

Trabajo Fin de Grado

# Grado Ingeniería en Tecnologías Industriales

## Diseño de una plataforma móvil autónoma para fines educativos

**Autor:** Sergio García Catalán

**Tutor:** José María Hinojo Montero

Abrahan Márquez Alcaide

Dpto. de Electrónica  
Escuela Técnica Superior de Ingeniería  
Universidad de Sevilla

Sevilla, 2023





Proyecto Fin de Carrera  
Ingeniería de las tecnologías industriales

# Diseño de una plataforma móvil autónoma para fines educativos

Autor:

Sergio García Catalán

Tutor:

José María Hinojo Montero

Investigador Postdoctoral Talento

Abraham Márquez Alcaide

Investigador Postdoctoral Talento

Dpto. de Ingeniería Electrónica  
Escuela Técnica Superior de Ingeniería  
Universidad de Sevilla

Sevilla, 2023



Trabajo Fin de Grado: Diseño de una plataforma móvil autónoma para fines educativos.

Autor: Sergio García Catalán

Tutor: José María Hinojo Montero

Abraham Márquez Alcaide

El tribunal nombrado para juzgar el Proyecto arriba indicado, compuesto por los siguientes miembros:

Presidente:

Vocales:

Secretario:

Acuerdan otorgarle la calificación de:

Sevilla, 2023

El Secretario del Tribunal



A mi familia

A mis maestros





# AGRADECIMIENTOS

---

Agradecer con estas líneas en mi último trabajo de la escuela a todas aquellas personas que me han acompañado durante mi formación, no solo durante mi época universitaria, si no, a todas aquellas personas que me mostraron su apoyo durante cualquier momento de mi época como estudiante.

A mis padres, sin vuestro apoyo nunca habría podido conseguir nada de lo que tengo. A ti mamá gracias por nunca ponerme frenos en cualquier cosa que me he propuesto, por abrirme las alas y hacerme ser como soy ahora. A ti papá por siempre confiar en mí incluso cuando yo menos creía, por hacerme ver que la vida de estudiante va mucho más allá de sentarte en una mesa a estudiar.

A mi hermano mayor, el cual siempre me sirvió como un punto de referencia, por ser un referente en todos mis estudios y acompañarme en mi etapa de independencia, sin ti no habría sido lo mismo.

A toda mi familia que siempre me mostró su apoyo.

A mis amigos, a los que conocí en la universidad y nunca pensé que serían de tantísima importancia en mi vida, aquellos amigos que siempre te dan lo mejor de ellos incluso cuando ellos también necesitan de tu apoyo, gracias porque sin vosotros tampoco podría haber conseguido todo lo que tengo. Y a todos mis amigos que siempre han estado ahí incluso antes de empezar mi época universitaria, aquellos que siempre ven lo mejor de mí y siempre me han apoyado.

Y a todas aquellas personas que han formado parte de esta época tan bonita para mí.

A mis tutores y a mis profesores que me enseñaron lo bonito que puede llegar a ser la ingeniería.

Sergio García Catalán

Sevilla, 2023



# RESUMEN

---

Desde hace varios años la fabricación de robots móviles como “Automated Guided Vehicles” (AGV) han evolucionado llegando a estar presente actualmente en casi cualquier ámbito industrial. Esta tecnología que comenzó en EEUU a mediados de 1950 ha llegado incluso a las aulas de diferentes universidades como una manera de introducir a todos los alumnos de una manera interactiva en el mundo de la programación.

En este trabajo se plantea el diseño y fabricación de una plataforma robótica móvil que pueda ser fácilmente ampliable y presente los periféricos indispensables para su correcto funcionamiento. De esta manera, el objetivo principal del trabajo es la reutilización del diseño del robot para que sea materia de estudio para todo interesado que esté empezando en la programación de vehículos autónomos.

Es por esto por lo que el DSP que será utilizado durante el proyecto será el TMS320F28335 de Texas Instruments, ya que este microprocesador es el empleado en algunas de las asignaturas de la titulación Grado de Ingeniería de las Tecnologías Industriales y su programación y utilización es familiar a todos aquellos estudiantes que cursen dicha titulación.

Este robot, además, incorporará tantos sensores como actuadores con diferentes interfaces de comunicación, por lo que se ha planteado un diseño compatible con este hecho, generando una base sobre la que trabajar y poder realizar un trabajo posterior.



# ABSTRACT

---

For several years the manufacture of mobile robots such as “Automated Guided Vehicles” (AGVs) have evolved and are currently present in almost any industrial field. This technology that began in the US in the mid-1950s has even reached the classrooms of different universities as a way to introduce all students in an interactive way to the world of programming.

This project proposes the design and manufacture of a mobile robotic platform that can be easily expandable, which present the essential peripherals for its proper functioning. In this way, the main objective of the work is the reuse of the robot design so that it is a subject of study for all interested parties who are starting in the programming of autonomous vehicles.

That is why the DSP that will be used during the project will be the Texas Instruments TMS320F28335, since this microprocessor is the one used in some of the subjects of the Degree in Engineering in Industrial Technologies and its programming and use is familiar. to all those students who study this degree.

This robot will also incorporate as many sensors as actuators with different communication interfaces, which is why a design compatible with this fact has been proposed, generating a base on which to work and to be able to carry out subsequent work.



# ÍNDICE

---

Agradecimientos .....	ix
Resumen.....	xi
Abstract .....	xiii
Índice.....	xv
ÍNDICE DE TABLAS .....	xvii
Índice de Figuras .....	xviii
<b>1 Introducción.....</b>	<b>11</b>
1.1 <i>Objetivos</i> .....	11
1.2 <i>Organización</i> .....	12
<b>2 Descripción de la plataforma móvil .....</b>	<b>13</b>
2.1 <i>Control Stick</i> .....	13
2.2 <i>Alimentación</i> .....	14
2.3 <i>Tren de Potencia</i> .....	15
2.3.1 <i>Ruedas</i> .....	15
2.3.2 <i>Driver</i> .....	16
2.3.3 <i>Encoder</i> .....	18
2.4 <i>Sistema de Posicionamiento</i> .....	19
2.4.1 <i>GPS - Exteriores</i> .....	19
2.4.2 <i>Sensor de ultrasonidos – Interiores</i> .....	19
2.5 <i>Bluetooth</i> .....	22
2.6 <i>Acelerómetro</i> .....	23
<b>3 Diseño de Tarjeta Impresa .....</b>	<b>24</b>
3.1 <i>Esquemático</i> .....	25
3.1.1 <i>Vista General</i> .....	25
3.1.2 <i>DSP</i> .....	26
3.1.3 <i>Condensadores</i> .....	27
3.1.4 <i>Alimentación</i> .....	28
3.1.5 <i>Tren de potencia</i> .....	31
3.1.6 <i>Bluetooth</i> .....	33
3.1.7 <i>GPS</i> .....	34
3.1.8 <i>Servo motor</i> .....	34
3.1.9 <i>Sensor ultrasónico</i> .....	35
3.1.10 <i>Acelerómetro</i> .....	36

3.2	<i>Implementación física</i> .....	36
3.2.1	Floorplanning.....	38
3.2.2	Rutado.....	41
<b>4</b>	<b>Programación</b> .....	<b>46</b>
4.1	Descripción General de la Aplicación. ....	47
4.2	Bluetooth .....	49
4.3	Test1: Servomotor .....	50
4.4	Test2: Sensor de proximidad .....	51
4.5	Test3: Motor .....	53
4.6	Test4: GPS.....	55
<b>5</b>	<b>Validación</b> .....	<b>58</b>
5.1	<i>Hardware</i> .....	58
5.2	<i>SOFTWARE</i> .....	59
5.3	<i>Correcciones</i> .....	59
<b>6</b>	<b>Conclusiones y Líneas Futuras</b> .....	<b>61</b>
6.1	<i>Líneas Futuras</i> .....	62
<b>7</b>	<b>Referencias</b> .....	<b>63</b>



# ÍNDICE DE TABLAS

---

Tabla 1. Características principales DSP.....	14
Tabla 2. Consumo del sistema.....	14
Tabla 3. Modo de funcionamiento drv8835 IN/IN.....	17
Tabla 4. Modo de funcionamiento drv8835 PH/EN.....	17
Tabla 5. Características técnicas del módulo GPS elegido. ....	19
Tabla 6. Características técnicas servo motor .....	21
Tabla 7. Características técnicas del HC-SR04:.....	22
Tabla 8. Características técnicas de HC-05.....	23
Tabla 9. GPIO motores.....	31
Tabla 10. Pines Encoder.....	33
Tabla 11. Pines Bluetooth .....	33
Tabla 12. Pines GPS.....	34
Tabla 13. Pines Servo motor .....	35
Tabla 14. Pines Sensor Sonar .....	35
Tabla 15. Pines Acelerómetro .....	36
Tabla 16. Plan de pruebas físicas .....	58
Tabla 17. Plan de pruebas Software .....	59

# ÍNDICE DE FIGURAS

---

Figura 1. Robot empleado en proyectos anteriores .....	12
Figura 2. Control Stick TMS320F28335.....	13
Figura 3. Funcionamiento Puente H.....	15
Figura 4. Ruedas sobre KIT.....	15
Figura 5. Kit de desarrollado suministrado por la compañía Pololu basado en el driver DRV8835.[3] ....	16
Figura 6. Esquemático del driver DRV8835[4].....	16
Figura 7. Modo de funcionamiento drv8835 IN/IN[4].....	17
Figura 8. Modo de funcionamiento drv8835 PH/EN[4].....	18
Figura 9. HC-020K[5] .....	18
Figura 10. Telo132 [6].....	19
Figura 11. Unidad 28BYJ-48[7].....	20
Figura 12. Funcionamiento Motor PP[7].....	20
Figura 13. PCB driver 28BYJ-48[7].....	21
Figura 14. Modelo HC-SR04[9].....	21
Figura 15. Funcionamiento Sensor ultrasónico [9].....	22
Figura 16. Módulo HC-05 .....	22
Figura 17. Ejemplo de símbolo creado en Altium Designer. ....	25
Figura 18. Ejemplo de huella creado en Altium Designer.....	25
Figura 19. Esquemático General .....	26
Figura 20. Esquemático DSP.....	26

Figura 21. Esquemático de los condensadores de estabilización y filtrado.....	27
Figura 22. Alimentación que incorporar el stick donde se localiza el DSP empleado[2]. .....	28
Figura 23. Esquemático LM2596-12 según fabricante[12].....	28
Figura 24. Valor Inductancia en función $V_{in}$ $I_{load}$ [12] .....	29
Figura 25. Esquemático LM317 .....	29
Figura 26. Pinout de Alimentación.....	30
Figura 27. Esquemático final de la etapa de alimentación. ....	30
Figura 28. Esquemático asociado al tren de potencia.....	31
Figura 29. Esquemático Driver de los motores .....	32
Figura 30. Esquemático de la línea de alimentación protegida para los motores.....	32
Figura 31. Modelo encoder HC-020K[5] .....	33
Figura 32. Esquemático Encoders .....	33
Figura 33. Esquemático Bluetooth .....	34
Figura 34. Esquemático GPS.....	34
Figura 35. Esquemático Servomotor .....	35
Figura 36. Esquemático de Sensor de proximidad .....	35
Figura 37. Esquemático del acelerómetro .....	36
Figura 38. Reglas de diseño para la PCB .....	37
Figura 39. Forma de la PCB.....	37
Figura 40. organización de capas .....	38
Figura 41. Floorplanning.....	39
Figura 42. Floorplanning zona trasera.....	40
Figura 43. Floorplanning zona delantera.....	40
Figura 44. Rutado capa superior.....	42
Figura 45. Rutada capa inferior. ....	43
Figura 46. Rutada capa de alimentación.....	44
Figura 47. Rutado del plano de tierra .....	45
Figura 48. Creación proyecto CCS.....	47
Figura 49. Diagrama de flujo MAIN.....	48

Figura 50. Código, inicialización Servo motor.....	50
Figura 51. Funcionamiento Servo motor.....	50
Figura 52. Diagrama de Flujo, test1 .....	51
Figura 53. Registros empleados para la Ecap[14] .....	52
Figura 54. Programación del módulo Ecap .....	52
Figura 55. Diagrama de Flujo, test2 .....	53
Figura 56. Código para la inicialización del motor. ....	54
Figura 57. Código, inicialización PWM.....	54
Figura 58. Diagrama de flujo, test3 .....	55
Figura 59. Información posible recibida por el GPS .....	56
Figura 60. Información útil recibida por GPS .....	56
Figura 61. Diagrama de flujo, GPS .....	57
Figura 62. Diagrama de Flujo, test4 .....	57
Figura 63. Robot Final.....	61
Figura 64. Robot final, superior.....	62





# 1 INTRODUCCIÓN

---

**E**ste trabajo engloba el diseño, su posterior fabricación y puesta en marcha de un PCB que actúa como la base de una plataforma móvil autónoma de bajo coste para su aplicación en la enseñanza de dispositivos de microprocesadores y algoritmos de guiado autónomo. Para ello, se dotará de diferentes periféricos, cuyo funcionamiento está probado. Asimismo, su diseño está pensado para poder adaptarse a nuevas aplicaciones mediante la adición de puertos de expansión.

La idea surge a través de la realización de diferentes prácticas en la Escuela Técnica Superior de Ingeniería (ETSI), donde ésta siempre pone a disposición del alumnado todo material necesario para que pueda desenvolverse en un ámbito más práctico. De esta manera, el diseño de esta PCB está centrado en que sea utilizado como nuevo material para futuras prácticas.

En el departamento de Electrónica existen muchos robots con diferentes funciones, los cuales han sido usados por muchos alumnos con el fin aprender sobre programación de una manera interactiva. Este trabajo está inspirado en uno de estos robots que el departamento pone a nuestra disposición. Un vehículo con 2 ruedas de tracción controladas por un DSP y una tercera rueda libre. Cabe indicar que esta estructura simplifica el proceso de control y operación de la plataforma móvil.

Este trabajo está inspirado en un proyecto anterior [1], el cual sirvió para la realización de las prácticas de la asignatura Electrónica industrial, adscrita al Grado de Ingeniería de las Tecnologías Industriales[1]. Aunque el objetivo del trabajo no era su posterior modificación a mí personalmente me sirvió como punto de referencia para mis prácticas y trabajo de la asignatura y, a continuación, este trabajo.

## 1.1 Objetivos

La idea inicial de este trabajo es la mejora del robot que se observa en la , diseñando una plataforma que simplifique, el conexionado y la alimentación de los diferentes componentes y que mejore la integridad de señal de estos. Además, se pretende que este trabajo sirva como guía técnica para todo aquel que quiera empezar a programar un nuevo robot móvil.

Por tanto, la simplificación de la plataforma móvil a desarrollar consistirá en la reducción del número de PCBs a emplear, ya que como se observa en la , la plataforma actual posee un chasis sobre el cual está montado el DSP. El primer objetivo es desarrollar un PCB que unifique la función de estos dos elementos para que el propio chasis del robot integre el DSP. Además, esta decisión evitará el uso de cableado largo en el robot, haciendo que su manipulación sea más cómoda y fácil, así como simplificar las conexiones a los periféricos que se usarán en este trabajo.

Para la realización de este trabajo usaremos el procesador TMS320F28335 [2] de *Texas Instruments*, ya que ha sido el utilizado en las prácticas de la asignatura Electrónica industrial. La elección de este DSP se debe a que posee una serie de periféricos hardware que serán de gran utilidad para el desarrollo del trabajo y su posterior continuación.

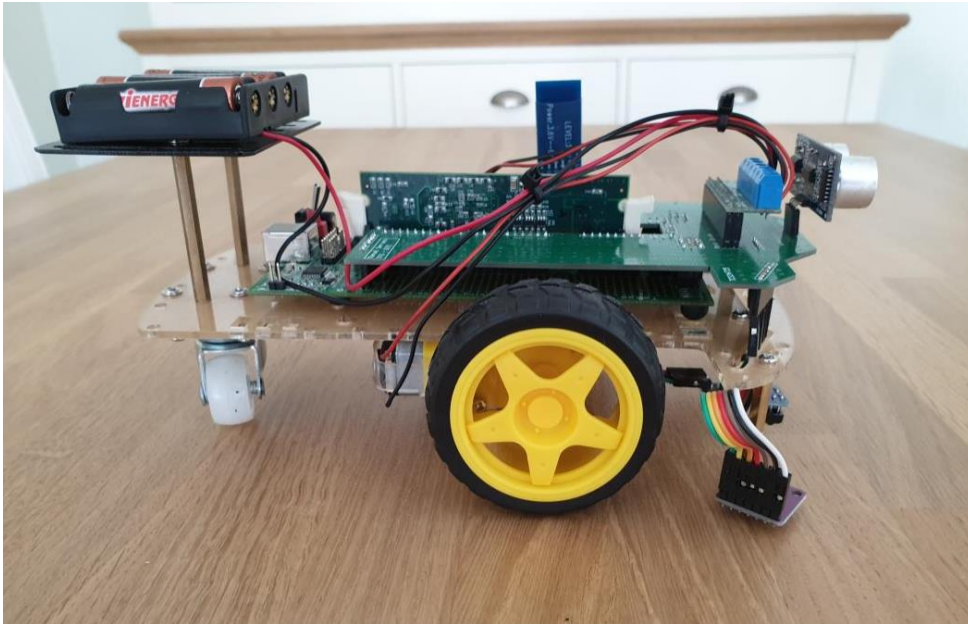


Figura 1. Robot empleado en proyectos anteriores

## 1.2 Organización

El presente documento se ha organizado de la siguiente manera:

- **Capítulo 1:** muestra la motivación del trabajo, así como los objetivos marcados y el planteamiento a seguir durante todo el trabajo.
- **Capítulo 2:** describe la plataforma autónoma a diseñar junto con todos los componentes empleados para su correcto funcionamiento.
- **Capítulo 3:** en este capítulo se detalla el proceso de diseño de la tarjeta impresa que será usada como la base del robot.
- **Capítulo 4:** recoge el código de programación empleado durante el trabajo, así como el funcionamiento principal del robot.
- **Capítulo 5:** presenta las diferentes pruebas, tanto a nivel hardware como software, realizadas para asegurar el correcto funcionamiento del sistema desarrollado.
- **Capítulo 6:** presenta las conclusiones extraídas del presente trabajo, así como las posible líneas futuras para continuar su desarrollo.



## 2 DESCRIPCIÓN DE LA PLATAFORMA MÓVIL

---

**D**urante este capítulo, se va a proceder a describir los diferentes periféricos integrados en la tarjeta impresa, recogiendo su funcionamiento individual, así como sus principales características técnicas.

El robot constara de diferentes sistemas periféricos para poder facilitar tanto la comunicación entre usuario y robot, así como otros periféricos que servirán para obtener información del entorno o bien actuar sobre el propio robot.

### 2.1 Control Stick

Como ya se ha mencionado el DSP empleado será el modelo de Texas Instruments TMS320F28335 [2], este viene incorporado en una Control Stick como se muestra en la Figura 2.



Figura 2. Control Stick TMS320F28335

Las principales características del DSP empleado se muestran en la Tabla 1.

Tabla 1. Características principales DSP

Característica	Valor
<b>Frecuencia máxima del reloj</b>	150MHz
<b>Memoria</b>	256k x 16 Flash + 34k x 16 SARAM
<b>Periféricos</b>	PWM, SCI, SPI, CAN, I2C, McBSP, XINTF, E/S paralela
<b>GPIOs</b>	88 pines reservados
<b>ADC</b>	12 bit, 16 canales

## 2.2 Alimentación

La fuente de alimentación que usará la tarjeta impresa está diseñada íntegramente en la propia tarjeta. Además, esta alimentación tendrá una conversión en cascada, ya que la tensión de alimentación será de aproximadamente 12V, pero los diferentes periféricos usarán una tensión de alimentación de 5V o de 3.3V.

Para esta conversión en cascada se ha optado por dos componentes, un primer DC/DC (LM2596-5) el cual transforma la tensión de entrada a una tensión de 5V. Esta tensión será usada por el DSP. Aunque la tarjeta control stick esté alimentada a 5V, el DSP TMS320F28335 está alimentado a 3.3V ya que dicho sistema de evaluación posee un seguidor de tensión integrado en la propia tarjeta.

A continuación, todos los periféricos deben de estar alimentados a la misma tensión que el microprocesador, es decir, a 3.3V. Es por eso por lo que se ha integrado un seguidor de tensión (LM317) el cual se ha ajustado para obtener una tensión de salida de 3.3V.

Esta conversión en cascada desde 12V viene debido a que la tensión máxima requerida por uno de los componentes del sistema son estos propios 12V. Los motores que son usados por el robot móvil están alimentados aproximadamente a unos 12V. La primera conversión de tensión a 5V es para poder alimentar el DSP correctamente. Por ultimo los 3.3V serán necesarios para la correcta comunicación entre los diferentes periféricos y el DSP.

A continuación, en la Tabla 2 se muestran los valores de consumo en mA por cada componente, junto al consumo total del sistema.

Tabla 2. Consumo del sistema

Componentes	Consumo (mA)
<b>HC-05</b>	40
<b>HC-SR04</b>	15
<b>28BYJ-48</b>	240
<b>DRV8835</b>	1500
<b>Telo132</b>	100
<b>HC-020K</b>	40
<b>Sistema</b>	1935

## 2.3 Tren de Potencia

El primer periférico será el tren de potencia para las ruedas del robot, así como todo lo relacionado para el correcto funcionamiento para este.

