para un proyecto de estas dimensiones. La estructura debe cumplir unas condiciones de tamaño y de peso más estrictas, y los componentes utilizados aumentan su precio cuanto más grandes son. Asimismo, existen razones ergonómicas por las que no se ha trabajado con un dispositivo de tamaño real.

Es esencial considerar que, si bien el exoesqueleto es una maqueta, el resto del sistema podría reutilizarse para cualquier tamaño de una estructura. El software, incluso, podría funcionar correctamente con cambios mínimos en su código fuente.

Se ha explicado, sin embargo, que el sistema desarrollado no es válido para todas las personas. Sin contar otros parámetros como una gran desviación de la altura o el peso del usuario con respecto a la media, existe un requisito especial que toda persona debe cumplir para poder utilizar el aparato: debe poder transmitir la fuerza de los pies a los sensores.

Esto limita el rango de usuarios que podrían aprovecharse de un sistema como este. Es necesario que, aunque no sean capaces de levantarse por sí solos, sí tengan una pequeña cantidad de fuerza en el tren inferior. Por consiguiente, el sector de población objetivo de uso de este modelo de exoesqueleto serían preferentemente personas mayores y aquellos que padezcan de algún trastorno que les impida moverse con soltura, como pacientes de esclerosis múltiple o distrofia muscular.

Estas conclusiones presentadas no sirven sino como un análisis de como este dispositivo sería de utilidad en un caso real. En la actualidad, la silla de ruedas se ha impuesto como medio de transporte habitual para aquellas personas que no pueden caminar por su propio pie, pues supone un medio relativamente seguro, barato y manejable para el usuario. Las sillas de ruedas, en cambio, presentan la desventaja de no rendir bien a la hora de cambiar de un asiento a otro, o para realizar transferencias de la silla a la cama, por ejemplo. En estos casos, un sistema ligero como el presentado en este documento sería de especial utilidad. No se trata de sustituir a la silla de ruedas, sino de utilizar un exoesqueleto para ayudar en las tareas en las que la otra no puede.

Existen en el mercado sillas de ruedas con capacidad de enderezar a su usuario, de modo que esté completamente erguido, pero unido a la silla. Esto resulta muy incómodo si se quieren realizar transferencias, y definitivamente no es aprovechable para personas que pueden caminar —aunque quizás con dificultades— pero no son capaces de levantarse o de sentarse de forma autónoma.

Por tanto, el dispositivo construido está ideado para sentarse y levantarse en un entorno controlado, como puede ser el propio hogar del usuario. Esto suprime el gran problema actual de los exoesqueletos, pues no es necesario alimentar el sistema con una batería externa, sino que podría funcionar con la alimentación fija de la casa. Con las convenientes ampliaciones para la asistencia en la marcha, este exoesqueleto podría servir a personas discapacitadas a moverse por recorridos prefijados a lo largo de toda su casa, así como sentarse y levantarse donde quieran, y cuando quieran.

6. LÍNEAS FUTURAS

6.1. Futuro de los exoesqueletos

El mundo de los exoesqueletos se ha desarrollado de una manera ciertamente creciente en los últimos años. Al ser una disciplina que ha nacido y que aún sigue alimentándose del campo de la robótica, su propia evolución está directamente ligada a los avances que vayan sucediéndose en el ámbito del hardware y software de los que se aprovechan los robots. Si bien las utilidades de uno y otro son distintas, y las dos ramas cada vez se van especializando, la robótica general y los exoesqueletos seguirán desarrollándose a la par, bebiendo uno del otro de los sistemas que vayan mejorando.

En un pasado no muy lejano, los artículos sobre exoesqueletos se reducían a unos pocos documentos teóricos acerca de sistemas que estaban por implementarse. Hoy en día, estos artículos siguen creciendo en número de forma exponencial, presentando tanto documentos de tecnologías punteras como de estudios de otros campos aplicables al campo de la robótica y de los exoesqueletos. Esta idea puede apreciarse en la figura 7.1. (Young, J. y Ferris, D., 2015), la cual muestra, sobre el total de 2014, la cantidad de artículos escritos sobre exoesqueletos desde 1997 hasta 2014, usando las palabras clave *robotic exoskeleton* para realizar la búsqueda. En las tres bases de datos presentadas se obtiene que la cantidad de publicaciones se dobló entre 2004 a 2009, y se dobló de nuevo entre 2009 y 2013. En adelante, este número seguirá creciendo cada vez más rápido.

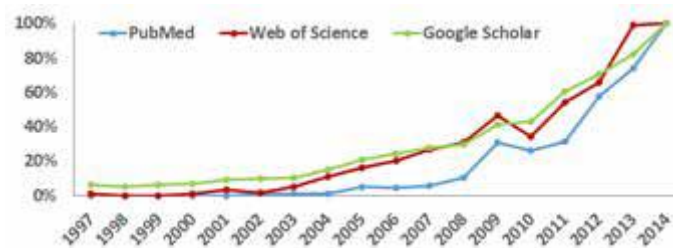


Figura 7.1. Crecimiento de publicaciones sobre exoesqueletos desde 1997 hasta 2014.

Actualmente, el desarrollo más avanzado sobre exoesqueletos lo llevan principalmente grandes empresas con tecnología puntera. Además, el Departamento de Defensa de los Estados Unidos (DoD), así como la NASA, trabajan en el desarrollo de sus propios dispositivos con los sistemas más novedosos.

Un ejemplo de exoesqueleto en desarrollo es el TALOS. Está diseñado con un propósito íntegramente militar. Se trata de un traje a prueba de balas, con armamento incorporado y con la capacidad de monitorizar las constantes vitales del soldado. También permite incrementar la fuerza del usuario por medio de motores incorporados, así como sensores para medir otro tipo de demandas militares. Sin embargo, este traje está pensado solo para unos pocos soldados de élite, pues el precio es elevado y, como la gran mayoría de exoesqueletos, demanda una gran cantidad de energía y, por consiguiente, cuenta con poca autonomía.

El TALOS es un ejemplo de exoesqueleto que se encuentra en vías de desarrollo, si bien la gran cantidad de proyectos son continuación de los ya existentes, tales como el HAL, de Cyberdyne, o el PowerLoader, de Panasonic. Hay numerosos diseños y prototipos de exoesqueletos, pero debe tenerse en cuenta que el sistema básico ya está inventado. El futuro de los exoesqueletos es, pues, la mejora y la optimización de los sistemas ya existentes.

El director del Laboratorio de Robótica de Bristol, Chris Melhuish, realizó unas interesantes declaraciones a la BBC acerca de este tema. Declaró que los tres temas principales que deben desarrollarse son los materiales, los sistemas de control y, más que nada, los sistemas de energía. La autonomía es el factor clave que condiciona a todo exoesqueleto inalámbrico.

“La evolución del exoesqueleto irá de la mano con la evolución de las baterías u otros sistemas de almacenamiento de alta densidad, además del desarrollo de materiales estructurales livianos.”

No obstante, este se trata de un campo en constante cambio y debe ser revisado continuamente. Dado que un papel muy importante de los exoesqueletos es ofrecer apoyo a personas mayores o lesionadas, la seguridad se convierte en un factor a tener en cuenta y debe revisarse en cada momento, tanto desde la perspectiva más tecnológica como en el ámbito legal.

6.2. Mejoras del sistema actual

Uno de los temas más interesantes a la hora de concluir un trabajo es definir cuáles serán, o podrían ser, las principales líneas que seguirían avanzando en el diseño del proyecto.

Se ha presentado un sistema que simula un exoesqueleto que recoge la información de la intención de un usuario a la hora de levantarse o de sentarse. Ha de recordarse que el objeto principal de este proyecto es tanto la implementación del sistema mencionado, así como un estudio general de la sensorización en exoesqueletos. Por tanto, el paso más lógico a la hora de continuar este trabajo lleva a ampliarse por la rama de la deambulacion.

Un sistema capaz de simular la marcha ofrece muchas posibilidades de estudio de los exoesqueletos. Tanto las ventajas como los inconvenientes de ampliar esta rama ya fueron explicados con detalle en la sección de Sensores en la deambulacion. De forma resumida: utiliza casi todos los procedimientos empleados para la bipedestacion, pero podría presentar problemas mecánicos al instalar los sensores en los pies.

Sin embargo, la implementación de la deambulacion no es la única vía de mejora del sistema actual. Una de las cuestiones más importantes a la hora de diseñar un exoesqueleto es la seguridad. Personas ancianas y/o lesionadas, con dificultades para mantener el equilibrio, pueden caerse a la hora de utilizar el exoesqueleto. Aunque este problema está especialmente marcado en la deambulacion, a la hora de levantarse también pueden surgir inconvenientes.

Es esto por lo que resulta especialmente importante el mantenimiento del equilibrio como aspecto fundamental de la seguridad del dispositivo. El elemento más utilizado para medir los cambios de inclinación y, por tanto, del propio equilibrio de la persona, son los giroscopios. No se pretende realizar un estudio detallado de los giroscopios, pues no es el tema que tratar en este documento, sino dar pistas sobre cuál podrían ser los siguientes pasos para ampliar el sistema.

El giroscopio es un aparato mecánico que sirve para medir, mantener o cambiar la orientación de algún dispositivo asociado a él. Los lugares habituales donde se suelen colocar en un exoesqueleto son a la altura de la rodilla y en las caderas, es decir, en los lugares donde inicialmente se esperan mayores cambios de inclinación.

De la respuesta de estos giroscopios deberían responder unos servomotores adicionales que otorgasen a la máquina dos grados de libertad más, que permitieran al exoesqueleto inclinarse a izquierda y derecha para conseguir estabilizarse.

Estas son las ideas presentadas que de forma lógica debería continuarse este trabajo. Sin embargo, las líneas futuras son innumerables y dependen de factores tales como el presupuesto, la tecnología disponible y el tiempo empleado para su mejora.

