



Universidad de Valladolid



**ESCUELA DE INGENIERÍAS
INDUSTRIALES**

UNIVERSIDAD DE VALLADOLID

ESCUELA DE INGENIERIAS INDUSTRIALES

Grado en Ingeniería en Tecnologías Industriales

Diseño de una plataforma movida por cuatro motores paso a paso controlados mediante ordenador

Autor:

Muñoz Martín, Alberto

Tutor(es):

**Parra Gonzalo, Eduardo Pedro
Departamento de Ingeniería
Eléctrica**

Valladolid, Julio de 2025.

Resumen

En este proyecto se pretende diseñar una plataforma móvil basada en el control de cuatro motores paso a paso mediante una tarjeta Raspberry Pi y LabVIEW.

La plataforma funcionará independientemente del ordenador usado para su programación, contando con alimentación propia mediante baterías y autonomía en el control dada la funcionalidad de la tarjeta Raspberry de cargar programas de LabVIEW en su memoria interna. Solo se interactuará con la plataforma mediante pulsadores físicos durante su funcionamiento.

Se llevará a cabo el diseño tanto de la plataforma, como de los circuitos necesarios para su funcionamiento, y un programa con varias trayectorias ejemplo.

Palabras Clave: Motor, Paso, LabVIEW, Raspberry

Abstract

The purpose of this project is the design of a mobile platform based on control of four stepper motors through a Raspberry Pi card and LabVIEW.

The platform will function independently of the computer used for its programming, having its own power supply through batteries and autonomy on control given the Raspberry's capability of loading LabVIEW programs to its internal memory. The platform will only be interacted with through physical push buttons while functioning.

The platform will be designed, as well as the circuits required for its operation, and a program with multiple example trajectories.

Keywords: Motor, Step, LabVIEW, Raspberry

Índice

1. Introducción y Objetivos	1
2. Descripción Teórica.....	1
2.1 Motores Paso a Paso	1
2.1.1 Motor paso a paso de imán permanente:	2
2.1.2 Motor de reluctancia variable:	3
2.1.3 Motor paso a paso híbrido:.....	4
2.1.4 Motores paso a paso Multi-stack.....	5
2.1.5 Motores paso a paso según su cableado.....	6
2.1.6 Control de motores paso a paso	7
2.2 Raspberry Pi.....	8
2.3 LabVIEW	9
2.4 KiCad	12
3. Desarrollo del TFG.....	15
3.1 Hardware.....	15
3.1.1 Raspberry Pi 4:	15
3.1.2 Driver DRV8833:	17
3.1.3 Motor SL42STH40-1684A-23:	19
3.1.4 Circuito de control:	19
3.1.5 Piezas de Impresión 3D:.....	21
3.1.6 Montaje Plataforma:	25
3.2 Software	27
3.2.1 LabVIEW 2025 Q1 (32bit):	27
3.2.2 Programa de control:	29
4. Presupuesto.....	33
5. Conclusiones	35
6. Bibliografía	37
Anexo 1: Hojas de datos	39
1.1 DRV8833 Dual H-Bridge Motor Driver Datasheet.....	39
1.2 SL42STH40-1684A-23 Datasheet.....	39
Anexo 2: Planos.....	39
1.1 Plano frontal circuito	39

1.2 Plano trasero circuito	39
1.3 Plano engranaje motor.....	39
1.4 Plano parte superior eje.....	39
1.5 Plano parte inferior eje.....	39
1.6 Plano clavija rueda	39
1.7 Plano sujeción rodamiento	39
1.8 Plano receptáculo circuito	39
1.9 Plano marco circuito.....	39
1.10 Plano sujeción Raspberry	39
Anexo 3: Código LabVIEW	39

Índice de figuras

Figura 1: Melkebeek (2018). Electrical Machines and Drives: Fundamentals and Advanced Modelling.	2
Figura 2: Melkebeek (2018). Electrical Machines and Drives: Fundamentals and Advanced Modelling.	3
Figura 3: Melkebeek (2018). Electrical Machines and Drives: Fundamentals and Advanced Modelling.	4
Figura 4: Melkebeek (2018). Electrical Machines and Drives: Fundamentals and Advanced Modelling.	5
Figura 5: Motores Unipolares. Recuperado el 17 de junio de 2025 de https://knowledge.ni.com/KnowledgeArticleDetails?id=kA00Z000000PAkPSAW&l=es-ES	6
Figura 6: Motor Bipolar. Recuperado el 17 de junio de 2025 de https://knowledge.ni.com/KnowledgeArticleDetails?id=kA00Z000000PAkPSAW&l=es-ES	7
Figura 7: Raspberry PI 4 Model B. Recuperado el 17 de Junio de 2025 de https://www.raspberrypi.com/products/raspberry-pi-4-model-b/	8
Figura 8: Panel frontal de un VI. Fuente: Elaboración propia	9
Figura 9: Diagrama de bloques de un VI. Fuente: Elaboración propia.....	10
Figura 10: VI Package Manager. Fuente: Elaboración propia.....	11
Figura 11: LabVIEW Hobbyist Toolkit. Reuperado el 17 de Junio de 2025 de https://www.ni.com/es/support/downloads/tools-network/download.labview-hobbyist-toolkit.html#559138	11
Figura 12: Hobbyist Toolkit Target Configuration. Fuente: Elaboración propia	12
Figura 13:. Editor de esquema KiCad. Fuente: Elaboración Propia	12
Figura 14: Editor de placas KiCad. Fuente: Elaboración propia	13
Figura 15: Editor de huellas KiCad. Fuente: Elaboración propia.....	14
Figura 16: Raspberry Pi Imager. Fuente: Elaboración propia	15
Figura 17: Pinout Raspberry Pi 4 Model B. Recuperado el 17 de Junio de 2025 de https://pinout.xyz	16
Figura 18: DRV8833. Recuperado el 17 de Junio de 2025 de https://www.ti.com/product/es-mx/DRV8833	17
Figura 19: Pinout Controlador. Fuente: Elaboración propia.....	18
Figura 20: Motor SL42STH40-1684A-23. Fuente: Elaboración Propia	19
Figura 21: Circuito de control. Fuente: Elaboración propia	19
Figura 22: Diagrama circuito de control. Fuente: Elaboración propia.....	20
Figura 23: Engranaje motor. Fuente: Elaboración propia	21
Figura 24: Eje ruedas. Fuente. Elaboración propia	22
Figura 25: Sujeción eje. Fuente: Elaboración propia.....	22
Figura 26: Sujeción rodamientos. Fuente: Elaboración propia	23
Figura 27: Receptáculo circuito. Fuente: Elaboración propia.....	23

Figura 28: Marco Superior. Fuente: Elaboración Propia	23
Figura 29: Soporte Raspberry. Fuente: Elaboración propia	24
Figura 30: Acople de motores. Fuente: Elaboración propia	25
Figura 31: Acople de ruedas. Fuente: Elaboración propia	25
Figura 32: Plataforma montada. Fuente: Elaboración propia	26
Figura 33: Plataforma cableada. Fuente: Elaboración propia	26
Figura 34: Proyecto LabVIEW. Fuente: Elaboración propia	27
Figura 35: Proyecto con VI creado. Fuente: Elaboración propia	28
Figura 36: Build Specifications. Fuente: Elaboración Propia	28
Figura 37: Trayectoria hacia adelante. Fuente: Elaboración propia	29
Figura 38: Trayectoria hacia atrás. Fuente: Elaboración propia	30
Figura 39: Lectura de pulsadores. Fuente: Elaboración propia	31
Figura 40: Programa completo. Fuente: Elaboración propia	31
Figura 41: Panel frontal programa. Fuente: Elaboración propia	32

1. Introducción y Objetivos

En este trabajo se busca diseñar y controlar una plataforma móvil con la intención de visualizar de manera didáctica el comportamiento y las posibles aplicaciones de los motores paso a paso.

Para conseguir esto, se busca que la plataforma no requiera cables externos de ningún tipo para facilitar su movimiento, y se controlará mediante un programa simple de fácil comprensión. Se programarán varias trayectorias de ejemplo, siendo estas seleccionadas mediante pulsadores cuando la plataforma esté en funcionamiento. El objetivo didáctico también se tendrá en cuenta en el montaje, dejando claramente visibles y marcadas las conexiones de los motores.

2. Descripción Teórica

2.1 Motores Paso a Paso

Los motores paso a paso se caracterizan por desplazarse en incrementos angulares discretos al ser energizadas sus bobinas, permitiendo un control fino de su giro y posición. Físicamente se caracterizan por contar con varias bobinas en el estátor, y material magnético tanto en el estátor como en el rotor.

En general, el estator está formado por dientes que tienen bobinas enrolladas y el rotor puede ser un imán permanente o un núcleo de hierro de reluctancia variable

Los bobinados se alimentan uno a continuación del otro, arrastrando el rotor de una posición a la siguiente, causando un desplazamiento angular pequeño, el cual depende del paso del motor. La dirección del giro depende de la secuencia de alimentación de las bobinas.

Estos motores también se pueden operar en medios pasos, es decir, no se retira la corriente de la primera bobina antes de energizar la siguiente, haciendo que el eje se sitúe en la posición media entre las dos bobinas. Esto divide el paso efectivo a la mitad, aunque requiere un control más fino de la alimentación al motor.

Estos motores son comúnmente clasificados por el tipo de rotor y el tipo de cableado de las bobinas.

2.1.1 Motor paso a paso de imán permanente:

Estos motores cuentan con un imán permanente en el rotor, el cual busca alinearse con el campo magnético de las bobinas al ser estas energizadas. El imán permanente cuenta solo con dos polos, como consecuencia de esta estructura el número de pasos por vuelta del motor dependerá solamente del número de polos del estator, sin embargo, al contar con un campo magnético permanente, el motor tiende a mantener su posición una vez retirada la corriente de las bobinas.

Estos motores son muy comunes debido a su bajo coste, sin embargo, sus desventajas comparado con otros tipos de motor paso a paso es la baja velocidad y menor resolución.

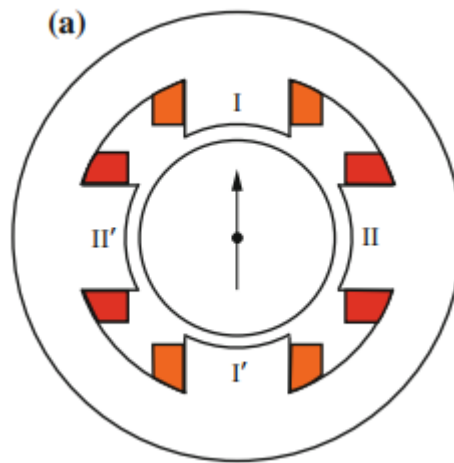


Figura 1: Melkebeek (2018). *Electrical Machines and Drives: Fundamentals and Advanced Modelling*.

Ya que el campo magnético del rotor es constante, la capacidad de invertir la polaridad de los polos del estator es obligatoria. Hay dos estructuras comunes en este tipo de motor, la primera una estructura unifilar en la que cada polo del estator está formado por una sola bobina, requiriendo de una fuente de alimentación capaz de invertir su polaridad, y la segunda una estructura bifilar, en la cual cada polo dispone de dos bobinas y por lo tanto el cambio de polaridad en ellos se puede realizar con un interruptor simple sin una fuente capaz de invertir su polaridad. [1]

La estructura bifilar es más simple de controlar, pero requiere más cobre, aumentando el precio y reduciendo la densidad de potencia del motor. [1]

2.1.2 Motor de reluctancia variable:

Al contrario de los motores de imán permanente, estos motores cuentan con un rotor formado por un material magnéticamente blando cuya geometría le permite alinearse con los campos magnéticos generados por el estator. Al energizar las bobinas del estator, el rotor tiende a reducir la permeabilidad magnética, girando para acercar los dientes del rotor a los polos de las bobinas estáticas.

En general puede alcanzar velocidades y resoluciones mayores que los motores de imán permanente, pero la ausencia del campo magnético del eje resulta en la incapacidad de mantener su posición una vez eliminada la corriente de las bobinas. Además, generalmente desarrollará un par inferior al motor de imán permanente.

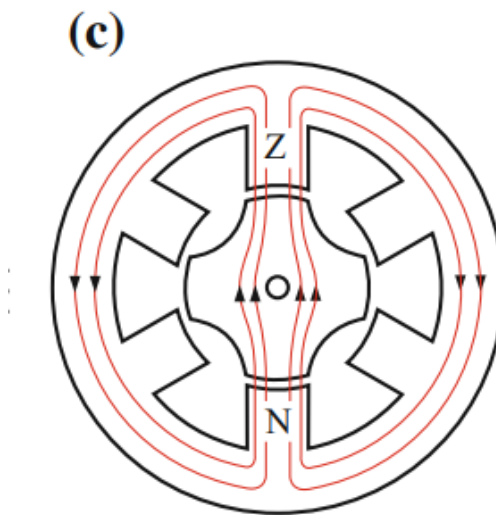


Figura 2: Melkebeek (2018). *Electrical Machines and Drives: Fundamentals and Advanced Modelling*.