### 2.3.1 Ruedas

El robot consta de dos tipos de ruedas. Un modelo de rueda rotatoria o móvil, (la cual será una rueda única situada en la parte trasera del robot móvil) y un segundo modelo de rueda, adherida al tren de potencia. Cabe indicar que de este último modelo existen dos ruedas, situadas en los laterales del robot.

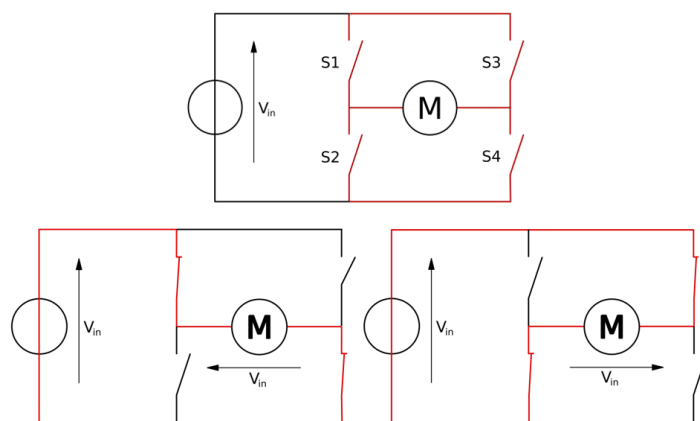


Figura 3. Funcionamiento Puente H

La rueda motorizada estará alimentada con una fuente DC, por lo que usaremos un puente H, como el que se muestra en Figura 3 para el control de las ruedas. El puente en H nos permite gracias a una fuente DC poder elegir el sentido de la rueda, así como poder frenar la rueda.

El montaje de las ruedas, como se ha comentado anteriormente, será dos ruedas motoras en la parte delantera del vehículo y una rueda libre en la parte trasera el vehículo. La Figura 4 muestra un ejemplo del montaje final. Cabe destacar que este está basado en un kit de desarrollo, el cual incluye todos los componentes necesarios para su montaje en el vehículo.



Figura 4. Ruedas sobre KIT

### 2.3.2 Driver

Para el diseño de nuestro puente H, se ha optado por el driver DRV8835 suministrado por la compañía Texas Instruments. Este ha sido seleccionado debido a que la versión original de la plataforma hacía uso de un controlador desarrollo por la empresa Pololu [3] [4].



Figura 5. Kit de desarrollo suministrado por la compañía Pololu basado en el driver DRV8835.[3]

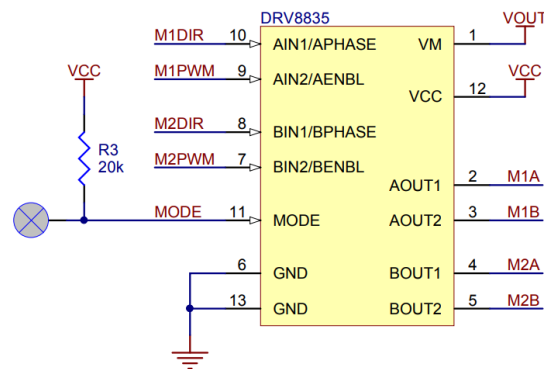


Figura 6. Esquemático del driver DRV8835[4]

En este diseño, se ha optado por usar el mismo driver que se usó en este kit de desarrollo ya que ha sido probado durante prácticas y trabajos anteriores y su funcionalidad y prestaciones son suficientes para el control de los motores DC empleados. Como se muestra en Figura 6 el driver consta de dos alimentaciones distintas, una alimentación de 3.3V, que sirve para alimentar los circuitos lógicos y de control internos del driver, y una alimentación de alta tensión, empleada para alimentar cada uno de los dos puentes en H que integra dicho driver. Por tanto, esta segunda tensión de alimentación será la suministrada a los motores. Además, consta de cuatro pines de entrada, que serán usados para el control de las dos ruedas. La función de estos pines vendrá determinada por el pin de MODE. En nuestro trabajo, usaremos un modo de funcionamiento Phase/Enable, por lo que tendremos que ajustar el pin de MODE al valor lógico '1'. Finalmente, tendrá cuatro salidas correspondientes a las conexiones directas con los motores.

De manera predeterminada el pin MODE tiene un pull-down interno que fuerza su valor a un cero lógico, '0'. En este modo de funcionamiento, las entradas se denominan IN1/IN2 y su valor lógico define el valor

de los puertos de salida OUT1 y OUT2. Los posibles valores que pueden tomar las salidas a partir de los valores asignados a IN1 e IN2 se muestran en la Tabla 3.

Tabla 3. Modo de funcionamiento drv8835 IN/IN

In1	In2	Out1	Out2	Función
0	0	Z	Z	Deslizamiento
0	1	L	H	Marcha atrás
1	0	H	L	Avanza
1	1	H	H	Freno

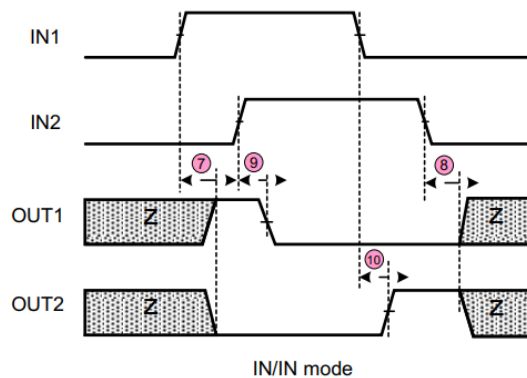


Figura 7. Modo de funcionamiento drv8835 IN/IN[4]

Como se puede observar en la Tabla 3, con el pin MODE a nivel bajo y las entradas actuando como IN/IN el DRV8835, posee hasta cuatro modos de respuesta frente a las dos variables lógicas de entrada. Si ambas variables de entrada se encuentran a nivel bajo el sistema podrá rodar con facilidad, siempre y cuando tenga una inercia propia. Con las dos variables a nivel alto, el sistema motor frena, aunque este lleve inercia. Por último, si una de las variables está a nivel alto y la otra se encuentra a nivel bajo el sistema avanzará o retrocederá en función de la variable que está a nivel alto.

En este proyecto se ha optado por un modo de funcionamiento del DRV8835 PH/EN, (Phase/Enable), el cual con las variables de entrada PH y EN controlan el funcionamiento de un puente H. Siendo la variable “Phase” una señal lógica que indica el sentido del puente H y la otra variable, “Enable”, controla la tensión a través de un PWM que entra en este puente H.

Como el pin correspondiente al “MODE” tiene una resistencia pull-down interna se diseñará una resistencia externa pull-up que pueda contrarrestar el efecto de la pull-down

Tabla 4. Modo de funcionamiento drv8835 PH/EN

In1 (enable)	In2 (Phase)	Out1	Out2	Función
0	X	L	L	Freno
PWM	1	L	H (PWM)	Marcha atrás
PWM	0	H (PWM)	L	Avanza

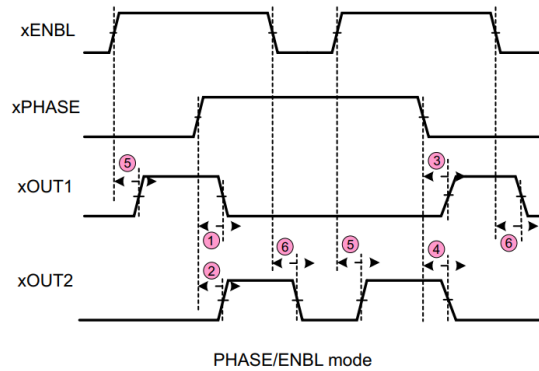


Figura 8. Modo de funcionamiento drv8835 PH/EN[4]

En la Tabla 4, se muestra el modo de funcionamiento que se empleará en el proyecto (PH/EN), mostrando como responder a el DRV8835 frente a dos entradas, una de ellas una señal lógica y la otra señal una señal PWM. En la tabla se muestra que sin importar el valor de entrada de PH, siempre que la señal EN este a cero el sistema frena. A continuación, podemos ver como el sistema puede avanzar o retroceder en función de si esta señal PH está a nivel alto o nivel bajo.

### 2.3.3 Encoder

Para el control de la velocidad sobre las ruedas, vamos a usar un encoder para cada rueda motora. El modelo de estos encoders son el HC-020K. Este modelo nos ofrece unos encoders ópticos de una alta resolución y una respuesta rápida[5].

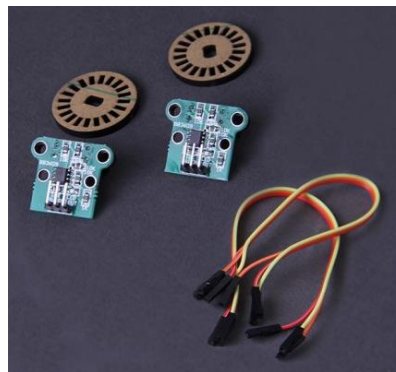


Figura 9. HC-020K[5]

Los encoders ópticos son una tecnología que nos ayuda a medir el giro de una rueda. En este proyecto, nos ayudarán a estimar la velocidad de cada rueda del robot.

El funcionamiento de los enconder ópticos es gracias a un circuito formado por un fotodiodo, un led y una rueda con aberturas. El led emite una luz que el fotodiodo recibe, cerrando así un circuito. Por tanto, si se interrumpe este paso de luz entre el led y el fotodiodo, el circuito permanece abierto; dando así una señal a nivel alto en la salida. Gracias a este mecanismo, al introducir la rueda con aberturas, como la que se muestra en Figura 9, que gira de forma solidaria con el rotor del motor, el circuito permite generar una señal a su salida que varía entre nivel alto y nivel bajo y cuya frecuencia es proporcional a la velocidad de rotación del propio motor.

## 2.4 Sistema de Posicionamiento.

El robot contará con dos sistemas independientes de posicionamiento, siendo uno de ellos diseñado para exteriores (GPS) y el otro sistema para interiores (ultrasonidos).

### 2.4.1 GPS - Exteriores

El GPS servirá como periférico para obtener las coordenadas de donde se encuentra la plataforma móvil. El modelo que se ha escogido para este periférico es el mostrado en la Figura 10 cuya referencia es “Telo132” [6].



Figura 10. Telo132 [6]

Las principales características técnicas del módulo GPS se recogen en la Tabla 5:

Tabla 5. Características técnicas del módulo GPS elegido.

Parámetro	Valor
<b>Voltaje de funcionamiento</b>	3.3V-5V
<b>Puerto de antena</b>	SMA e IPEX
<b>Sensibilidad de seguimiento (arranque/estándar)</b>	-148dBm / -162dBm
<b>Precisión</b>	2.5m
<b>Tiempo respuesta</b>	32 segundos (depende del entorno)
<b>Consumo</b>	<25mA
<b>Rango de velocidad en baudios</b>	4800-115200 (bps)

### 2.4.2 Sensor de ultrasonidos – Interiores

En el proyecto se plantea la localización para interiores mediante el uso de dos elementos: un actuador, compuesto por un driver y un motor paso a paso, y un sensor de proximidad.

La idea es ubicar el sensor de ultrasonidos sobre el eje del motor paso a paso para que, moviendo un ángulo preciso el motor paso a paso, podamos determinar la distancia de la plataforma móvil a los objetos cercanos. De esta forma, se pretende conocer el entorno que rodea a dicha plataforma con el objetivo de determinar la posible ruta a seguir.

### 2.4.2.1 Motor paso a paso

El motor seleccionado para usarse durante el proyecto ha sido el modelo 28BYJ-48. Este es un motor paso a paso con cuatro pines de entrada y un pin extra de alimentación [7].



Figura 11. Unidad 28BYJ-48[7]

Este motor paso a paso está compuesto por dos bobinas que presentan tres terminales cada una de ellas, como se puede observar en la Figura 12. El segundo terminal de cada una de las bobinas es conectado a la tensión de alimentación, VDD=5V, mientras que el resto de los terminales son conectados a diferentes líneas con el objetivo de crear el flujo magnético que permita la rotación del motor en el sentido deseado.

Los pines 1 y 3 de la figura facilitan la creación de un campo magnético vertical en la Figura 12, que combinado con el campo magnético horizontal que se crea gracias a los pines 2 y 4, se puede realizar un giro de motor central.

Finalmente, este motor paso a paso estará acompañado de una caja reductora para obtener un ángulo de giro por cada pulso  $5.625^\circ/64=0.0879^\circ$ .

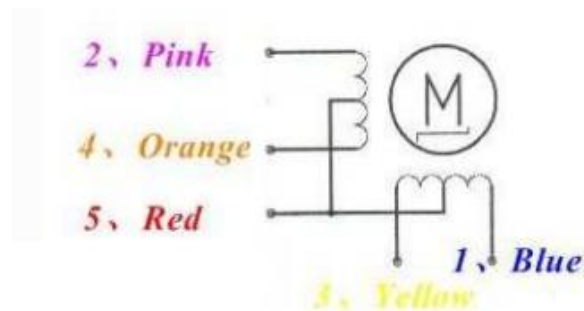


Figura 12. Funcionamiento Motor PP[7]

Para el control del motor paso a paso, se ha usado el driver ULN2003 [8], por su simplicidad y buen funcionamiento. Este driver es el encargado de enviar los pulsos de 0.6V necesarios para avanzar un paso.

El driver tiene hasta 6 entradas y sus respectivas salidas, de las cuales sólo 4 de estos pares serán usados para el control del motor paso a paso. Para poder observar de manera experimental estos pulsos generados, a la salida se ha incorporado un conjunto de leds.



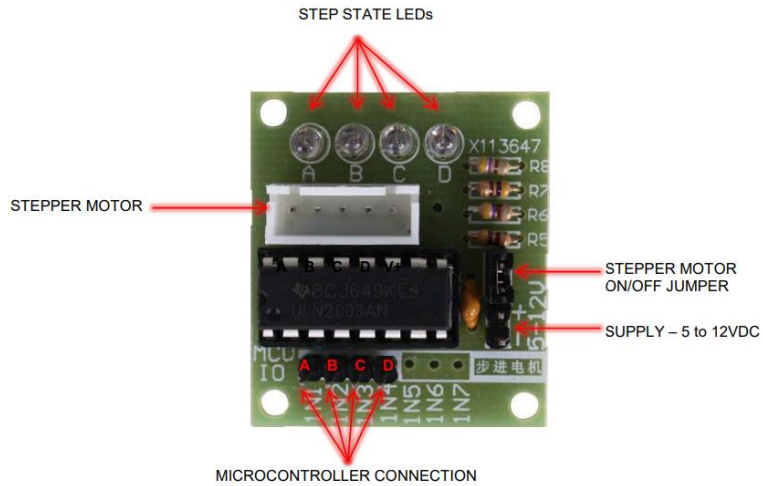


Figura 13. PCB driver 28BYJ-48[7]

La Tabla 6. recoge las principales características técnicas:

Tabla 6. Características técnicas servo motor

Parámetro	Valor
Tensión de funcionamiento	5-12V
Consumo	500mA
Angulo de giro	5.625°/64
Frecuencia	100Hz

#### 2.4.2.2 Sensor ultrasónico

Por último, el sensor que se acoplará al motor paso a paso será un sensor ultrasónico, también llamado sensor de proximidad. El modelo de este sensor es HC-SR04 [9].



Figura 14. Modelo HC-SR04[9]

Estos sensores ultrasónicos funcionan a través del eco producido por la emisión de un ultrasonido. El sensor emite este ultrasonido cuando el usuario active la señal de trigger, este rebotará en los objetos y es recibido de nuevo por el sensor, por lo que se puede calcular la distancia entre objeto y sensor sabiendo la velocidad del sonido. Esta viene determinada por la fórmula

$$Distancia = \frac{T_{echo}^{nivel\ alto} * v_{sonido}}{2}$$

donde  $T_{echo}^{nivel\ alto}$  es el tiempo que la señal de eco está a nivel alto y  $v_{sonido}$  representa la velocidad del sonido en el aire (valor).

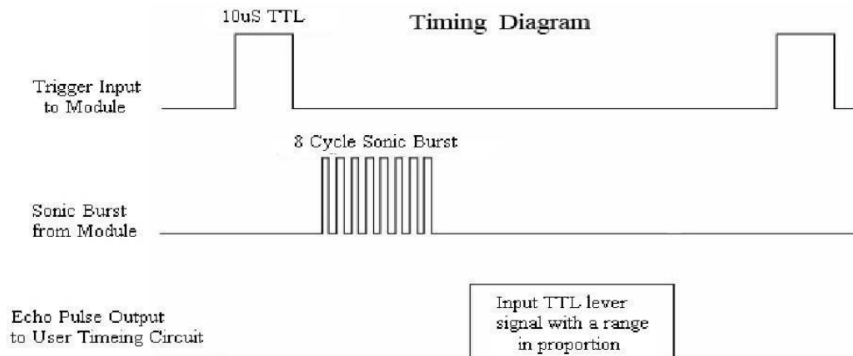


Figura 15. Funcionamiento Sensor ultrasónico [9].

La Tabla 7 recoge las características principales del módulo HC-SR04.

Tabla 7. Características técnicas del HC-SR04:

Parámetros	Valor
Tensión de trabajo	5V
Consumo	15mA
Frecuencia de trabajo	40Hz
Rango	2cm-4m
Ángulo de trabajo	15°

## 2.5 Bluetooth

La comunicación inalámbrica será a través del módulo bluetooth HC-05. Este será el encargado de la transmisión de información entre el usuario y el microprocesador [6].



Figura 16. Módulo HC-05

El módulo bluetooth HC-05 requiere una interfaz de comunicación de tipo UART. Para este modo de comunicación, son necesarios dos pines (TX, para transmitir información del módulo al DSP, y RX, para enviar información del DSP al módulo) y otros dos pines usados para la alimentación y tierra.

Tabla 8. Características técnicas de HC-05

Parámetros	Valor
<b>Voltaje de funcionamiento</b>	5(V) adaptado a 3.3(V)
<b>Consumo</b>	30-40 (mA)
<b>Chip Radio</b>	CSR BC41743
<b>Baudios por defecto</b>	9600
<b>Frecuencia</b>	2.4 GHz, banda ISM
<b>Alcance</b>	5-10 (m)
<b>Sensibilidad</b>	-80dBm

## 2.6 Acelerómetro

Durante este proyecto se ha incorporado este periférico, aunque no haya sido evaluado a lo largo del propio trabajo. Esto se debe a que uno de los objetivos de esta tarjeta impresa es ser un material educativo se ha diseñado la incorporación de un acelerómetro que use un protocolo de comunicación distinto al GPS y a Bluetooth.

Este acelerómetro usará un protocolo de comunicación I2C. El modelo no ha sido escogido ya que se deja a disposición del próximo usuario la elección de este módulo.

---

## 3 DISEÑO DE TARJETA IMPRESA

---

**P**ara la realización de esta PCB se ha utilizado el software Altium Designer. Altium es un software de circuitos impresos reconocido a nivel mundial, el software, además, integra diferentes funciones que han sido de gran utilidad durante el desarrollo del proyecto como el gestor de materiales y espesores que posee para estimar el ancho de las pistas en función de la impedancia y corriente que debe soportar.

La idea del diseño es recrear el chasis de metacrilato de la plataforma original, que se usa de manera frecuente en los robots móviles, en un PCB que sirva como soporte del DSP TMS320F28335. Para esto, sabemos, además, que la control stick que se usará es la versión R2.2.

Para la creación de un proyecto autocontenido se ha diseñado una librería, la cual integra todos los componentes necesarios para la creación de este.

La creación de una nueva librería de componentes que se usarán en el diseño de la PCB es primordial, para esto, hemos usado agregadores de componentes, como es SnapEDA [10]. Asimismo, se tenía que generar componentes individuales, creando tanto la huella como el símbolo haciendo uso del propio software. Esta librería está incorporada en el proyecto de manera que todos los componentes empleados en el desarrollo del trabajo están incluidos dentro del proyecto.

Las librerías de Altium se dividen en símbolo y huella, siendo el símbolo una representación lógica que asocia los pines del dispositivo con su funcionalidad y que se usará en el esquemático. Por ejemplo, en la creación de un diodo, se dibuja el símbolo de un diodo Figura 17 y se asocia a cada terminal de este con el pin que corresponde al cátodo y al ánodo.

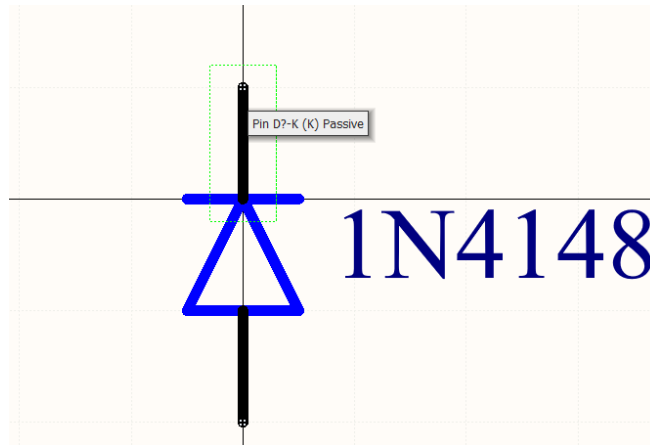


Figura 17. Ejemplo de símbolo creado en Altium Designer.

La librería PCB contiene la huella asociada al componente que será empleado, posteriormente, en el diseño del PCB. Esta huella se representa tanto en 2D como en 3D, siendo esta última especialmente útil para comprobar el volumen y la altura final del prototipo. En esta librería es importante asociar los pad a los pines anteriormente mencionados, es decir, en el caso del diodo indicar si el pad corresponde al cátodo o al ánodo.