7. ANEXOS

Microcontrolador Arduino Mega 2560

8-bit Atmel Microcontroller with 16/32/64KB In-System Programmable Flash

Features

- High Performance, Low Power Atmel® AVR® 8-Bit Microcontroller
- Advanced RISC Architecture
 - 135 Powerful Instructions – Most Single Clock Cycle Execution
 - 32 × 8 General Purpose Working Registers
 - Fully Static Operation
 - Up to 16 MIPS Throughput at 16MHz
 - On-Chip 2-cycle Multiplier
- High Endurance Non-volatile Memory Segments
 - 64K/128K/256KBytes of In-System Self-Programmable Flash
 - 4Kbytes EEPROM
 - 8Kbytes Internal SRAM
 - Write/Erase Cycles: 10,000 Flash/100,000 EEPROM
 - Data retention: 20 years at 85° C/ 100 years at 25° C
 - Optional Boot Code Section with Independent Lock Bits
- In-System Programming by On-chip Boot Program
- True Read-While-Write Operation
- Programming Lock for Software Security
- Endurance: Up to 64Kbytes Optional External Memory Space • Atmel® QTouch® library support
 - Capacitive touch buttons, sliders and wheels
 - QTouch and QMatrix acquisition
 - Up to 64 sense channels
- JTAG (IEEE® std. 1149.1 compliant) Interface
 - Boundary-scan Capabilities According to the JTAG Standard
 - Extensive On-chip Debug Support
 - Programming of Flash, EEPROM, Fuses, and Lock Bits through the JTAG Interface
- Peripheral Features
 - Two 8-bit Timer/Counters with Separate Prescaler and Compare Mode
 - Four 16-bit Timer/Counter with Separate Prescaler, Compare- and Capture Mode
 - Real Time Counter with Separate Oscillator
 - Four 8-bit PWM Channels
 - Six/Twelve PWM Channels with Programmable Resolution from 2 to 16 Bits (ATmega1281/2561, ATmega640/1280/2560)
 - Output Compare Modulator
 - 8/16-channel, 10-bit ADC (ATmega1281/2561, ATmega640/1280/2560)
 - Two/Four Programmable Serial USART (ATmega1281/2561, ATmega640/1280/2560)
 - Master/Slave SPI Serial Interface
 - Byte Oriented 2-wire Serial Interface
 - Programmable Watchdog Timer with Separate On-chip Oscillator
 - On-chip Analog Comparator
 - Interrupt and Wake-up on Pin Change
- Special Microcontroller Features
 - Power-on Reset and Programmable Brown-out Detection
 - Internal Calibrated Oscillator
 - External and Internal Interrupt Sources
 - Six Sleep Modes: Idle, ADC Noise Reduction, Power-save, Power-down, Standby, and Extended Standby
- I/O and Packages
 - 54/86 Programmable I/O Lines (ATmega1281/2561, ATmega640/1280/2560)
 - 64-pad QFN/MLF, 64-lead TQFP (ATmega1281/2561)
 - 100-lead TQFP, 100-ball CBGA (ATmega640/1280/2560)
 - RoHS/Fully Green
- Temperature Range:
 - -40° C to 85° C Industrial
- Ultra-Low Power Consumption
 - Active Mode: 1MHz, 1.8V: 500µA
 - Power-down Mode: 0.1µA at 1.8V
- Speed Grade:
 - ATmega640V/ATmega1280V/ATmega1281V:
 - 0 - 4MHz @ 1.8V - 5.5V, 0 - 8MHz @ 2.7V - 5.5V
 - ATmega2560V/ATmega2561V:
 - 0 - 2MHz @ 1.8V - 5.5V, 0 - 8MHz @ 2.7V - 5.5V
 - ATmega640/ATmega1280/ATmega1281:
 - 0 - 8MHz @ 2.7V - 5.5V, 0 - 16MHz @ 4.5V - 5.5V
 - ATmega2560/ATmega2561:
 - 0 - 16MHz @ 4.5V - 5.5V

Parallax Standard Servo

Parallax Standard Servo (#900-00005)

The Parallax Standard Servo provides 180° range of motion and position control to your project. Great for animatronics and robotics applications.

Features

- Holds any position between 0 and 180 degrees
- 38 oz-in torque at 6 VDC
- Accepts four mounting screws
- Easy to interface with any Parallax microcontroller or PWM-capable device
- Simple to control with the PULSOUT command in PBASIC
- High-precision gear made of POM (polyacetal) resin makes for smooth operation with no backlash
- Weighs only 1.55 oz (44 g)



Key Specifications

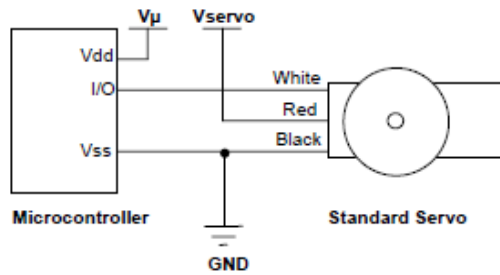
- Power requirements: 4 to 6 VDC*; Maximum current draw is 140 +/- 50 mA at 6 VDC when operating in no load conditions, 15 mA when in static state
- Communication: Pulse-width modulation, 0.75–2.25 ms high pulse, 20 ms intervals
- Dimensions approx 2.2 x 0.8 x 1.6 in (5.58 x 1.9 x 40.6 cm) excluding servo horn
- Operating temperature range: 14 to 122 °F (-10 to +50 °C)

***Power Requirement Notes**

The manufacturer specifies 4-6 VDC for this servo. However, we find that this servo is tolerant of a 9 V battery for very brief periods of time when there is no load, as used in some activities in the Stamps in Class series of tutorials. (Slight jittering may be observed when batteries are fresh; this does not cause damage). Do not use this servo with an unregulated wall-mount supply, or a regulated wall mount supply exceeding 6 VDC.

Servo current draw can spike while under load. Be sure that your application's power supply and voltage regulator is prepared to supply adequate current for all servos used. Do not try to power this servo directly from a BASIC Stamp module's or any microcontroller's Vdd or Vin pins; do not connect the servo's Vss line directly to the BASIC Stamp module's or any microcontroller's Vss pin.

Quick-Start Circuit



V_{μ} = microcontroller voltage supply

V_{servo} = 4 to 6 VDC, regulated or battery

I/O = PWM TTL or CMOS output signal from microcontroller: 3.3 to 5 V, not to exceed $V_{servo} + 0.2$ V

Specifications

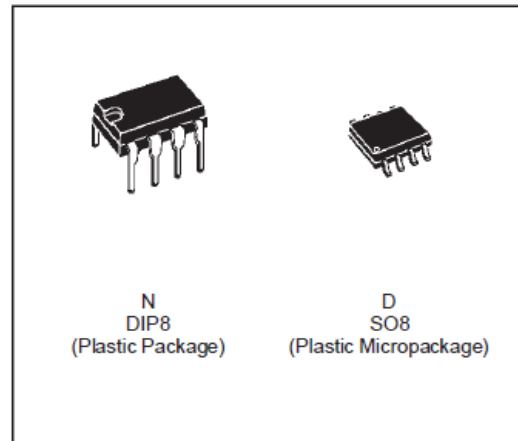
| Pin | Name | Description | Minimum | Typical | Maximum | Units |
|-----------|-------------|--------------------|---------|---------|-------------------|-------|
| 1 (White) | Signal | Input; TTL or CMOS | 3.3 | 5.0 | $V_{servo} + 0.2$ | V |
| 2 (Red) | V_{servo} | Power Supply | 4.0 | 5.0 | 6.0 | V |
| 3 (Black) | V_{ss} | Ground | | 0 | | V |

Power Precautions

- Do not use this servo with an unregulated wall-mount supply. Such power supplies may deliver variable voltage far above the stated voltage.
- Do not power this servo through the BASIC Stamp Module's Vdd pin.
- Servo current draw can spike while under peak load; be sure your application's regulator is prepared to supply adequate current for all servos used in combination.
- Some Stamps in Class tutorials, such as *What's a Microcontroller?* instruct the user to briefly power these servos with a 9 V battery when using a HomeWork Board and no load; this does not cause damage.

TL081CN

- WIDE COMMON-MODE (UP TO V_{CC}^+) AND DIFFERENTIAL VOLTAGE RANGE
- LOW INPUT BIAS AND OFFSET CURRENT
- OUTPUT SHORT-CIRCUIT PROTECTION
- HIGH INPUT IMPEDANCE J-FET INPUT STAGE
- INTERNAL FREQUENCY COMPENSATION
- LATCH UP FREE OPERATION
- HIGH SLEW RATE : $16V/\mu s$ (typ)



DESCRIPTION

The TL081, TL081A and TL081B are high speed J-FET input single operational amplifiers incorporating well matched, high voltage J-FET and bipolar transistors in a monolithic integrated circuit.

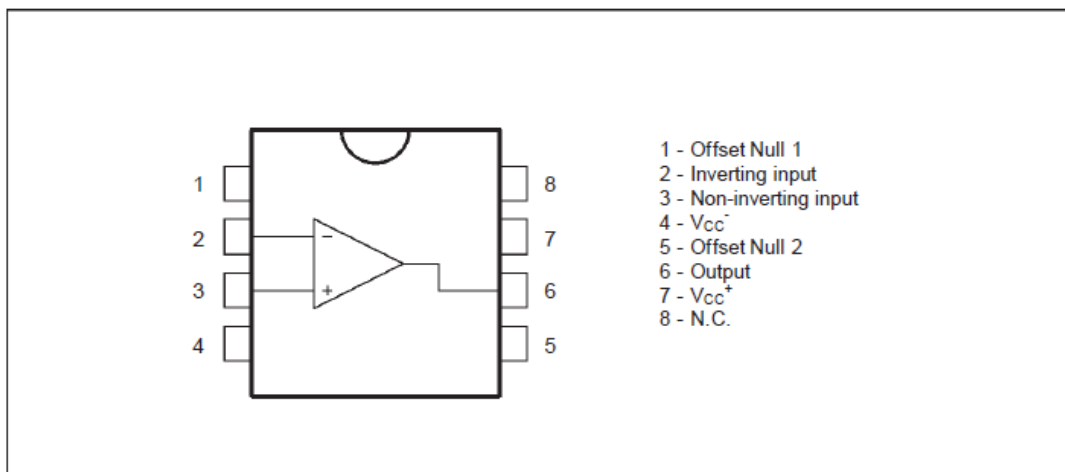
The devices feature high slew rates, low input bias and offset currents, and low offset voltage temperature coefficient.

