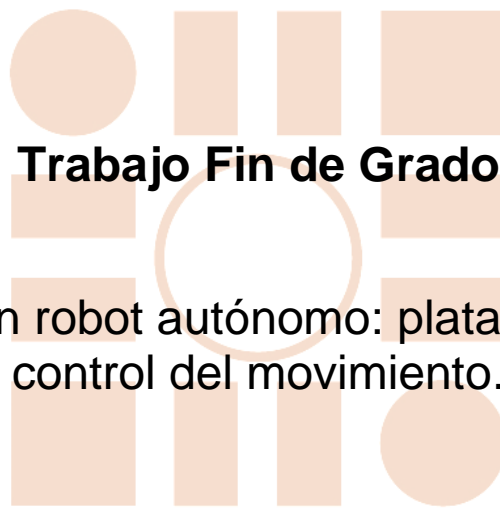


**Grado en Ingeniería en Electrónica y Automática
Industrial**



Trabajo Fin de Grado

Diseño de un robot autónomo: plataforma base y control del movimiento.

ESCUELA POLITECNICA
SUPERIOR

Autor: Miguel Tradacete Ágreda

Tutor: Julio Pastor Mendoza

2018

UNIVERSIDAD DE ALCALÁ
Escuela Politécnica Superior

**Grado en Ingeniería en Electrónica y Automática
Industrial**

Trabajo Fin de Grado
Diseño de un robot autónomo: plataforma base y
control del movimiento.

Autor: Miguel Tradacete Ágreda

Tutor: Julio Pastor Mendoza

TRIBUNAL:

Presidente: D. José Luis Martín Sánchez

Vocal 1º: D. Pedro Alfonso Revenga De Toro

Vocal 2º: D. Julio Pastor Mendoza

Fecha:

Agradecimientos

Este trabajo de fin de grado está dedicado especialmente a mi familia, mis padres y mi hermana, por el amor y apoyo incondicional que me han dado siempre durante toda mi vida; a mis compañeros y amigos del equipo Utopian Robotics Rodrigo, Gonzalo y Mario por toda su ayuda; a mis también compañeros de carrera y amigos Eduardo, Juan, Pablo y Sergio los cuales han hecho de mi paso por la carrera una de las mejores etapas de mi vida; a Julio por darme la oportunidad de realizar este proyecto y lanzarme al mundo de la robótica; a Pedro por toda su ayuda; y a aquellas personas que después de días enteros de trabajo me recargaban de energía y de amor.

Índice

1. RESUMEN	3
2. ABSTRACT	3
3. PALABRAS CLAVE	3
4. RESUMEN EXTENDIDO	4
5. MEMORIA	8
5.1 INTRODUCCIÓN	8
5.1.1 <i>Presentación</i>	8
5.1.2 <i>La competición y objetivos</i>	8
5.1.3 <i>Organización del libro</i>	10
5.2 DISEÑO MECÁNICO.....	1
5.2.1 <i>Modelado dinámico y cinemático</i>	1
5.2.1.1 <i>Aceleración máxima en las fases de aceleración y frenado</i>	3
5.2.2 <i>Cálculo de motores</i>	7
5.2.3 <i>Encoders</i>	12
5.2.3.1.1 <i>Encoder v.1</i>	13
5.2.3.1.2 <i>Encoder v.2</i>	18
5.2.4 <i>Ruedas motrices</i>	19
5.2.5 <i>Estructura</i>	21
5.3 DISEÑO ELECTRÓNICO.....	28
5.3.1 <i>Mini DK2: LPC 1768</i>	28
5.3.2 <i>Tarjetas de potencia: Puente en H y protección de Mini-DK2</i>	29
5.3.3 <i>Lectura y acondicionamiento de los pulsos de los encoders</i>	33
5.3.4 <i>Circuito de alimentación</i>	38
5.3.4.1 <i>Batería</i>	38
5.3.4.2 <i>Circuito de encendido y seguridad</i>	39
5.3.4.3 <i>Estación de 5V</i>	40
5.3.5 <i>Medidor del nivel de la batería</i>	42
5.3.6 <i>Sensores de distancia</i>	42
5.3.7 <i>Esquema general de conexiones de Mini-DK2</i>	43
5.4 DISEÑO SOFTWARE	45
5.4.1 <i>Software de Control Central</i>	45
5.4.1.1 <i>Comportamiento Sensores de distancia</i>	45
5.4.1.2 <i>Librería de control del Puente en H</i>	48
5.4.1.3 <i>Comportamiento secuencia inicial</i>	49
5.4.1.4 <i>Comportamiento periódico del control central</i>	50
5.4.1.5 <i>Software de Control PI</i>	51
5.4.1.5.1 <i>Lectura de los encoders: cálculo de la velocidad</i>	51
5.4.1.5.2 <i>Controlador PI</i>	52
5.4.1.5.3 <i>Limitador de la consigna</i>	55
5.4.1.5.4 <i>Adaptación al nivel de batería</i>	56
5.4.2 <i>Modelado y simulación de la planta del robot y su controlador</i>	58
5.4.2.1 <i>Recogida de datos y su procesamiento</i>	58
5.4.2.2 <i>Modelado de planta y simulación</i>	63
5.4.2.3 <i>Modelado, ajuste y simulación del controlador</i>	69
5.4.3 <i>Comunicación con la tarjeta de estrategia</i>	74
5.4.3.1 <i>Protocolo y tiempos</i>	74
5.4.4 <i>Estructuración y prioridades de las interrupciones</i>	77
5.4.5 <i>Trayectorias y odometría</i>	80
5.4.5.1 <i>Lógica de las trayectorias</i>	81
5.4.5.2 <i>Odometría</i>	82
5.5 CONCLUSIONES.....	94
5.6 TRABAJOS FUTUROS	95