Ya que el núcleo de rotor no presenta un campo magnético permanente, la polaridad los polos del estator es indiferente. Esto significa que se puede manipular estos motores sin una fuente capaz de invertir su polaridad. Además, ya que el rotor no posee solo dos polos es posible alterar el paso del motor cambiando tanto el número de polos del estator como el número de polos del rotor. El número de dientes del estator afectará al par y resolución, pero no influye en el paso del motor. [1]

2.1.3 Motor paso a paso híbrido:

Estos motores son una variante de los motores de reluctancia variable que busca eliminar la necesidad de un sistema mecánico para mantener la posición del eje una vez eliminada la corriente. Para conseguirlo, se añade un imán permanente al rotor orientado axialmente. Cuando se energiza una bobina, el polo sur del rotor se es atraído al polo norte del estator, y el sur del estator atrae el polo norte del rotor.

Esto consigue conservar las ventajas de ambos motores, manteniendo la posición en desconexión y permite variar los pasos por vuelta modificando la geometría del rotor.

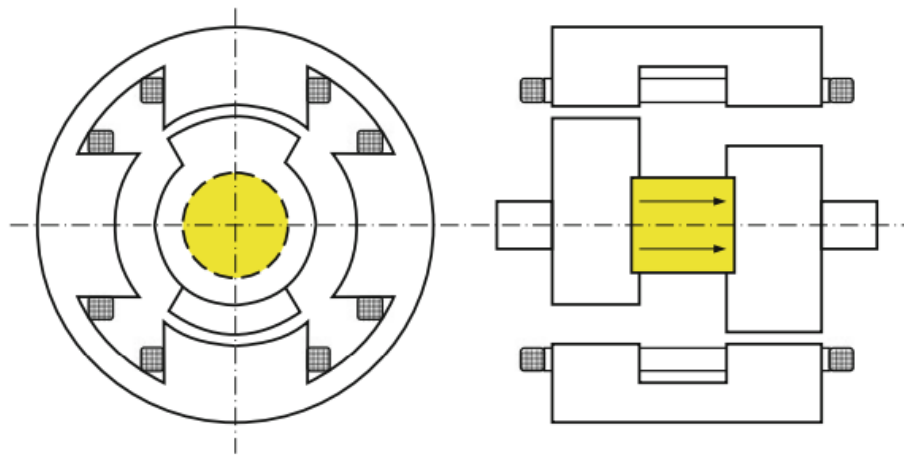


Figura 3: Melkebeek (2018). *Electrical Machines and Drives: Fundamentals and Advanced Modelling*.

Estos motores requieren el uso de alimentación bipolar o unifilar como es el caso en los de imán permanente. También es importante en la estructura que los dientes del polo norte del rotor y los del polo sur estén desfasados en ángulo, en ciertos casos se desfasan los dientes del estator en lugar del rotor. Aunque el imán permanente es crucial para generar un par, el paso del motor dependerá de la estructura dentada del rotor y estator. Los distintos polos del estator deben tener una posición relativa al rotor distinta. De no ser esto así, al alinearse el rotor con un polo del estator, se alinearían todos ellos y el motor no se movería al energizar las bobinas. [1]

2.1.4 Motores paso a paso Multi-stack

Todos los motores mencionados previamente tienen una sola capa o stack. En tales motores el comportamiento magnético es igual en toda la longitud del motor.

Otra estructura posible para los motores paso a paso es la estructura multi-stack, o estructura cascada. En estos motores cada capa tiene un circuito ferromagnético propio, y cada capa corresponde a una fase del estator.

Hay dos tipos de motores multi-stack. Los motores homopolares, en los que cada capa contiene dos planos, uno para los polos sur y otro para los polos norte, y los motores heteropolares, en los que cada capa tiene sus polos norte y sur en el mismo plano de manera alternada.

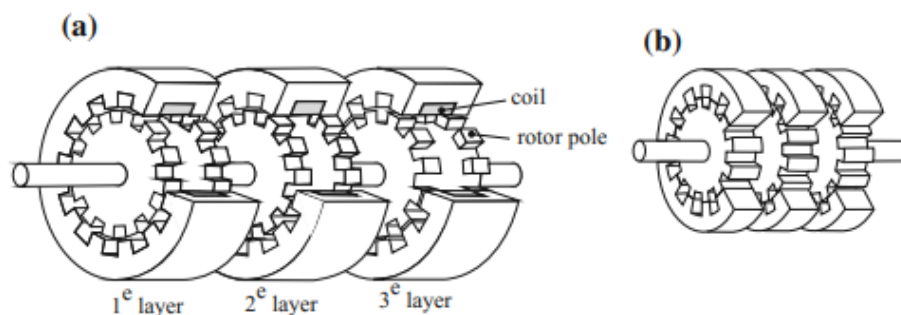


Figura 4: Melkebeek (2018). *Electrical Machines and Drives: Fundamentals and Advanced Modelling*.

Los motores homopolares funcionan de manera similar a los motores híbridos y pueden alimentarse con una bobina simple en cada capa, mientras que los heteropolares requieren una bobina en cada polo, dificultando la conexión y requiriendo más espacio. [1]

El rotor de este tipo de motores puede o no contener un imán permanente, cambiando las necesidades de la alimentación. [1]

2.1.5 Motores paso a paso según su cableado

Independientemente de la estructura del eje, los motores paso a paso se pueden clasificar según el número de hilos usados para alimentarlos.

En su mayoría los motores se pueden dividir en dos categorías, motores unipolares y motores bipolares.

Los motores unipolares disponen de un cable en la mitad de la bobina que puede ser usado como alimentación o como tierra. En este caso, energizar la bobina requiere solamente la conexión de uno de sus extremos, ya sea a alimentación, si el cable intermedio es tierra, o a tierra si el cable intermedio es alimentación.

Su control es más sencillo, sin embargo, ya que solo se utiliza la mitad de la bobina, el campo magnético generado es más débil. La secuencia de operación de estos motores sería A+, B+, A-, B- en un sentido y A+, B-, A-, B+ en el sentido opuesto.

Estos motores pueden presentar 5, 6 y 8 hilos externos dependiendo si la toma intermedia de ambas bobinas esta interconectada (5), si cada una tiene un hilo (6) o si cada media bobina cuenta con su hilo propio (8). Un motor de 8 hilos de imán permanente correspondería a la estructura bifilar mencionada anteriormente. [2]

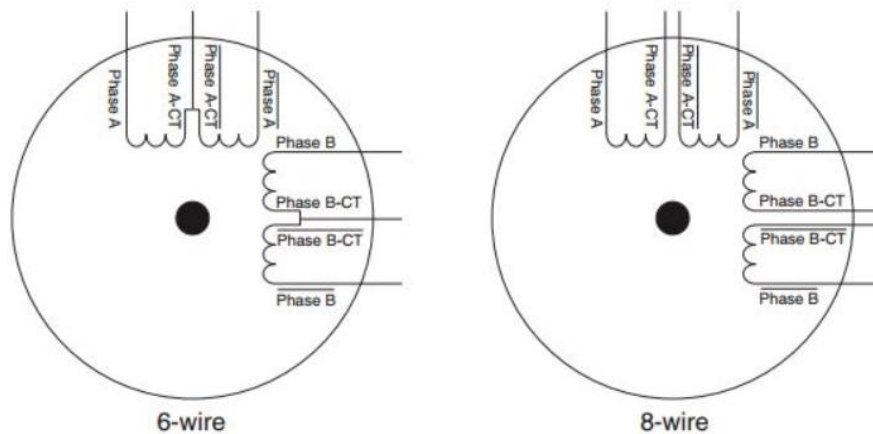


Figura 5: Motores Unipolares. Recuperado el 17 de junio de 2025 de <https://knowledge.ni.com/KnowledgeArticleDetails?id=kA00Z000000PAkPSAW&l=es-ES>

Los motores bipolares no disponen de una toma intermedia en sus bobinas, aprovechando su longitud entera para generar el campo magnético. Esto significa que para controlarlos es necesario invertir la polaridad de las bobinas, complicando su manejo. La secuencia de operación de estos motores sería igual a la de los motores unipolares. [2]

Estos motores solo presentan 4 hilos externos y corresponderían con la estructura unifilar mencionada anteriormente.

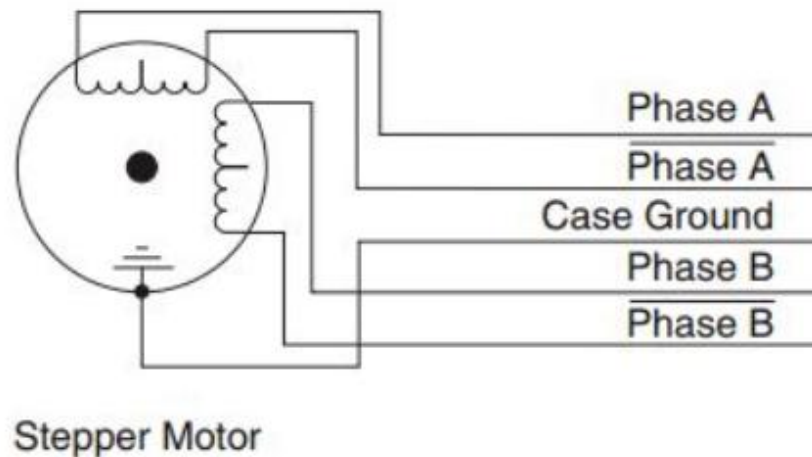


Figura 6: Motor Bipolar. Recuperado el 17 de junio de 2025 de <https://knowledge.ni.com/KnowledgeArticleDetails?id=kA00Z000000PAkPSAW&l=es-ES>

2.1.6 Control de motores paso a paso

Ya que los motores paso a paso requieren un control basado en pulsos de alimentación, se recurre a circuitos integrados como controladores.

Hay varios tipos de controladores para motores paso a paso en el mercado, dependiendo del tipo de motor a controlar y como que quiera operar. Algunos tipos comunes son.

- **Step/Direction:** Este tipo de controlador dispone de un pin de dirección y un pin de paso. Al mandar un pulso al pin de paso, el motor avanzará en la dirección determinada por el pin de dirección. [3]
- **Phase/Enable:** Este tipo de controlador dispone de dos entradas. La entrada de fase, una por cada bobina, determina la dirección de la corriente en la bobina conectada y la entrada de activación da la orden de alimentar dicha bobina. El controlador usado en este proyecto es de este tipo. [3]
- **PWM:** Estos controladores manipulan el voltaje en las bobinas mediante modulación de pulsos, pudiendo alimentarlas con diferentes voltajes efectivos con un solo voltaje nominal de entrada. [3]

Los controladores también se pueden clasificar dependiendo de su capacidad para limitar voltaje o intensidad.

Un controlador que solo limita voltaje hará funcionar al motor siempre en estado nominal, por lo tanto, su par y velocidad dependerán solo de las características del motor. [3]

Un controlador más avanzado podría limitar la intensidad, lo que permite hacer un control dinámico del par y el sistema. [3]

2.2 Raspberry Pi

La tarjeta Raspberry Pi es un ordenador monoplaca diseñado en su origen como plataforma didáctica para la enseñanza de informática. Sin embargo, debido a su software libre y cómodo uso, se ha popularizado como controlador en proyectos de robótica y automatismo. Tiene una interfaz extensa y varios modelos con distintas capacidades de memoria RAM.

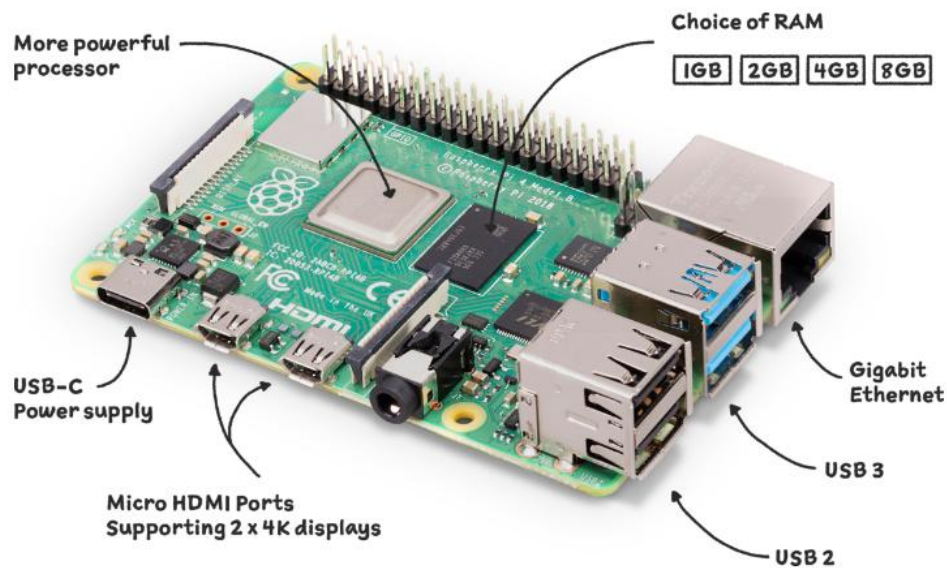


Figura 7: Raspberry Pi 4 Model B. Recuperado el 17 de Junio de 2025 de <https://www.raspberrypi.com/products/raspberry-pi-4-model-b/>

Al tratarse de un ordenador completo con sistema operativo basado en Linux, Raspberry Pi puede ejecutar directamente programas de LabVIEW sin compilarlos en un lenguaje propio como sería necesario en las tarjetas Arduino.