ORDER CODES

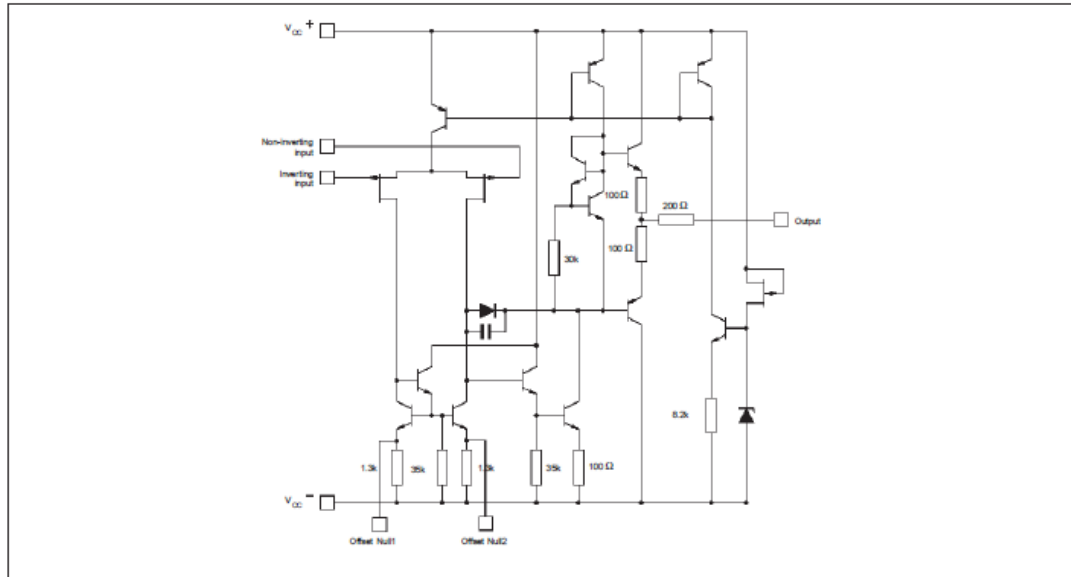
| Part Number | Temperature Range | Package | |
|--------------|-------------------|---------|---|
| | | N | D |
| TL081M/AM/BM | -55°C, +125°C | • | • |
| TL081I/AI/BI | -40°C, +105°C | • | • |
| TL081C/AC/BC | 0°C, +70°C | • | • |

Examples : TL081CD, TL081IN

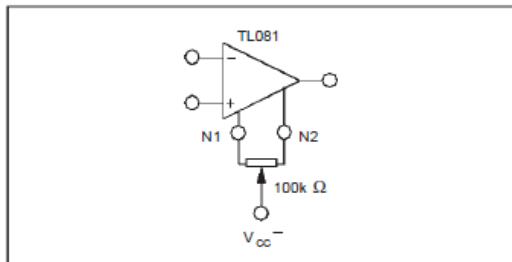
PIN CONNECTIONS (top view)



SCHEMATIC DIAGRAM



INPUT OFFSET VOLTAGE NULL CIRCUITS



ABSOLUTE MAXIMUM RATINGS

| Symbol | Parameter | Value | Unit | |
|------------|--|--|---|--------------------|
| V_{CC} | Supply Voltage - (note 1) | ± 18 | V | |
| V_i | Input Voltage - (note 3) | ± 15 | V | |
| V_{id} | Differential Input Voltage - (note 2) | ± 30 | V | |
| P_{tot} | Power Dissipation | 680 | mW | |
| | Output Short-circuit Duration - (note 4) | Infinite | | |
| T_{oper} | Operating Free Air Temperature Range | TL081C,AC,BC TL081I,AI,BI TL081M,AM,BM | 0 to 70 -40 to 105 -55 to 125 | $^{\circ}\text{C}$ |
| T_{stg} | Storage Temperature Range | | -65 to 150 | $^{\circ}\text{C}$ |

- Notes :
1. All voltage values, except differential voltage, are with respect to the zero reference level (ground) of the supply voltages where the zero reference level is the midpoint between V_{CC+} and V_{CC-} .
 2. Differential voltages are at the non-inverting input terminal with respect to the inverting input terminal.
 3. The magnitude of the input voltage must never exceed the magnitude of the supply voltage or 15 volts, whichever is less.
 4. The output may be shorted to ground or to either supply. Temperature and /or supply voltages must be limited to ensure that the dissipation rating is not exceeded.

ELECTRICAL CHARACTERISTICS

V_{CC} = ±15V, T_{amb} = 25°C (unless otherwise specified)

| Symbol | Parameter | TL0811,M,AC,AI, AM,BC,BI,BM | | | TL081C | | | Unit |
|------------------|--|--------------------------------|-------------|------------------------------|----------|------------|------------------|----------|
| | | Min. | Typ. | Max. | Min. | Typ. | Max. | |
| V _{io} | Input Offset Voltage (R _S = 50Ω) T _{amb} = 25°C T _{min.} ≤ T _{amb} ≤ T _{max.} TL081 TL081A TL081B TL081 TL081A TL081B | | 3 3 1 | 10 6 3 13 7 5 | | 3 | 10 13 | mV |
| DV _{io} | Input Offset Voltage Drift | | 10 | | | 10 | | μV/°C |
| I _{io} | Input Offset Current * T _{amb} = 25°C T _{min.} ≤ T _{amb} ≤ T _{max.} | | 5 | 100 4 | | 5 | 100 4 | pA nA |
| I _{ib} | Input Bias Current * T _{amb} = 25°C T _{min.} ≤ T _{amb} ≤ T _{max.} | | 20 | 200 20 | | 20 | 400 20 | pA nA |
| A _{vd} | Large Signal Voltage Gain (R _L = 2kΩ, V _O = ±10V) T _{amb} = 25°C T _{min.} ≤ T _{amb} ≤ T _{max.} | 50 25 | 200 | | 25 15 | 200 | | V/mV |
| SVR | Supply Voltage Rejection Ratio (R _S = 50Ω) T _{amb} = 25°C T _{min.} ≤ T _{amb} ≤ T _{max.} | 80 80 | 86 | | 70 70 | 86 | | dB |
| I _{CC} | Supply Current, no Load T _{amb} = 25°C T _{min.} ≤ T _{amb} ≤ T _{max.} | | 1.4 | 2.5 2.5 | | 1.4 | 2.5 2.5 | mA |
| V _{icm} | Input Common Mode Voltage Range | ±11 | +15 -12 | | ±11 | +15 -12 | | V |

| | | | | | | | | |
|-------------------|---|----------------------|------------------|----------|----------------------|------------------|----------|------------------------|
| CMR | Common Mode Rejection Ratio (R _S = 50Ω) T _{amb} = 25°C T _{min.} ≤ T _{amb} ≤ T _{max.} | 80 80 | 86 | | 70 70 | 86 | | dB |
| I _{os} | Output Short-circuit Current T _{amb} = 25°C T _{min.} ≤ T _{amb} ≤ T _{max.} | 10 10 | 40 | 60 60 | 10 10 | 40 | 60 60 | mA |
| ±V _{OPP} | Output Voltage Swing T _{amb} = 25°C T _{min.} ≤ T _{amb} ≤ T _{max.} R _L = 2kΩ R _L = 10kΩ R _L = 2kΩ R _L = 10kΩ | 10 12 10 12 | 12 13.5 | | 10 12 10 12 | 12 13.5 | | V |
| SR | Slew Rate (V _{in} = 10V, R _L = 2kΩ, C _L = 100pF, T _{amb} = 25°C, unity gain) | 8 | 16 | | 8 | 16 | | V/μs |
| t _r | Rise Time (V _{in} = 20mV, R _L = 2kΩ, C _L = 100pF, T _{amb} = 25°C, unity gain) | | 0.1 | | | 0.1 | | μs |
| K _{OV} | Overshoot (V _{in} = 20mV, R _L = 2kΩ, C _L = 100pF, T _{amb} = 25°C, unity gain) | | 10 | | | 10 | | % |
| GBP | Gain Bandwidth Product (f = 100kHz, T _{amb} = 25°C, V _{in} = 10mV, R _L = 2kΩ, C _L = 100pF) | 2.5 | 4 | | 2.5 | 4 | | MHz |
| R _i | Input Resistance | | 10 ¹² | | | 10 ¹² | | Ω |
| THD | Total Harmonic Distortion (f = 1kHz, A _V = 20dB, R _L = 2kΩ, C _L = 100pF, T _{amb} = 25°C, V _O = 2V _{PP}) | | 0.01 | | | 0.01 | | % |
| e _n | Equivalent Input Noise Voltage (f = 1kHz, R _S = 100Ω) | | 15 | | | 15 | | $\frac{nV}{\sqrt{Hz}}$ |
| ∅m | Phase Margin | | 45 | | | 45 | | Degrees |

* The input bias currents are junction leakage currents which approximately double for every 10°C increase in the junction temperature.

TL084CN

JFET Input Operational Amplifiers

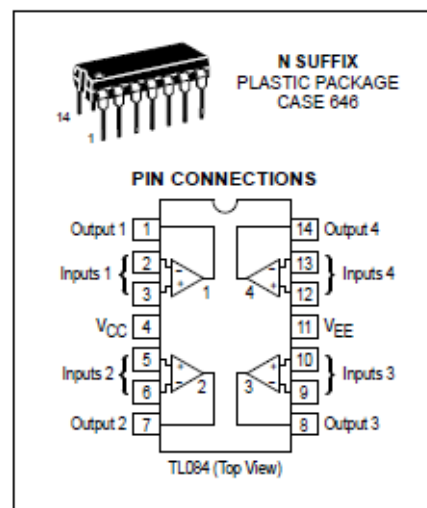
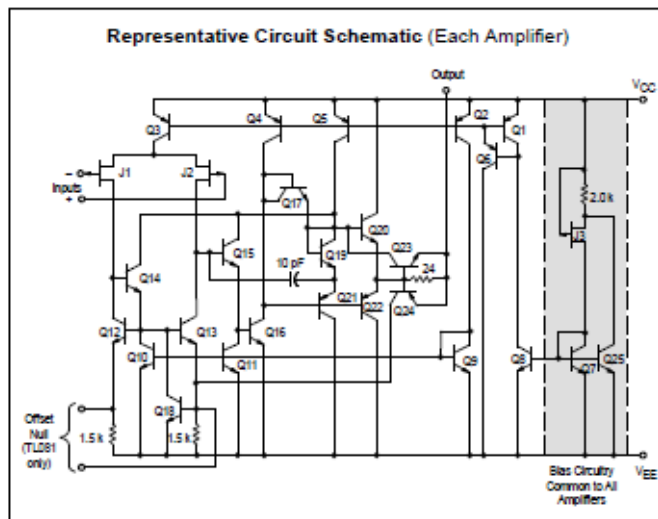
These low-cost JFET input operational amplifiers combine two state-of-the-art linear technologies on a single monolithic integrated circuit. Each internally compensated operational amplifier has well matched high voltage JFET input devices for low input offset voltage. The BIFET technology provides wide bandwidths and fast slew rates with low input bias currents, input offset currents, and supply currents.

These devices are available in single, dual and quad operational amplifiers which are pin-compatible with the industry standard MC1741, MC1458, and the MC3403/LM324 bipolar products.

- Input Offset Voltage Options of 6.0 mV and 15 mV Max
- Low Input Bias Current: 30 pA
- Low Input Offset Current: 5.0 pA
- Wide Gain Bandwidth: 4.0 MHz
- High Slew Rate: 13 V/ μ s
- Low Supply Current: 1.4 mA per Amplifier
- High Input Impedance: $10^{12} \Omega$

ORDERING INFORMATION

| Op Amp Function | Device | Operating Temperature Range | Package |
|-----------------|--------------|--|-------------|
| Single | TL081CD | $T_A = 0^\circ$ to $+70^\circ\text{C}$ | SO-8 |
| | TL081ACP | | Plastic DIP |
| Dual | TL082CD | $T_A = 0^\circ$ to $+70^\circ\text{C}$ | SO-8 |
| | TL082ACP | | Plastic DIP |
| Quad | TL084CN, ACN | $T_A = 0^\circ$ to $+70^\circ\text{C}$ | Plastic DIP |



MAXIMUM RATINGS

| Rating | Symbol | Value | Unit |
|---|-------------------------------------|-------------|-------------|
| Supply Voltage | V _{CC} V _{EE} | 18 -18 | V |
| Differential Input Voltage | V _{ID} | ±30 | V |
| Input Voltage Range (Note 1) | V _{IDR} | ±15 | V |
| Output Short Circuit Duration (Note 2) | t _{SC} | Continuous | |
| Power Dissipation | | | |
| Plastic Package (N, P) Derate above T _A = +47°C | P _D 1/θ _{JA} | 680 10 | mW mW/°C |
| Operating Ambient Temperature Range | T _A | 0 to +70 | °C |
| Storage Temperature Range | T _{stg} | -65 to +150 | °C |

NOTES: 1. The magnitude of the input voltage must not exceed the magnitude of the supply voltage or 15 V, whichever is less.
 2. The output may be shorted to ground or either supply. Temperature and/or supply voltages must be limited to ensure that power dissipation ratings are not exceeded.
 3. ESD data available upon request.