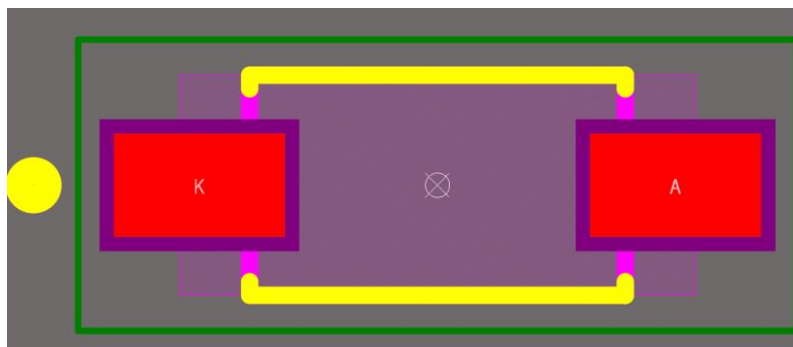


Figura 18. Ejemplo de huella creado en Altium Designer.

Además, sobre esta huella podemos situar pistas que no sean de cobre, como son en este caso las líneas amarillas que se muestran en la Figura 18 que se corresponde con la capa Silkscreen (capa empleada para mostrar la serigrafía sobre la máscara de soldadura). También permite añadir capas mecánicas para mostrar algún aspecto importante del componente en cuestión. Esto se suele utilizar para indicar el primer pin de cada componente, o el nombre del componente.

### 3.1 Esquemático

A continuación, procederemos a describir la organización del esquemático del proyecto.

#### 3.1.1 Vista General

Para la representación del esquemático, se ha optado por hacer uso de un diseño jerárquico, de forma que la primera hoja del esquemático actúe como una representación a nivel de diagrama de bloques de todo el sistema. Para ello, hemos usado una de las herramientas que nos permite el software de Altium como es la creación de conjuntos, estos conjuntos representarán los diferentes periféricos presentados en la sección anterior. De esta manera, la primera hoja del esquemático muestra de una forma rápida y sencilla la composición general del proyecto.

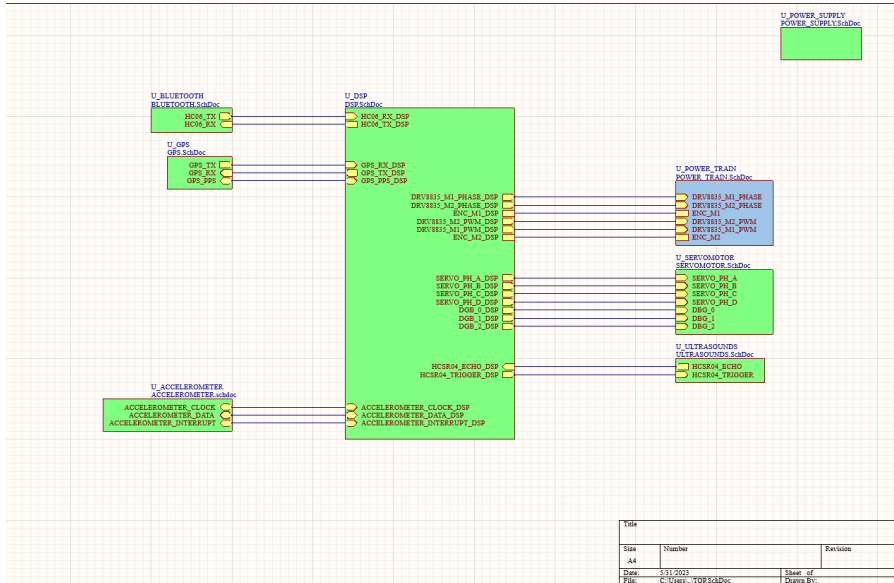


Figura 19. Esquemático General

En la Figura 19 se puede observar de manera rápida como es la disposición de los componentes principales del proyecto. En la posición central encontramos el DSP de donde salen diferentes conexiones a los periféricos que hemos nombrado anteriormente. Además de estas conexiones desde el DSP podemos observar como también tenemos situada la alimentación del sistema en la esquina superior de la derecha.

### 3.1.2 DSP

Dentro de la carpeta del DSP tendremos todas las conexiones que estén directamente conectadas a nuestra control stick, en este caso es un DIM100.

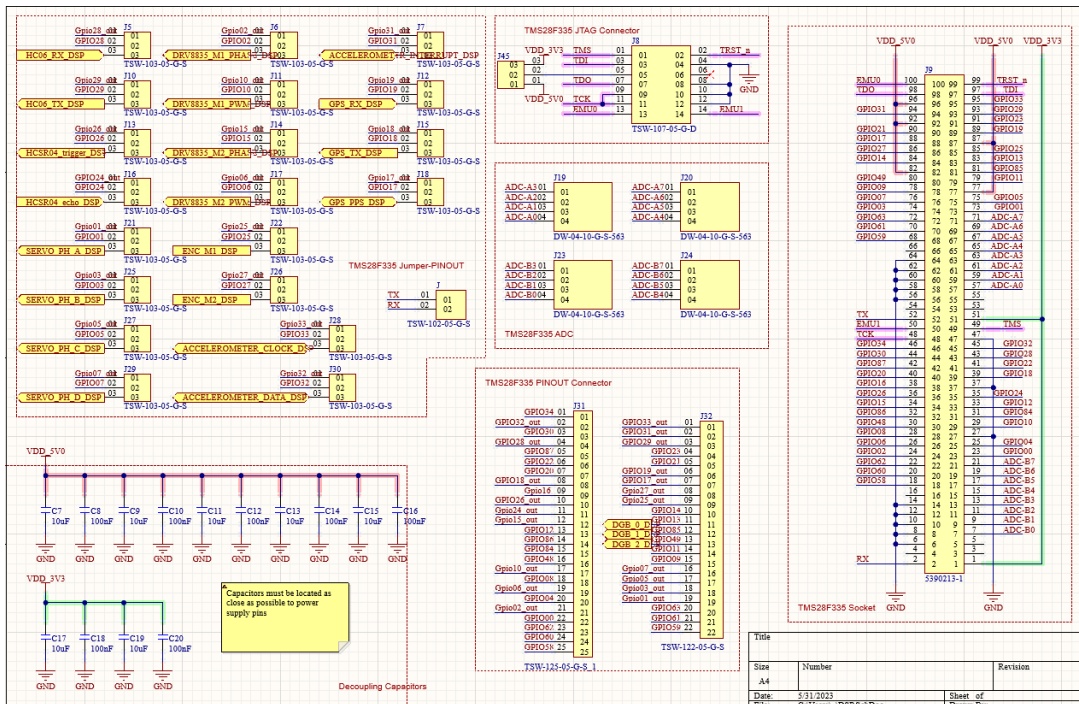


Figura 20. Esquemático DSP

Como se puede observar en la Figura 20 dentro del documento del esquemático (.SchDoc) referido al DSP se ha dividido a su vez en diferentes subgrupos que se explicaran a continuación.

Empezando por la derecha como base tenemos el DIM100, este es el conector que se usará para insertar nuestra control stick donde estará en microprocesador, es por eso por lo que tiene un alto número de conexiones y todas están referidas a los pines del TMS320F28335. Continuando nos encontramos con tres grupos de conectores situados en la parte central de la imagen. Siendo estos grupos los conectores del JTAG, ADCs y GPIOs. Se ha considerado incorporar un pinout a todas las conexiones por si fuera necesario utilizar alguna de las funciones que dispone el DSP.

En cuanto al JTAG, este es un conector específico. Este es utilizado para la programación de nuestro microprocesador. Su conexión es de vital importancia y es por eso por lo que como podemos observar uno de sus pines está al aire. Esto se debe a que el conector del JTAG tiene un pin ciego, para evitar mal conexiones y provocar algún tipo de fallo en el sistema [11].

En la esquina superior izquierda podemos encontrar un total de 20 conectores Jumpers. Todos estos Jumpers tienen la misma función, la de seleccionar el pinout que tendrá nuestro DSP. En nuestro diseño como se ha comentado antes dejaremos la opción de tener un pinout auxiliar para conectar a todos los pines del DSP, esto puede incorporar problemas cuando queramos utilizar nuestros periféricos, ya que estas hileras de pines que estará al descubierto pueden actuar como receptor de alguna señal y provocar un ruido en la señal de nuestro periférico. Es decir, todos estos jumpers tienen la opción de facilitar la conexión directa desde nuestro DSP hasta el periférico o a nuestras hileras de pines auxiliares, aislando el que no hayamos elegido y evitando que sufra ruido electrónico.

### 3.1.3 Condensadores

Se han situado una pareja de condensadores de 10uF y 100nF por cada periférico de manera que, si existiera un pico de tensión, en cualquier punto del sistema no afectase al resto de componentes. Esta manera de proteger el sistema hace que los cambios de tensión provocados de manera no deseada sean unos cambios lentos en el tiempo aportando de esta manera una mejora de estabilidad al sistema. Por esto es importante situar una pareja de condensadores, uno de ellos actuará de manera rápida y el otro de una manera más lenta, pero con mayor capacidad de actuación. Además, es importante que estos condensadores se sitúen cerca del posible fallo, es decir, estas parejas están situadas lo más próximas posibles a todos los periféricos.

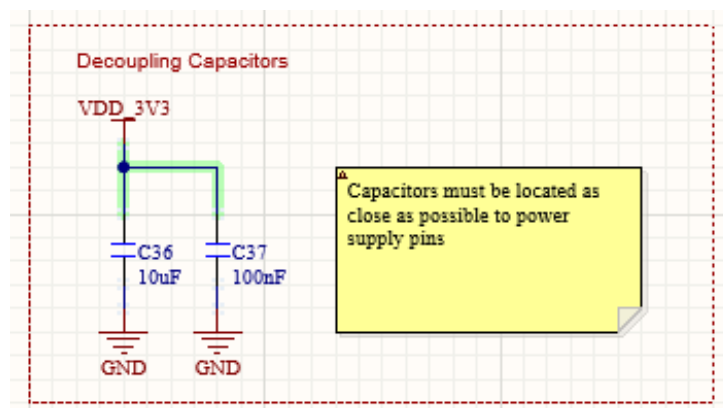


Figura 21. Esquemático de los condensadores de estabilización y filtrado.

### 3.1.4 Alimentación

Para la alimentación de la plataforma móvil, se ha optado por una alimentación a través de un conjunto de baterías que suministre 12V. A continuación, estos 12V se convierten, gracias a un DC/DC, a 5V que serán utilizados por el DSP. Finalmente, los periféricos y una parte del DSP serán alimentados a 3.3V; por lo que será necesario una conversión de 5V-3.3V, a través de un regulador lineal. Hemos elegido un regulador lineal para reducir el posible ruido que el convertidor DC/DC introduzca en la línea de 5V.

El motivo de la alimentación de 3.3V sobre los periféricos es debido a que el microprocesador estará alimentado dentro de control stick a estos mismo 3.3V gracias a un DC/DC que viene incorporado a la propia stick.

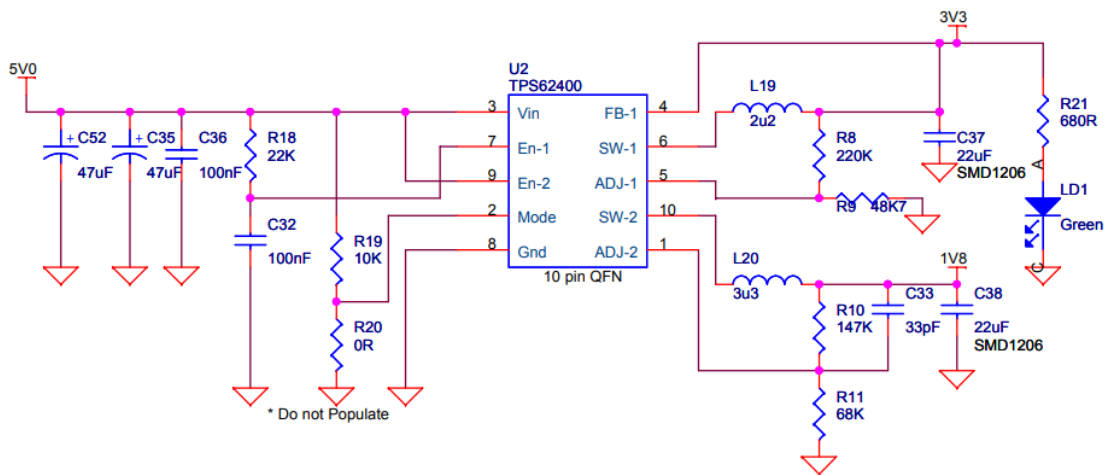


Figura 22. Alimentación que incorporar el stick donde se localiza el DSP empleado[2].

Para esta conversión en cascada se han elegido dos DC/DC, el primero LM2596-5 cuyo esquemático ha sido obtenido de la propia hoja de especificaciones del fabricante como se muestra a continuación [12].

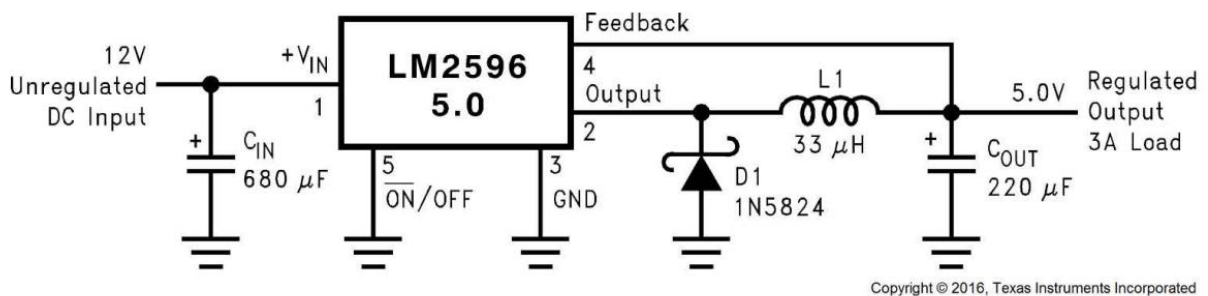


Figura 23. Esquemático LM2596-12 según fabricante[12]



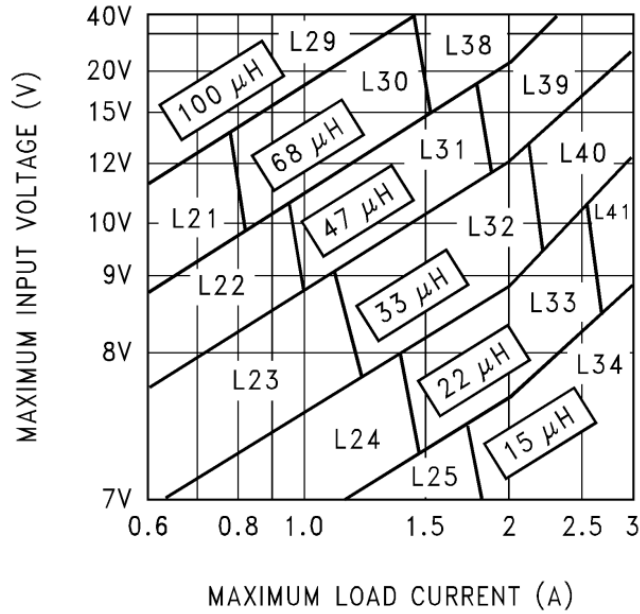


Figura 24. Valor Inductancia en función Vin Iload[12]

Como se puede observar en la gráfica, para una tensión de entrada de 12V y con los cálculos de potencias presentados en el capítulo anterior, Tabla 2, se ha previsto un consumo menor de 2A la inductancia adecuada es de 33uH (L32)

A continuación, para la conversión de 5V a 3.3V se ha elegido el regulador de tensión lineal LM317 que al igual que el anterior se ha elegido su esquemático basándose en el datasheet del fabricante [13].

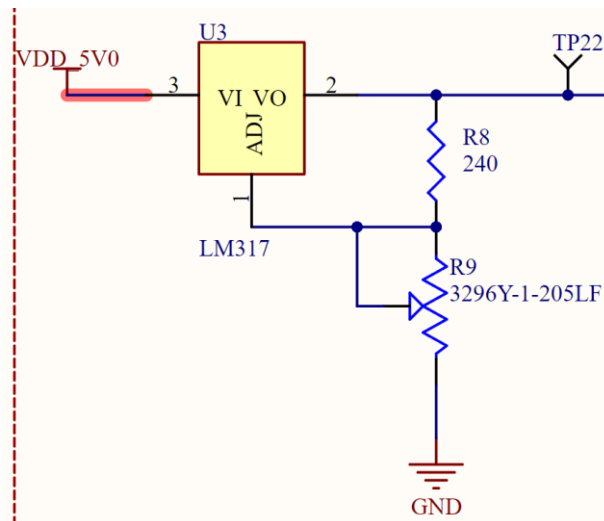


Figura 25. Esquemático LM317

Los valores de las resistencias R8 y R9 son necesarios para ajustar el valor de la tensión de salida en función de la tensión de referencia. Para poder ajustar esta tensión de salida con facilidad a los 3.3V se ha elegido un potenciómetro como Resistencia variable. Una vez montado el circuito se ajusta este valor de la Resistencia R9 para obtener en la salida una tensión de 3.3V.

Además de todo lo mencionado, se ha añadido como medida de seguridad un jumper a la salida de cada DC/DC el cual no alimentara la PCB si no está colocado. De esta manera, se podrán realizar la prueba de la tarjeta de forma fiable y evitar que posibles cortocircuitos provoquen la destrucción de los componentes.

Finalmente, se han añadido dos condensadores para evitar grandes variaciones de la tensión de salida que provoquen un mal funcionamiento del resto de componentes del sistema.

Además del esquemático de los dos DC/DC se ha incorporado un Pinout auxiliar de alimentación. Este diseño se ha realizado con la idea de poder conectar si fuera necesario en algún proyecto posterior algún dispositivo nuevo. Estos pines servirían como la alimentación de todos estos nuevos periféricos.

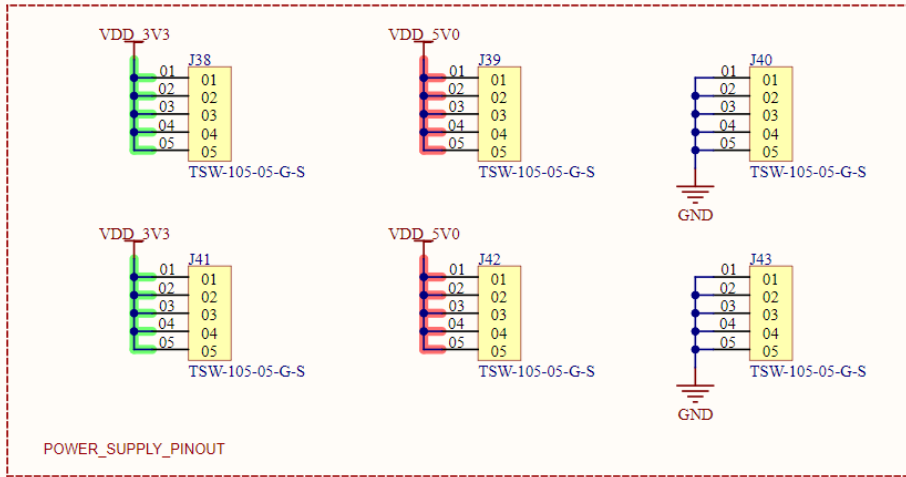


Figura 26. Pinout de Alimentación

Finalmente, integrando todo lo relacionado con la alimentación del sistema se ha añadido todo al mismo documento de esquemático donde a su vez se separa en pequeñas secciones para poder estudiar cada punto por separado.

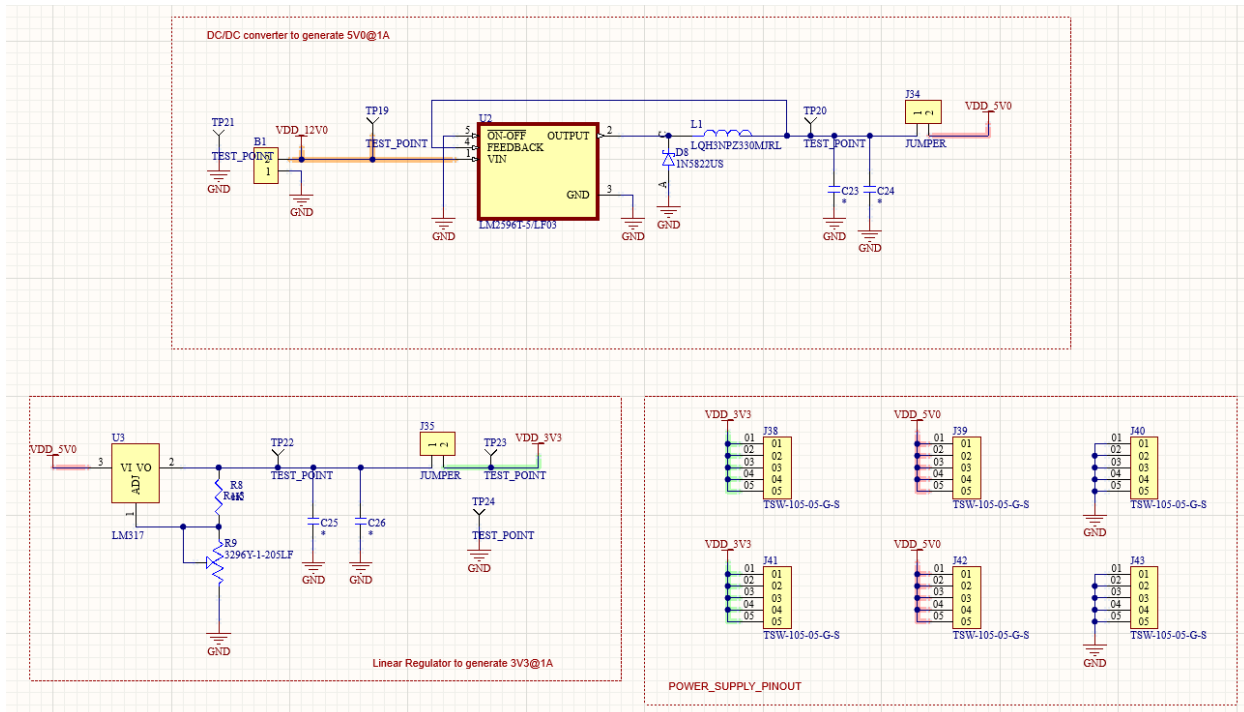


Figura 27. Esquemático final de la etapa de alimentación.