El control de los motores se realizará mediante los pines de E/S (I/O) de la tarjeta, utilizando módulos de control para traducir la señal digital a una alimentación a las bobinas correctas con polaridad apropiada.

2.3 LabVIEW

LabVIEW es un entorno de programación gráfico diseñado para brindar la funcionalidad de una programación compleja a cualquier usuario, sin ser necesarios conocimientos de programación en lenguajes de menor nivel.

Su principal enfoque era el control de instrumentación, sin embargo, con el tiempo se ha expandido y actualizado su funcionalidad, permitiendo controlar y programar cualquier equipo electrónico compatible.

El entorno se centra en dos ventanas distintas, una el panel frontal del programa y la otra el diagrama de bloques que describe su comportamiento:

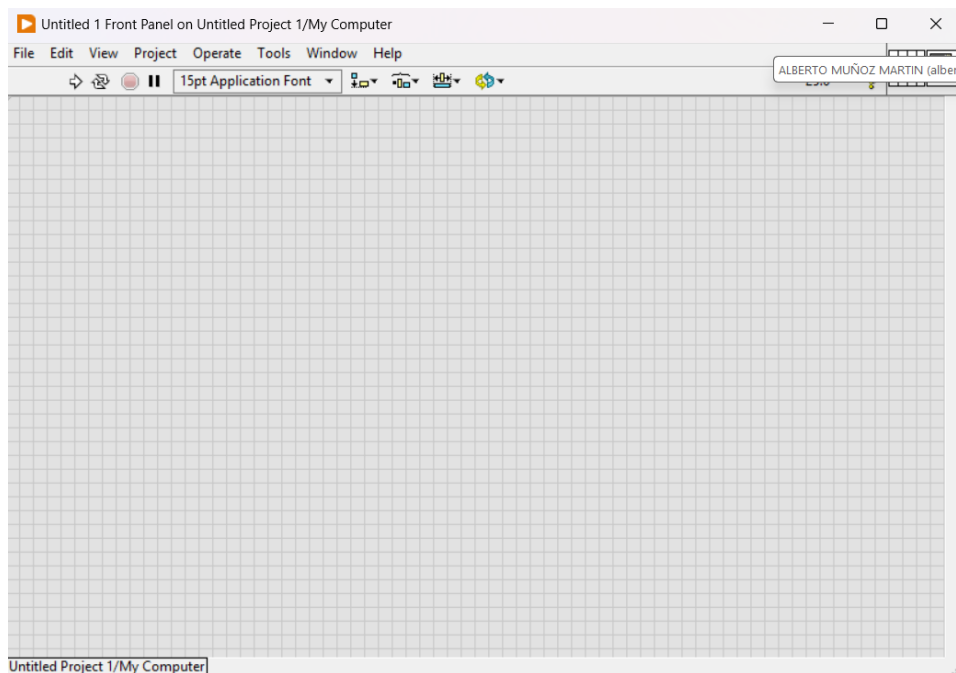


Figura 8: Panel frontal de un VI. Fuente: Elaboración propia

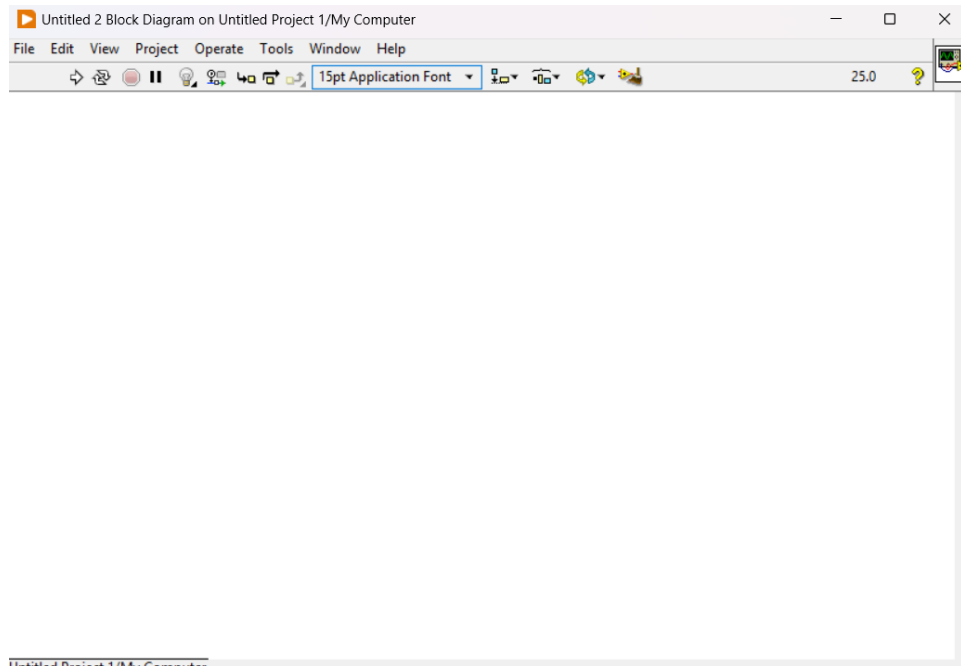


Figura 9: Diagrama de bloques de un VI. Fuente: Elaboración propia

En el diagrama de bloques se realizará la programación, mientras que en el panel frontal se podrá interactuar con los indicadores y las entradas virtuales del sistema.

La programación se realiza con elementos discretos unidos por líneas de proceso, siendo los bucles representados por marcos que envuelven a los bloques funcionales. Todos estos componentes se encuentran en una librería interna, accesible en todo momento.

El entorno cuenta con multitud de elementos base, y además permite la instalación de librerías y añadidos mediante el software VI Package Manager, siendo VI (Instrumento Virtual) la designación de los programas realizados en LabVIEW.

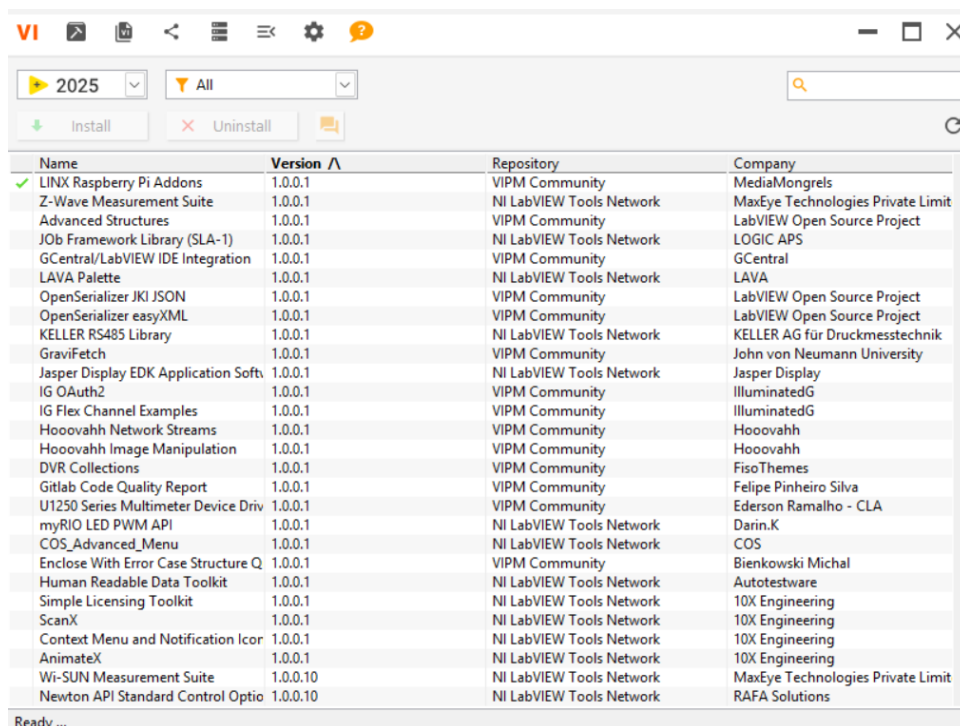


Figura 10: VI Package Manager. Fuente: Elaboración propia

Para la comunicación con la tarjeta Raspberry Pi han existido múltiples soluciones dependiendo de la versión de LabVIEW, generalmente mediante librerías, tanto oficiales como de terceros. Para realizar esta comunicación con LabVIEW 2025 Q1 es necesario instalar el LabVIEW Hobbyist Toolkit, siendo este una versión actualizada de la librería LINX utilizada en versiones anteriores. [4]



Figura 11: LabVIEW Hobbyist Toolkit. Recuperado el 17 de Junio de 2025 de <https://www.ni.com/es/support/downloads/tools-network/download.labview-hobbyist-toolkit.html#559138>

Este toolkit introduce una herramienta de conexión con tarjetas Raspberry Pi y Beaglebone, la cual permite instalar la versión apropiada de LabVIEW en la tarjeta para poder comunicarse con ella y ejecutar los programas de manera nativa. Además, incluye nuevos bloques funcionales en el entorno de desarrollo relacionados con las tarjetas conectadas. [4]

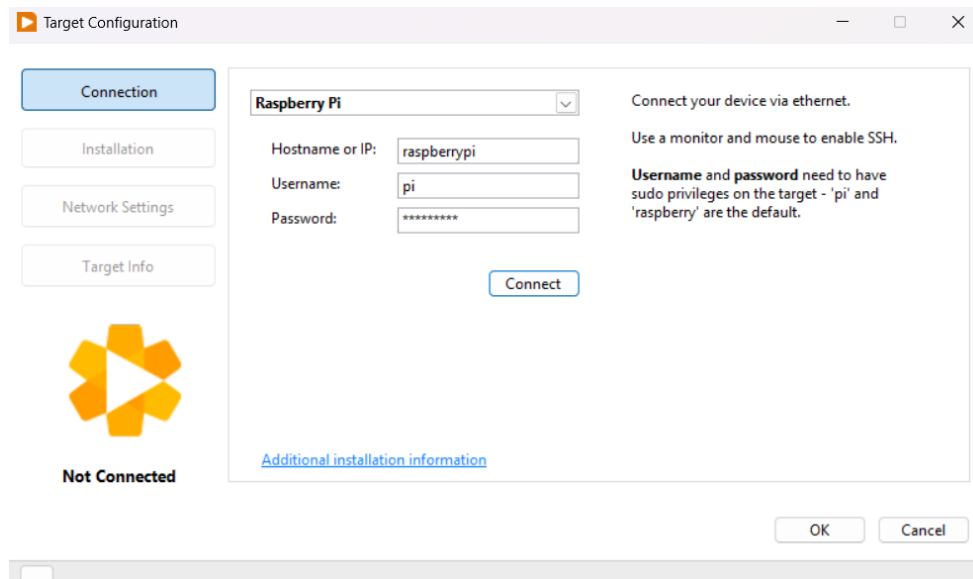


Figura 12: Hobbyist Toolkit Target Configuration. Fuente: Elaboración propia

2.4 KiCad

KiCad es un programa libre de diseño de placas de circuito impreso de fácil manejo. Al empezar un proyecto, se crean automáticamente un archivo de esquema y un archivo de placa. En el esquema se añaden y conectan los elementos para asegurarse de que el circuito es funcional y las conexiones son correctas.