ELECTRICAL CHARACTERISTICS (V_{CC} = 15 V, V_{EE} = -15 V, T_A = T_{low} to T_{high} [Note 1].)

| Characteristics | Symbol | Min | Typ | Max | Unit |
|--|------------------|----------|--------|-----------------|------|
| Input Offset Voltage (R _S ≤ 10 k, V _{CM} = 0) TL081C, TL082C TL084C TL08_AC | V _{IO} | - | - | 20 20 7.5 | mV |
| Input Offset Current (V _{CM} = 0) (Note 2) TL08_C TL08_AC | I _{IO} | - | - | 5.0 3.0 | nA |
| Input Bias Current (V _{CM} = 0) (Note 2) TL08_C TL08_AC | I _{IB} | - | - | 10 7.0 | nA |
| Large-Signal Voltage Gain (V _O = ±10 V, R _L ≥ 2.0 k) TL08_C TL08_AC | A _{VOL} | 15 25 | - - | - - | V/mV |
| Output Voltage Swing (Peak-to-Peak) (R _L ≥ 10 k) (R _L ≥ 2.0 k) | V _O | 24 20 | - - | - - | V |

NOTES: 1. T_{low} = 0°C for TL081AC,C T_{high} = 70°C for TL081AC
 TL082AC,C TL082AC,C
 TL084AC,C TL084AC,C

2. Input Bias currents of JFET Input op amps approximately double for every 10°C rise in Junction Temperature as shown in Figure 3. To maintain junction temperature as close to ambient temperature as possible, pulse techniques must be used during testing.

ELECTRICAL CHARACTERISTICS ($V_{CC} = 15\text{ V}$, $V_{EE} = -15\text{ V}$, $T_A = 25^\circ\text{C}$, unless otherwise noted.)

| Characteristics | Symbol | Min | Typ | Max | Unit |
|--|--------------------------|----------------------|--------------------|-----------------|------------------------------|
| Input Offset Voltage ($R_S \leq 10\text{ k}$, $V_{CM} = 0$) TL081C, TL082C TL084C TL08_AC | V_{IO} | – | 5.0 5.0 3.0 | 15 15 6.0 | mV |
| Average Temperature Coefficient of Input Offset Voltage $R_S = 50\ \Omega$, $T_A = T_{low}$ to T_{high} (Note 1) | $\Delta V_{IO}/\Delta T$ | – | 10 | – | $\mu\text{V}/^\circ\text{C}$ |
| Input Offset Current ($V_{CM} = 0$) (Note 2) TL08_C TL08_AC | I_{IO} | – | 5.0 5.0 | 200 100 | μA |
| Input Bias Current ($V_{CM} = 0$) (Note 2) TL08_C TL08_AC | I_{IB} | – | 30 30 | 400 200 | μA |
| Input Resistance | r_i | – | 10^{12} | – | Ω |
| Common Mode Input Voltage Range TL08_C TL08_AC | V_{ICR} | ± 10 ± 11 | 15, –12 15, –12 | – – | V |
| Large Signal Voltage Gain ($V_O = \pm 10\text{ V}$, $R_L \geq 2.0\text{ k}$) TL08_C TL08_AC | A_{VOL} | 25 50 | 150 150 | – – | V/mV |
| Output Voltage Swing (Peak-to-Peak) ($R_L = 10\text{ k}$) | V_O | 24 | 28 | – | V |
| Common Mode Rejection Ratio ($R_S \leq 10\text{ k}$) TL08_C TL08_AC | CMRR | 70 80 | 100 100 | – – | dB |
| Supply Voltage Rejection Ratio ($R_S \leq 10\text{ k}$) TL08_C TL08_AC | PSRR | 70 80 | 100 100 | – – | dB |
| Supply Current (Each Amplifier) | I_D | – | 1.4 | 2.8 | mA |
| Unity Gain Bandwidth | BW | – | 4.0 | – | MHz |

| | | | | | |
|---|-------|---|-----|---|------------------------------|
| Slew Rate (See Figure 1) $V_{in} = 10\text{ V}$, $R_L = 2.0\text{ k}$, $C_L = 100\text{ pF}$ | SR | – | 13 | – | $\text{V}/\mu\text{s}$ |
| Rise Time (See Figure 1) | t_r | – | 0.1 | – | μs |
| Overshoot ($V_{in} = 20\text{ mV}$, $R_L = 2.0\text{ k}$, $C_L = 100\text{ pF}$) | OS | – | 10 | – | % |
| Equivalent Input Noise Voltage $R_S = 100\ \Omega$, $f = 1000\text{ Hz}$ | e_n | – | 25 | – | $\text{nV}/\sqrt{\text{Hz}}$ |
| Channel Separation $A_V = 100$ | CS | – | 120 | – | dB |

NOTES: 1. $T_{low} = 0^\circ\text{C}$ for TL081AC,C TL082AC,C TL084AC,C $T_{high} = 70^\circ\text{C}$ for TL081AC TL082AC,C TL084AC,C

2. Input Bias currents of JFET input op amps approximately double for every 10°C rise in Junction Temperature as shown in Figure 3. To maintain junction temperature as close to ambient temperature as possible, pulse techniques must be used during testing.

IPS 2303 Power Supply

2. SPECIFICATIONS

2-1. General

Main Supply (switch selectable) : 100V/120V/220V = 10%(230V +10%~-6%) 50/60Hz.
 Operation Environment : Indoor use.
 Altitude up to 2000m.
 Ambient temperature 0°C to 40°C.
 Relative humidity 80% (maximum).
 Installation category II.
 Pollution degree 2.
 Storage Temperature & Humidity : -10°C to 70°C,70% (maximum).
 Accessories : Operation Manual x1.

Table 2-1:

| Model | OUTPUT | | | REPLACEMENT FUSE TYPE(20,5mmHBC) | | RATED INPUT | | TEST LEADS (QTY) |
|----------|-------------------|-----------|-----------|----------------------------------|----------------|-------------|-----|------------------|
| | Independent | Series | Para | 100V/120V | 220V/230V | WATTS | VA | |
| IPS-2303 | 0-30Vx2 0-3Ax2 | 60V 3A | 30V 6A | T6A 250V | T3 15A 250V | 350 | 450 | 2 |
| IPS-3303 | 0-30Vx2 0-3Ax2 | 60V 3A | 30V 6A | T6.3A 250V | T3 15A 250V | 430 | 550 | 3 |
| IPS-4303 | 0-30Vx2 0-3Ax2 | 60V 3A | 30V 6A | T6.3A 250V | T3 15A 250V | 430 | 550 | 4 |

Dimensions : 255(W) x145(H) x265(D) mm
 Weight (kg) : 7.0 kg.

2-2. Operation Mode

- (1) Independent : Two independent outputs and CH3: 2.2-5.2V output for IPS-4303, Fixed 5V for IPS-3303, CH4: 8-15V output for IPS-4303. Output from 0 to rated voltage and 0 to rated current. Output from 0 to rated voltage at rated current each. Output from 0 to twice rated voltage at rated current. Output from 0 to twice rated current at rated volts.
- (2) Series
- (3) Parallel
- 2-3. Constant Voltage Operation
- (1) Output voltage range : 0 to rated voltage, continuously adjustable.
 : Line regulation $\leq 0.01\% + 3mV$.
 : Load regulation $\leq 0.01\% + 3mV$ (rated current $\leq 3A$).
 : Load regulation $\leq 0.02\% + 5mV$ (rated current $\leq 3A$).
 : $\leq 100 \mu s$ (50% load change, minimum load 0.5A).
- (2) Regulation
- (3) Recovery time : $\leq 1mVrms$ (5Hz-1MHz).
 : $\leq 300ppm/^\circ C$.
- (4) Ripple & Noise
- (5) Temperature coefficient

2-4. Constant Current Operation

- (1) Output current range : 0 to rated current continuously adjustable.
 (2) Regulation : Line regulation $\leq 0.2\% + 3mA$.
 : Load regulation $\leq 0.2\% + 3mA$.
 (3) Ripple current : $\leq 3mArms$.

2-5. Tracking Operation

- (1) Parallel Operation Regulation : Line regulation $\leq 0.01\% + 3mV$.
 : Load regulation $\leq 0.01\% + 3mV$ (rating current $\leq 3A$).
 : $\leq 0.02\% + 5mV$ (rating current $\leq 3A$).
 (2) Series Operation Regulation : Line regulation $\leq 0.01\% + 5mV$.
 : Load regulation $\leq 300mV$.
 A. Positive and Negative supply (Fig 5-4) CH2 tracking error $\leq 0.5\% + 10mV$ of the CH1 (No load, with load add load regulation $\leq 300mV$)
 B. Single supply (Fig 5-3)

2-6. Meter

- A. Display A : 3 digits panel meter x2 (0.5" Red LED display).
 Display V : 3 digits panel meter x 2 (0.5" Green LED display).
 B. Accuracy : OUTPUT ON $\pm 0.5\%$ of rdg + 2 digits
 : OUTPUT OFF $\pm 0.5\%$ of rdg + 8 digits (not for IPS-2303)
 C. Voltmeter : 99.9V full scale.
 D. Ammeter : 9.99A full scale.

2-7. CH3 Output Specifications

- (1) Regulation : Line regulation $\leq 5mV$, load regulation $\leq 15mV$.
 (2) Ripple & Noise : $\leq 2mVrms$.
 (3) Output Voltage range : IPS-4303: 2.2-5.2V $\pm 8\%$ continuous adjustment.
 : IPS-3303 fixed 5V $\pm 8\%$ continuous adjustment.
 : 3A for IPS-3303, 1A for IPS-4303.

2-8. CH4 Output Specifications

- (1) Regulation : Line regulation $\leq 5mV$, load regulation $\leq 10mV$.
 (2) Ripple & Noise : $\leq 2mVrms$.
 (3) Output Voltage range : IPS-4303: 8-15V $\pm 8\%$ continuous adjustment.
 (4) Output current : 1A

2-9. Insulation

- Between chassis and output terminal : 20M Ω or above (DC 500V).
 Between chassis and AC cord : 30M Ω or above (DC 500V).