6.	PLANOS	96
6.1	PLANOS MECÁNICOS	96
6.2	PLANOS DE LA ELECTRÓNICA.....	113
6.2.1	<i>Protección de Mini-DK2</i>	<i>113</i>
6.2.2	<i>Lectura y acondicionamiento de Encoder v.1.....</i>	<i>113</i>
6.2.3	<i>Circuito de alimentación</i>	<i>114</i>
6.2.4	<i>Estación de 5V.....</i>	<i>114</i>
6.2.5	<i>Medidor del nivel de batería.....</i>	<i>115</i>
7.	PLIEGO DE CONDICIONES.....	116
7.1	CONDICIONES DE DISEÑO	116
7.2	CONDICIONES DE MATERIALES Y EQUIPOS	116
7.2.1	<i>Electrónica</i>	<i>116</i>
7.2.2	<i>Elementos mecánicos y estructurales</i>	<i>116</i>
7.2.3	<i>Baterías.....</i>	<i>116</i>
7.2.4	<i>Cableado.....</i>	<i>117</i>
7.2.5	<i>Equipos mecánicos</i>	<i>117</i>
7.2.6	<i>Equipos electrónicos.....</i>	<i>117</i>
7.2.7	<i>Equipos informáticos.....</i>	<i>117</i>
7.2.7.1	<i>Recursos hardware</i>	<i>118</i>
7.2.7.2	<i>Recursos software.....</i>	<i>118</i>
7.2.8	<i>Seguridad e Higiene</i>	<i>118</i>
7.2.9	<i>Personal.....</i>	<i>118</i>
7.2.10	<i>Condiciones de ejecución.....</i>	<i>119</i>
8.	PRESUPUESTO	120
8.1	INTRODUCCIÓN	120
8.2	COSTE MANO DE OBRA.....	120
8.3	PRESUPUESTO MATERIAL	121
8.3.1	<i>Elementos Mecánicos de Tracción</i>	<i>121</i>
8.3.2	<i>Encoder v.1.....</i>	<i>122</i>
8.3.3	<i>Encoder v.2.....</i>	<i>122</i>
8.3.4	<i>Estructura Externa</i>	<i>122</i>
8.3.5	<i>Tarjeta de Control Central</i>	<i>123</i>
8.3.6	<i>Tarjetas de Potencia.....</i>	<i>123</i>
8.3.7	<i>Electrónica Encoder v.1</i>	<i>124</i>
8.3.8	<i>Alimentación.....</i>	<i>124</i>
8.3.9	<i>Electrónica Medición de Batería.....</i>	<i>125</i>
8.3.10	<i>Sensores de Distancia</i>	<i>125</i>
8.4	COSTE DE LAS AMORTIZACIONES	126
8.5	COSTE GLOBAL	126
9.	MANUAL DE USUARIO.....	127
9.1	MEDIDAS DE SEGURIDAD	127
9.2	MANUAL DE USUARIO	128
9.2.1	<i>Encendido y conexionado.....</i>	<i>128</i>
9.2.2	<i>Configuración del control PI.....</i>	<i>131</i>
9.2.3	<i>Protocolo de comunicación.....</i>	<i>131</i>
10.	BIBLIOGRAFÍA	133
11.	ANEXOS.....	135
11.1	PLANOS ELECTRÓNICA: TARJETAS ADQUIRIDAS.....	135

1. Resumen

Este proyecto tiene su origen en la unión de un grupo de estudiantes de la Escuela Politécnica Superior de la Universidad de Alcalá con el objetivo de participar en una competición internacional de robótica en la *Charles University* de Praga. La participación en ésta requería del diseño y construcción de un robot totalmente autónomo, capaz de realizar distintas tareas y cumplir ciertas características definidas en las bases de la competición. Se decidió dividir el robot en distintas partes, asignando a cada componente del grupo el desarrollo de una de ellas.