El símbolo utilizado en el esquema no tiene por qué coincidir con el elemento real ya que se trata de un esquema orientativo. La asociación del componente con sus dimensiones reales se realiza asignando una huella a dicho componente. [5]

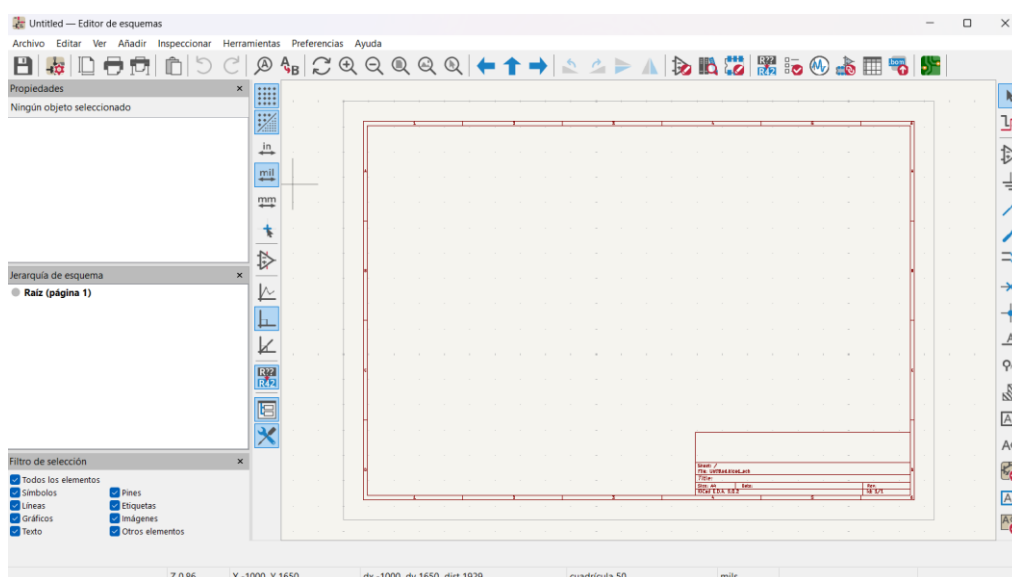


Figura 13: Editor de esquema KiCad. Fuente: Elaboración Propia

Una vez creado el esquema, se pueden importar los componentes directamente a la placa, y quedan introducidos con sus dimensiones y conectores reales asociados a su huella para poder ser colocados y orientados en el espacio. [5]

Una vez organizados, el programa dispone de marcas indicando las conexiones realizadas en el esquema como guía para crear las trazas. Las trazas se crean automáticamente siguiendo una rejilla de proporciones constantes, con un espacio de seguridad en su entorno para evitar cortocircuitos y trayectorias que evitan esquinas rectas. [5]

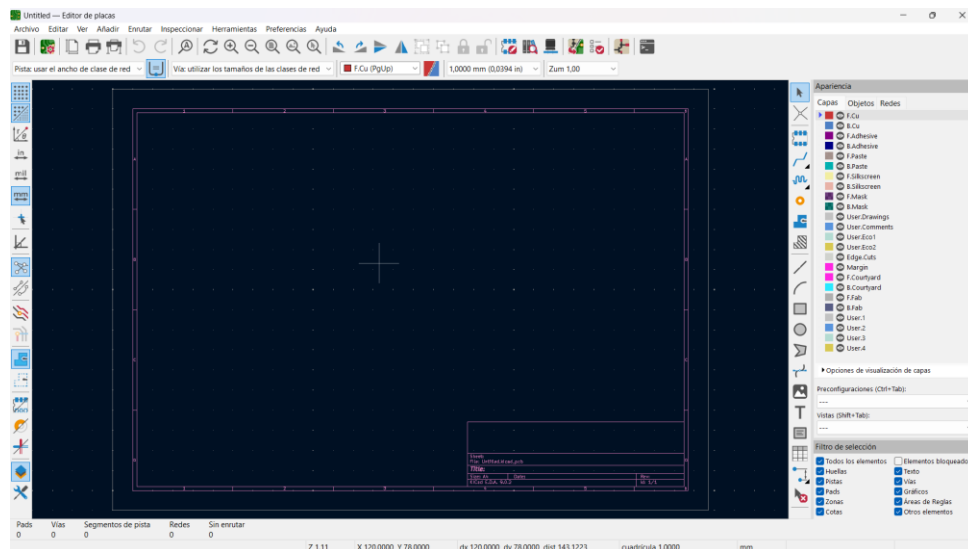


Figura 14: Editor de placas KiCad. Fuente: Elaboración propia

Todos los componentes salvo los controladores se encontraban en los catálogos base que vienen instalados en el programa.

Para los controladores se ha creado una huella tomando las medidas reales de los módulos con un calibre. El editor de huellas permite crear componentes con cualquier geometría, primero se añaden los conectores, ya sean superficiales o atraviesen la placa, y luego se marca la forma exterior del módulo. Rodeando al módulo se añade un perímetro que indicará su lugar en la placa, y alrededor de este perímetro, un área libre para asegurar que los elementos no harán colisión con otros al ser montados en la placa. [5]

Como norma se suele crear el pin 1 en el centro de coordenadas. [5]

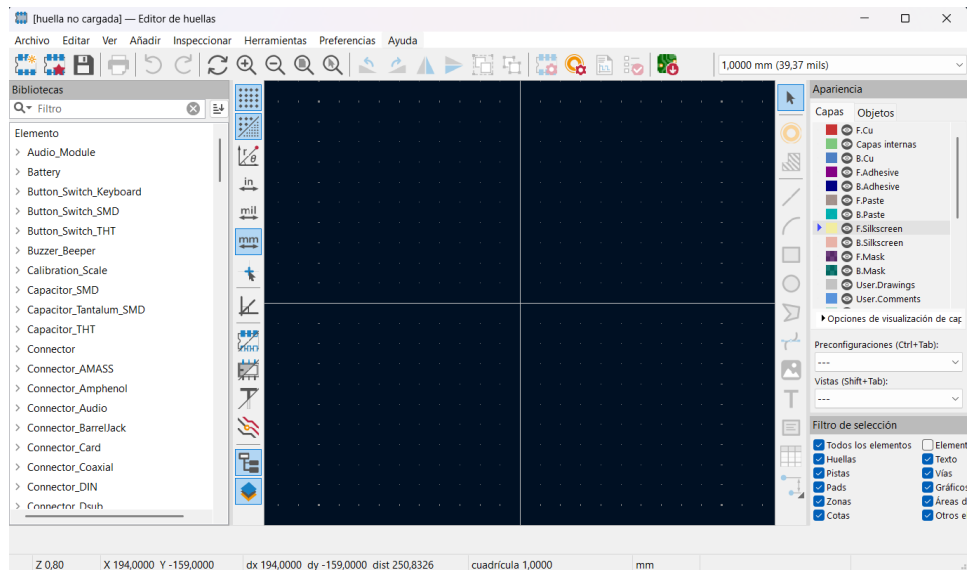


Figura 15: Editor de huellas KiCad. Fuente: Elaboración propia

3. Desarrollo del TFG

3.1 Hardware

3.1.1 Raspberry Pi 4:

Para el desarrollo de este proyecto se ha utilizado una tarjeta Raspberry Pi 4 Model B de 2GB de memoria RAM.

Al tratarse de un ordenador Linux, se ha instalado un sistema operativo libre de forma manual, en este caso el sistema Raspberry OS creado y puesto a disposición por el fabricante. La compatibilidad con LabVIEW ha requerido el uso de la versión Legacy del sistema operativo, ya que las versiones actualizadas no eran reconocidas por la herramienta de conexión incorporada en el toolkit utilizado.

La instalación se ha realizado con la herramienta oficial de Raspberry, Raspberry Pi Imager. Esta herramienta obtiene automáticamente los archivos de Raspberry OS de internet, aunque también puede usarse para instalar otros sistemas operativos si se descargan previamente de manera manual. [6]

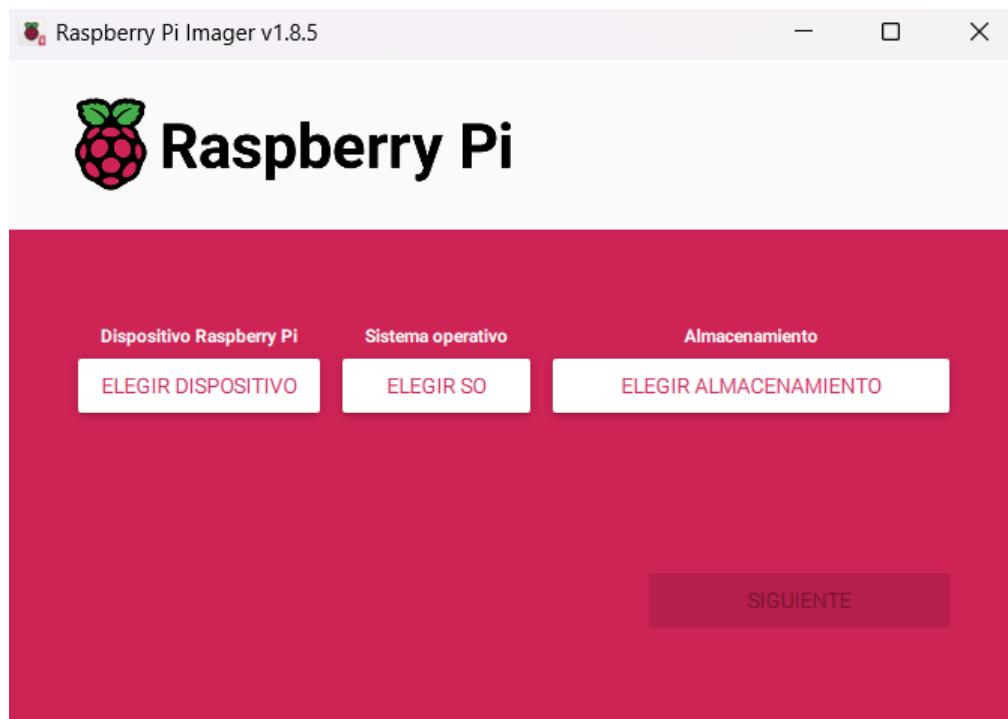


Figura 16: Raspberry Pi Imager. Fuente: Elaboración propia

El control de los motores se ha realizado mediante los pines de E/S (I/O) de la tarjeta. El modelo 4 B cuenta con 40 pines de uso general, sin embargo, 4 de ellos son pines de alimentación, 8 son pines de tierra, 2 están reservados (EEPROM) y no pueden manipularse y 15 están activados para protocolos de comunicación industriales por defecto (I2C, UART, SPI). [7]

3v3 Power	1		2	5v Power
GPIO 2 (I2C1 SDA)	3		4	5v Power
GPIO 3 (I2C1 SCL)	5		6	Ground
GPIO 4 (GPCLK0)	7		8	GPIO 14 (UART TX)
Ground	9		10	GPIO 15 (UART RX)
GPIO 17 (SPI1 CE1)	11		12	GPIO 18 (SPI1 CE0)
GPIO 27	13		14	Ground
GPIO 22	15		16	GPIO 23
3v3 Power	17		18	GPIO 24
GPIO 10 (SPI0 MOSI)	19		20	Ground
GPIO 9 (SPI0 MISO)	21		22	GPIO 25
GPIO 11 (SPI0 SCLK)	23		24	GPIO 8 (SPI0 CE0)
Ground	25		26	GPIO 7 (SPI0 CE1)
GPIO 0 (EEPROM SDA)	27		28	GPIO 1 (EEPROM SCL)
GPIO 5	29		30	Ground
GPIO 6	31		32	GPIO 12 (PWM0)
GPIO 13 (PWM1)	33		34	Ground
GPIO 19 (SPI1 MISO)	35		36	GPIO 16 (SPI1 CE2)
GPIO 26	37		38	GPIO 20 (SPI1 MOSI)
Ground	39		40	GPIO 21 (SPI1 SCLK)

Figura 17: Pinout Raspberry Pi 4 Model B. Recuperado el 17 de Junio de 2025 de <https://pinout.xyz>

Desactivando los protocolos I2C y SPI se puede conseguir acceso a 13 de los pines de comunicación industrial en la Raspberry, sin embargo, el Hobbyist Toolkit de LabVIEW no reconoce 7 (I2C, SPI0, UART) de ellos como manipulables, dejando un total de 17 pines útiles.

Debido a este limitado número de pines, se ha tomado la decisión de controlar los motores en grupos según el lado de la plataforma, utilizando 4 pines para cada 2 motores en lugar de uno. Esto reduce el número de drivers necesarios y permite utilizar más pines como entradas para el control de la plataforma. En este caso, 3 pines se utilizarán como entradas digitales permitiendo elegir la trayectoria a llevar a cabo. La alimentación de la Raspberry se llevará a cabo con una batería USB capaz de aportar los 3A necesarios para su operación.

3.1.2 Driver DRV8833:

Para traducir las salidas digitales de la Raspberry al control de los motores paso a paso se utilizará drivers basados en el circuito integrado DRV8833. Este circuito cuenta con 2 puentes H implementados con MOSFETs y permite generar dos salidas analógicas de polaridad reversible. El funcionamiento basado en MOSFETs hace que este circuito sea más eficiente energéticamente y genere menos calor que el circuito LN298, el cual es extremadamente común en aplicaciones de control de motores mediante Arduino o Raspberry. [8]

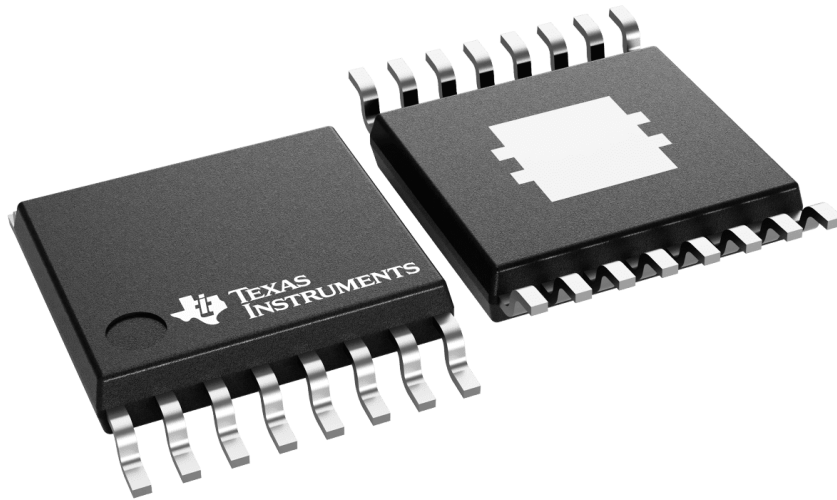


Figura 18: DRV8833. Recuperado el 17 de Junio de 2025 de <https://www.ti.com/product/es-mx/DRV8833>

Comercialmente es muy común encontrar controladores para motores paso a paso ya montados con una interfaz externa para no requerir conocimientos avanzados de micro soldadura o electrónica para hacer uso de estos circuitos.