### 3.1.5 Tren de potencia

La Figura 28 muestra la hoja de esquemático que integra el driver DRV8835, responsable de controlar los motores DC que mueven la plataforma desarrollada.

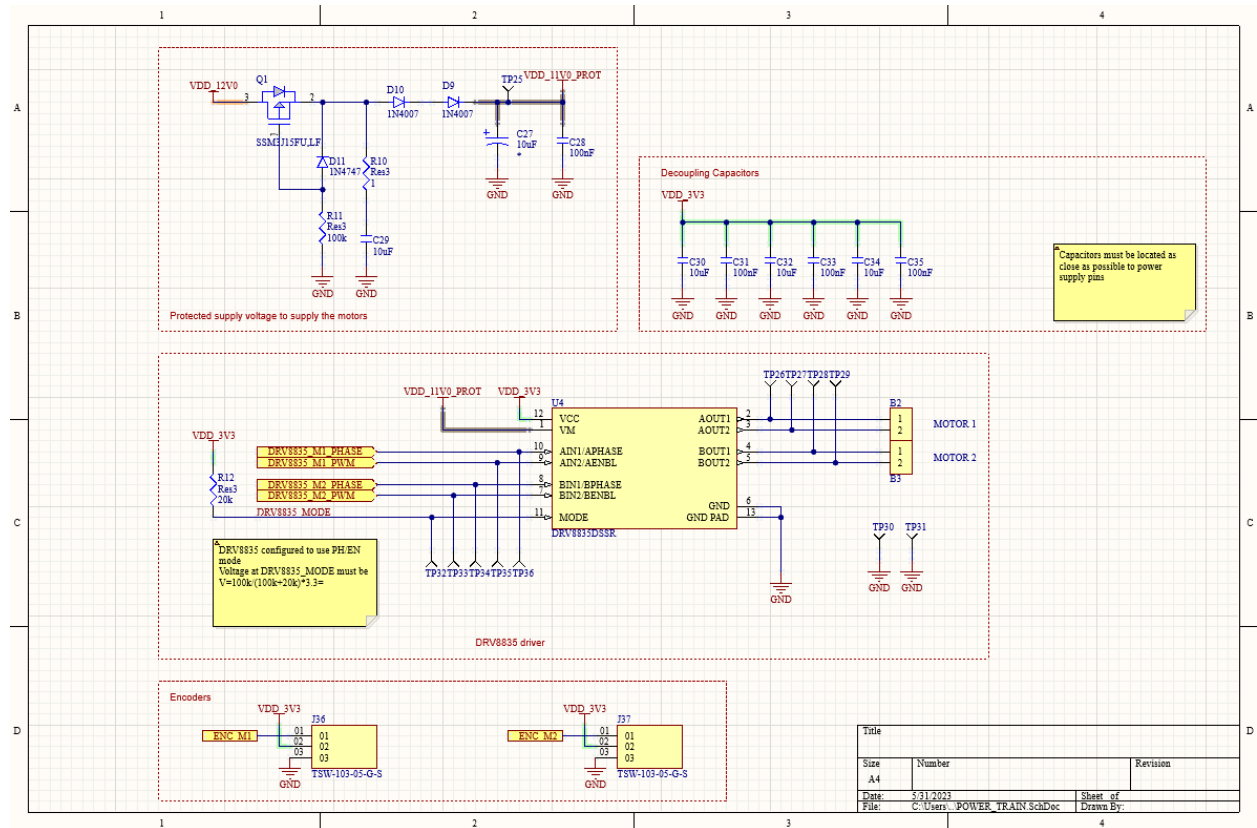


Figura 28. Esquemático asociado al tren de potencia

#### 3.1.5.1 Driver

Para el correcto funcionamiento del robot móvil, tendremos en cuenta el primer periférico, el driver para los motores de las ruedas. Este driver usa cuatro pines distintos del microprocesador, siendo dos pines de salida lógica y los otros dos pines con una función de PWM. Estos pines son necesarios para el control del puente H que usará el DRV8835 para el control de las ruedas.

Tabla 9. GPIO motores

Puerto	Terminal
<b>GPIO02</b>	M1-PHASE
<b>GPIO10</b>	M1-PWM
<b>GPIO06</b>	M2-PWM
<b>GPIO15</b>	M2-PHASE

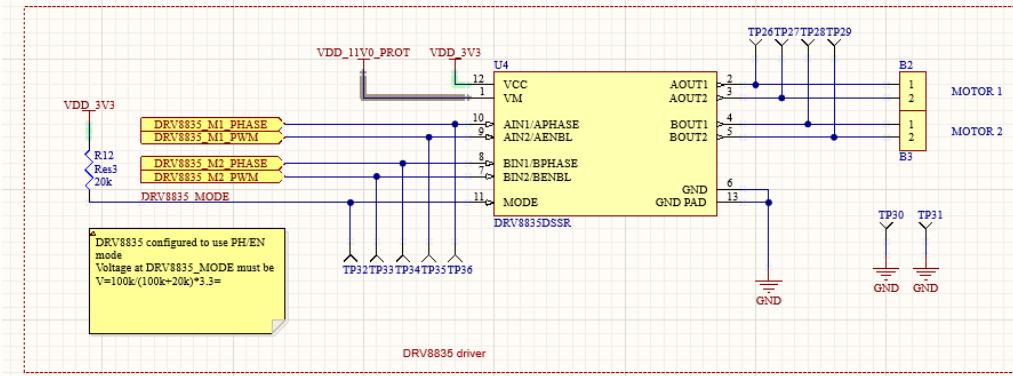


Figura 29. Esquemático Driver de los motores

### 3.1.5.2 Alimentación protegida

La tensión que usa el DRV8835 para la alimentación de los motores tiene una medida de seguridad adicional. Esta protección añadida protege al resto del sistema si ocurre algún problema en el puente H. Se ha diseñado un transistor que separa la tensión del sistema de la tensión que alimenta el puente H, de manera que un fallo en el puente H no afecte al resto del sistema. Además, se han situado dos diodos en serie a la salida de este sistema de protección, ya que la tensión máxima para el componente DRV8835 no puede superar los 11V.

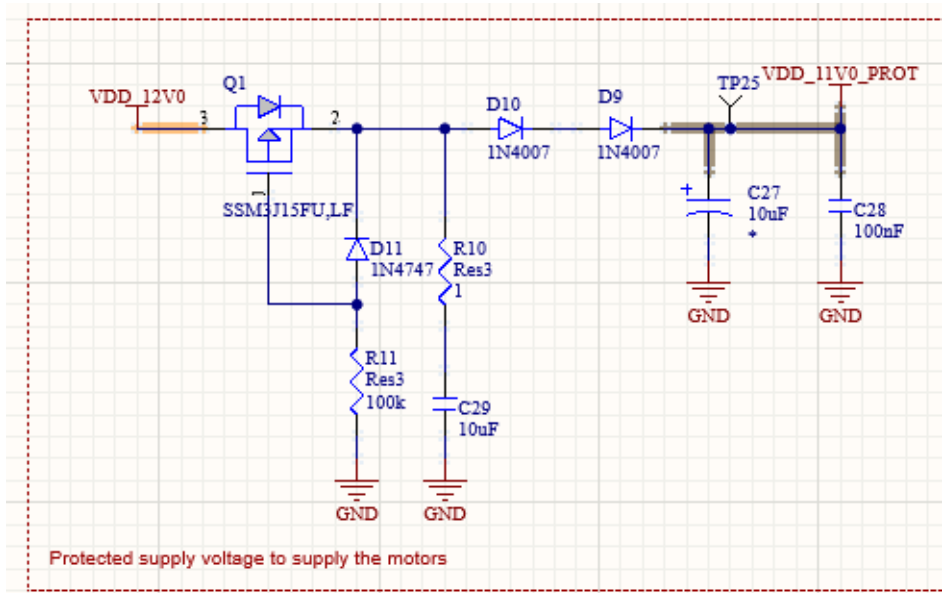


Figura 30. Esquemático de la línea de alimentación protegida para los motores

### 3.1.5.3 Encoder

El modelo que usará para el cálculo de la velocidad en ambas ruedas del robot será HC-020K

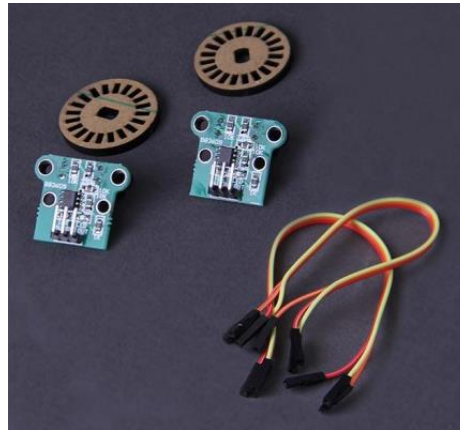


Figura 31. Modelo encoder HC-020K[5]

Los pines elegidos para los dos encoders serán dos que puedan ser usados con función de Ecap

Tabla 10. Pines Encoder

Puerto	Terminal
<b>GPIO25</b>	Out1
<b>GPIO27</b>	Out2

El esquemático en este caso solo será una conexión con los propios módulos.

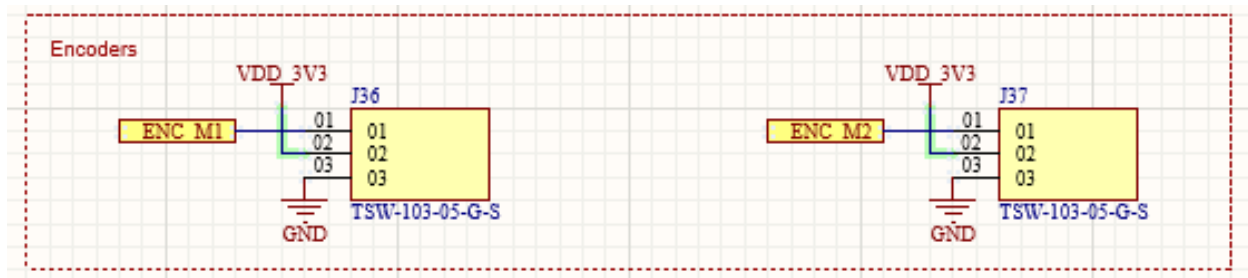


Figura 32. Esquemático Encoders

### 3.1.6 Bluetooth

Otro periférico que se usará en el Proyecto es un dispositivo Bluetooth, este usara una comunicación con el microprocesador SCI. El módulo bluetooth que se usará será HC-06.

Este periférico será usado para una comunicación con el controlador del vehículo de manera inalámbrica. La comunicación se realizará a través de una aplicación móvil que pueda enviar y recibir caracteres con el módulo HC-06 para que este se comunique con el microprocesador a través de la comunicación SCI.

Según el datasheet los pines reservados para la comunicación SCI son los GPIO 28 para TX y el GPIO 29 RX

Tabla 11. Pines Bluetooth

Puerto	Terminal
<b>GPIO28</b>	TX_BLUETOOTH
<b>GPIO29</b>	RX_BLUETOOTH

Es importante fijarse que para que esta comunicación sea correcta los pines de comunicación TX y RX deben de estar intercambiados para que el pin de transmisión del microprocesador sea el pin de recepción del Bluetooth.

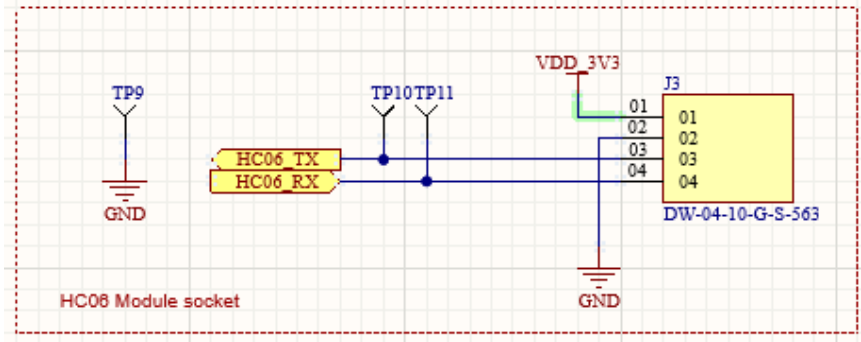


Figura 33. Esquemático Bluetooth

### 3.1.7 GPS

El Proyecto tendrá dos periféricos de comunicación, este Segundo será un módulo GPS, que usará el mismo protocolo de comunicación que el módulo Bluetooth.

Este módulo será de referencia para indicar la posición del robot autónomo en todo momento, teniendo una comunicación tipo SCI con el microprocesador. Esta comunicación tiene reservado los GPIOs 18 TX y GPIO 19 RX de la SCI-B.

Tabla 12. Pines GPS

Puerto	Terminal
<b>GPIO19</b>	TX-GPS
<b>GPIO18</b>	RX-GPS
<b>GPIO17</b>	PPS

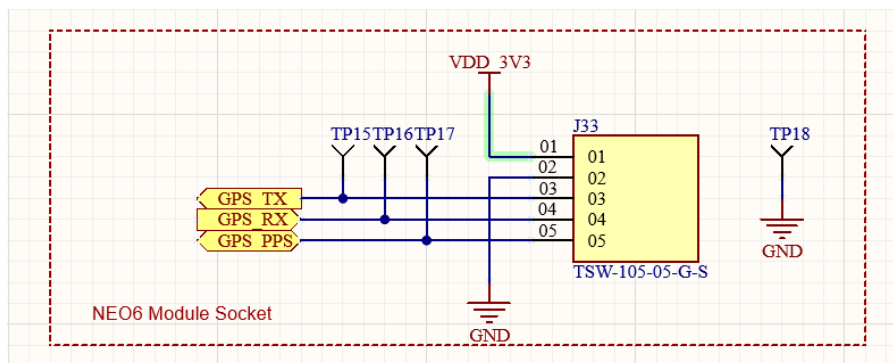


Figura 34. Esquemático GPS

### 3.1.8 Servo motor

Para el control del motor paso será necesario además de la propia alimentación del motor un driver para el control directo a través del microprocesador. Como se ha indicado en la descripción del sistema este driver es el ULN2003.

En este caso como todos los pines serán GPIOs de salida no hay ningún criterio de selección.

Tabla 13. Pines Servo motor

Puerto	Terminal
<b>GPIO01</b>	INPUT1
<b>GPIO03</b>	INPUT2
<b>GPIO05</b>	INPUT3
<b>GPIO07</b>	INPUT4

Para el diseño del esquemático será necesario el propio Driver además del conector del motor paso a paso como se indicó en los periféricos. Además, estarán acompañados de unos leds para poder realizar pruebas con estos y ver cómo realmente funcionan.

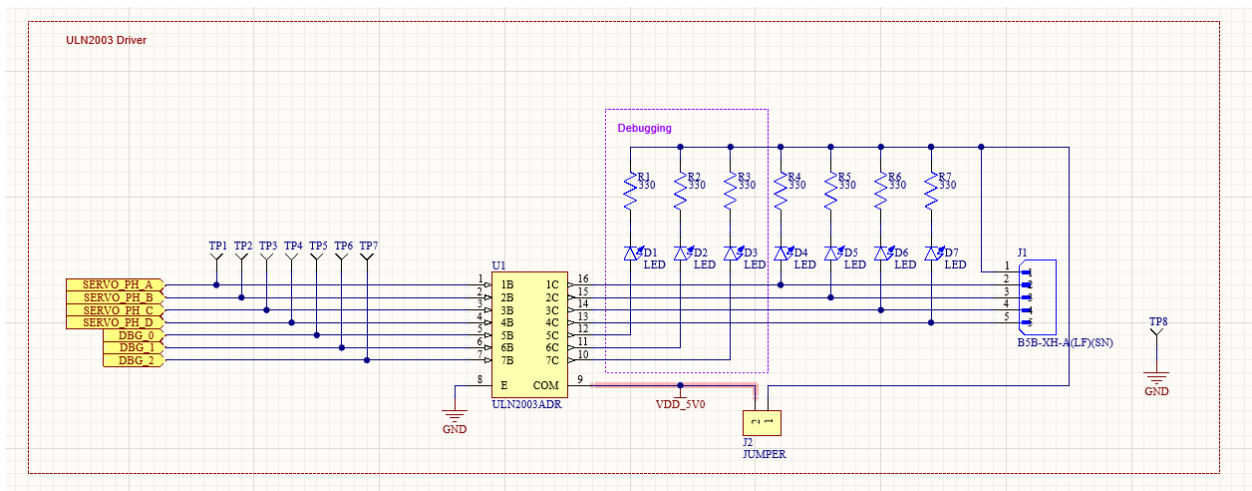


Figura 35. Esquemático Servomotor

### 3.1.9 Sensor ultrasónico

Los pines elegidos para el módulo HC SR04 serán para el trigger un GPIO 26 de salida y para el echo será relevante que la GPIO 24 tenga la función de ecap

Tabla 14. Pines Sensor Sonar

Puerto	Terminal
<b>GPIO24</b>	Ecap
<b>GPIO26</b>	Trigger

El esquemático en este caso solo será una conexión con el propio módulo.

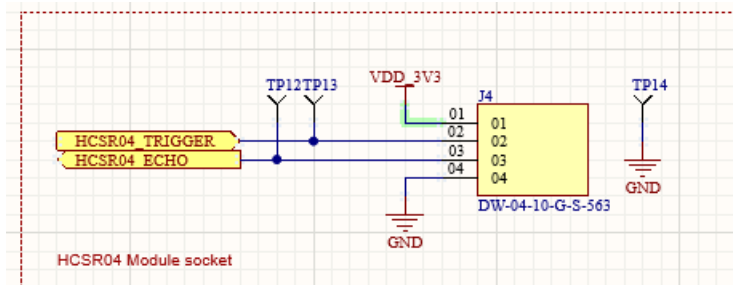


Figura 36. Esquemático de Sensor de proximidad

### 3.1.10 Acelerómetro

Este será un periférico adicional que no se usará en el proyecto, pero se dejará preparado para poder ser usado más adelante. Este tipo de periférico usa una comunicación I2C.

Como el periférico usa una comunicación I2C los pines deben tener este tipo de función, además de un tercer pin de interrupción que será un pin genérico de entrada.

Tabla 15. Pines Acelerómetro

Puerto	Terminal
<b>GPIO33</b>	clock open-drain bidirectional port
<b>GPIO32</b>	data open-drain bidirectional port
<b>GPIO31</b>	interrupt

El esquemático en este caso solo será una conexión con el futuro módulo del acelerómetro.

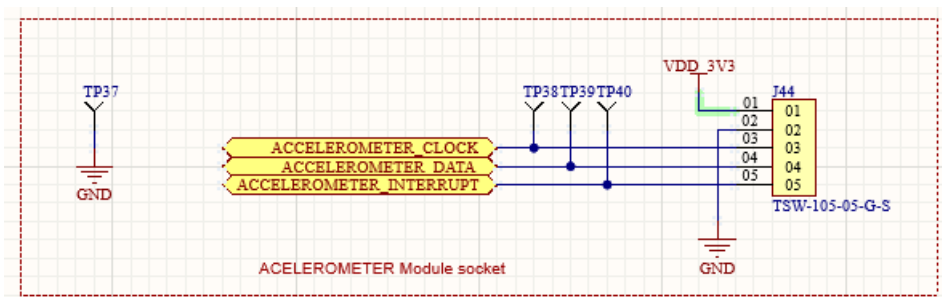


Figura 37. Esquemático del acelerómetro

## 3.2 Implementación física

Una vez realizado todo el esquemático y con las librerías asociadas a cada componente con una huella específica comienza el proceso de ruteo. La PCB ha sido enviada a fabricar a través de la página web JLCPCB por lo que se ha tenido que revisar sus normas de fabricación y adaptarlas a nuestra propia PCB. Esto se realiza dentro de Altium donde te deja cambiar las normas de diseño ajustando así en función de donde se mande a fabricar.