4. PANEL CONTROLS AND INDICATORS

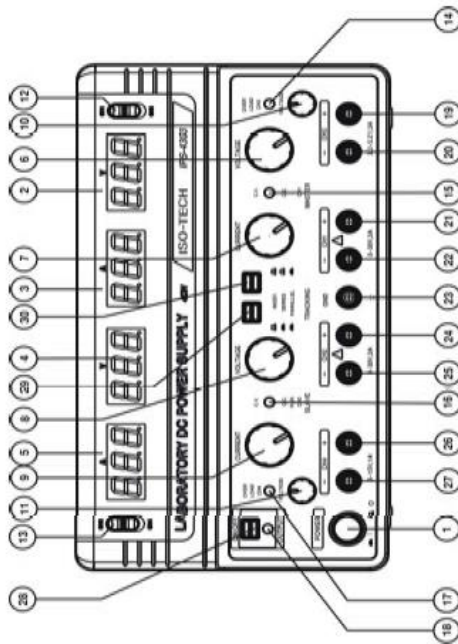


Fig. 4-1 Front Panel

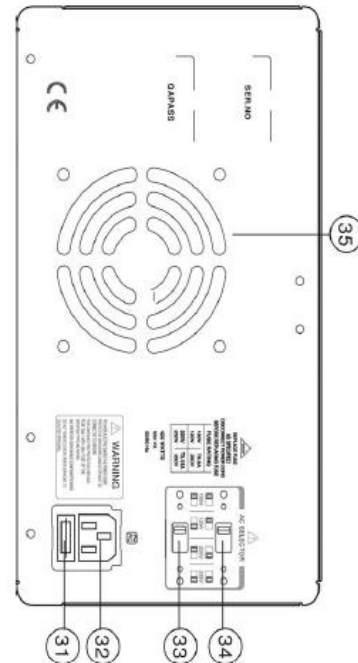
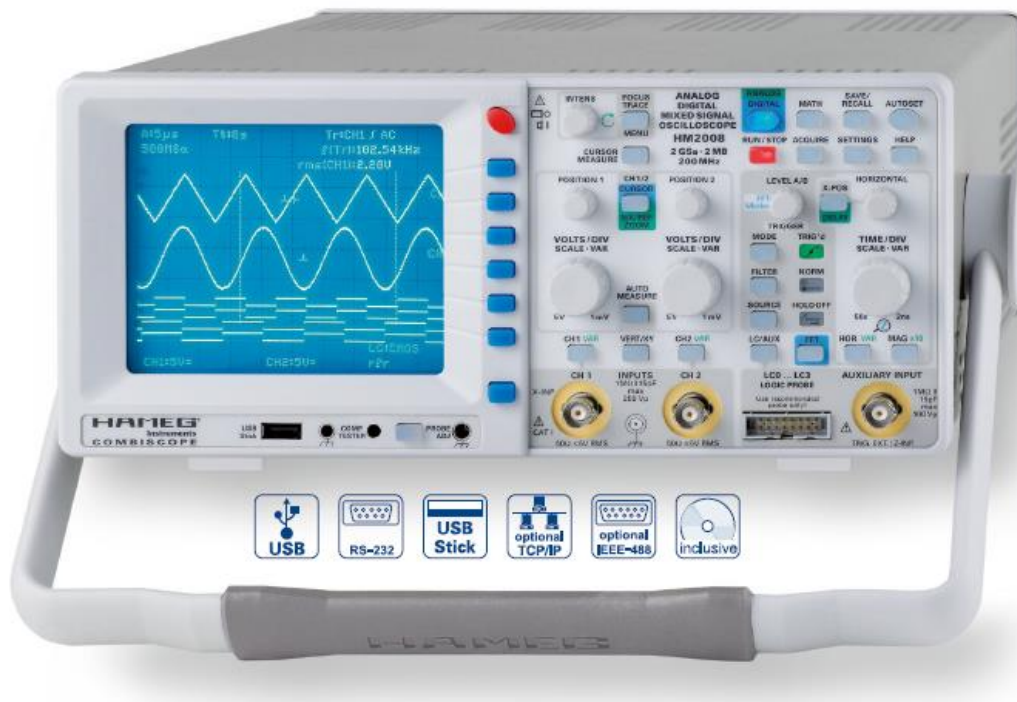


Fig. 4-2 Rear Panel

4-1. Front Panel

- | | | | |
|----------------------------|---|-------------------------|---|
| (1) Power switch | : Turns the power input ON/OFF | (15) CV&CC Indicator | : The CV light (green LED) is on when the CH1 supply is in constant voltage mode, or when both the CH1 and CH2 supplies are in constant voltage mode in either series or parallel tracking mode. : The CC light (red LED) is on when the CH1 supply is in constant current mode. |
| (2) Meter V | : Indicates the CH1 or CH3 output voltage. | (16) CV&CC Indicator | : The CV light (green LED) is on when the CH2 supply is in constant voltage mode. : The CC light (red LED) is on when the CH2 supply is in constant current mode or in parallel tracking mode. |
| (3) Meter A | : Indicates the CH1 or CH3 output current. | (17) Overload Indicator | : Lights when CH4 output load is larger than rated value (not IPS-2303 & 3303). |
| (4) Meter V | : Indicates the CH2 or CH4 output voltage. | (18) Output Indicator | : Lights when the output is turned on. |
| (5) Meter A | : Indicates the CH2 or CH4 output current. | (19) + output terminal | : Positive polarity terminal for the CH3 supply (not IPS-2303). |
| (6) Voltage Control | : Adjusts the output voltage of the CH1 supply. Also the adjustment control for the maximum output voltage of the CH2 supply when in either parallel or series tracking mode. | (20) - output terminal | : Negative polarity output terminal for the CH3 supply (not IPS-2303). |
| (7) Current Control | : Adjusts the output current of the CH1 supply. Also the adjustment control for the maximum output current of the CH2 supply when in either parallel or series tracking mode. | (21) + output terminal | : Positive polarity output terminal for the CH1 supply. |
| (8) Voltage Control | : Adjusts the output voltage of the CH2 supply when in independent operation. | (22) - output terminal | : Negative polarity output terminal for the CH1 supply. |
| (9) Current Control | : Adjusts the output current of the CH2 supply. | (23) GND terminal | : Earth and chassis ground. |
| (10) Voltage Control | : Adjusts the output voltage of the CH3 supply (not IPS-2303 & 3303). | (24) + output terminal | : Positive polarity output terminal for the CH2 supply. |
| (11) Voltage Control | : Adjusts the output voltage of the CH4 supply (not IPS-2303 & 3303). | (25) - output terminal | : Negative polarity output terminal for the CH2 supply. |
| (12) CH1/CH3 select switch | : Selects CH1 or CH3 output voltage or current (not IPS-2303 & 3303). | (26) + output terminal | : Positive polarity output terminal for the CH4 supply (not IPS-2303 & 3303). |
| (13) CH2/CH4 select switch | : Selects CH2 or CH4 output voltage or current (not IPS-2303 & 3303). | (27) - output terminal | : Negative polarity output terminal for the CH4 supply (not IPS-2303 & 3303). |
| (14) Overload Indicator | : Lights when CH3 output load is greater than rated output (not IPS-2303). | (28) Output switches | : Turns the output ON/OFF. |

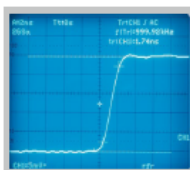
200 MHz Mixed Signal Combiscope HM2008



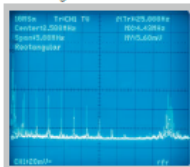
Logic Probe H02010



Rise Time Measurement in DSO Mode with 2ns/div., 2GSa/s



Frequency Analysis of a Video Signal with FFT



- ✓ 2GSa/s Real Time Sampling, 20GSa/s Random Sampling
- ✓ 2MPts Memory per Channel, Memory Zoom up to 100,000:1
- ✓ FFT for Spectral Analysis
- ✓ 2 Channels + 4 Logic Channels with Option H02010 (MSO)
- ✓ Deflection Coefficients 1mV/div....5V/div., with adjustable DC Offset Voltage; Time Base 2ns/div....50s/div.
- ✓ Acquisition Modes: Single, Refresh, Average, Envelope, Roll, Peak-Detect
- ✓ Front USB-Stick Connector for Screenshots
- ✓ USB/RS-232, optional: IEEE-488 (GPIB) or Ethernet/USB
- ✓ Signal Display: Yt, XY and FFT; Interpolation: Sinx/x, Pulse, Dot Join (linear)
- ✓ Adjustable Input Impedance 1M Ω /50 Ω
- ✓ See HM2005-2 for Analog Mode

200 MHz Mixed Signal CombiScope® HM2008

All data valid at 23 °C after 30 minutes warm-up.

| Vertical Deflection | |
|---|---|
| Channels: | |
| Analog: | 2 |
| Digital: | 2 + (additionally with Option HO2010) |
| 4 Logic Channels | |
| Operating Modes: | |
| Analog: | CH 1 or CH 2 separate, DUAL (CH 1 and CH 2 alternate or chopped), Addition |
| Digital: | Analog Signal Channels: CH 1 or CH 2 separate, DUAL (CH 1 and CH 2) or Addition. Logic Signal Channels (LCH 0...3) switchable. CH 1 |
| X in XY-Mode: | |
| Invert: | CH 1, CH 2 |
| Bandwidth (-3 dB): | 2 x 0...200 MHz |
| Rise time: | < 1.75 ns |
| Bandwidth Limiter (switchable): approx. 20 MHz (1 mV/div...5V/div.) | |
| Deflection Coefficients (CH 1, 2): 12 calibrated steps | |
| 1...2 mV/div.: | ±3% (0...100 MHz (-3 dB)) |
| 5 mV...5V/div.: | ±3% (1-2-5 sequence) |
| variable (uncalibrated): | > 1 mV/div...5V/div., continuous |
| Inputs CH 1, 2: | |
| Impedance: | 1 MΩ 13 pF |
| Coupling: | DC, AC, 50Ω, GND (ground) |
| Offset control: | |
| 1 mV, 2 mV | ±0.2V |
| 5...50 mV | ±1V |

| | |
|--|--|
| 100 mV...5V | ±20V |
| Max. Input Voltage: | 250V (DC + peak AC), 500 <math><V_{rms}</math> |
| Y Delay Line (analog): | 70 ns |
| Measuring Circuits: | Measuring Category I |
| Analog mode only: | |
| Auxiliary input: | |
| Function (selectable): | Ext. Trigger, Z (unblank in analog mode) |
| Coupling (Ext. Trig./Z): | all/AC, DC |
| Max. input voltage: | 100V (DC + peak AC) |
| Digital mode only: | |
| Logic Channels in combination with Option HO2010: | |
| Quantity | 4 (LCH 0...3) |
| Select. switching thresholds: | TTL, CMOS, ECL (common for all) |
| User definable thresholds: | 2 |
| within the range: | -2...+8V (common for all) |