Se ha utilizado uno de estos controladores prefabricados para el control de los motores, del tipo Phase/Enable, ya que requieren un control más directo y permiten observar el funcionamiento de los motores más claramente.

El controlador tiene 4 entradas digitales, 4 salidas analógicas, 1 pin de error, 1 pin de suspensión y dos pines para su alimentación. También cuenta con un saltador en su parte posterior. [8]

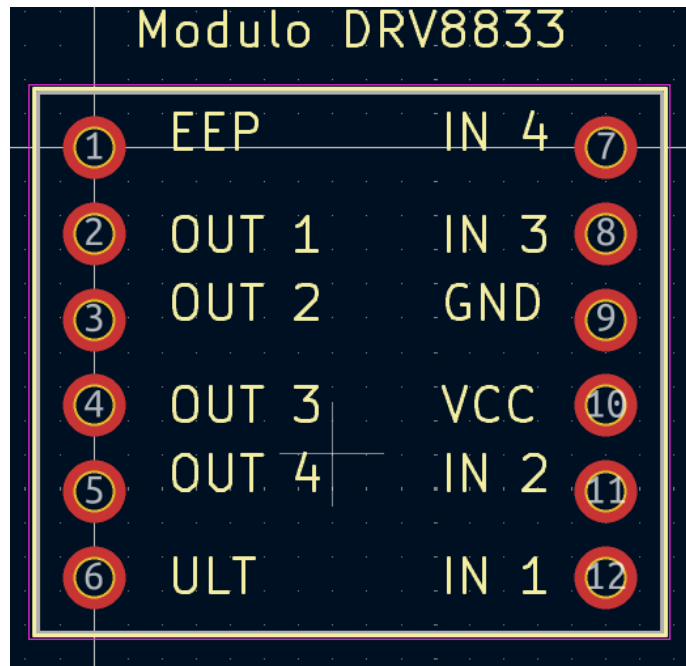


Figura 19: Pinout Controlador. Fuente: Elaboración propia

Las entradas 1 y 2 controlan las salidas 1 y 2. Activar la entrada 1 generará una corriente de la salida 1 a la 2, y activar la entrada 2 generará una corriente de la salida 2 a la 1. El funcionamiento es idéntico en las salidas y entradas 3 y 4. [8]

Ya que el controlador está en modo de suspensión por defecto, es necesario puentear el saltador J que se encuentra en su parte posterior para cambiarlo a activado por defecto y no requerir una entrada digital en el pin de suspensión para funcionar. [8]

Este controlador puede manejar 2 motores de corriente continua o 1 motor paso a paso de 4 hilos, como los que se han utilizado en este proyecto. Para realizar el control, es necesario energizar la primera bobina con una polaridad, luego la segunda bobina con esa misma polaridad, luego la primera bobina con la polaridad opuesta y finalmente la segunda bobina con la polaridad opuesta. [8]

Esto, asumiendo que la primera bobina está conectada a las salidas 1 y 2, y la segunda bobina a las salidas 3 y 4, se consigue activando las entradas en el orden 1-3-2-4 para girar en un sentido, o 1-4-2-3 para el sentido opuesto. [8]

3.1.3 Motor SL42STH40-1684A-23:

En este proyecto se han utilizado motores SL42STH40-1684A-23 rescatados de una impresora 3D.



Figura 20: Motor SL42STH40-1684A-23. Fuente: Elaboración Propia

Estos son motores NEMA 17 de 1,8° de paso. Funcionan a 3,6 V y un máximo de 1,6 A. Su estructura de 4 hilos hace posible controlarlos con el Driver escogido, además, su bajo consumo permite controlar más de uno con cada Driver. Se alimentarán con baterías de litio de 3,7V, una por cada motor.

3.1.4 Circuito de control:

El montaje de los componentes electrónicos necesarios se ha realizado sobre una placa base sencilla. Esta placa recibe las señales de la tarjeta Raspberry a través de un conector IDC de 40 pines.

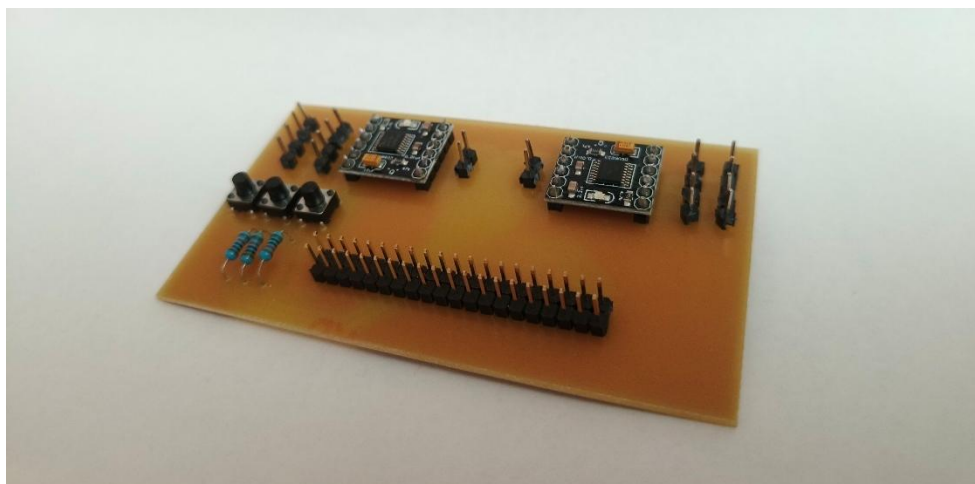


Figura 21: Circuito de control. Fuente: Elaboración propia

Como entradas se utilizarán botones simples conectados a pines en la tarjeta. Cada botón conectará un pin I/O (E/S) con un pin de alimentación de 3V3 en la tarjeta. Al presionar el botón, el pin de entrada pasará de tener un nivel de lógica bajo (0V) a uno alto (3V3) y se comportará como una entrada booleana. Por motivos de seguridad, es necesario añadir resistencias de 10k ohmios en serie a cada botón. [9]

El circuito dispone de 3 botones para seleccionar las trayectorias deseadas, las resistencias necesarias en serie con cada botón, y 2 módulos de control, uno para la derecha de la plataforma y otro para la izquierda. Cada módulo tiene dos pines para la alimentación, uno positivo y otro de tierra, y 8 pines de salida, conectados como 4 pares en serie. Cada módulo controlará 2 motores.

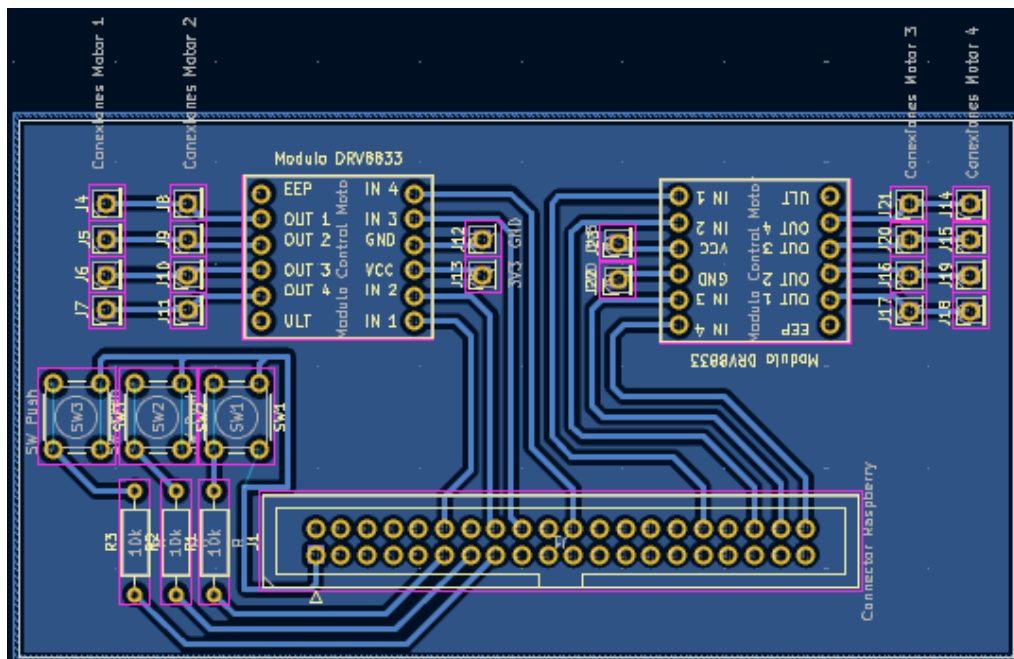


Figura 22: Diagrama circuito de control. Fuente: Elaboración propia

El diseño del circuito se ha realizado con el software KiCad, siendo necesario crear la huella de los controladores de los motores al no encontrarse esta en los catálogos base. El resto de los componentes son estándar y disponían de entradas en dichos catálogos.

3.1.5 Piezas de Impresión 3D:

Para comunicar la rotación de los ejes de los motores a los ejes de las ruedas ha sido necesario añadir componentes de fabricación propia.

Las piezas se han diseñado en Tinkercad, un programa online gratuito de uso simple. Su funcionalidad es bastante limitada, pero suficiente para las necesidades de este proyecto. Una vez diseñadas las piezas, sus archivos STL se han convertido a código G mediante Ultimaker Cura, un programa slicer de libre acceso.

En este caso, no todos los motores cuentan con engranaje propio en su eje, así que se ha recreado la pieza mediante impresión 3D de PLA para el resto de los motores.

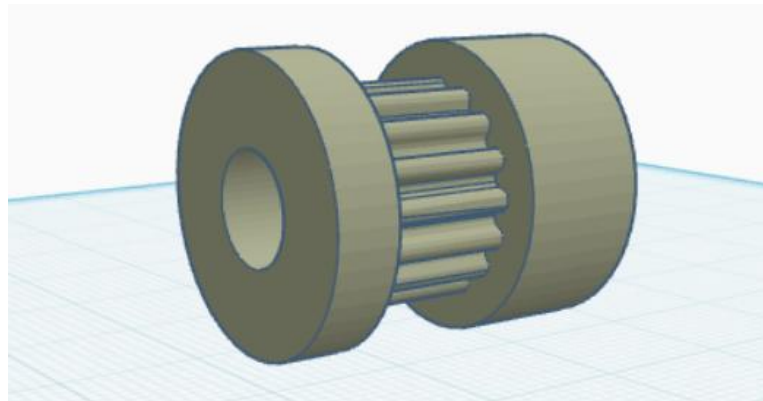


Figura 23: Engranaje motor. Fuente: Elaboración propia

También se han diseñado los ejes de las ruedas, ya que estas no tienen eje propio. Las ruedas escogidas tienen un acoplamiento hexagonal que hará que el eje fabricado se sujete firmemente a la rueda. Los rodamientos utilizados para unir los ejes a la plataforma son rodamientos normalizados 608ZZ, comunes para ruedas de patines y otros usos varios. Para que los ejes sean lo más cortos posible y queden bien sujetos, se harán en dos partes, dejando el engranaje entre los dos rodamientos. La pieza que encaje en los rodamientos se acoplará después a la pieza que tenga la cabeza hexagonal para conectar con la rueda.

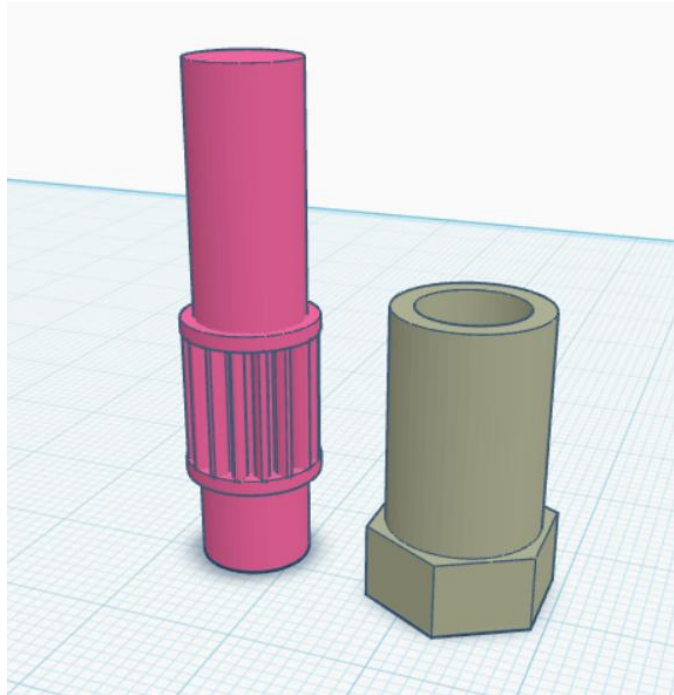


Figura 24: Eje ruedas. Fuente: Elaboración propia

Una pequeña pieza se insertará en la parte superior del eje a través de la rueda, asegurando su unión.