El autor de este trabajo e integrante de este grupo ha sido el encargado del diseño y construcción de la parte motriz y todo el software de control que ello conlleva. Se crea así una plataforma móvil de carácter modular, la cual sustenta al resto del robot. Se comienza con el diseño y estimación de los parámetros del robot, siguiendo con la construcción de la plataforma consecuentemente. Finalmente se trabaja en el control software de los motores y la odometría, desde la simulación a la aplicación real.

2. Abstract

This project is the result of the collaboration between a four-student team from the Higher Polytechnic School of University of Alcalá (Universidad de Alcalá) to participate in an international robotic competition in Charles University of Prague. The competition required teams to design and construct a fully autonomous robot capable of carrying out a variety of tasks that met the competition criteria. Each team member was assigned a part of the robot to design and construct for the competition.

The author of this work designed and built the driving parts of the robot, and control software. As a result, a mobile platform with a modular nature was created that supported the rest of the robot. The first part of the process was designing and estimating robot parameters, while continuing with the platform construction. Finally, motor software control and odometry were developed, starting with simulation and later to real application.

3. Palabras clave

Robot autónomo.

Análisis dinámico.

Controlador PID para motores y adaptación de batería.

Encoder y odometría.

4. Resumen extendido

Este proyecto tiene su origen en la unión de un grupo de cuatro estudiantes de la Escuela Politécnica Superior de la Universidad de Alcalá con el objetivo de participar en la competición internacional de robótica “*Robotic Day*” en la *Charles University* de Praga, en la especialidad “*Roadside Assistance*” [1]. La participación en ésta requería del diseño y construcción de un robot totalmente autónomo, capaz de realizar distintas tareas y cumplir ciertas características definidas en las bases de la competición. Se decidió dividir el robot en distintas partes, asignando a cada componente del grupo el desarrollo de una de ellas. Estas partes son: desarrollo de la estrategia, sistema de posicionamiento por balizado, diseño de los brazos robóticos, y el diseño y construcción de la plataforma base y el control del movimiento.

Así pues, este proyecto nace de esa división, siendo éste la parte de la plataforma base y el control del movimiento del robot por el campo de juego. En el presente trabajo se expone el diseño y construcción de la parte motriz y todo el software de control que ello conlleva. La finalidad del proyecto es la de crear una estructura robusta, versátil en cuanto al control de los movimientos y de carácter modular que permitiera al resto de compañeros desarrollar el resto del robot en paralelo.