| Triggering | |
|---|--|
| Analog and Digital Mode | |
| Automatic (Peak to Peak): | |
| Min. signal height: | 5 mm |
| Frequency range: | 10 Hz...250 MHz |
| Level control range: | from Peak- to Peak+ |
| Normal (without peak): | |
| Min. signal height: | 5 mm |
| Frequency range: | 0...250 MHz |
| Level control range: | -10...+10 div. |
| Operating modes: Slope/Video/Logic | |
| Slope: | Rising, falling, both |
| Sources: | CH 1, CH 2, alt. CH 1/2 (≥8 mm, analog mode only), Line, Ext. |
| Coupling: | |
| AC: | 10 Hz...250 MHz |
| DC: | 0...250 MHz |
| HF: | 30 kHz...250 MHz |
| LF: | 0...5 kHz |

| | |
|--|--|
| Video: | Noise Rej. switchable pos./neg. Sync. Impulse |
| Standards: | 525 Line/60 Hz Systems 625 Line/50 Hz Systems |
| Field: | even/odd/both |
| Line: | all/line number selectable |
| Source: | CH 1, CH 2, Ext. |
| Indicator for trigger action: LED | |
| External Trigger via: | AUXILIARY INPUT (0.3V _{pp} , 0...200 MHz) |
| Coupling: | AC, DC |
| Max. input voltage: | 100V (DC + peak AC) |
| Digital mode: | |
| Pre/Post Trigger: | -100...+400% relative to complete memory |
| Logic (with Option HO2010): | AND/OR, TRUE/FALSE |
| Source: | Logic Channel 0...3 |
| State: | X, H, L |
| Analog mode: | |

| | |
|----------------------|----------------|
| 2nd Trigger | |
| Min. signal height: | 5 mm |
| Frequency range: | 0...250 MHz |
| Coupling: | DC |
| Level control range: | -10...+10 div. |

| Horizontal Deflection | |
|---|--|
| Analog Time Base | |
| Operating modes: | A, ALT (alternating A/B), B |
| Time base A: | 20 ns/div...0.5 s/div. (1-2-5 sequence) |
| Time base B: | 20 ns/div...20 ms/div. (1-2-5 sequence) |
| Accuracy A and B: | ±3% |
| X Magnification x10: | to 2 ns/div. |
| Accuracy: | ±5% |
| Variable time base A/B: | cont. 1:2.5 |
| Hold Off time: | var. 1:10 (LED-Indication) |
| Analog XY Mode | |
| Bandwidth X-Amplifier: | 0...3 MHz (-3 dB) |
| XY phase shift: | <3° <220 kHz |
| Digital Time Base | |
| Time base range (1-2-5 sequence) | |
| Refresh Mode: | 2 ns/div...50 s/div. |
| with Peak Detect: | 500 ns/div...50 s/div. (min. Pulse Width 10 ns) |
| Roll Mode: | 50 ms/div...50 s/div. |
| Accuracy time base | |
| Time coefficient: | 50 ppm |
| Display: | ±1% |
| Memory Zoom: | max. 100,000:1 |
| Digital XY Mode | |
| Bandwidth X-Amplifier: | 0...200 MHz (-3 dB) |
| XY phase shift: | <3° <200 MHz |

| Digital Storage | |
|---|--|
| Sampling Rate (real time): | Analog channels: 2 x 1 GSa/s or 1 x 2 GSa/s (interleaved); Logic Channels: max. 4 x 500 MSa/s |
| Sampling Rate (random sampling): | 20 GSa/s (1-Channel mode) |
| Bandwidth: | 25 GSa/s (2-Channel mode) |
| Memory: | 2 x 0...200 MHz (Random) |
| Operating modes: | 2 x 2 MPts (analog); 4 x 2 MPts (logic) |
| | Refresh, Average, Envelope, Roll; Free Run/Triggered, Peak-Detect |
| Resolution (vertical): | 8 Bit (25 Pts/div.) |
| Resolution (horizontal): | |
| Yt: | 11 Bit (200 Pts/div.) |
| XY: | 8 Bit (25 Pts/div.) |
| Interpolation: | Sinx/x, Dot Join (linear) |
| Delay: | 2 Million x (1/Sampling Rate; max.) 8 Million x (1/Sampling Rate; max.) |
| Display refresh rate: | max. 170/s at 2 MPts |
| Display: | Dots (acquired points only), Vectors (interpolation), Optimal (complete memory weighting and vector display) |
| Reference Memories: | 9 with 2kPts each (for recorded signals) |
| Display: | 2 signals of 9 (freely selectable) |

| FFT Mode | |
|-------------------------------|------------------------------------|
| Display X: | Frequency Range |
| Display Y: | True rms value of spectrum |
| Scaling: | Linear or logarithmic |
| Level display: | dBV, V |
| Window: | Square, Hanning, Hamming, Blackman |
| Control: | Center frequency, Span |
| Marker: | Frequency, Amplitude |
| Zoom (frequency axis): | up to x20 |

| Operation/Measuring/Interfaces | |
|---|--|
| Operation: Menu (multilingual), Autoset, Help functions (multilingual) | |
| Save/Recall internal: | |
| analog: | 9 Instrument parameter settings |
| digital: | 9 Signals (each 2k) incl. instrument parameters |
| Signal sources: | CH 1, CH 2, LCH 0...3, ZOOM, Reference 1...9 or Mathematics |
| Signal display: | max. 6 traces |
| USB Memory-Stick: | |
| Save/Recall external: | |
| Instrument settings and Signals: | CH 1, CH 2, LCH 0...3, ZOOM, Reference 1...9 or Mathematics |
| Screen-shot: | as Bitmap |
| Signal display data (2k per channel): | Binary (SCPI-Data), Text (ASCII-Format), CSV (Spread Sheet) |
| Frequency counter: | |

| | |
|-----------------------------|--|
| 6 digit resolution: | 1...250 MHz |
| 5 digit resolution: | 0.5Hz...1 MHz |
| Accuracy: | 50 ppm |
| Auto Measurements: | |
| Analog mode: | Frequency, Period, V_{dc} , V_{pp} , V_{p+} , V_{p-} |
| plus in digital mode: | V_{rms} , V_{avg} |
| Cursor Measurements: | |
| Analog mode: | Δt , $1/\Delta t$ (f), tr, ΔV , V to GND, ratio X, ratio Y |
| plus in digital mode: | V_{pp} , V_{p+} , V_{p-} , V_{avg} , V_{rms} , pulse count |
| Resolution Readout/Cursor: | 1000 x 2000 Pts, Signals: 250 x 2000 |
| Interfaces (plug-in): | USB/RS-232 (HO720) |
| Optional: | IEEE-488, Ethernet/USB |

Mathematic functions

| | |
|-------------------------|---|
| Number of Formula Sets: | 5 with 5 formulas each |
| Sources: | CH 1, CH 2, Math 1–Math 5 |
| Targets: | 5 math. memories (Math 1...5) |
| Functions: | ADD, SUB, 1/X, ABS, MUL, DIV, SQ, POS, NEG, INV |
| Display: | max. 2 math. memories (Math 1...5) |

Display

| | |
|--------------------------------|------------------|
| CRT: | D14-375GH |
| Display area (with graticule): | 8 div. x 10 div. |
| Acceleration voltage: | approx. 14 kV |

General Information

| | |
|-------------------------|--|
| Component tester | |
| Test voltage: | approx. $7V_{rms}$ (open circuit), approx. 50 Hz |
| Test current: | max. $7mA_{rms}$ (short circuit) |
| Reference Potential: | Ground (safety earth) |
| Probe ADJ Output: | 1 kHz/1 MHz square wave signal $0.2V_{pp}$ (tr <4 ns) |
| Trace rotation: | electronic |
| Line voltage: | 105...253V, 50/60 Hz $\pm 10\%$, CAT II |
| Power consumption: | 48 Watt at 230V, 50Hz |
| Protective system: | Safety class I (EN61010-1) |
| Operating temperature: | +5...+40 °C |
| Storage temperature: | -20...+70 °C |
| Rel. humidity: | 5...80% (non condensing) |
| Dimensions (W x H x D): | 285 x 125 x 380 mm |
| Weight: | 5.6 kg |

IDM201N Digital Multimeter

3.1 General Specifications

Display: 3-1/2 digit (2000 count), full annunciators, automatic polarity indication

Measurement Rate: 1.5 times per second

Over range Indication: OL is displayed for Ω , Hz, \pm functions. Actual value is shown for A, and V functions

Backlight: Electroluminescent panel

Operating Temperature: 0°C to 50°C

Storage Temperature: -30°C to 70°C (batteries removed)

Temperature Stability: 0.15 x (Spec. Accuracy) per °C, < 18°C or > 28°C

Relative Humidity: Maximum RH 80% for temperatures up to 31°C decreasing linearly to 50% RH at 40°C

Safety: EN (IEC) 61010-1:2001 specifications

Maximum Voltage to Earth: 600V dc or ac peak to any terminal

Power Requirements: ac; 90 ~ 264V, 50/60Hz < 10VA, dc; Qty 6 x 1.5V AA LR6 or AM3 battery

Battery Life: 1200 hours without backlight (alkaline) or 80 hours with continuous backlight (alkaline)

Size (mm): 218(W) x 195(L) x 73(H) without carrying strap

Weight: 1.3 Kg without power cord

Mains Fuse: 80mA/250V, FAST, 5x20mm HBC Ceramic

Overload Protection: 1A 600V IR 10KA fuse (Bussmann BBS-1 or equivalent) for μ A and mA input
15A 600V IR 10KA fuse (Bussmann KTK 15 or equivalent) for A input

Environment Conditions

The equipment was designed for use in the following environmental conditions:

- Indoor use only.
- Maximum Altitude: 2000 meters.
- Installation Category (Measurement Category): IEC 61010-1, 600V CAT II, 300V CAT III.
- Pollution degree: 2

NOTE:

Measurement Category I is for measurements performed on circuits not directly connected to mains. Examples include: Measurements on battery powered equipment and specially protected (internal) mains-derived circuits.

Measurement Category II is for measurements on circuits directly connected to the low voltage installation. Examples include: Household appliances, portable tools and similar equipment.

Measurement Category III is for measurements performed in the building installation. Examples include measurements on distribution boards, junction boxes, socket-outlets and wiring and cables in the fixed installation.

Measurement Category IV is for measurements performed at the source of the low-voltage installation. Examples include measurements on primary overcurrent protection devices and electricity meters.

3.2 Electrical characteristics

Accuracy is \pm (% reading + number of digits) at 23°C \pm 5°C, less than 80% R.H.

(1) DC Volts

| Range | Resolution | Accuracy | Over voltage protection |
|-------|-------------|-------------------|-------------------------|
| 200mV | 100 μ V | \pm (0.5% + 2d) | 1100V dc or 1100Vp-p |
| 2V | 1mV | | |
| 20V | 10mV | | |
| 200V | 100mV | | |
| 600V | 1V | | |

Input Impedance: 10M Ω (over 1000M Ω in 200mV range).

(2) AC Volts

| Range | Resolution | Accuracy | Over voltage protection |
|-------|------------|-----------------------------|-------------------------|
| 200mV | 100µV | Unspecified | 1100V dc or 1100Vp-p |
| 2V | 1mV | ± (1.5% + 5d) 50Hz to 500Hz | |
| 20V | 10mV | | |
| 200V | 100mV | | |
| 600V | 1V | | |

Input Impedance: 10MΩ // less than 100pF.