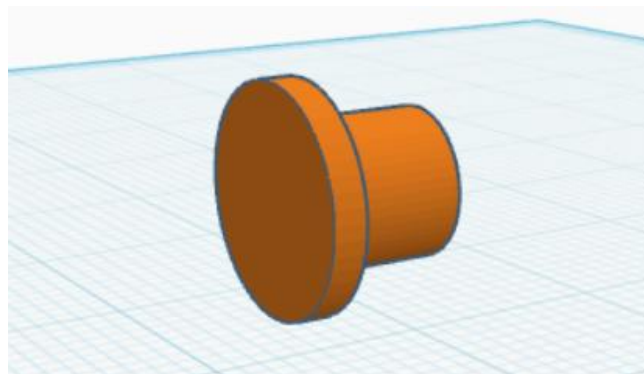


Figura 25: Sujeción eje. Fuente: Elaboración propia

La sujeción de los rodamientos a la plataforma también se hará con una pieza de impresión 3D. Las piezas permitirán insertar los rodamientos por uno de sus lados y quedarán sujetas a la plataforma con tornillos. Una vez montadas, dos de estas piezas estarán juntas para cada rueda, impidiendo que los rodamientos salgan de su posición.

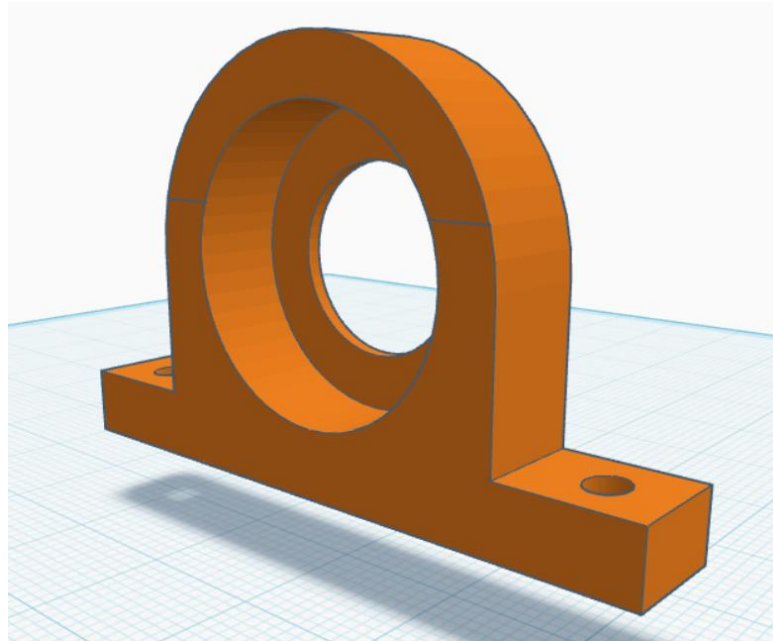


Figura 26: Sujeción rodamientos. Fuente: Elaboración propia

También se han diseñado sujeciones para el circuito de control y la Raspberry Pi, para que queden acopladas a la plataforma. En el caso del circuito la sujeción trata de un receptáculo inferior y un marco superior en el que el circuito queda colocado y sujeto.

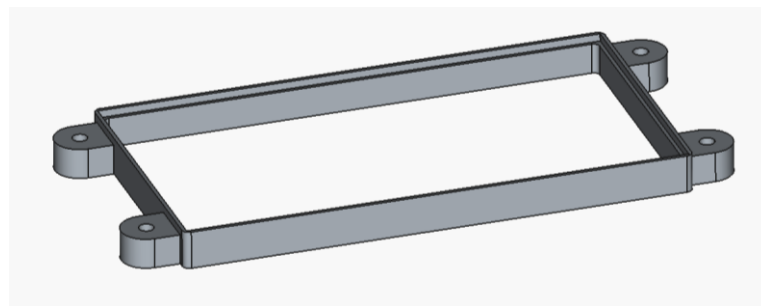


Figura 27: Receptáculo circuito. Fuente: Elaboración propia

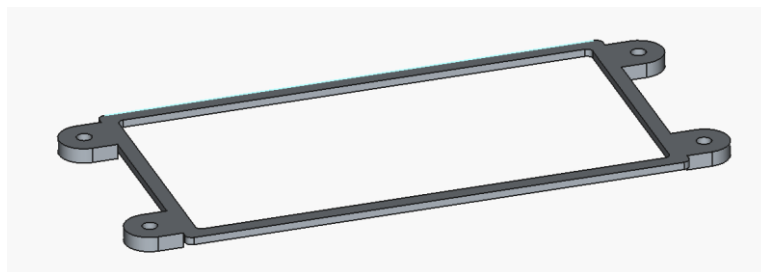


Figura 28: Marco Superior. Fuente: Elaboración Propia

Ya que la Raspberry cuenta con agujeros para su montaje de fábrica, solo se ha hecho un soporte que le aporta la separación necesaria de la plataforma.

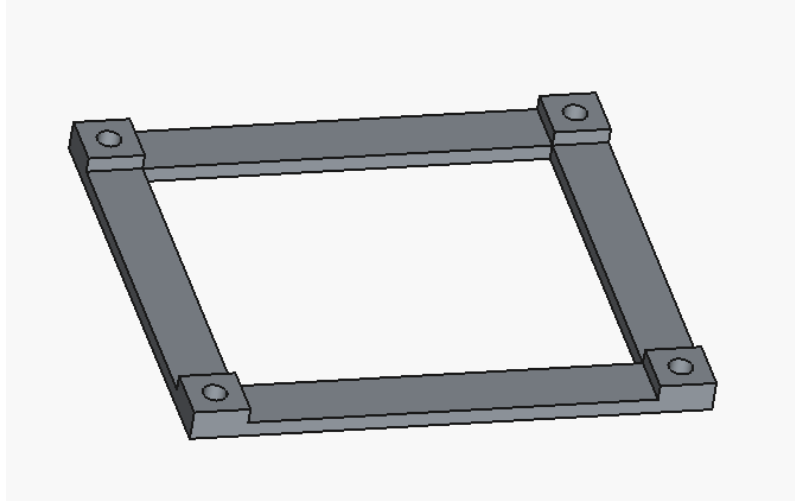


Figura 29: Soporte Raspberry. Fuente: Elaboración propia

3.1.6 Montaje Plataforma:

El montaje de la plataforma se ha realizado en un panel MDF de 1cm de grosor, de superficie igual a un formato A4. Los motores se han acoplado a la plataforma con bridas para evitar añadir material pesado o que ocupe mucho espacio.

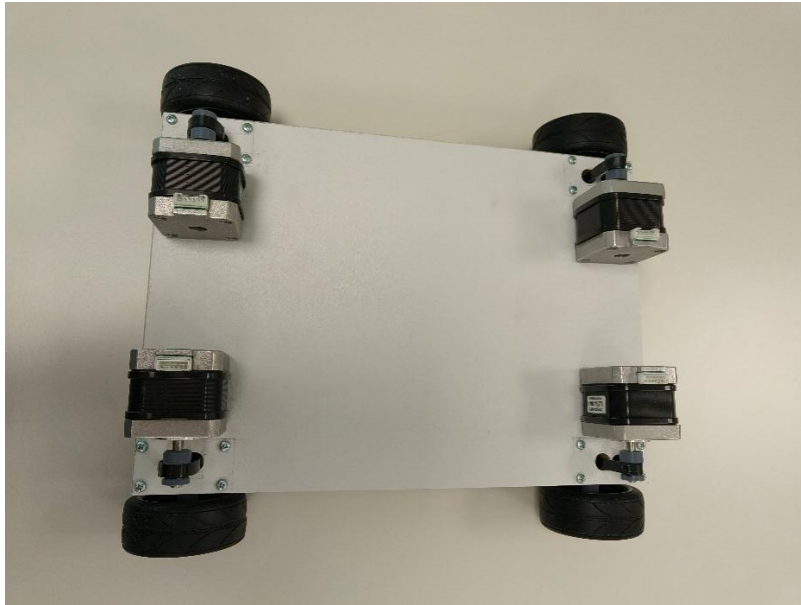


Figura 30: Acople de motores. Fuente: Elaboración propia

Los rodamientos se han añadido atornillando las sujeciones fabricadas con tornillos M3x30, y se han colocado los ejes en su posición correspondiente. También se han acoplado las ruedas a los ejes, y se han unido ambas mitades del eje.

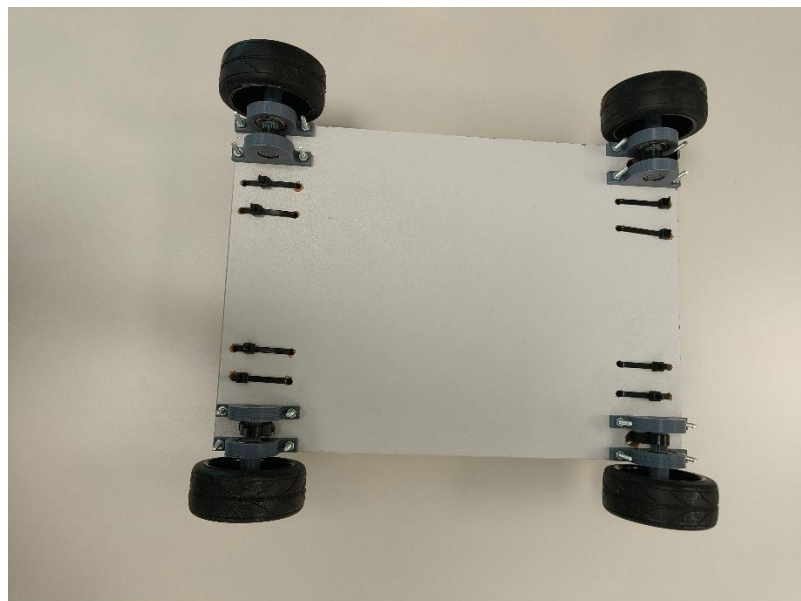


Figura 31: Acople de ruedas. Fuente: Elaboración propia

Las correas que transmiten el movimiento del motor a las ruedas se han hecho a partir de una correa recuperada de una impresora 3D. Se ha cortado la longitud necesaria y se han unido los extremos con hilo y cinta aislante.

Una vez montados los componentes mecánicos se han montado los componentes electrónicos: el circuito de control, la tarjeta Raspberry, una batería portátil para esta misma, y baterías de lito para los motores.

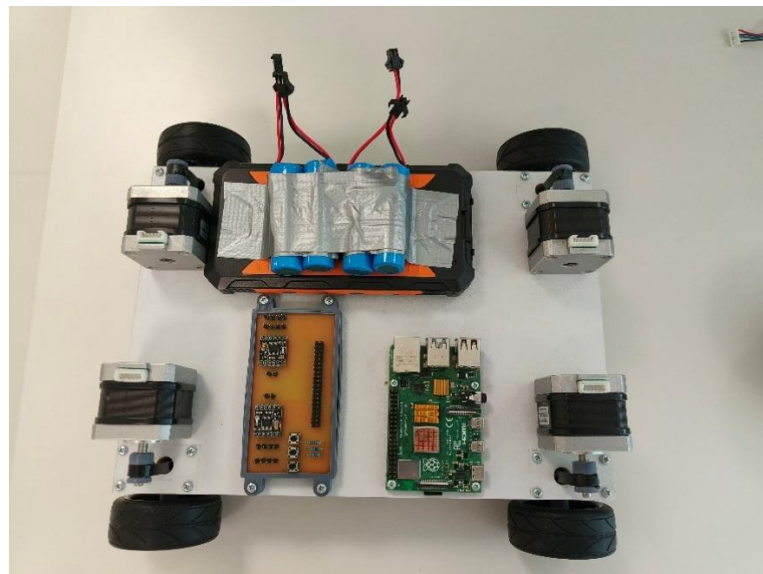


Figura 32: Plataforma montada. Fuente: Elaboración propia

Posteriormente se han añadido los cables, conectando el circuito de control con la Raspberry, los motores al circuito, y las alimentaciones con sus baterías correspondientes. En este caso, la batería USB portátil a la Raspberry y las baterías de lito a los controladores en el circuito de control.