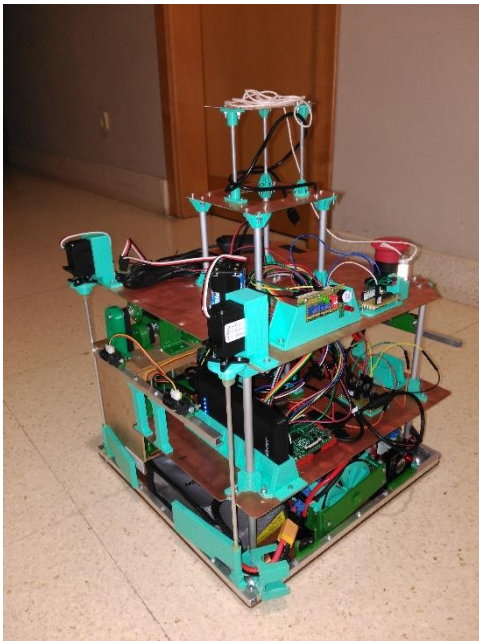


Figura 2. Robot Alan. Vista general

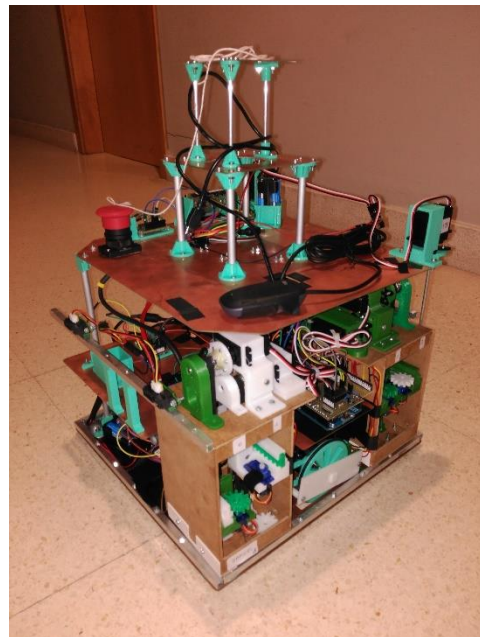


Figura 1. Robot Alan. Vista general 2

Este trabajo se divide en 3 bloques principales: diseño mecánico, diseño electrónico y diseño software. En el primer bloque del diseño mecánico se describen los cálculos de los parámetros dinámicos del robot, de la elección de los motores y del diseño de todos los elementos mecánicos del proyecto. Especialmente se analizan la aceleración máxima que debe alcanzar el robot, la potencia requerida por los motores y todo el diseño de los elementos mecánicos y estructurales. Destacar que se diseñan unos encoders propios (*Encoders v.1*) cuyas piezas son impresas en 3D, aunque finalmente se compran unos comerciales ya que aumentan las prestaciones del control del movimiento.

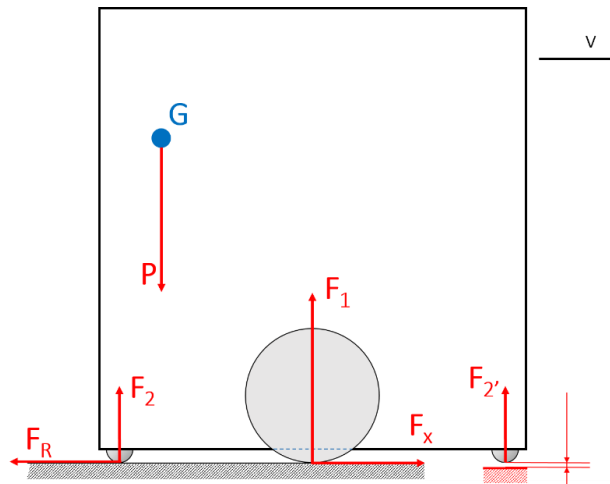


Figura 7. Esquema de fuerzas del robot. Consideración un solo apoyo

El segundo bloque, diseño electrónico, muestra toda la electrónica desarrollada e implementada en este proyecto. Se realiza un seguimiento de todos los cálculos y consideraciones necesarias para comprender la elección de esta electrónica. Dentro de ésta encontramos las tarjetas de potencia (puente en H y protección de la Mini-DK2/LPC1768), la electrónica de acondicionamiento de los *Encoders v.1*, el circuito de alimentación, el medidor del nivel de la batería y el conexionado de los sensores de distancia. Finalmente se indican todas las conexiones de transmisión de datos entre todos estos elementos.