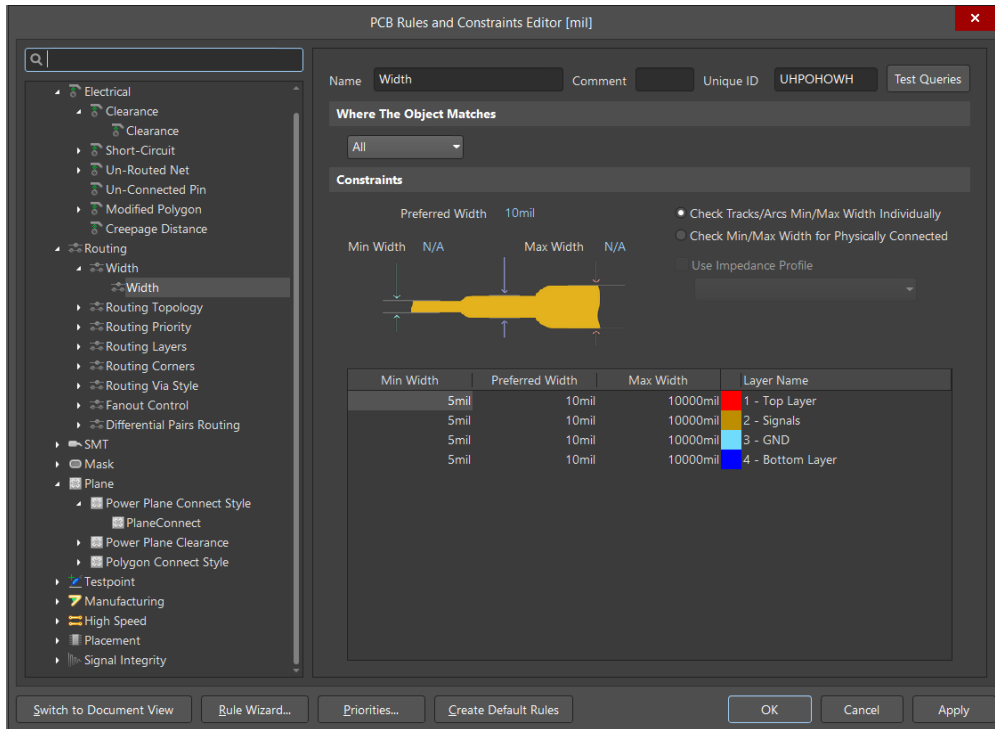


Figura 38. Reglas de diseño para la PCB

Continuando con las reglas de diseño adecuadas debemos ajustar la forma que tendrá el PCB final ya que de manera predeterminada Altium nos ofrece un rectángulo. Como ya hemos mencionado nuestro PCB tendrá la forma del chasis que poseen muchos de robots móviles en el mercado.

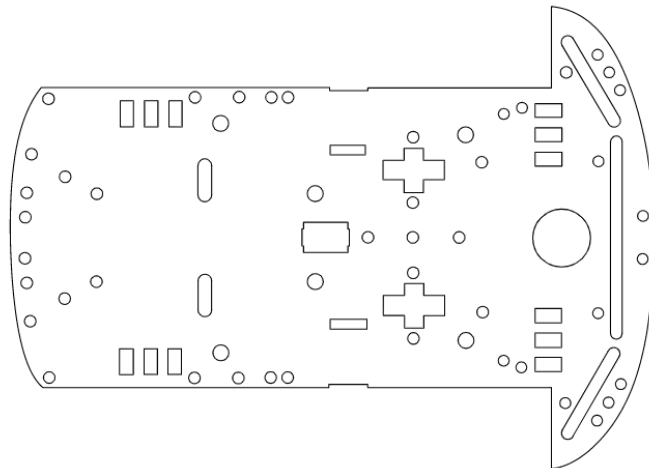


Figura 39. Forma de la PCB

Una vez tenemos diseñada la forma que tendrá nuestra PCB será implementada en el software de Altium. Además de manera predeterminada Altium no ofrece un stack de capas de 2, la cara superior y la cara inferior. Para el Desarrollo del trabajo se ha optado por una PCB de cuatro 4 capas, es decir la capa superior e inferior y dos capas adicionales posicionadas entre estas dos.

Teniendo en cuenta que realizaremos un trabajo en cuatro capas es razonable reservar una de estas capas como un plano de tierra para tener desde cualquier punto del diseño una rápida conexión a tierra en caso de fallo. El resto de las capas se han organizado de tal manera que la capa superior se intente reservar para aquellos componentes que sea del modelo SMD, es decir que no atraviesan la PCB si no que descansan sobre esta, y una de las capas internas intentar reservarla lo máximo posible para las conexiones de alimentación.

#	Name	Material	Type	Thickness	Weight	Dk	Df
	Top Overlay		Overlay				
	Top Solder	Solder Resist	Solder Mask	0.4mil		3.5	
1	Top Layer		Signal	1.4mil	1oz		
	Dielectric 2	PP-006	Prepreg	2.8mil		4.1	0.02
2	GND	CF-004	Plane	1.378mil	1oz		
	Dielectric 3	PP-006	Prepreg	2.8mil		4.1	0.02
3	VDD	CF-004	Signal	1.378mil	1oz		
	Dielectric 1	FR-4	Dielectric	12.6mil		4.8	
4	Bottom Layer		Signal	1.4mil	1oz		
	Bottom Solder	Solder Resist	Solder Mask	0.4mil		3.5	
	Bottom Overlay		Overlay				

Figura 40. organización de capas

### 3.2.1 Floorplanning

Una vez realizados los pasos previos al diseño de nuestra PCB, como han sido la elección de las normas de diseño y la elección de los planos y su funcionalidad, debemos llevar a cabo el floorplanning de los diferentes componentes.

La idea del floorplanning es distribuir los componentes sobre la PCB con el objetivo de minimizar el número de cruces entre los diferentes elementos y las posibles interferencias; así como garantizar que aquellos elementos que deben ser accesibles están ubicados en una zona que permite este hecho. Esto se realiza en función de sus tamaños, de sus conexiones o de otros datos relevantes como consumo, disipación térmica o funcionalidad. Por ejemplo, la bornera de conexión a los motores de continua que mueven las ruedas deben situarse donde se vayan a localizar las ruedas para así evitar conexiones largas a través de pistas.

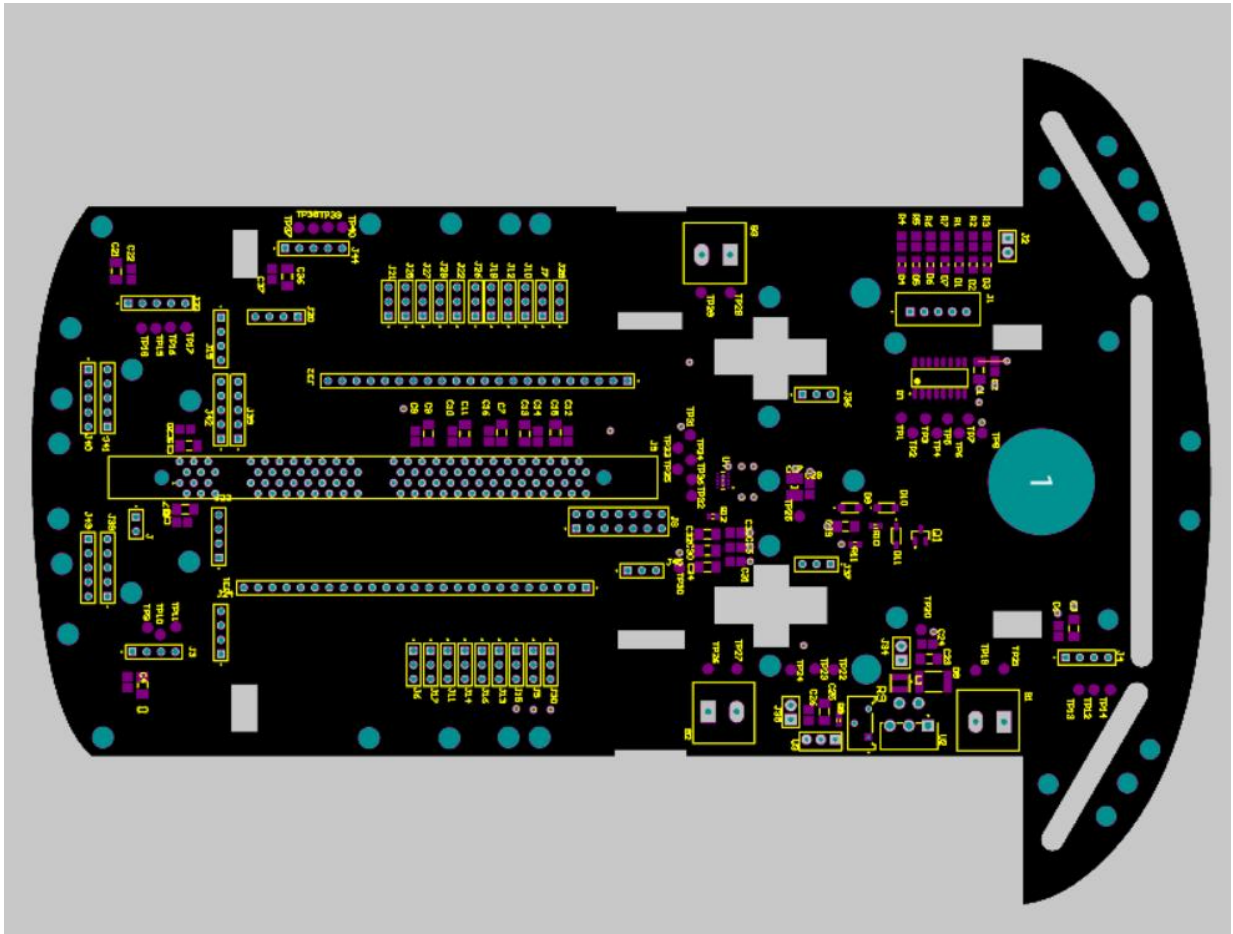


Figura 41. Floorplanning

Analizando la Figura 1Figura 41 detalladamente, podemos observar como el DIM100 está situado en la zona central ya que de este componente es de donde salen más conexiones y debe estar próximo de todas ellas. Además, se ha situado en la parte posterior del robot para evitar los agujeros necesarios para la sujeción de las ruedas.

Alrededor de la control stick podemos ver cómo están todos los conectores auxiliares que se nombraron en el diseño del esquemático. Siendo las hileras más largas las de las GPIOs y las hileras de pines situadas en la zona inferior las destinadas a los ADCs o la Alimentacion. Hay que destacar como en la zona exterior de la hilera de pines de las GPIOs podemos observar los jumpers. Estos Jumpers están diseñados para que si queremos utilizar todos los periféricos que dispone el robot debamos ubicar el jumper entre el pin central y el situado más externo del centro de la placa.

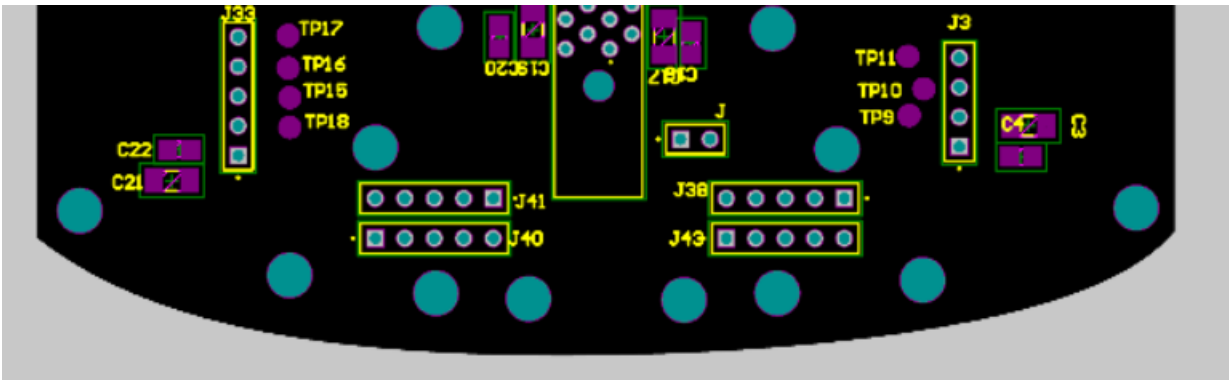


Figura 42. Floorplanning zona trasera

En la Figura 42 se muestra que en zona inferior del vehículo están situados J3 y J33 que serán los conectores del GPS y del Bluetooth. Se ha elegido esta ubicación ya que en la zona delantera del vehículo debido a que la ubicación de las ruedas y el servomotor será en la zona delantera del vehículo y estos componentes metálicos pueden provocar algún tipo de interferencia sobre el GPS se ha decidido situarlos en la zona contraria.

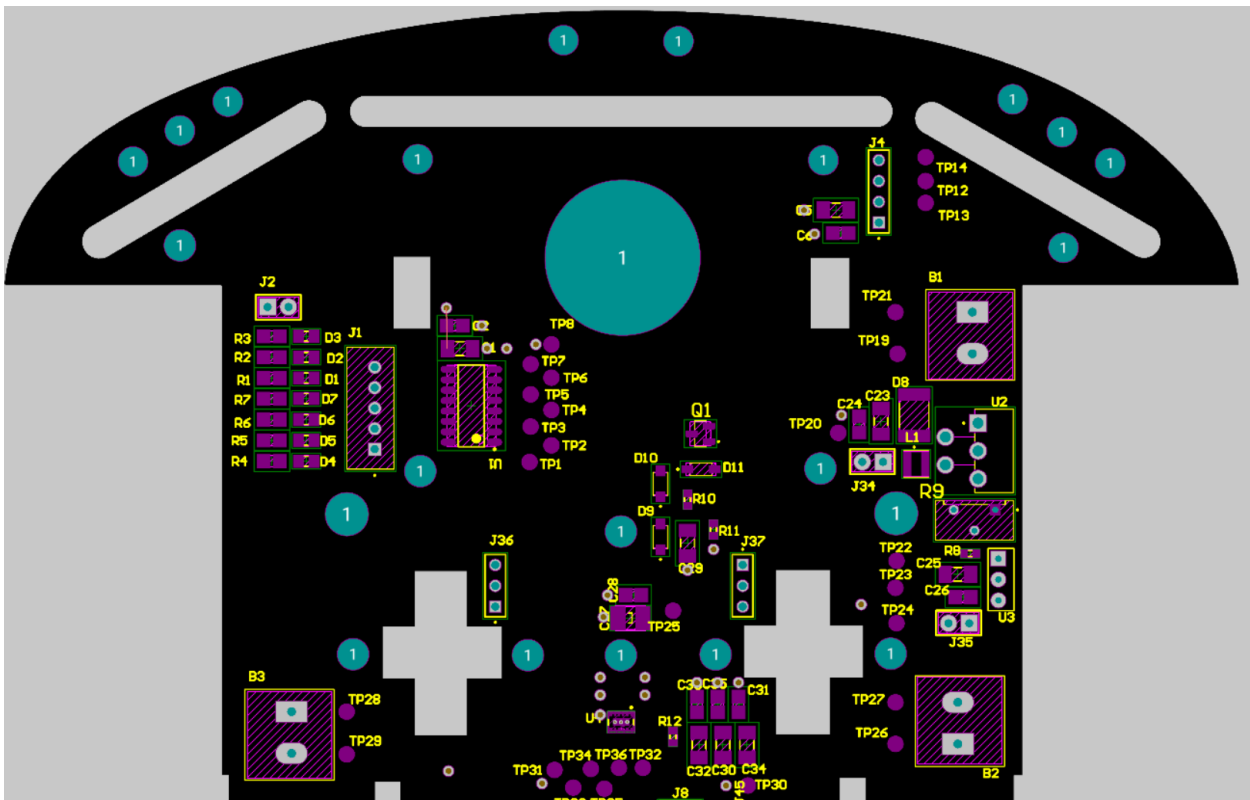


Figura 43. Floorplanning zona delantera

La Figura 43 muestra la zona delantera del robot podemos encontrar la alimentación de este situada en el borde de la placa siendo la conexión a la batería la bornera B1.

También podemos encontrar el conector del sensor de proximidad (J4) y el conector del motor paso a paso (J1) ya que este se situará sobre una plataforma en la zona delantera.

Finalmente hay que destacar la ubicación de las borneras que alimentan la ruedas que estarán situadas justo encima de la ubicación de la rueda. Además, el driver (U4) estará situado en el centro de estas dos para que la conexión desde driver y rueda sea la mínima posible.

Otro detalle del diseño son los condensadores, que como se mencionaron en el diseño del esquemático, todos estarán situados próximos a la alimentación de un periféricos. Además, se pueden observar Test Point, estos puntos de cobre que se encuentran en la capa superior y servirán para poder probar la placa y sacar datos y conclusiones con el osciloscopio.

Una vez ya se ha diseñado el floorplaning se debe se realizar el ruteo en las diferentes capas de cobre que se han elegido previamente sabiendo que una de estas capas estará reservada a un plano de tierra.

### 3.2.2 Rutado

Para el diseño de las pistas de cobre se deben de tener en cuenta una serie de criterios como que la distancia que conecte los dos puntos deseados sea mínima, ya que esta pista de cobre actúa como si fuera una resistencia. Además, por esta misma analogía eléctrica de que las pistas de cobre actúen como pequeñas resistencias se ha decidido crear sobre las mismas capas algunos planos de misma tensión de esta manera la resistencia disminuye.

También cabe destacar la importancia de no realizar giros en las pistas de cobre de 90° ya que esto provoca una subida de la resistencia en estos puntos, es por eso por lo que se han ajustado para que los giros de las pistas sean de 45°.

Por último, se deben evitar ajustar las pistas de cobre a los bordes exteriores, esto es debido a que durante el proceso de fabricación pueda existir algún tipo de corte provocando que estas pistas pierdan toda su funcionalidad.

Sobre la capa superior se ha diseñado situar todos los componentes del robot, esto es debido a la simplicidad que aporta a la hora de la visualización final del dispositivo. De esta manera se pueden observar todos los componentes agrupados en función de los esquemáticos anteriormente mencionados.

Cabe destacar que existen algunos componentes cuya huella sobre la PCB es Through Hole, es decir, una huella que una vía sobre la PCB, esto facilita que se puede conectar a cualquiera de las capas de cobre disponibles, pero habrá otros componentes con una huella SMD. Estos componentes tienen una huella que solo estará disponible sobre la capa superior, suponiendo así una dificultad si queremos conectar dos componentes con una huella SMD ya que nos vemos obligado a utilizar vías sobre la PCB o utilizar únicamente la capa superior.

Estos componentes con huella SMD pertenecen en mayor parte en su mayoría a la sección de distribución de la alimentación, por lo tanto, la zona cercana a la alimentación se ha elegido usar la capa superior para el ruteo de las pistas. Además de usar esta capa superior se han situado próximas a las huellas SMD que pertenece a la tensión de tierra una vía para realizar una conexión directa a tierra, aprovechando que nuestra PCB tiene un plano de tierra y así evitando conexiones excesivamente largas que puedan suponer un problema si existiese algún tipo de mal conexión.

Por último, en esta capa superior existen algunas conexiones de una tensión más elevada como es la alimentación de los 12V que llega desde la bornera hasta el driver de los motores. En este tipo de pistas se ha decidido realizar planos de tensión que se adapten mejor a la forma de la pista aprovechando al máximo el espacio disponible y minimizando así la resistencia que provoca una pista de cobre.

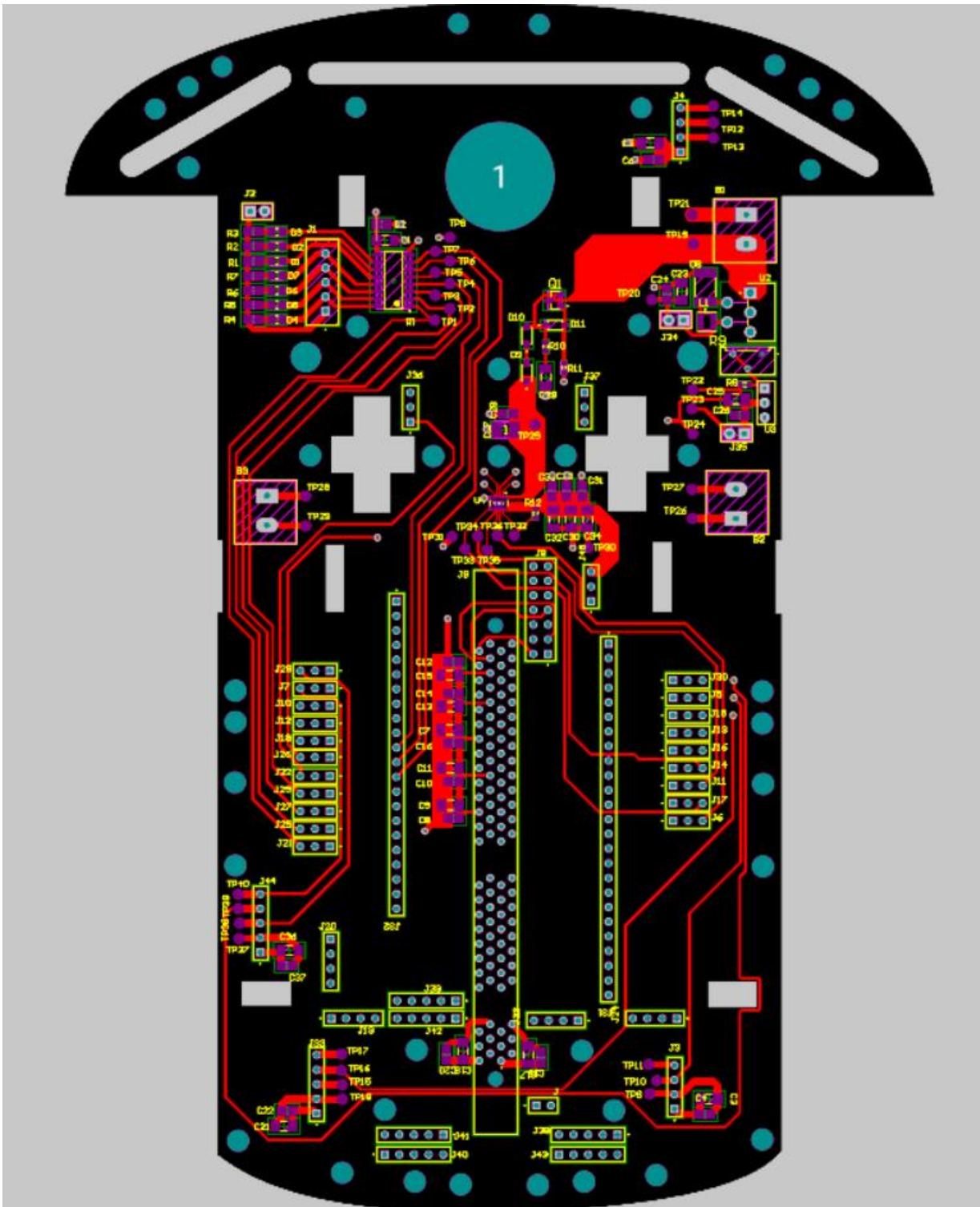


Figura 44. Rutado capa superior.