AC Conversion Type: AC conversions are ac-coupled true rms responding and calibrated to the rms value sine-wave input.

(3) Resistance

| Range | Resolution | Accuracy | Over voltage protection |
|-------|------------|-----------------|-------------------------|
| 200Ω | 0.1Ω | ±(0.7% + 3d) *1 | 600V dc or 600V rms |
| 2KΩ | 1Ω | ±(0.7% + 3d) | |
| 20KΩ | 10Ω | | |
| 200KΩ | 100Ω | ±(1.0% + 3d) | |
| 2MΩ | 1KΩ | | |
| 20MΩ | 10KΩ | ±(2.5% + 3d) *2 | |

Open circuit Voltage: -1.3V approx.

*1 < 10dgt rolling.

*2 < 100dgt rolling.

(4) Diode Check

| Resolution | Accuracy | Max. Test Current | Max. Open Current Voltage | Overload protection |
|------------|----------------------------------|-------------------|---------------------------|------------------------|
| 10mV | ±(1.5% + 5d) For 0.4V to 0.8V | 1.5mA | 3V | 600V dc or 600V rms |

Internal beeper activates if the voltage of the circuit under test is less than 0.25V.

(5) Continuity

The internal sounder activates if the resistance of the circuit under test is less than 20Ω. It will then turn off if the resistance is increased beyond 50Ω. Response time is approximately 50 msec. Overload Protection: 600V dc/ac rms max.

(6) Capacitance

| Range | Resolution | Accuracy | Over voltage protection |
|--------|------------|-----------------|-------------------------|
| 2nF *3 | 1pF | ±(1.9% + 20d) | 600V dc or 600V rms |
| 20nF | 10pF | ±(1.9% + 8d) | |
| 200nF | 100pF | | |
| 2µF | 1nF | | |
| 20µF | 10nF | | |
| 200µF | 100nF | | |
| 2mF *1 | 1µF | ±(1.9% + 8d) *2 | |

*1 < 100 dgt of reading rolling.

*2 The accuracy is specified with the backlight not activated.

*3 For accurate readings, do not use the mains power supply when measuring low-value capacitance.

*4 In order to ensure the best accuracy in measurement of a low value capacitance, note and subtract the stray capacitance of the open-circuit test leads from subsequent measurements.

(7) Frequency

| Resolution | Resolution | ** Sensitivity | Accuracy | Overload protection |
|------------|------------|----------------|------------|---------------------|
| 2000Hz | 1Hz | 100mV rms * | 0.01% ± 2d | 600V rms |
| 20KHz | 10Hz | | | |
| 200KHz | 100Hz | | | |
| 2MHz | 1KHz | 250mV rms | | |
| 20MHz | 10KHz | 1V rms | | |

* Less than 20Hz the sensitivity is 1.5V rms.

** Max. Sensitivity: <5 Vac rms.

(8) DC Current

| Range | Resolution | Accuracy | Voltage Burden |
|---------|------------|--------------|----------------|
| 200.0μA | 0.1μA | ±(1.0% + 2d) | 800mV max |
| 2000μA | 1μA | | |
| 20.00mA | 0.01mA | | |
| 200.0mA | 0.1mA | | |
| 10A | 10mA | ±(1.5% + 4d) | 1V max |

10A Range: 30 seconds maximum for current greater than 10A.

(9) AC Current

| Range | Resolution | Accuracy | Voltage Burden |
|---------|------------|--------------|----------------|
| 200.0μA | 0.1μA | ±(1.5% + 5d) | 800mV max |
| 2000μA | 1μA | | |
| 20.00mA | 0.01mA | | |
| 200.0mA | 0.1mA | | |
| 10A | 10mA | ±(2.5% + 5d) | 1V max |

Frequency Response: 50Hz to 500Hz

10A Range: 30 seconds maximum for current greater than 10A.

AC Conversion Type: AC conversions are ac-coupled true rms responding and calibrated to the rms value sine-wave input.

(10) Auto Power-off (APO)

If the instrument is not used for more than 10 minutes, it automatically turns the power off. The state of the instrument is stored and it can be turned on again by pressing any button or changing the rotary switch position. If a button is pressed, the LCD displays the saved state; press the Hold button to disable the hold state. The instrument will alarm 15 seconds before automatically turning power off, but any button press or rotary switch movement will reset Auto Power-off.

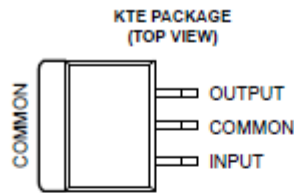
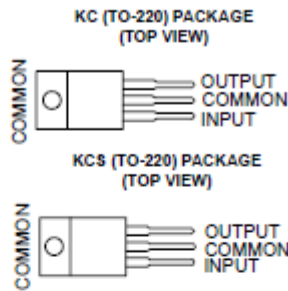
(11) Disable Auto Power-off

To disable the auto power-off function, press and hold any button (except the Hold button) whilst turning on the instrument.

LM7805

- 3-Terminal Regulators
- Output Current up to 1.5 A
- Internal Thermal-Overload Protection

- High Power-Dissipation Capability
- Internal Short-Circuit Current Limiting
- Output Transistor Safe-Area Compensation

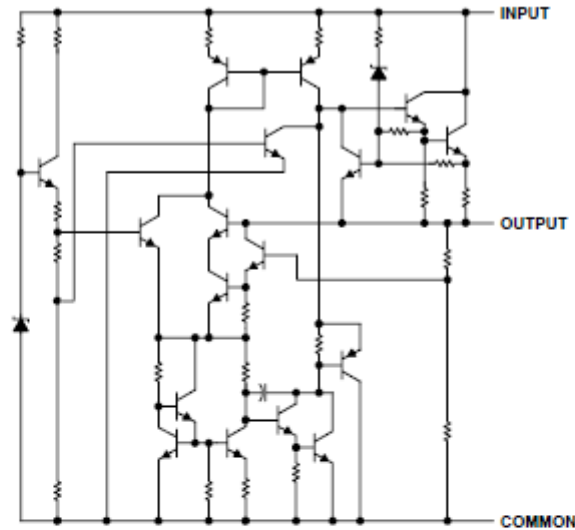


description/ordering information

This series of fixed-voltage integrated-circuit voltage regulators is designed for a wide range of applications. These applications include on-card regulation for elimination of noise and distribution problems associated with single-point regulation. Each of these regulators can deliver up to 1.5 A of output current. The internal current-limiting and thermal-shutdown features of these regulators essentially make them immune to overload. In addition to use as fixed-voltage regulators, these devices can be used with external components to obtain adjustable output voltages and currents, and also can be used as the power-pass element in precision regulators.

ORDERING INFORMATION

| T _J | V _{O(NOM)} (V) | PACKAGE† | | ORDERABLE PART NUMBER | TOP-SIDE MARKING |
|----------------|----------------------------|------------------------------|--------------|--------------------------|---------------------|
| 0°C to 125°C | 5 | POWER-FLEX (KTE) | Reel of 2000 | μA7805CKTER | μA7805C |
| | | TO-220 (KC) | Tube of 50 | μA7805CKC | μA7805C |
| | | TO-220, short shoulder (KCS) | Tube of 20 | μA7805CKCS | |
| | 8 | POWER-FLEX (KTE) | Reel of 2000 | μA7808CKTER | μA7808C |
| | | TO-220 (KC) | Tube of 50 | μA7808CKC | μA7808C |
| | | TO-220, short shoulder (KCS) | Tube of 20 | μA7808CKCS | |
| | 10 | POWER-FLEX (KTE) | Reel of 2000 | μA7810CKTER | μA7810C |
| | | TO-220 (KC) | Tube of 50 | μA7810CKC | μA7810C |
| | | TO-220, short shoulder (KCS) | Tube of 20 | μA7810CKCS | |
| | 12 | POWER-FLEX (KTE) | Reel of 2000 | μA7812CKTER | μA7812C |
| | | TO-220 (KC) | Tube of 50 | μA7812CKC | μA7812C |
| | | TO-220, short shoulder (KCS) | Tube of 20 | μA7812CKCS | |
| | 15 | POWER-FLEX (KTE) | Reel of 2000 | μA7815CKTER | μA7815C |
| | | TO-220 (KC) | Tube of 50 | μA7815CKC | μA7815C |
| | | TO-220, short shoulder (KCS) | Tube of 20 | μA7815CKCS | |
| | 24 | POWER-FLEX (KTE) | Reel of 2000 | μA7824CKTER | μA7824C |
| | | TO-220 (KC) | Tube of 50 | μA7824CKC | μA7824C |



absolute maximum ratings over virtual junction temperature range (unless otherwise noted)†

| | |
|--|----------------|
| Input voltage, V_i : $\mu A7824C$ | 40 V |
| All others | 35 V |
| Operating virtual junction temperature, T_J | 150°C |
| Lead temperature 1,6 mm (1/16 inch) from case for 10 seconds | 260°C |
| Storage temperature range, T_{stg} | -65°C to 150°C |

† Stresses beyond those listed under "absolute maximum ratings" may cause permanent damage to the device. These are stress ratings only, and functional operation of the device at these or any other conditions beyond those indicated under "recommended operating conditions" is not implied. Exposure to absolute-maximum-rated conditions for extended periods may affect device reliability.

package thermal data (see Note 1)

| PACKAGE | BOARD | θ_{JC} | θ_{JA} |
|------------------|-------------------|---------------|---------------|
| POWER-FLEX (KTE) | High K, JESD 51-5 | 3°C/W | 23°C/W |
| TO-220 (KC/KCS) | High K, JESD 51-5 | 3°C/W | 19°C/W |

NOTE 1: Maximum power dissipation is a function of $T_J(\max)$, θ_{JA} , and T_A . The maximum allowable power dissipation at any allowable ambient temperature is $P_D = (T_J(\max) - T_A) / \theta_{JA}$. Operating at the absolute maximum T_J of 150°C can affect reliability.