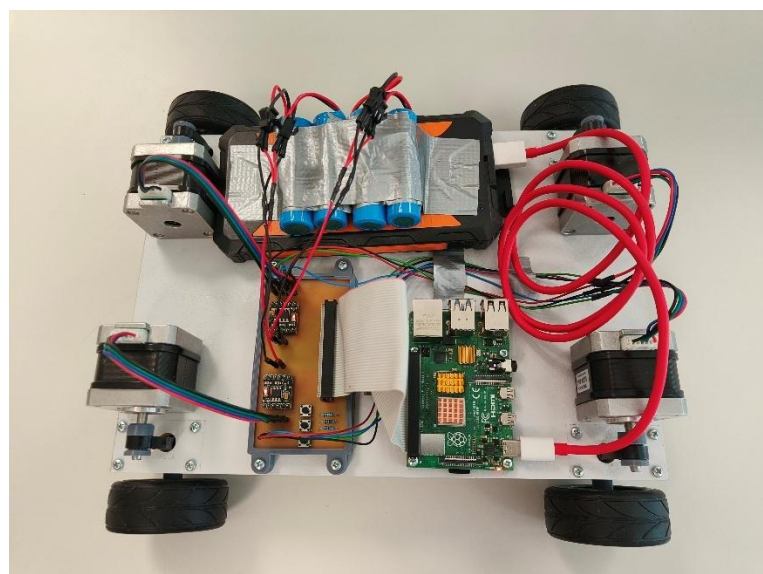


Figura 33: Plataforma cableada. Fuente: Elaboración propia

3.2 Software

3.2.1 LabVIEW 2025 Q1 (32bit):

Se ha utilizado la versión de 32bit de LabVIEW, ya que algunas librerías solo funcionan en esta, y puede hacer uso de las librerías de 64bit sin problema.

En la versión de 2025, la funcionalidad de conexión con Raspberry Pi se encuentra en un añadido oficial llamado Hobbyist Toolkit. Una vez instalado LabVIEW y el toolkit, es necesario conectar la tarjeta Raspberry al ordenador y a internet. Una vez conectado se puede utilizar la herramienta Target Configuration del toolkit para comunicarse con la tarjeta e instalar la versión adecuada de LabVIEW en ella. De este modo la tarjeta puede comunicarse con LabVIEW para ejecutar los programas, y también pueden cargarse en su memoria para que funcione independiente del ordenador usado para programarlo. [4]

Una vez configurada la Raspberry, es necesario crear un proyecto nuevo en LabVIEW, y añadir la tarjeta. Al disponer del Hobbyist Toolkit, la herramienta de adición de objetivos buscará automáticamente la Raspberry bajo la categoría LINX, en caso de no encontrarla se le podría proporcionar la dirección IP de la tarjeta. LabVIEW asignará la dirección IP a la tarjeta como identificador, ya sea la conexión por WiFi o Ethernet.

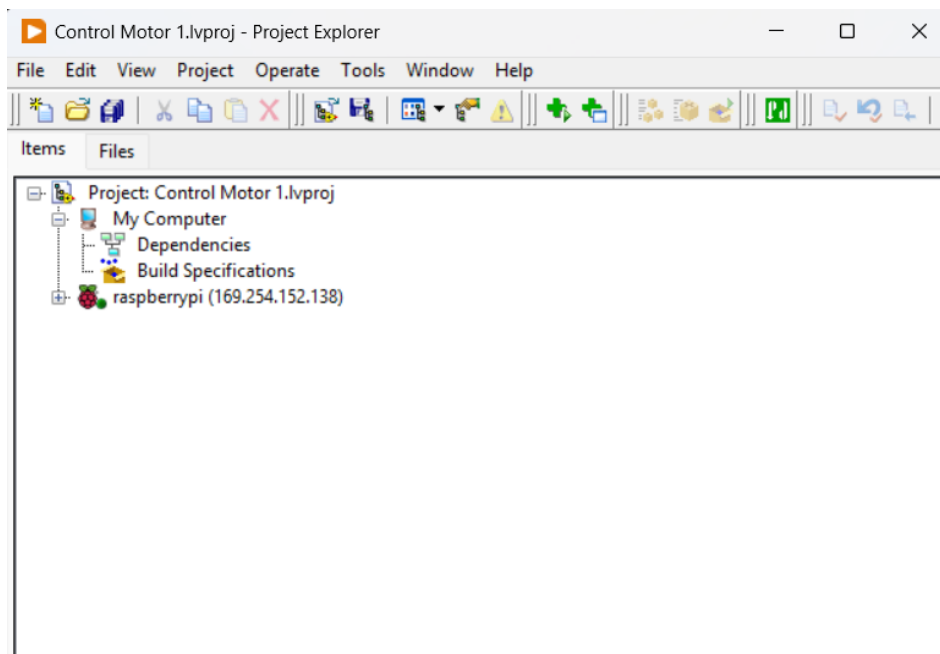


Figura 34: Proyecto LabVIEW. Fuente: Elaboración propia

Una vez añadida la tarjeta, recibirá una dirección IP para comunicarse con LabVIEW. Ahora se crea un VI (Instrumento Virtual) bajo la tarjeta, el cual contendrá el código a ejecutar.

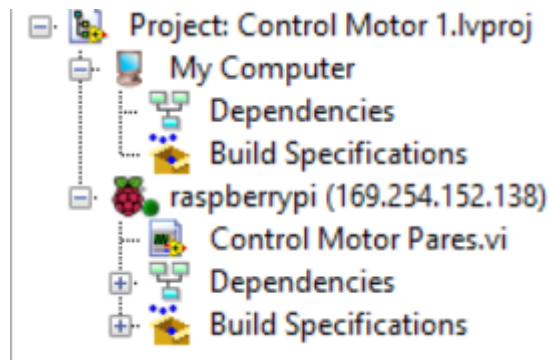


Figura 35: Proyecto con VI creado. Fuente: Elaboración propia

Para cargar el código a la tarjeta, se selecciona el apartado Build Specification y se crea una nueva aplicación en tiempo real. En este caso, simplemente es necesario ir al apartado de archivos fuente y añadir el VI a la lista de VIs de inicio. Esto hará que el VI se ejecute automáticamente al iniciarse la Raspberry. [4]

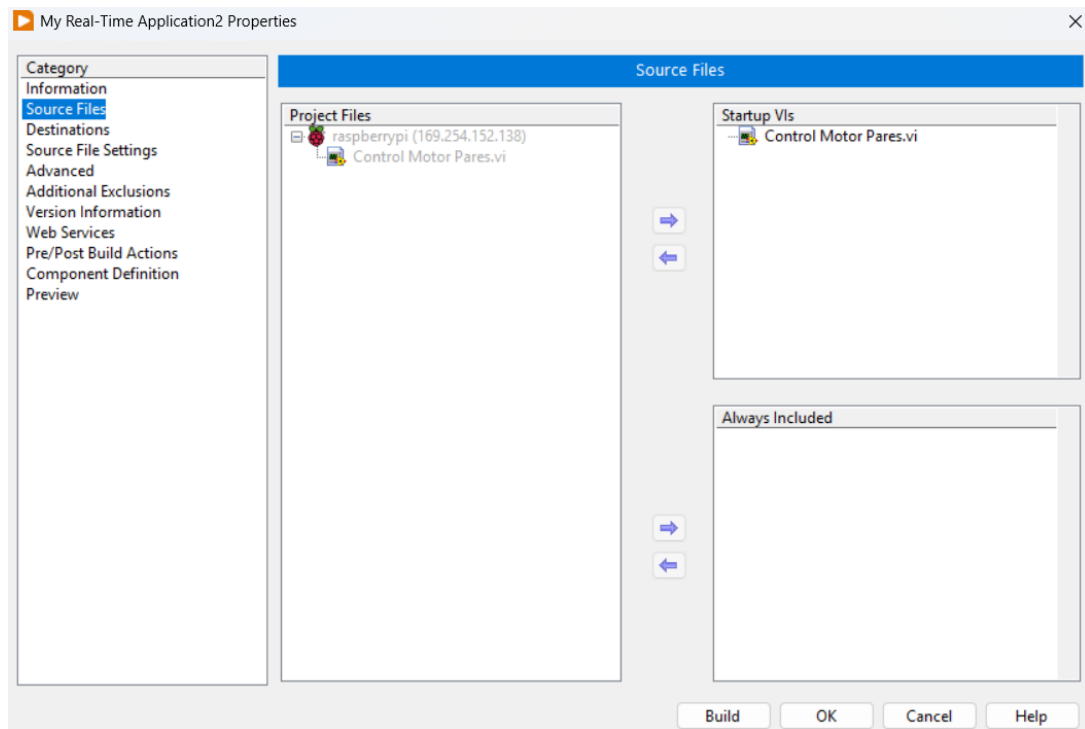


Figura 36: Build Specifications. Fuente: Elaboración Propia

Una vez creada la aplicación, simplemente se selecciona y se despliega a la tarjeta.

3.2.2 Programa de control:

El programa de control manipula los motores mandando las señales digitales por los pines que se comunican con los Drivers con el orden y frecuencia adecuado.

Se han programado 3 trayectorias básicas en este caso.

Cada trayectoria es un bucle while con un contador como condición de parada, es decir, se repetirán las ordenes en su interior hasta que pase un periodo determinado de tiempo. Dentro de este ciclo hay 5 bloques secuenciales, con un retardo de unos milisegundos entre ellos. Los 4 primeros bloques mandan una señal por cada uno de los 4 pines usados para controlar los motores, utilizando un bloque Write N Chan, mientras que el quinto bloque desactiva todos los pines para evitar que quede una señal de salida al acabar el bucle.

La primera trayectoria será rectilínea hacia el frente de la plataforma. En este caso se envía, las señales con el mismo retardo a ambos controladores, asegurando que los motores de ambos lados de la plataforma se mueven a la misma velocidad.

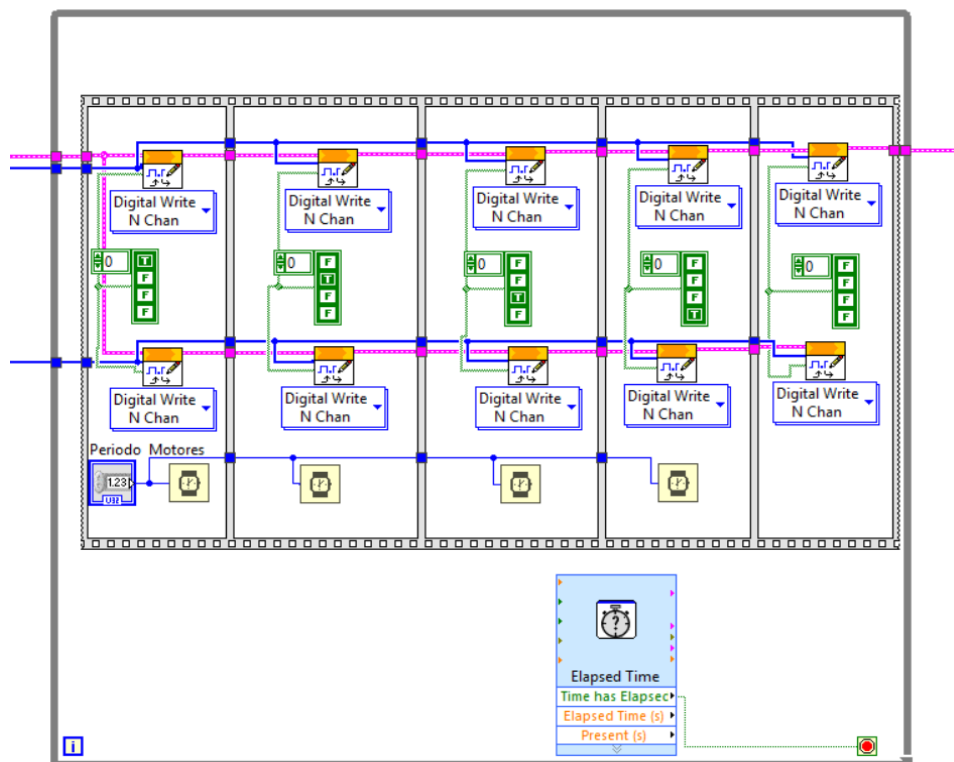


Figura 37: Trayectoria hacia adelante. Fuente: Elaboración propia

La segunda trayectoria será similar, invirtiendo el sentido de los motores para moverse hacia la parte trasera de la plataforma. En este caso bastaría con cambiar la secuencia de los bloques Write N Chan de 1-2-3-4 a 1-4-3-2.

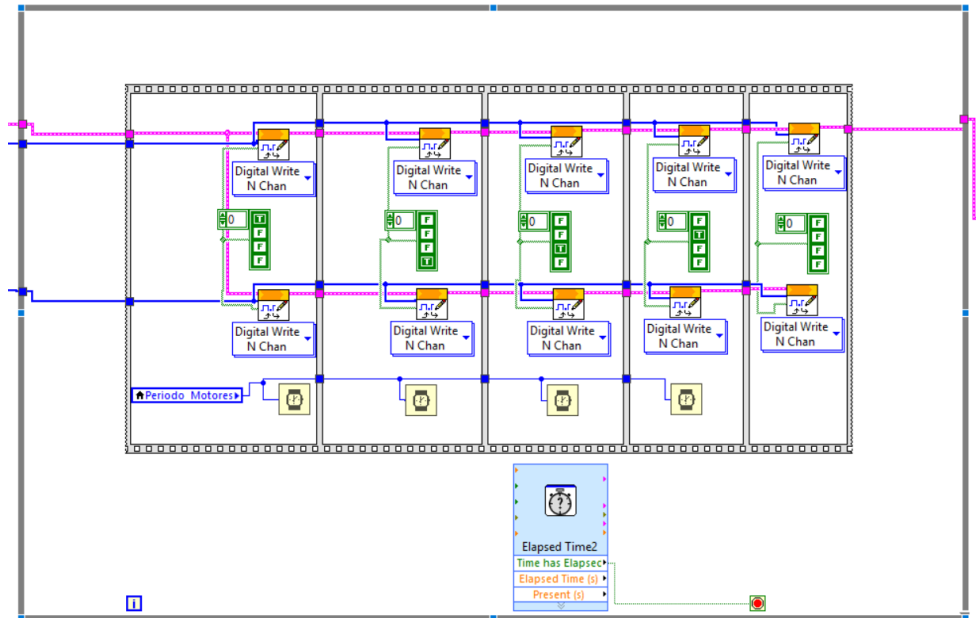


Figura 38: Trayectoria hacia atrás. Fuente: Elaboración propia

La tercera trayectoria realizará ambos movimientos, cambiando de sentido sin parar el programa. Esto se consigue poniendo las trayectorias anteriores en serie.