Sobre esta capa, aprovechando que no existen componentes con huella SMD, se ha decido realizar la pista entre los componentes que tengan ambos una huella Through Hole. De esta manera se minimizan las pistas de cobre en la capa superior que necesitara una mayor cantidad de pistas de cobre debido a las huellas SMD.



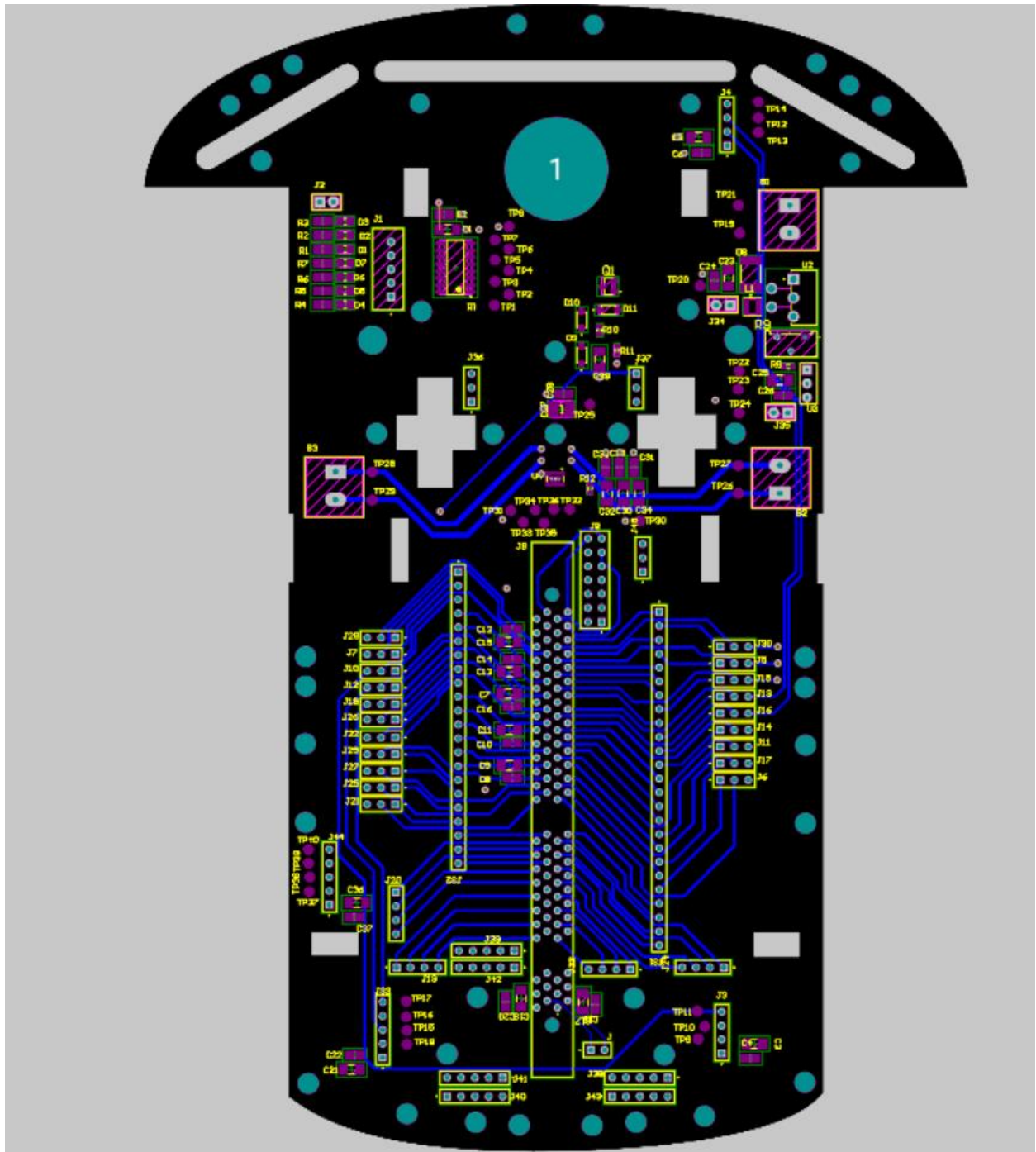


Figura 45. Rutada capa inferior.

La capa de alimentación se encuentra entre la capa superior y la capa inferior y se ha destinado de manera exclusiva a las conexiones de alimentación. En esta capa como se puede observar no se han usado pista de cobre si no planos. Con estos planos se minimizarán las resistencias entre los diferentes puntos de conexión.

Existen dos grandes planos de alimentación uno en la zona exterior (Plano verde) de la placa destinado a los 3.3V que usarán los periféricos que se usarán durante el trabajo. El plano central (plano rosa) tiene una tensión de 5V y la alimentación del microprocesador, así como de servo motor.

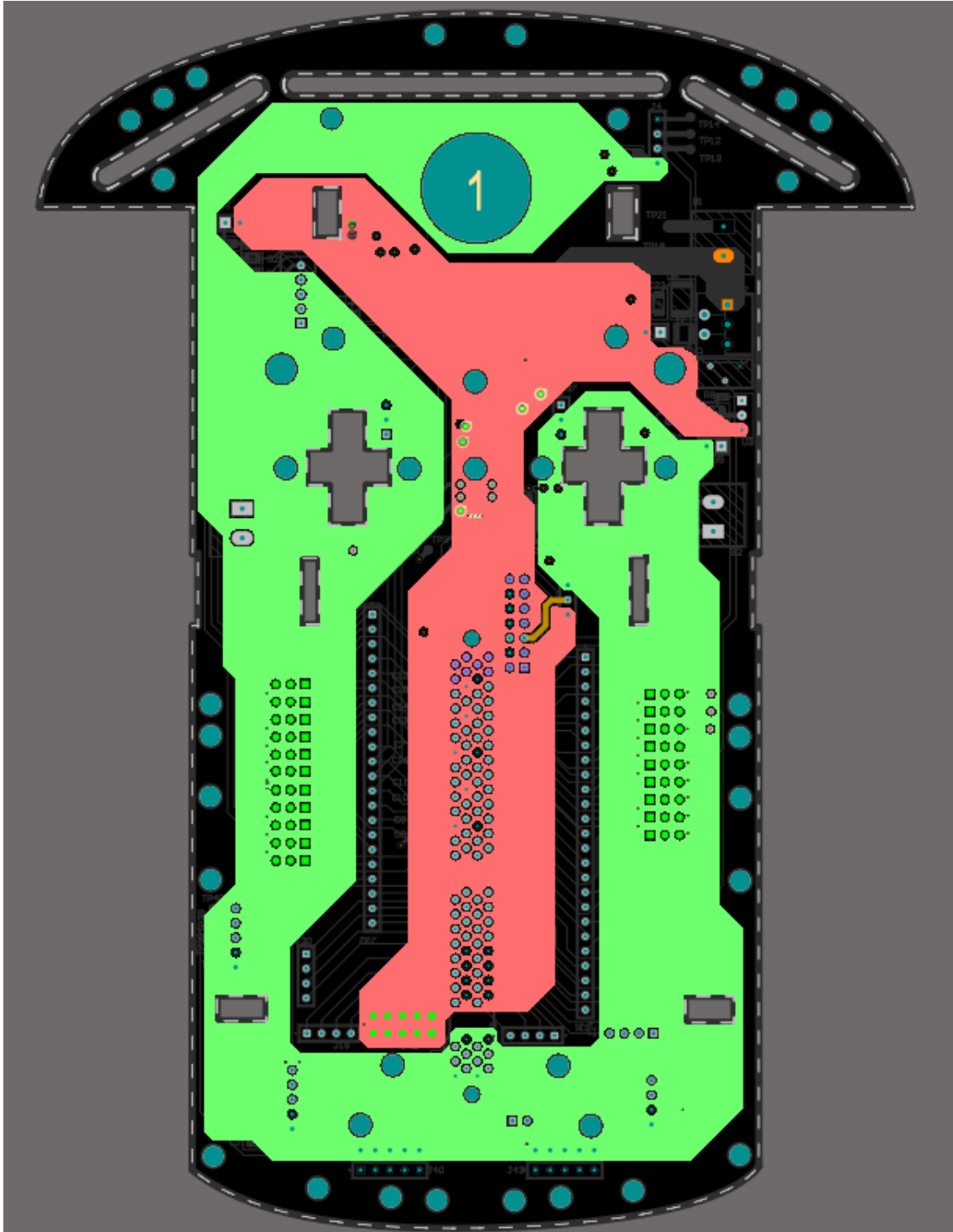


Figura 46. Rutada capa de alimentación.

Por último, existe un plano de tierra. Esta capa tiene conexiones directas desde cualquier vía a la tensión de tierra. De esta manera se realizan conexiones rápidas y desde cualquier punto de la PCB a GND. Cabe indicar que en la Figura 47, la representación del plano de tierra es negativa. Es decir, aquellas zonas coloreadas de color verde reflejan que no hay presencia de cobre mientras que el resto de la capa se corresponde con la presencia del cobre.



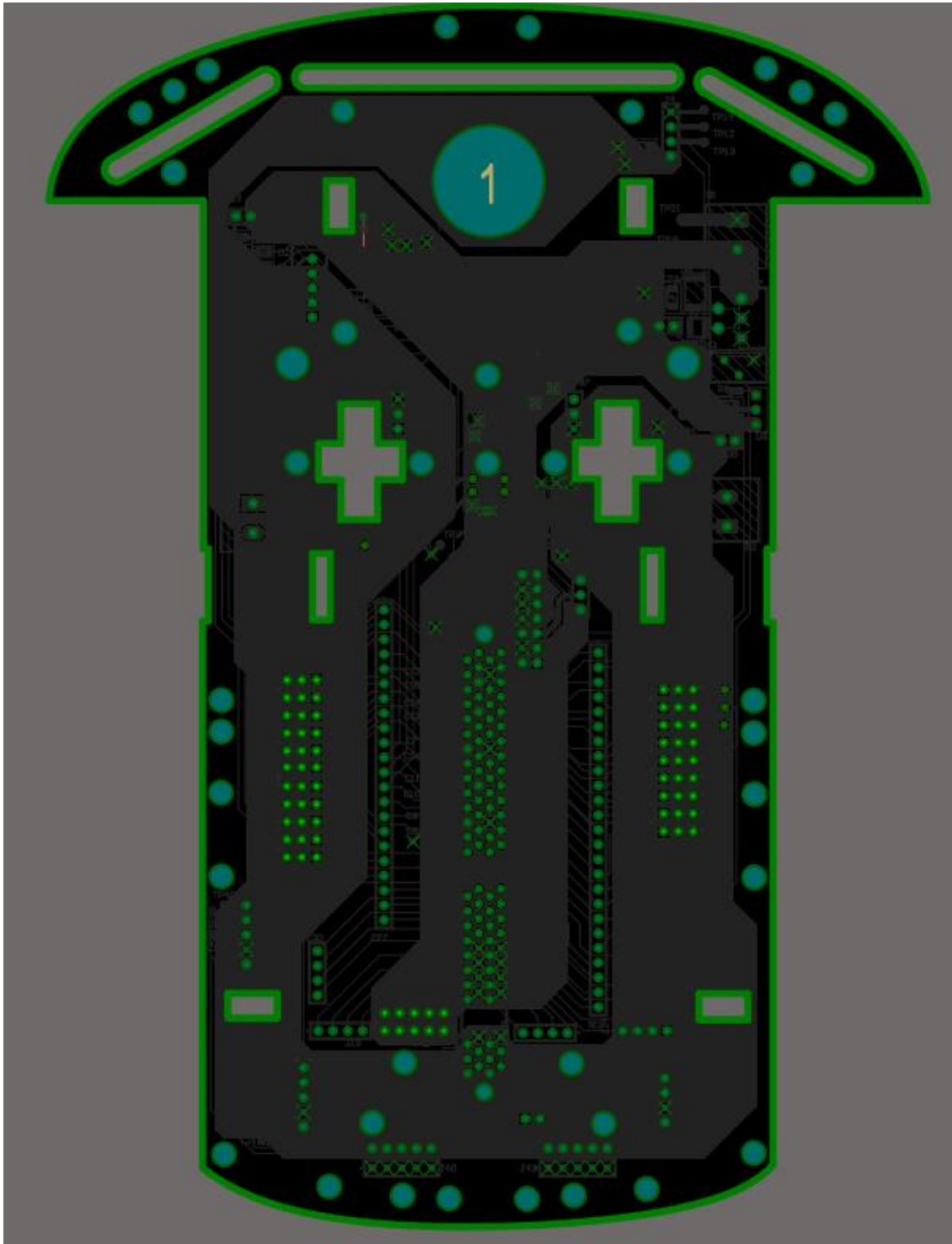


Figura 47. Rutado del plano de tierra

---

## 4 PROGRAMACIÓN

---

La programación del robot móvil se ha realizado a través del software CodeComposerStudio (CCS). Además, se ha usado la librería C2000 Ware que suministra funciones en un lenguaje de programación C y ha sido escogido ya que es el recomendado por el fabricante, además de aportar algunas librerías, para la programación de los DSP de la empresa Texas Instruments como es nuestro caso.

Además del uso de algunas de las librerías que ofrece el software, han sido necesaria la creación de hasta cuatro nuevas librerías para la correcta estructuración del código desarrollado, simplificando el mantenimiento y actualización del mismo.

La idea del código es crear un fichero principal, main.c, que sirva como una esqueleto central donde se programe las necesidades básicas del microprocesador y crear cuatro ficheros diferentes donde cada uno se centrará en la programación de los diferentes periféricos. De esta manera, se puede visualizar de manera cómoda y simple como se ha realizado la programación de cada subsistema sin la necesidad de conocer el resto de ellos.

Para la programación, como ya se ha mencionado, se ha utilizado el DSP de Texas Instruments TMS320F28335, y para agilizar la fase de desarrollo de la aplicación, se ha usado el kit de desarrollo Docking-Stn USB-EMU [R3] junto con la plataforma móvil desarrollada. Por tanto, en una primera fase del proyecto se ha empleado una docking base del fabricante que incorpora el programador y depurador. Una vez que el código ha sido probado en esta plataforma de Texas Instruments y el robot ha sido fabricado, se ha utilizado la propia base del robot para su programación. Cabe indicar que el robot ha sido diseñado para ser programado a través de la interfaz JTAG. Para ello, hemos incorporado un conector en la parte central del robot.

En cuanto a la creación del proyecto en el software CCS, este incluye todas las necesidades básicas para la programación de los DSP del fabricante, por lo tanto, se ha seleccionado un nuevo proyecto que tenga las características de nuestro robot.

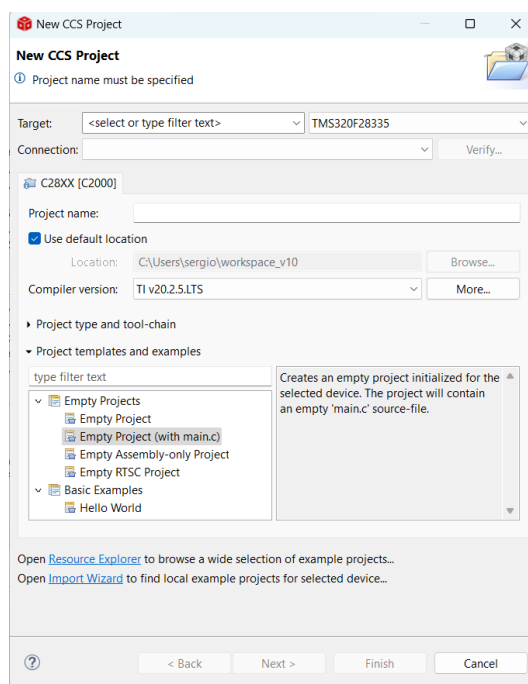


Figura 48. Creación proyecto CCS

En la Figura 48 se muestra como durante la creación del proyecto se selección del microprocesador objetivo para la programación además del tipo de conexión que se empleará. Por último, también se seleccionará la memoria que usará nuestro robot. De entre las opciones, las relevantes son una memoria tipo RAM o tipo Flash. Siendo la primera la que se ha empleado durante una fase previa de programación ya que nos permite entrar en un modo “Debug” del propio software y poder analizar paso por paso el código creado pudiendo depurar el código.

Una vez se ha realizado el código y podemos asegurar que funciona correctamente se ha empleado una memoria Flash como se observa en la parte de Validación.

La programación del fichero principal, main.c, ha sido lo más breve posible y consta de una inicialización de todos los periféricos que usara el DSP, además se incluye en el bucle principal las opciones de probar todos los periféricos disponibles.

#### 4.1 Descripción General de la Aplicación.

La aplicación consta de un esqueleto central como será en código main, este tendrá hasta cuatro procesos independientes creados en estos ficheros llamados “test”.

En el código main el encargado de mover al usuario por este código y comenzar los diferentes test será el módulo bluetooth. Comenzando previamente con una comprobación de estado del propio modulo, esto se realiza una única vez al principio y consiste en una mensaje con respuesta, donde el usuario envíe la palabra “test” y el módulo responda con un “OK”.

Durante todo el código el módulo bluetooth indicara con un “OK” si se ha recibido un tipo de información, de esta manera el usuario recibirá un continuo feedback del funcionamiento.

El main se separa en estos cuatro test, donde el usuario, puede enviar un número para comenzar así el test correspondiente.

El primero, moverá el motor paso a paso y para esto se enviará por el módulo bluetooth el sentido de giro y el ángulo de giro. Una vez recibido el test comienza y una vez finalizado se enviará "FINTEST", pudiendo comenzar otro test.

El segundo test hace uso del sensor de proximidad, durante este test se mostrará por pantalla la distancia en milímetros del objeto más próximo y al cabo de un tiempo se enviará "FINTEST".

El tercer test muestra el funcionamiento del motor del robot, al igual que en el primer test serán necesario más comandos para este test, siendo el primero el sentido, ya sea para girar o avanzar, y además enviar un número del 0-100 donde se indique la potencia de funcionamiento de los motores. Esta prueba durara un pequeño periodo de tiempo y finaliza enviando "FINTEST".

El ultimo test, pondrá en funcionamiento el módulo GPS, una vez comience se mostrará por pantalla las coordenadas, indicando además el hemisferio, por último, se enviará "FINTEST" indicando que se puede comenzar otro test.

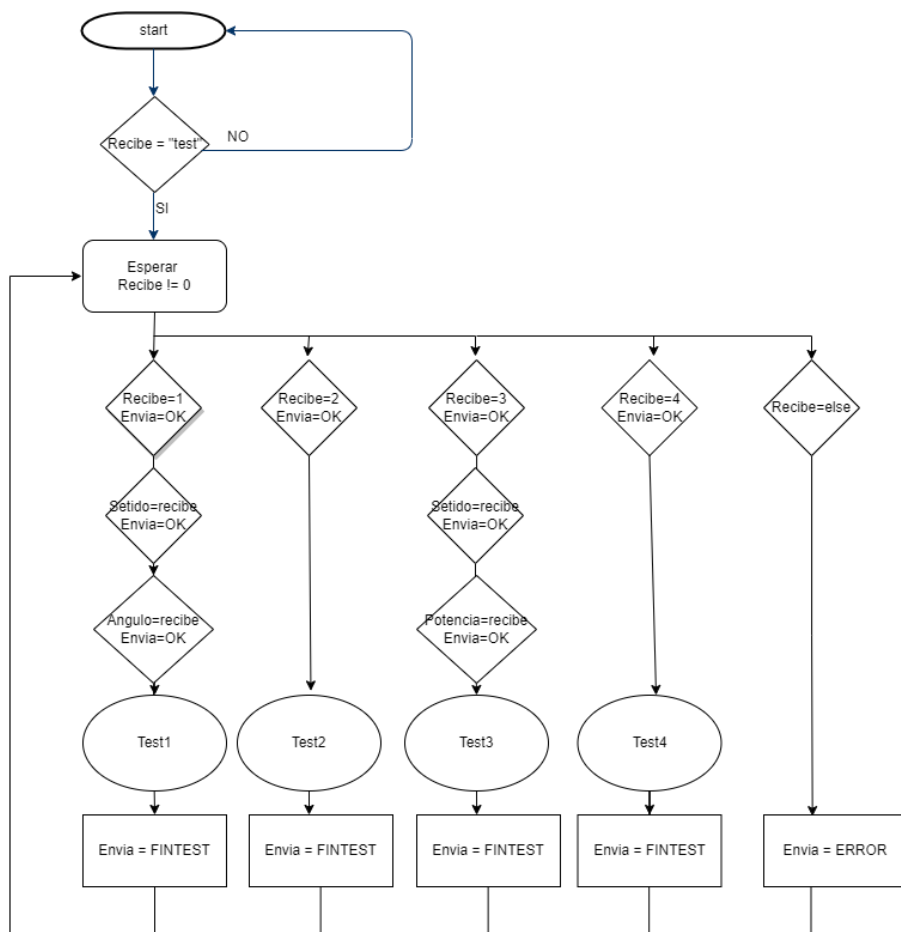


Figura 49. Diagrama de flujo MAIN

## 4.2 Bluetooth

El módulo bluetooth HC-06 usa un protocolo de comunicación basado en UART (Universal Asynchronous receiver / transmitter) cuyas siglas define como funciona este protocolo de una manera bastante sencilla.