| | | MIN | MAX | UNIT | |
|----------------|--|----------------|------|------|----|
| V _I | Input voltage | μA7805C | 7 | 25 | V |
| | | μA7808C | 10.5 | 25 | |
| | | μA7810C | 12.5 | 28 | |
| | | μA7812C | 14.5 | 30 | |
| | | μA7815C | 17.5 | 30 | |
| | | μA7824C | 27 | 38 | |
| I _O | Output current | | 1.5 | A | |
| T _J | Operating virtual junction temperature | μA7800C series | 0 | 125 | °C |

electrical characteristics at specified virtual junction temperature, V_I = 10 V, I_O = 500 mA (unless otherwise noted)

| PARAMETER | TEST CONDITIONS | T _J † | μA7805C | | | UNIT |
|---|---|------------------|---------|-----|------|-------|
| | | | MIN | TYP | MAX | |
| Output voltage | I _O = 5 mA to 1 A, V _I = 7 V to 20 V, P _D ≤ 15 W | 25°C | 4.8 | 5 | 5.2 | V |
| | | 0°C to 125°C | 4.75 | | 5.25 | |
| Input voltage regulation | V _I = 7 V to 25 V | 25°C | 3 | | | mV |
| | V _I = 8 V to 12 V | | 1 | | | |
| Ripple rejection | V _I = 8 V to 18 V, f = 120 Hz | 0°C to 125°C | 62 | 78 | | dB |
| Output voltage regulation | I _O = 5 mA to 1.5 A | 25°C | 15 | | | mV |
| | I _O = 250 mA to 750 mA | | 5 | | | |
| Output resistance | f = 1 kHz | 0°C to 125°C | 0.017 | | | Ω |
| Temperature coefficient of output voltage | I _O = 5 mA | 0°C to 125°C | -1.1 | | | mV/°C |
| Output noise voltage | f = 10 Hz to 100 kHz | 25°C | 40 | | | μV |
| Dropout voltage | I _O = 1 A | 25°C | 2 | | | V |
| Bias current | | 25°C | 4.2 | 8 | | mA |
| Bias current change | V _I = 7 V to 25 V | 0°C to 125°C | 1.3 | | | mA |
| | I _O = 5 mA to 1 A | | 0.5 | | | |
| Short-circuit output current | | 25°C | 750 | | | mA |
| Peak output current | | 25°C | 2.2 | | | A |

† Pulse-testing techniques maintain the junction temperature as close to the ambient temperature as possible. Thermal effects must be taken into account separately. All characteristics are measured with a 0.33-μF capacitor across the input and a 0.1-μF capacitor across the output.

| PARAMETER | TEST CONDITIONS | T _J † | μA7808C | | | UNIT |
|---|--|------------------|---------|-----|-----|-------|
| | | | MIN | TYP | MAX | |
| Output voltage | I _O = 5 mA to 1 A, V _I = 10.5 V to 23 V, P _D ≤ 15 W | 25°C | 7.7 | 8 | 8.3 | V |
| | | 0°C to 125°C | 7.6 | | 8.4 | |
| Input voltage regulation | V _I = 10.5 V to 25 V | 25°C | 6 | | | mV |
| | V _I = 11 V to 17 V | | 2 | | | |
| Ripple rejection | V _I = 11.5 V to 21.5 V, f = 120 Hz | 0°C to 125°C | 55 | 72 | | dB |
| Output voltage regulation | I _O = 5 mA to 1.5 A | 25°C | 12 | | | mV |
| | I _O = 250 mA to 750 mA | | 4 | | | |
| Output resistance | f = 1 kHz | 0°C to 125°C | 0.016 | | | Ω |
| Temperature coefficient of output voltage | I _O = 5 mA | 0°C to 125°C | -0.8 | | | mV/°C |
| Output noise voltage | f = 10 Hz to 100 kHz | 25°C | 52 | | | μV |
| Dropout voltage | I _O = 1 A | 25°C | 2 | | | V |
| Bias current | | 25°C | 4.3 | 8 | | mA |
| Bias current change | V _I = 10.5 V to 25 V | 0°C to 125°C | 1 | | | mA |
| | I _O = 5 mA to 1 A | | 0.5 | | | |
| Short-circuit output current | | 25°C | 450 | | | mA |
| Peak output current | | 25°C | 2.2 | | | A |

† Pulse-testing techniques maintain the junction temperature as close to the ambient temperature as possible. Thermal effects must be taken into account separately. All characteristics are measured with a 0.33-μF capacitor across the input and a 0.1-μF capacitor across the output.

8. REFERENCIAS

- Alzate, E. et al. (2007). “Medidores de deformación por resistencia: galgas extensiométricas”. *Scientia et Technia* (NO. 34). DOI: 10.22517/23447214.5519.
- Banala, K. et al. (2009). “Robot Assisted Gait Training With Active Leg Exoskeleton (ALEX)”. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, VOL. 17 (NO. 1). DOI: 10.1109/TNSRE.2008.2008280
- Contreras-Vidal, J. et al. (2016). “Powered exoskeletons for bipedal locomotion after spinal cord injury”. *Journal of Neural Engineering*. DOI: 10.1088/17412560/13/3/031001.
- García Casado, FJ. (2010). “Galgas Extensiométricas: Funcionamiento y Circuitos de Medida”. Universitat Politècnica de València. Escuela Técnica Superior de Ingenieros Industriales.
- Jang, E. et al. (2010). “Recognition of Walking Intention Using Multiple Bio/Kinesthetic Sensors for Lower Limb Exoskeletons”. *Control, Automation and Systems*. DOI: 10.1109/ICCAS.2010.5669799.
- Jung, P. et al. (2015). “A Wearable Gesture Recognition Device for Detecting Muscular Activities Based on Air-Pressure Sensors”. *IEEE Transactions on Industrial Informatics*, DOI: 10.1109/TII.2015.2405413.
- Kawamoto, H. et al. (2003). “Power Assist Method for HAL-3 Using EMG-based Feedback Controller”. *IEEE International Conference on Systems, Man and Cybernetics*. DOI: 10.1109/ICSMC.2003.1244649.
- Kong, K. y Tomizuya, M. (2008). “Smooth and Continuous Human Gait Phase Detection Based on Foot Pressure Patterns”. *IEEE International Conference on Robotics and Automation*. DOI: 10.1109/ROBOT.2008.4543775.
- Marco, C. (2001). “Cinesiología de la marcha humana normal”. *Universidad de Zaragoza*.

Rea, R. et al. (2013). "X1: a Robotic Exoskeleton for In-Space Countermeasures and Dynamometry". AIAA SPACE Conference and Exposition, 2013. Oceanering Space Systems, Houston.

Schenkman, M. et al. (1990). "Whole-Body Movements During Rising to Standing from Sitting". *Physical Therapy*. VOL. 70 (NO. 10).

Tsukahara, A. (2010). "Sit-to-Stand and Stand-to-Sit Transfer Support for Complete Paraplegic Patients with Robot Suit HAL", *Adv. Robot.*, VOL. 24, pp. 1615-1638. DOI: 10.1163/016918610X512622.

Vukobratovic M., et al. (1974), "Development of Active Anthropomorphic Exoskeletons". *Medical and Biological Engineering*, VOL. 12, (NO 1). DOI: 10.1007/BF02629836.

Vukobratovic, M. (2007). "When were active exoskeletons actually born?" Mihajlo Pupin Institute, Robotics Department. Belgrado, Serbia & Montenegro.

Wu, H. et al. (2016). "Recognition of Stance Phase Using Flexible Pressure Sensors". *2016 8th International Conference on Intelligent Human-Machine Systems and Cybernetics (IHMSC)*. DOI: 10.1109/IHMSC.2016.218.

Young, J. y Ferris, D. (2015). "State-of-the-art and Future Directions for Lower Limb Robotic Exoskeletons". *IEEE Transactions on Neural Systems and Rehabilitation Engineering*. DOI 10.1109/TNSRE.2016.2521160.

Zeilig, G. et al. (2012). "Safety and Tolerance of the ReWalk™ Exoskeleton Suit for Ambulation by People with Complete Spinal Cord Injury: A Pilot Study". *The Journal of Spinal Cord Medicine*, VOL. 35. DOI: 10.1179/2045772312Y.0000000003.

Zoss, A., Kazerooni, H. y Chu, A. (2006). "Biomechanical Design of the Berkeley Lower Extremity Exoskeleton (BLEEX)". *IEEE/ASME Transactions on Mechatronics*, VOL. 11 (NO. 2), DOI: 10.1109/TMECH.2006.871087.

9. GLOSARIO

Se adjunta a continuación una serie de términos utilizados a lo largo del documento, acompañados de su explicación según su contexto.

- **Bíceps femoral.** También llamado bíceps crural, es un músculo de la zona externa en la región posteroexterna del muslo, es decir, en la parte superior trasera del muslo. Es especialmente importante en la flexión de la pierna sobre el muslo.
- **Bipedestación.** Alude a las acciones de levantarse y/o sentarse de una persona, así como mantenerse de pie.
- **Cuádriceps femoral.** Músculo situado en la parte superior de la pierna encargado de soportar nuestro peso, y que nos permite caminar, sentarnos, correr...
- **Deambulación.** También denominado marcha, se refiere al acto de caminar de una persona.
- **Dispositivo.** Mecanismo o artificio para producir una acción prevista. Puede referirse tanto a una parte del exoesqueleto como a todo su conjunto en sí.
- **Entorno de desarrollo.** También conocido como IDE, se trata de una aplicación informática dirigida a un desarrollo software eficiente por parte del programador.
- **Estado del arte.** Del inglés, *state-of-the-art*, define lo más avanzado en investigación y desarrollo en un ámbito concreto.
- **Exoesqueleto.** Estructura ajustable alrededor de una o varias extremidades de una persona. Por medio de sensores, motores y otros componentes electrónicos ofrece al usuario una ayuda parcial o completa para realizar movimientos que a priori podría ejecutar, o con mayor dificultad.
- **Galga extensiométrica.** Denominado también extensómetro, es un sensor que mide la deformación, la presión o la carga basado en el efecto piezorresistivo.
- **Giroscopio.** Dispositivo mecánico diseñado para medir, mantener o cambiar la orientación de algún aparato asociado, como un exoesqueleto.

- **Grado de libertad (DOF).** Está referido al número de parámetros (posición, rotación...) que son necesarios para definir los movimientos posibles de un mecanismo o sistema.
- **Marcha.** Equivalente a la deambulaci3n, se refiere a la acci3n de una persona de caminar.
- **Microcontrolador.** Se trata de un circuito integrado programable que es capaz de cumplir unas tareas espec3ficas ya programadas en su memoria.
- **Momento.** Magnitud vectorial referida como el producto del vector de posici3n de la aplicaci3n de una fuerza por ese vector de fuerza. Tambi3n es llamado momento dinámico o momento de fuerza.
- **Paciente.** En este documento, persona con capacidades motoras reducidas que no es capaz de realizar todos los movimientos convencionales de forma fluida y sin ayudas, independientemente de sus dolencias o su edad.
- **Piezorresistividad.** Propiedad de algunos materiales, conductores y semiconductores, cuya resistencia var3a cuando se les somete a una cierta presi3n o esfuerzo que los deforma.
- **Placa.** Las placas PCB son lámimas conductoras de cobre que se utilizan para construir circuitos con una fiabilidad mayor que en una placa *proto board*. Están compuestas de v3as o pistas que recorren la placa de forma horizontal, con agujeros a lo largo de todas estas pistas donde soldar los componentes.
- **Sensor.** Dispositivo que se ocupa de detectar cambios físicos o químicos y transformarlos en magnitudes eléctricas. La variable que recoge puede ser de diferentes características: temperatura, distancia, humedad, intensidad lumínica, movimiento, presi3n...
- **Usuario.** Persona con cualquier capacidad que utiliza un objeto o servicio. En este documento, se diferencia del término paciente en que no se discierne su capacidad motora, mientras que un paciente sí necesita de ayuda externa.