Una vez se dispone de los bucles para las 3 trayectorias deseadas, es necesario incluir un sistema que permita escoger cual se ejecutará, leyendo entradas digitales en la Raspberry.

Para esto se utiliza una estructura de caso con entrada numérica. En caso de no recibir entradas, se ejecutará el bloque 0 (default) que finaliza el programa.

Para escoger las trayectorias, se leen los valores de los 3 pines utilizados de entrada en la Raspberry. Una vez leídos sus valores, se introducen en un array booleano, dando lugar a una cadena de 3 bits. La lectura y creación del array se realiza en un solo paso, utilizando el bloque Read N Chan. Esta cadena tendrá 3 posibles valores en función de que entrada se encuentre activada, siendo 001 el correspondiente a la entrada 1, 010 el correspondiente a la entrada 2 y 100 el correspondiente a la entrada 3.

La cadena se convierte a un valor en base 10 con un bloque Boolean Array to Num y se introduce en la entrada de control de la estructura de caso. Los 3 valores booleanos corresponden con los valores 1, 2 y 4 en base 10, por lo

tanto, se crean casos que se ejecuten al recibir esos valores de entrada y se introducen los bucles de las trayectorias en su caso correspondiente. [10]

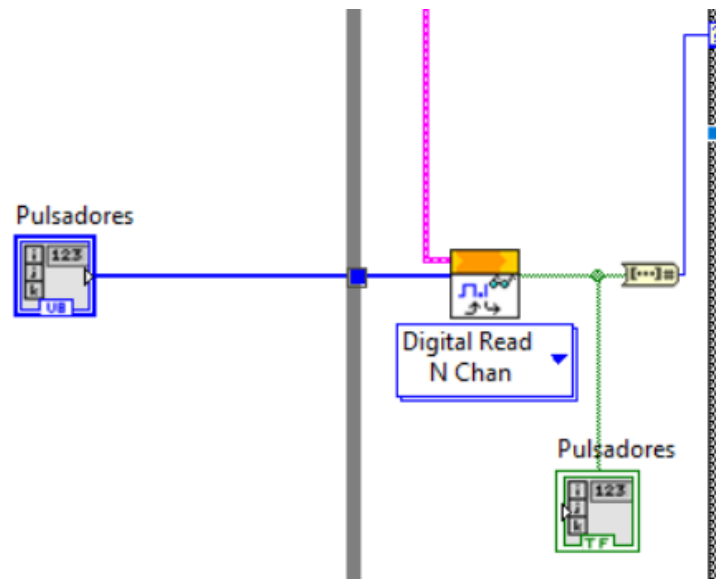


Figura 39: Lectura de pulsadores. Fuente: Elaboración propia

Una vez creada la estructura de control, se introduce en un bucle permanente para que el código se repita siempre que la Raspberry esté encendida. También es necesario empezar el programa con un bloque Open con su opción de Local I/O seleccionada, que abrirá la comunicación con la Raspberry mediante los pines de uso general, y acabarlo con un bloque Close, que cerrará la comunicación con la Raspberry.

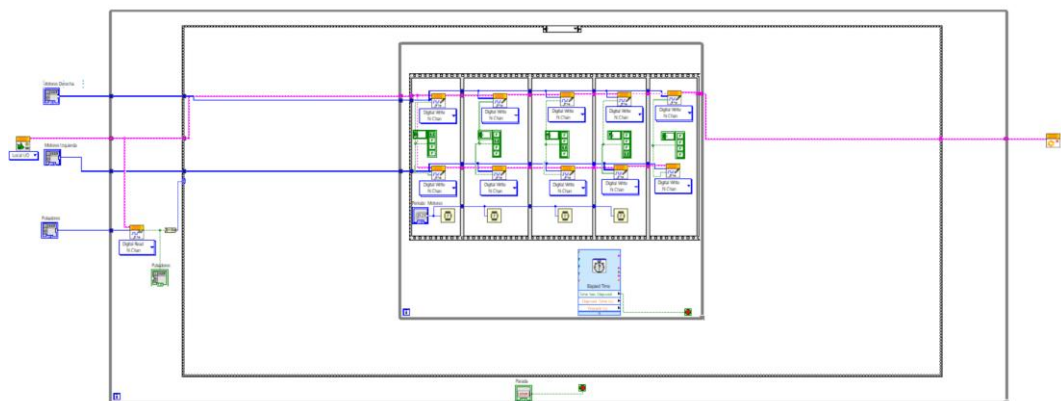


Figura 40: Programa completo. Fuente: Elaboración propia

En este caso, para facilitar la manipulación de las constantes utilizadas (Dirección de Pin, Periodo de los motores) se han incluido bloques con arrays manipulables en el panel frontal del programa en lugar de introducir valores directamente en el código.

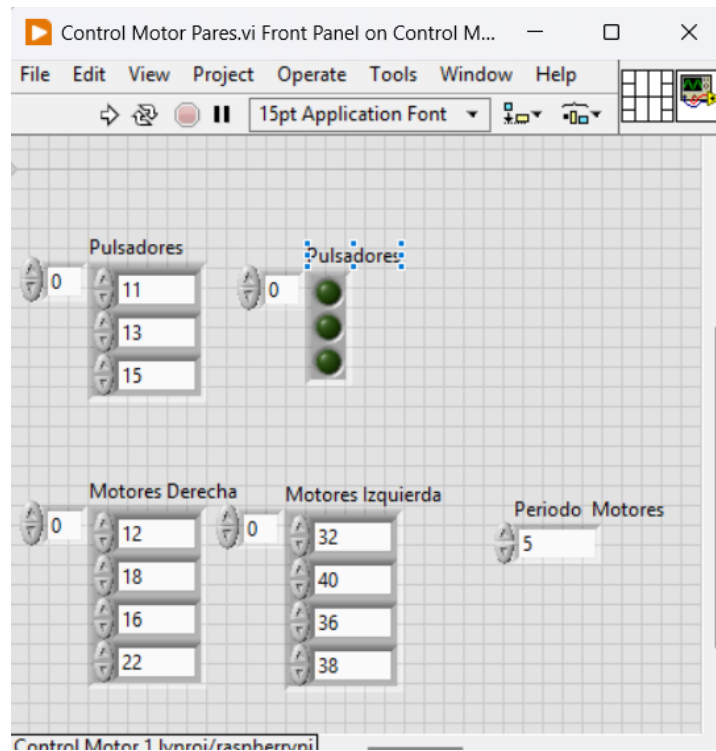


Figura 41: Panel frontal programa. Fuente: Elaboración propia

4. Presupuesto

Material	Unidades	Precio/u	Precio
SL42STH40-1684A-23 (Nema 17)	4	20,94 €	83,76 €
Kit Raspberry PI 4 Model B 2GB	1	95,99 €	95,99 €
Pulsadores	3	0,14 €	0,42 €
Resistencia 10kΩ	3	0,11 €	0,33 €
Driver DRV8833	2	0,41 €	0,82 €
Pin Macho	20	0,0375 €	0,75 €
Conector IDC 40 pines	1	0,40 €	0,40 €
Cable IDC 40 pines	1	1,10 €	1,10 €
Placa de cobre con resina fotosensible 100x60mm	1	0,60 €	0,60 €
Impresora 3D Artillery Genius	1	300 €	300,00 €
Bobina de filamento PLA 1kg	1	16,22 €	16,22 €
Soldador de mesa	1	19,99 €	19,99 €
Bobina de estaño para soldar	1	3,99 €	3,99 €
Ruedas	4	1,17 €	4,68 €

Rodamientos	8	0,21 €	1,68 €
Tornillos M3x30	24	0,10 €	2,40 €
Batería USB	1	29,99 €	29,99 €
Baterías Litio 3,7V	4	4,24 €	16,99 €
Programación LabVIEW	35 h	35 €	1225 €
Diseño e impresión piezas 3D	50 h	35 €	1700 €
Montaje y prueba de la plataforma	20 h	35 €	700 €
TOTAL			4.255,11 €

5. Conclusiones

Se han conseguido de manera razonable los objetivos planteados para el proyecto, obteniendo una plataforma móvil operada por motores paso a paso controlados mediante LabVIEW. La plataforma no requiere de ninguna conexión externa para funcionar, y todos sus cables quedan a la vista para poder observar su disposición. El programa de control es, también, fácil de observar y entender.

Sin embargo, la plataforma tiene ciertas carencias que, pese a no estar estrictamente en los objetivos, reducen su funcionamiento:

- Falta de maniobrabilidad. Debido al peso y el sistema utilizado para motorizar las ruedas, la plataforma no puede girar de manera independiente.
- Dificultades mecánicas. Ya que las correas de los motores se han hecho reciclando una correa de mayores dimensiones, su tolerancia y tensión no son completamente satisfactorias, causando patinaje en los engranajes de vez en cuando.
- Intermitencia en la alimentación. Las baterías utilizadas sufren variaciones en su corriente. Una bajada de corriente coincidente con un paso del motor puede hacer que el paso no se realice correctamente y el motor quede atrapado en un bucle sin moverse.

Todas estas carencias se podrían solucionar rediseñando la plataforma. Aun así, ya que el objetivo principal es la observación del funcionamiento de los motores paso a paso y no un funcionamiento ideal de la plataforma, se puede considerar que el proyecto ha alcanzado su objetivo.

6. Bibliografía

1. Melkebeek, J. A. (2018). *Electrical Machines and Drives: Fundamentals and Advanced Modelling*. Springer.
2. Diferencia Entre Motores Paso a Paso de 4, 6 y 8 Cables - NI. (2018, 23 enero). Recuperado 17 de junio de 2025, de <https://knowledge.ni.com/KnowledgeArticleDetails?id=kA00Z000000PAkPSAW&l=es-ES>
3. Stepper Motors Basics: Types, Uses, and Working Principles. (s. f.). Recuperado 17 de junio de 2025, de <https://www.monolithicpower.com/en/learning/resources/stepper-motors-basics-types-uses?srsId>
4. Getting Started with Raspberry Pi and LabVIEW Community Edition. (s. f.). NI Learning Center - NI. Recuperado 17 de junio de 2025, de <https://learn.ni.com/learn/article/getting-started-with-raspberry-pi-and-labview-community-edition>
5. Getting Started in KiCad | 9.0 | English | Documentation | KiCad. (s. f.). Recuperado 17 de junio de 2025, de https://docs.kicad.org/9.0/en/getting_started_in_kicad/getting_started_in_kicad.html
6. Raspberry Pi Foundation. (s. f.). Getting started with your Raspberry Pi. Raspberry Pi. Recuperado 30 de junio de 2025, de <https://www.raspberrypi.com/documentation/computers/getting-started.html>
7. Raspberry Pi GPIO Pinout. (s. f.). Recuperado 30 de junio de 2025, de <https://pinout.xyz/>
8. LME Editorial Staff. (2023, 27 mayo). Controlling DC Motors with DRV8833 Motor Driver and Arduino. Last Minute Engineers. <https://lastminuteengineers.com/drv8833-arduino-tutorial/>
9. Soren. (2021, 8 mayo). Using a push button with Raspberry Pi GPIO | Raspberry Pi HQ. Raspberry Pi HQ. Recuperado 30 de junio de 2025, de <https://raspberrypihq.com/use-a-push-button-with-raspberry-pi-gpio/#:~:text=Connecting%20the%20Raspberry%20Pi%27s%20general.pin%201%20using%20a%20resistor.>
10. LabVIEW Programming Reference Manual - NI. (2023, 19 julio). Recuperado 17 de junio de 2025, de <https://www.ni.com/docs/en-US/bundle/labview-api-ref/page/intro.html>

Anexo 1: Hojas de datos

1.1 DRV8833 Dual H-Bridge Motor Driver Datasheet

1.2 SL42STH40-1684A-23 Datasheet

Anexo 2: Planos

1.1 Plano frontal circuito

1.2 Plano trasero circuito

1.3 Plano engranaje motor

1.4 Plano parte superior eje

1.5 Plano parte inferior eje

1.6 Plano clavija rueda

1.7 Plano sujeción rodamiento

1.8 Plano receptáculo circuito

1.9 Plano marco circuito

1.10 Plano sujeción Raspberry

Anexo 3: Código LabVIEW