Dentro del DSP este protocolo se le asocia a los periféricos como “Serial Communications Interface (SCI) Modules” donde se encuentran hasta 3 UART diferentes A,B,C.

Todos estos periféricos tienen asociados dos pines del microprocesador como SCI-TXD y SCI-RXD, reservando un pin para la transmisión de datos y otro para la recepción. Además, estos pines pueden asociarse a una interrupción de microprocesador. Esto es necesario ya que al ser un protocolo de comunicación asíncrono la interrupción viene determinando por la recepción de algún mensaje y no por el reloj asociado al microprocesador.

Para el módulo Bluetooth se ha elegido de entre las 3 UART disponible la A ya que esta UART además posee la característica de que sobre las placa de pruebas que ofrece Texas Instruments está asociada al puerto USB de la propia placa de pruebas, por lo tanto, podemos depurar el código a través de puerto USB. De esta manera podemos verificar que el código propuesto es correcto y si existiera algún tipo de fallo debe de ser debido por algún problema del módulo bluetooth.

Más concretamente se ha seleccionado los pines asociados a la GPIO 28 para SCIRXDA y la GPIO 29 para SCITXDA, dentro del código se ha tenido que seleccionar dentro una función que estas GPIOs se usen el periférico SCI, esto se realiza a través de la función dentro del MAIN.c llamada GPIO\_SELECT()

Además, se ha habilitado una rutina de interrupción asociada a la SCI\_A tanto RXD como TXD, de esta manera cada vez que llegue un mensaje a través del bluetooth entrada en la función llamada, “SCIA\_RX\_isr” y una función “SCIA\_TX\_isr” para la transmisión de mensajes.

Esta inicialización del periféricos SCI se realiza dentro del main en la función “SCIA\_init” y en la inicialización de funciones.

Una vez está todo inicializado, el funcionamiento de este periférico es simple. La información que se envía desde un dispositivo bluetooth y que es recibida por el microprocesador a través de este periférico se guarda en el registro dentro de SCIA, es decir, los mensajes enviados se pueden observar en SciaRegs.SCIRXBUF.bit.RXDT. Esta información se guarda en un buffer de memoria interna creada dentro de un variable Struct.

Y para los mensajes enviados SciaRegs.SCITXBUF.

De esta manera en el código solo debemos observar la variable struct llamada “data\_com” dentro de esta variable struct esta tanto el buffer de transmisión como de recepción, así como un contador para ambos buffers. Para esto se usa una función llamada “strncmp” que pertenece a la librería <string.h>. Esta función compara dos cadenas de caracteres y de esta manera podemos saber si el mensaje que se encuentra en el buffer es el esperado. Avanzando poco a poco por el código y pudiendo responder cada vez que se recibe un mensaje que sea el esperado.

### 4.3 Test1: Servomotor

La programación del motor paso a paso se realiza a través del driver de direccionamiento ULN2003. Para la programación de este driver se ha estudiado el datasheet y sus necesidades. Para este caso la programación es simple, ya que constara de cuatro pines de entrada que actuaran como señal lógica.

La inicialización de los pines necesarios para la programación de este test1 se ha realizado en la función Gpio\_select().

En las siguientes líneas de programación se ha diseñado el uso de las GPIOs 01, 03, 05, 07 como puerto de entrada/salida usando el comando GpioCtrlRegs.GPAMUX1, como se indica en el datasheet y a continuación se ha seleccionado a través de GpioCtrlRegs.GPADIR. que se usaran como puerto de salida.

```
//Gpios para servo motor
GpioCtrlRegs.GPAMUX1.bit.GPIO1 = 0; // GPIO17
GpioCtrlRegs.GPAMUX1.bit.GPIO3 = 0; // GPIO21
GpioCtrlRegs.GPAMUX1.bit.GPIO5 = 0; // GPIO29
GpioCtrlRegs.GPAMUX1.bit.GPIO7 = 0; // GPIO33

GpioCtrlRegs.GPADIR.bit.GPIO1 = 1; // GPIO17
GpioCtrlRegs.GPADIR.bit.GPIO3 = 1; // GPIO21
GpioCtrlRegs.GPADIR.bit.GPIO5 = 1; // GPIO29
GpioCtrlRegs.GPADIR.bit.GPIO7 = 1; // GPIO33
```

Figura 50. Código, inicialización Servo motor

Para la programación de ha creado un fichero que estará incluido en la librería test1.h, el cual incluye una función que servirá para la programación del driver.

Esta función tendrá 4 argumentos de entrada, como son el sentido de giro, el ángulo que se desea girar, un contador y el ángulo actual de giro.

Tanto el sentido de giro como el ángulo objetivo a girar son indicados por el usuario a través del módulo bluetooth, esto se realiza antes de comenzar el test. Las otras variables se modifican en la interrupción del reloj. El contador se modifica con cada interrupción y el ángulo actual se modifica cada vez que este contador llegue a 2880 que será cuando el motor paso a paso hay realizado 8 pasos, cada uno de ellos cada 360us.

La función del test en un bucle swithh cuyo argumento de entrada es el contador y cambia de estado en estado cada vez que el contador incremente 360, es decir, cada 3600us el servo motor da un paso ya que en estos casos se modifican las señales de salida de las GPIOs creando así un bucle.

Step	Command	IN4	IN3	IN2	IN1
A	01H	0	0	0	1
AB	03H	0	0	1	1
B	02H	0	0	1	0
BC	06H	0	1	1	0
C	04H	0	1	0	0
CD	0CH	1	1	0	0
D	08H	1	0	0	0
DA	09H	1	0	0	1

Figura 51. Funcionamiento Servo motor

Dentro de esta función se encuentran dos bucles switch cuyos estados son inversos, esto servirá para indicar el sentido de giro del servo motor.

Para que el test finalice el ángulo de giro actual debe superar al ángulo objetivo, devolviendo la función un valor "1" indicando que se ha finalizado el test, por último, el módulo bluetooth indica al usuario que el test ha finalizado y que puede elegir otro test diferente.

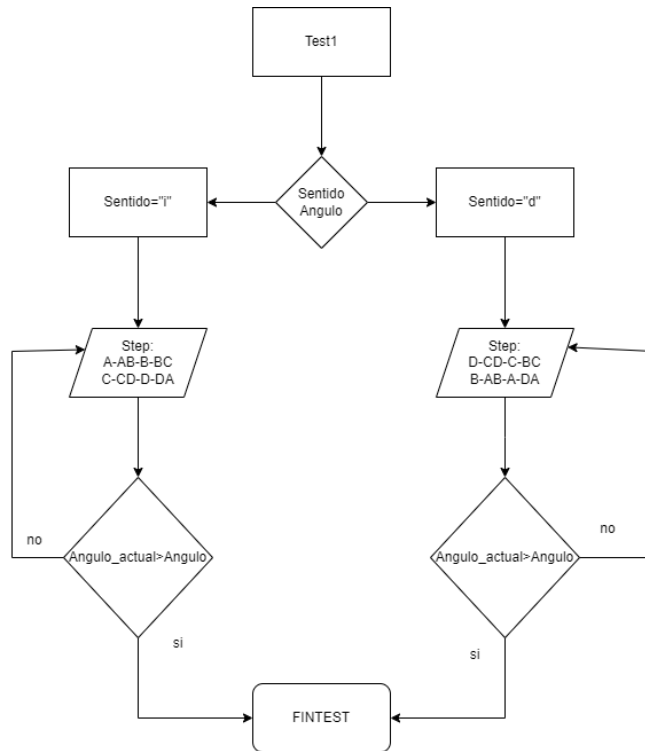


Figura 52. Diagrama de Flujo, test1

#### 4.4 Test2: Sensor de proximidad

Para la programación será necesario inicializar las GPIOs necesarias dentro de la función Gpio\_Select(). Sabiendo que uno de los pines actuará como salida y el otro pin, conectado al puerto Echo, deberá usar uno de los puertos compatibles con el periférico ECAP que ofrece el DSP.

El periférico ECAP se utiliza para el cálculo de tiempo a nivel alto de la señal Echo. Se ha inicializado en una nueva función llamada my\_InitEcap(). Para ello, es necesario modificar los registros del DSP ECCTL1 y ECCTL2

Figure 5-24. ECCTL1 Register

15	14	13	12	11	10	9	8
FREE_SOFT				PRESCALE			CAPLDEN
R/W-0h				R/W-0h			R/W-0h
7	6	5	4	3	2	1	0
CTRRST4	CAP4POL	CTRRST3	CAP3POL	CTRRST2	CAP2POL	CTRRST1	CAP1POL
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

Figure 5-25. ECCTL2 Register

15	14	13	12	11	10	9	8	
RESERVED						APWMPOL	CAP_APWM	SWSYNC
R-0h						R/W-0h	R/W-0h	R-0/W1S-0h
7	6	5	4	3	2	1	0	
SYNCO_SEL	SYNCI_EN	TSCTRSTOP	REARM	STOP_WRAP	CONT_ONESHOT			
R/W-0h	R/W-0h	R/W-0h	R-0/W1S-0h	R/W-3h			R/W-0h	

Figura 53. Registros empleados para la Ecap[14]

Para el correcto funcionamiento del módulo ECAP, se configurado para que actúe en 2 eventos: el primero corresponde con el flanco de subida de la señal Echo y el segundo, con el flanco de bajada. De esta forma, al detectar el primer evento, el módulo inicia un contador que estima el número de ciclos de reloj transcurridos hasta que se produce el segundo evento. La detección del flanco de bajada provoca la parada de dicho contador. El valor devuelto por el módulo ECAP corresponde con el número de ciclos de reloj transcurridos en los que la señal Echo ha estado a nivel alto. Este dato, convertido a tiempo, será usado para determinar la distancia.

```

ECap1Regs.ECEINT.all = 0x0000;           // Disable all capture interrupts
ECap1Regs.ECCLR.all = 0xFFFF;           // Clear all CAP interrupt flags
ECap1Regs.ECCTL1.bit.CAPLDEN = 0;       // Disable CAP1-CAP4 register loads
ECap1Regs.ECCTL2.bit.TSCTRSTOP = 0;     // Make sure the counter is stopped
//
// Configure peripheral registers
ECap1Regs.ECCTL1.bit.PRESCALE = 0;
ECap1Regs.ECCTL1.bit.CAP1POL = 0;       // rising edge
ECap1Regs.ECCTL1.bit.CAP2POL = 1;       // falling edge
ECap1Regs.ECCTL1.bit.CTRRST1 = 0;       // Difference operation
ECap1Regs.ECCTL1.bit.CTRRST2 = 1;       // Difference operation
ECap1Regs.ECCTL1.bit.PRESCALE = 0;

//
ECap1Regs.ECCTL2.bit.CAP_APWM = 0;
ECap1Regs.ECCTL2.bit.REARM = 0;          // arm one-shot
ECap1Regs.ECCTL2.bit.CONT_ONESHOT = 0;  // One-shot
ECap1Regs.ECCTL2.bit.STOP_WRAP = 1;     // Stop at 2 events
ECap1Regs.ECCTL2.bit.SYNCI_EN = 0;      // Enable sync in
ECap1Regs.ECCTL2.bit.SYNCO_SEL = 2;     // Pass through

ECap1Regs.ECCTL2.bit.TSCTRSTOP = 1;     // Start Counter
ECap1Regs.ECCTL1.bit.CAPLDEN = 1;       // Enable CAP1-CAP4 register loads
ECap1Regs.ECEINT.bit.CEVT1 = 0;         // 2 events = interrupt
ECap1Regs.ECEINT.bit.CEVT2 = 1;         // 2 events = interrupt

```

Figura 54. Programación del módulo Ecap

En la Figura 54 se muestra cómo se han inicializado los registros correspondientes a la Figura 53. Los primeros 16 bits correspondientes al registro ECCTL1 nos han servido para indicar cuántos eventos vamos a analizar y si queremos asociar este evento a uno flanco de subida o a un flanco de bajada. Estos son los bits 0,2,4,6. Inicializando a 0 si queremos que actúen como flanco de subida y a 1 como flanco de bajada



Los bits 1.3.5.7 están reservados para indicar con que evento se reinicia el contador. En nuestro caso como solo queremos calcular el tiempo a nivel alto podemos reiniciar el contador con el evento de bajada.

Analizando los 16 bits reservados para el registro ECCTL2, siendo 1 bit 0 para indicar que trabajamos en un modo continuo. Los bits 1 y 2 están reservados para indicar cuando se para el contado, que se ha seleccionado con el segundo evento. Los bits 6 y 7 se han seleccionado ambos como 1 para que el contador no sincronice con una señal externa.

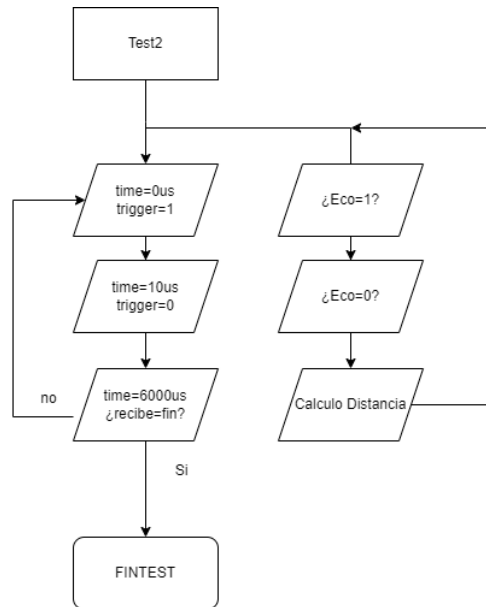


Figura 55. Diagrama de Flujo, test2

#### 4.5 Test3: Motor

El tercer test se encarga de la programación de los motores de la plataforma móvil, así como los encoders asociados. Para esto, se hará uso de 4 GPIOs de salida para el control de los motores del robot y 2 GPIOs adicionales de entrada para la información recibido por los dos encoders asociados a cada rueda.

Comenzando por el control de los motores, este se realiza a través del DRV8835, el cual como se ha presentado en la sección de periféricos, tiene dos modos de funcionamiento. Durante el trabajo se empleará un modo de uso Phase/Enable. Bajo este modo de operación, los pines de Phase indicaran la dirección de vehículo en función de tensión a nivel alto o nivel bajo, y el pin Enable el cual se inicializará como un pin PWM ajustará la velocidad de cada rueda individualmente.

La GPIOs seleccionadas son “GPIO02” “GPIO10” para los pines Enables del drv8835 que hay que inicializar como PWM y “GPIO06” “GPIO15” como pines de salida que servirán para los pines de Phase del driver.

```

380 // para el motor
381 GpioCtrlRegs.GPAMUX1.bit.GPIO2 = 1; // GPIO2 como EPWM2A
382 GpioCtrlRegs.GPAMUX1.bit.GPIO10 = 1; // GPIO10 como EPWM6A
383 GpioCtrlRegs.GPAPUD.bit.GPIO6 = 0;
384 GpioCtrlRegs.GPAMUX1.bit.GPIO6 = 0; // GPIO6 como E/S
385 GpioCtrlRegs.GPADIR.bit.GPIO6 = 1; // GPIO6 como salida
386 GpioCtrlRegs.GPAPUD.bit.GPIO15 = 0;
387 GpioCtrlRegs.GPAMUX1.bit.GPIO15 = 0; // GPIO15 como E/S
388 GpioCtrlRegs.GPADIR.bit.GPIO15 = 1; // GPIO15 como salida

```

Figura 56. Código para la inicialización del motor.

Además de esta inicialización de los pines, se ha creado una función para inicializar los PWMs seleccionados, “Setup\_PWM2”. Los PWM elegidos se corresponden con PWM2 y PWM6. La Figura 57 muestra la configuración del PWM2. Cabe indicar que la configuración del PWM6 es análoga.

```

541 // Configuración del módulo Time-Base
542 EPwm2Regs.TBCTL.bit.CLKDIV = 0; // Pre-escalador CLKDIV = 1
543 EPwm2Regs.TBCTL.bit.HSPCLKDIV = 1; // Pre-escalador HSPCLKDIV = 2
544 EPwm2Regs.TBCTL.bit.CTRMODE = 2; // Modo up - down
545 EPwm2Regs.TBCTL.bit.PRDL = 0; // Shadow activado
546 EPwm2Regs.TBCTL.bit.SYNCOSEL = 0; // Sincronización deshabilitada
547 EPwm2Regs.TBPRD = 37500; // Periodo para 1KHz de fPWM

556 // Configuración del módulo Compare-Counter
557 EPwm2Regs.CMPCTL.bit.SHDWAMODE = 0; // shadow activado para CMPA
558 EPwm2Regs.CMPCTL.bit.SHDWBMODE = 0; // shadow activado para CMPB
559 EPwm2Regs.CMPCTL.bit.LOADAMODE = 0; // shadow de CMPA cargado en TBCTR = 0
560 EPwm2Regs.CMPCTL.bit.LOADBMODE = 0; // shadow de CMPB cargado en TBCTR = 0

567 // Configuración del módulo Action Qualifier
568 EPwm2Regs.AQCTLA.all = 0;
569 EPwm2Regs.AQCTLA.bit.CAD = 2; // EPWM1A a low en CMPA down
570 EPwm2Regs.AQCTLA.bit.CAU = 1; // EPWM1A a high en CMPA up
571 EPwm2Regs.AQCTLB.all = 0;
572 EPwm2Regs.AQCTLB.bit.CBD = 2; // EPWM1B a low en CMPB down
573 EPwm2Regs.AQCTLB.bit.CBU = 1; // EPWM1B a high CMPB up

582 // Configuración del módulo Dead Band
583 EPwm2Regs.DBCTL.bit.OUT_MODE = 0; // Módulo descativado
584 // Configuración del módulo PWM-Chopper
585 EPwm2Regs.PCCTL.bit.CHPEN = 0; // Módulo descativado
586 // Configuración del módulo Trip Zone
587 EPwm2Regs.TZCTL.bit.TZA = 3; // No hacer nada
588 EPwm2Regs.TZCTL.bit.TZB = 3; // No hacer nada
589 EPwm2Regs.TZEINT.all = 0; // interrupción deshabilitada

597 // Configuración del módulo Event Trigger
598 EPwm2Regs.ETSEL.bit.SOCAEN = 0; // Generación de SOCA deshabilitado
599 EPwm2Regs.ETSEL.bit.SOCBEN = 0; // Generación de SOCB deshabilitado
600 EPwm2Regs.ETSEL.bit.INTEN = 0; // Generación de señal de interrupción deshabilitado

```

Figura 57. Código, inicialización PWM

Para inicializar el periférico ePWM se ha diseñado como se muestra en la Figura 57, donde cabe destacar algunos de los comandos que se han empleado para inicializar. Como periodo para la función PWM se ha escogido 1khz. Se inicializan dos shadow que actuaran como un doble buffer y su base de tiempo se inicializa a cero.

Para el desarrollo de la prueba se ha creado un nuevo fichero “.c” donde se han definido una serie de funciones para el correcto funcionamiento del robot. La función principal de este fichero recibe dos variables una para la potencia de los motores que vamos a suponer que siempre será la misma, tanto para la rueda izquierda como para la rueda derecha, así como el funcionamiento de las ruedas. Estos modos de funcionamiento pueden ser girar en cualquier sentido, avanzar o retroceder. Todos estos datos son obtenidos gracias a modulo bluetooth que pide antes de comenzar la prueba que se envíen los comandos necesarios.

Dentro de la función de la prueba, identificamos el modo de funcionamiento y se han creado una serie de funciones para controlar las ruedas. Esto se realiza durante un tiempo que se ha programado dentro de la función. Una vez que este tiempo llegue la prueba ejecuta una última función de frenar y devuelve un “1”.

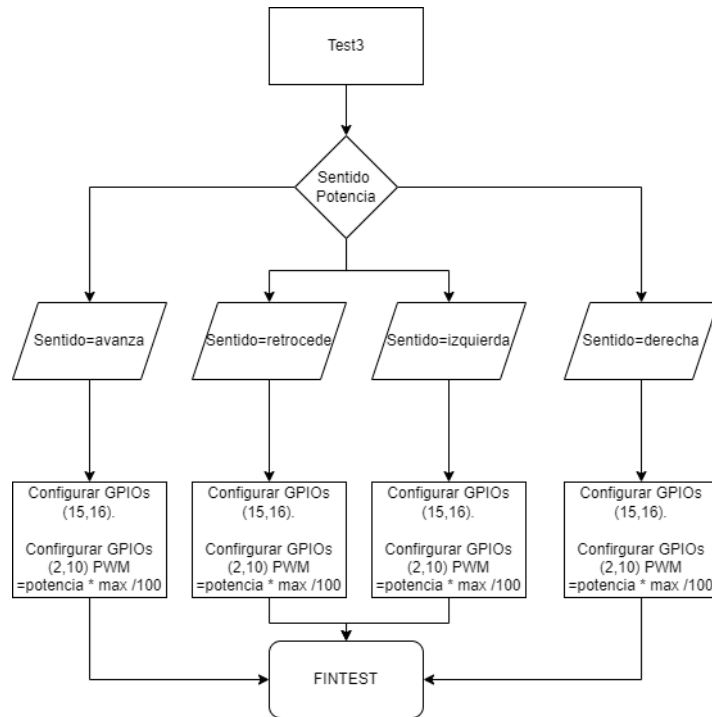


Figura 58. Diagrama de flujo, test3

#### 4.6 Test4: GPS

El ultimo test servirá para comprobar el funcionamiento del módulo GPS. Este módulo usa un modo de comunicación asíncrona, es decir, se comunicará con el robot a través de una comunicación UARTs. En este caso se ha elegido la UARTB, que es la única disponible.

Al igual que con el resto de periféricos, se han seleccionado una serie de pines del microprocesador para la correcta programación del robot. En este caso como se ha indicado será necesario el uso del periférico SCI del DSP.

La inicialización será semejante a la empleada con el Bluetooth ya que usa el mismo periférico. En este caso, se han elegido los pines “GPIO18” para el puerto SCIBTXD y el pin “GPIO19” para la señal SCIRXD. La inicialización de ambos se realiza dentro de la función Gpio\_select.

La inicialización del periférico SCI se realizará de la misma manera que la del módulo bluetooth en una función nueva llamada SCIB\_init.

El funcionamiento del dispositivo se ha estudiado según su datasheet, donde indica que este enviara a través de los pines TXD periférico una serie de caracteres como se muestra en la siguiente Figura 59

```

$GNGGA,084852.000,2236.9453,N,11408.4790,E,1,05,3.1,89.7,M,0.0,M,,*48
$GNGLL,2236.9453,N,11408.4790,E,084852.000,A,A*4C
$GPGSA,A,3,10,18,31,,,,,,,,,6.3,3.1,5.4*3E
$BDGSA,A,3,06,07,,,,,,,,,6.3,3.1,5.4*24
$GPGSV,3,1,09,10,78,325,24,12,36,064,,14,26,307,,18,67,146,27*71
$GPGSV,3,2,09,21,15,188,,24,13,043,,25,55,119,,31,36,247,30*7F
$GPGSV,3,3,09,32,42,334,*43
$BDGSV,1,1,02,06,68,055,27,07,82,211,31*6A
$GNRMC,084852.000,A,2236.9453,N,11408.4790,E,0.53,292.44,141216,,,A7.5 $GNVTG,292.44,T,M,0.53,N,0.98,K,A2D
$GNZDA,084852.000,14,12,2016,00,00*48
$GPTXT,01,01,01,ANTENNA OK*35

```

Figura 59. Información posible recibida por el GPS

De toda la información enviada la necesaria para obtener la latitud y longitud de la posición actual será las que estén seguidas del código \$GNRMC

(9) Format of latitudes and longitudes

```
$GNRMC,084852.000,A,2236.9453,N,11408.4790,E,0.53,292.44,141216,,,A*7.5
```

Data format: ddd°mm.mmm' Convert to the format of Google or Baidu Map. Latitude: ddm.mmmmm, Northern Latitude 2236.9453,  $22+(36.9453/60)= 22.615755$  Longitude: dddmm.mmmmm, East Longitude 11408.4790,  $114+(08.4790/60)=114.141317$

Figura 60. Información útil recibida por GPS

La idea de la programación para este último test es la creación de una función de interrupción para la comunicación con el GPS, y dentro de esta función almacenar los datos enviados en un buffer de información. Además, como se ha indicado de toda esta información enviada solo será de utilidad las que este continuada de los caracteres \$GNRMC, es por eso por lo que se comparara, y si la información es útil, es decir, se ha detectado que ha una coincidencia en el buffer con los caracteres se almacenara en un buffer de DATA.

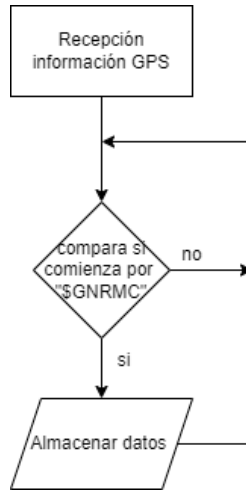


Figura 61. Diagrama de flujo, GPS

Para continuar con el test4 se ha creado un nuevo archivo “.c” donde se han creado hasta 4 funciones, estas reciben como entrada el buffer de información útil y devuelven exclusivamente la cabeza del puntero de la latitud o de la longitud.

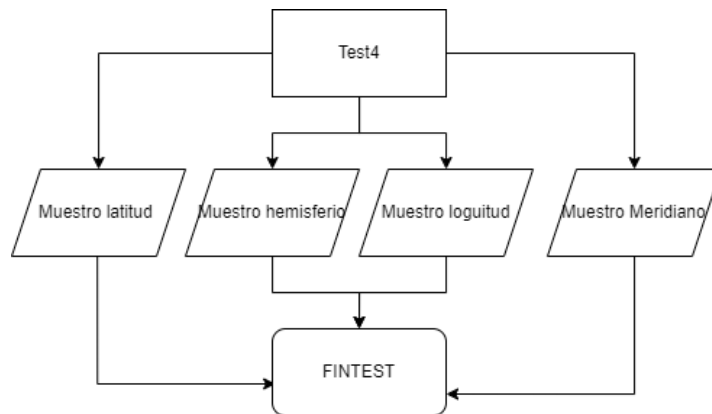


Figura 62. Diagrama de Flujo, test4

## 5 VALIDACIÓN

Una vez el robot ya ha sido diseñado y fabricado, se ha realizado una validación posterior para asegurar que la plataforma desarrollada funciona como se espera. Esta validación paso por paso será detallada a continuación. Cabe destacar que si se detecta algún error durante las pruebas, este se registrará y, posteriormente, se resolverá en el proyecto de la tarjeta impresa y/o software, generando una nueva versión.

Para ello, será necesario el uso de los laboratorios, así como de su instrumentación electrónica para obtener de manera experimental datos que prueben el correcto funcionamiento del robot y que los diferentes módulos integrados operan dentro de las especificaciones esperadas.

### 5.1 Hardware

Dentro del laboratorio de electrónica se realizarán una serie de pruebas para comprobar que la placa diseñada no presenta errores de esquemático. Para agilizar el proceso durante la fase de diseño se utilizaron una serie de TestPoint sobre la PCB para poder obtener información electrónica de cada punto del diseño.

A continuación, se muestra en la siguiente Tabla 16 el plan de pruebas físicas realizado sobre la tarjeta impresa, estas pruebas han seguido el mismo orden que el que es mostrado en la tralal de manera que no se pueden realizar algunas de estas pruebas sin haber validado la anterior.

Tabla 16. Plan de pruebas físicas

Prueba	Descripción
1.	Validar la correcta fabricación de la tarjeta impresa. Revisando que no existan puntos de ruptura o zonas indeseadas
2.	Comprobación de continuidad eléctrica del PCB recibido.
3.	Revisar el orden de las diferentes capas que componen la tarjeta de circuito impreso.
4.	Validar el correcto funcionamiento del DC/DC utilizado.
5.	Validar el correcto funcionamiento del regulador lineal empleado.
6.	Comprobar que la tensión medida en los pines de alimentación de los diferentes componentes es la correcta

Durante la ejecución de este plan de pruebas, se han encontrado una serie de problemas asociados con la fabricación de la tarjeta impresa. A continuación, se explican cuáles han sido estos problemas y la solución aportada para su resolución.

Durante el desarrollo de la prueba 2, se ha detectado un fallo en la continuidad de la tensión correspondiente a 3.3V, la cual tiene un cortocircuito a la tierra. Se ha revisado y esto es debido a una de las vías situadas próximas del DRV8835. Como medida resolutive se ha eliminado esta vía para poder continuar con el proyecto y se ha sustituido por unos cables que unen uno de los condensadores con la tierra del sistema.

Durante la prueba 3, se ha detectado un problema en cuanto a la huella de ambos DC/DC. El problema en cuanto al LM2596 ha sido que la huella esta invertida. Para solucionar este inconveniente durante el proyecto se ha situado en la parte inferior de la placa. El problema referido a la LM317 es que el taladro empleado en su huella no corresponde con su tamaño real. De esta manera el regulador lineal no entra totalmente en sus pines, aunque no haya supuesto un problema para continuar con el trabajo.

El resto de las pruebas han sido evaluadas exitosamente, pudiendo continuar con un siguiente plan de estudios para el sistema software.

## 5.2 SOFTWARE

De la misma manera que se ha planteado un plan de pruebas físicas para validar el hardware del sistema, se ha creado un plan de pruebas del software. Estas han sido evaluadas de manera experimental con la plataforma montada. Este plan de pruebas se recoge en la Tabla 17. Plan de pruebas Software:

Tabla 17. Plan de pruebas Software

Prueba	Descripción
1.	Comprobación de JTAG para la programación
2.	Comprobación del módulo bluetooth (tanto envío como respuesta de mensajes).
3.	Comprobación test1. Estos incluyen tanto el código descrito como el servomotor
4.	Comprobación test2. Estos incluyen tanto el código descrito como el sensor sonar
5.	Comprobación test3. Estos incluyen tanto el código descrito como los encoder y los motores del robot
6.	Comprobación test4. Estos incluyen tanto el código descrito como el módulo GPS

Todas las pruebas han sido realizadas con éxito finalizando la validación de todos los sistemas que incluye el proyecto.

## 5.3 Correcciones

Durante esta validación se ha comprobado que existen algunos componentes donde han ocurrido alguna serie de inconvenientes. Aunque la solución aportada para la continuidad del proyecto ha funcionado correctamente se han realizado una serie de cambios en los archivos correspondientes al trabajo.

Por una parte, las huellas existentes en el proyecto de ALTIUM, para el diseño de la tarjeta impresa que son erróneas, como han sido las huellas la de ambos DC/DC empleados han sido sustituidas de manera correcta, por si en un futuro hiciese falta avanzar en el diseño de este trabajo o por si fuera necesario la fabricación de más modelos.

Además, las vías que han provocado un fallo en la continuidad eléctrica del sistema han sido ajustada de manera que no deba existir este problema en la próxima fabricación del robot.



## 6 CONCLUSIONES Y LÍNEAS FUTURAS

---

Con el Robot diseñado, fabricado y evaluado correctamente se puede afirmar que los objetivos de este trabajo como eran la simplificación del kit de desarrollo para vehículos; así como la unión entre este kit de desarrollo junto a uno de los DSP como es el TMS320F28335 han sido un éxito.

Además, después de la validación del sistema, se ha comprobado que su funcionalidad es correcta. Asimismo, se ha comprobado que su utilización y programación resulta intuitiva, aspecto clave para los futuros usuarios de este robot.

Por tanto, se puede afirmar que todos los objetivos marcados para el proyecto se han cumplido dejando la plataforma autónoma móvil para su uso como material educativo y para futuras ampliaciones.

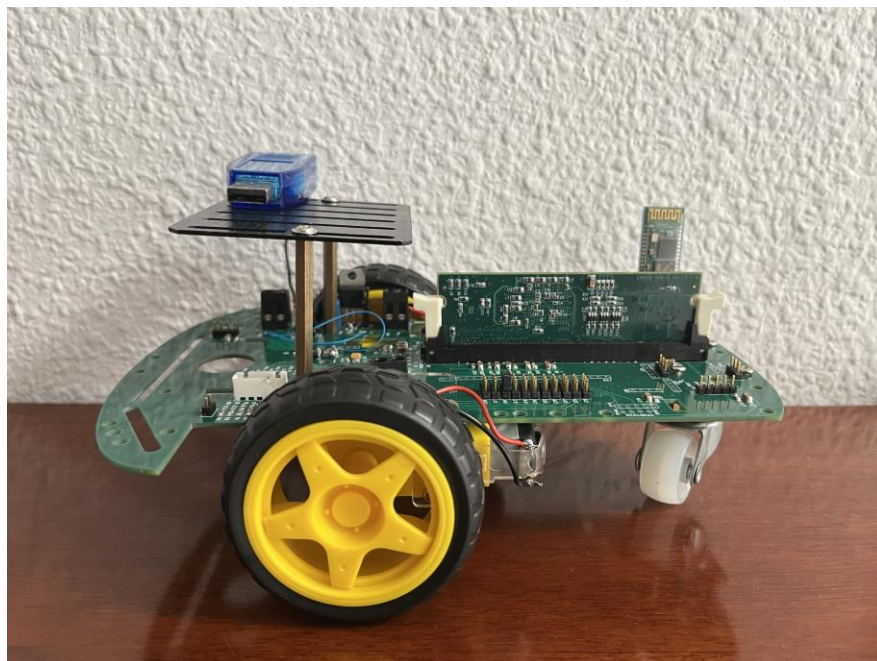


Figura 63. Robot Final

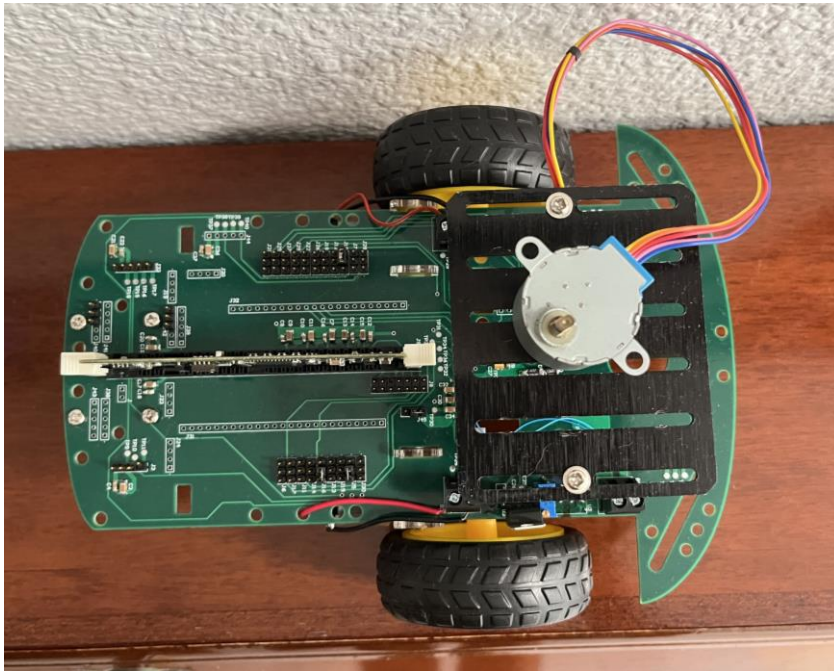


Figura 64. Robot final, superior

## 6.1 Líneas Futuras

Uno de los objetivos del proyecto era dejar vía libre a futuros usuarios para la modificación del robot como se ha diseñado. Es por eso por lo que se indican una serie de recomendaciones por donde se pueden continuar con el desarrollo del este robot.

- Durante la fase de diseño, se implementó conector para un periférico no empleado que usa un protocolo de comunicación I2C. Este periférico, podría ser un acelerómetro/giroscopio, además se ha dejado la sección de descripción del sistema tanto su esquemático, para saber que GPIOs usaran este nuevo periférico, como un conector sobre la propia placa impresa con sus correspondientes puntos de prueba para evaluar el correcto funcionamiento.
- Durante este trabajo se ha indicado que existe un dispositivo de localización para interiores, aunque durante la fase de validación se han utilizado ambos periféricos, como son el sensor sonar y el servomotor, no se ha descrito en código de programación para esto. Se puede ajustar el sensor de ultrasonidos para que gire de manera solidaria con el motor paso a paso, pudiendo generar un mapa de obstáculos en el interior de cualquier edificio.

## 7 REFERENCIAS

---

- [1] Manuel Serrano Rodríguez, “Robot Auxiliar de Guiado basado en TMS320F28335 de Texas Instruments.”.
- [2] “TMS320F2833x, TMS320F2823x Real-Time Microcontrollers datasheet .” [https://www.ti.com/lit/ds/symlink/tms320f28234.pdf?ts=1688121123337&ref\\_url=https%253A%252F%252Fwww.google.com%252F](https://www.ti.com/lit/ds/symlink/tms320f28234.pdf?ts=1688121123337&ref_url=https%253A%252F%252Fwww.google.com%252F) (accessed Jun. 30, 2023).
- [3] “Pololu DRV8835 Dual Motor Driver Kit for Raspberry Pi,” <https://www.pololu.com/product/2753>. <https://www.pololu.com/product/2753> (accessed Jun. 30, 2023).
- [4] “DRV8835 Dual Low-Voltage H-Bridge IC.” DRV8835 Dual Low-Voltage H-Bridge IC (accessed Jun. 30, 2023).
- [5] “Double Speed Measuring Module with Photoelectric Encoders Model: HC-020K.” <https://mrelectrobot.com/wp-content/uploads/2021/09/HC-020K.pdf> (accessed Jun. 30, 2023).
- [6] “Gunagzhou HC Information Technology CO., Ltd.” <https://www.mouser.com/datasheet/2/758/hc06-1142930.pdf> (accessed Jun. 30, 2023).
- [7] “28BYJ48 stepping motor.” <https://datasheetspdf.com/datasheet/28BYJ48.html> (accessed Jun. 30, 2023).
- [8] “ULN200x, ULQ200x High-Voltage, High-Current Darlington Transistor Arrays datasheet (Rev. Q).” <https://www.ti.com/lit/ds/symlink/ulq2003a.pdf?ts=1688082318630> (accessed Jun. 30, 2023).
- [9] “Ultrasonic Ranging Module HC - SR04.” <https://www.farnell.com/datasheets/3422740.pdf> (accessed Jun. 30, 2023).
- [10] “SnapEDA.” <https://www.snapeda.com/> (accessed Jul. 04, 2023).
- [11] “JTAG/MSPD Emulation”, Accessed: Jun. 30, 2023. [Online]. Available: [https://www.ti.com/lit/ug/spdu079a/spdu079a.pdf?ts=1688037004083&ref\\_url=https%253A%252F%252Fwww.ti.com%252Fsitesearch%252Fen-us%252Fdocs%252Funiversalsearch.tsp%253FflangPref%253Den-US%2526searchTerm%253DMSP430%2526bjtag%2526nr%253D31300](https://www.ti.com/lit/ug/spdu079a/spdu079a.pdf?ts=1688037004083&ref_url=https%253A%252F%252Fwww.ti.com%252Fsitesearch%252Fen-us%252Fdocs%252Funiversalsearch.tsp%253FflangPref%253Den-US%2526searchTerm%253DMSP430%2526bjtag%2526nr%253D31300), “JTAG/MPSD Emulation.” [https://www.ti.com/lit/ug/spdu079a/spdu079a.pdf?ts=1688037004083&ref\\_url=https%253A%252F%252Fwww.ti.com%252Fsitesearch%252Fen-](https://www.ti.com/lit/ug/spdu079a/spdu079a.pdf?ts=1688037004083&ref_url=https%253A%252F%252Fwww.ti.com%252Fsitesearch%252Fen-)

- us%252Fdocs%252Funiversalsearch.tsp%253FlangPref%253Den-US%2526searchTerm%253DMSP430%2Bjtag%2526nr%253D31300 (accessed Jun. 30, 2023).
- [12] “LM2596 SIMPLE SWITCHER® Power Converter 150-kHz 3-A Step-Down Voltage Regulator datasheet”, Accessed: Jun. 30, 2023. [Online]. Available: <https://www.ti.com/lit/ds/symlink/lm2596.pdf>
- [13] “LM317 3-Terminal Adjustable Regulator.” [https://www.ti.com/lit/ds/symlink/lm317.pdf?ts=1688107261839&ref\\_url=https%253A%252F%252Fwww.ti.com%252Fproduct%252Fde-de%252FLM317](https://www.ti.com/lit/ds/symlink/lm317.pdf?ts=1688107261839&ref_url=https%253A%252F%252Fwww.ti.com%252Fproduct%252Fde-de%252FLM317) (accessed Jun. 30, 2023).
- [14] “TMS320x2833x, TMS320x2823x Technical Reference Manual.” [https://www.ti.com/lit/ug/sprui07/sprui07.pdf?ts=1688447817958&ref\\_url=https%253A%252F%252Fwww.ti.com%252Fsitesearch%252Fen-us%252Fdocs%252Funiversalsearch.tsp%253FlangPref%253Den-US%2526searchTerm%253DTMS320x2833x%2BTMS320x2823x%2BTechnical%2BReference%2BManual%2526nr%253D50](https://www.ti.com/lit/ug/sprui07/sprui07.pdf?ts=1688447817958&ref_url=https%253A%252F%252Fwww.ti.com%252Fsitesearch%252Fen-us%252Fdocs%252Funiversalsearch.tsp%253FlangPref%253Den-US%2526searchTerm%253DTMS320x2833x%2BTMS320x2823x%2BTechnical%2BReference%2BManual%2526nr%253D50) (accessed Jul. 04, 2023).