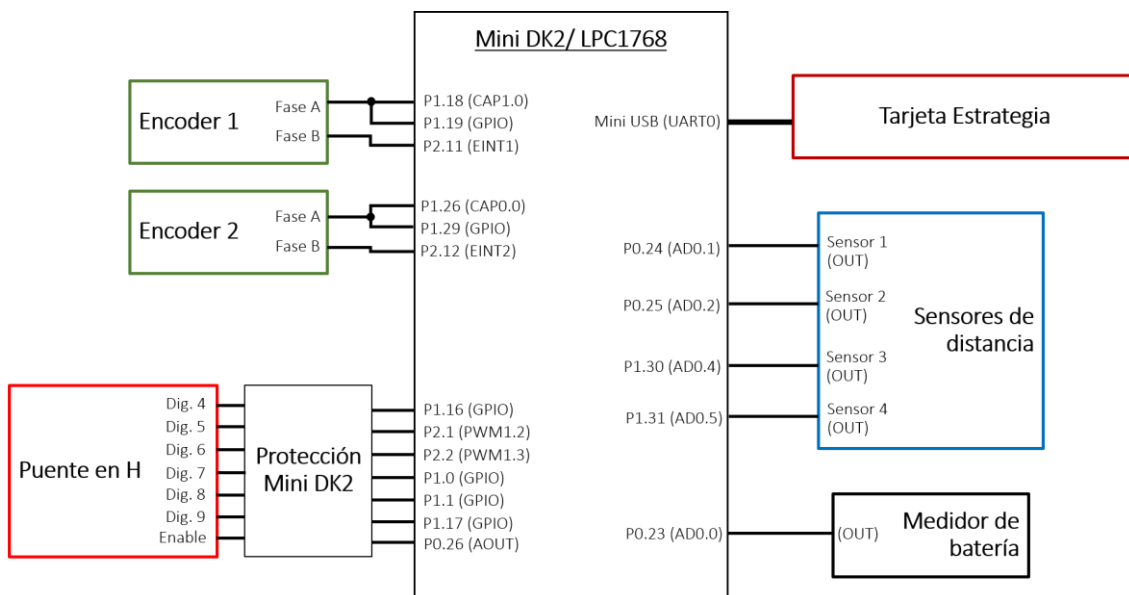


Figura 69. Esquema del conexionado de los elementos electrónicos a la Mini-DK2

En el diseño software, y tercer bloque principal de este trabajo, se describe todo el diseño a nivel software del proyecto y las correspondientes consideraciones tenidas en cuenta. Se desarrolla toda la programación de la Mini-DK2/LPC1768, tarjeta basada en el microcontrolador LPC1768, Cortex-M3 de ARM. Con ella se realiza toda la

implementación software de control central, viéndose la forma de interacción con los elementos electrónicos antes mencionados, sus prioridades asociadas, las mediciones del hardware y el control de todo el sistema. La CPU trabaja a 100 MHz y sus periféricos a 25MHz, suficiente velocidad para el correcto funcionamiento de este proyecto.

El controlador PI para los motores se desarrolla en torno a la simulación del modelo de la planta de cada motor. Estos modelos se obtienen mediante el procesamiento en Matlab de los datos recogidos de los encoders. Tras este proceso se comprueba el correcto funcionamiento de los controladores en el robot real.

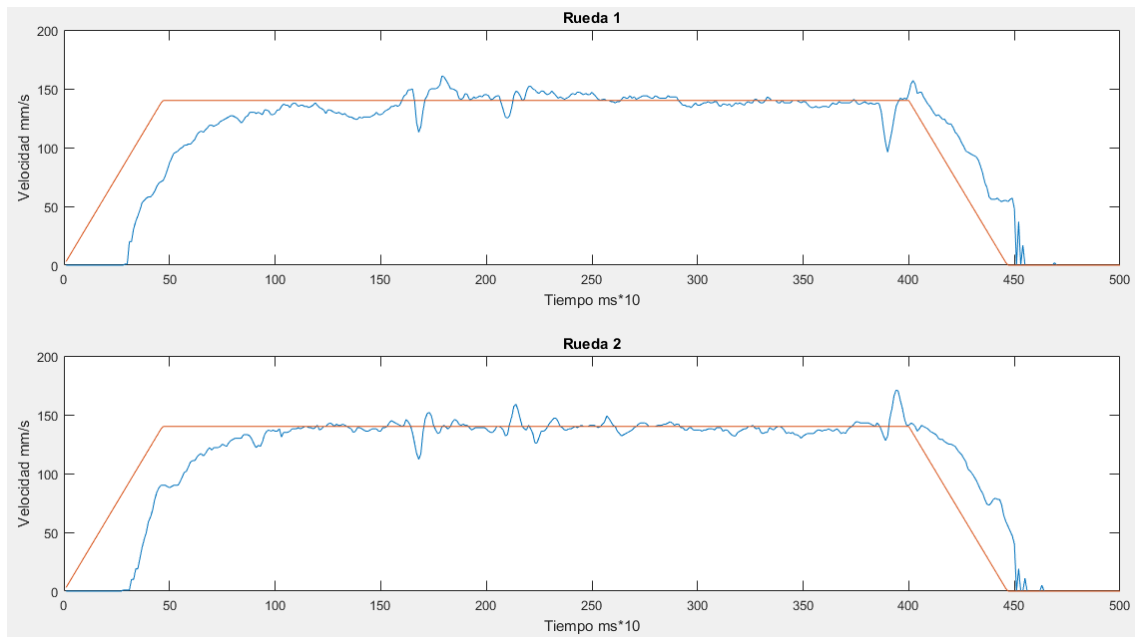


Figura 102. Perfil de velocidades de las ruedas en línea recta y referencia de 140 mm/s

Finalmente se describe la lógica de comunicación con la tarjeta de estrategia y la odometría desarrollada para el cálculo de la posición del robot en el campo de juego.

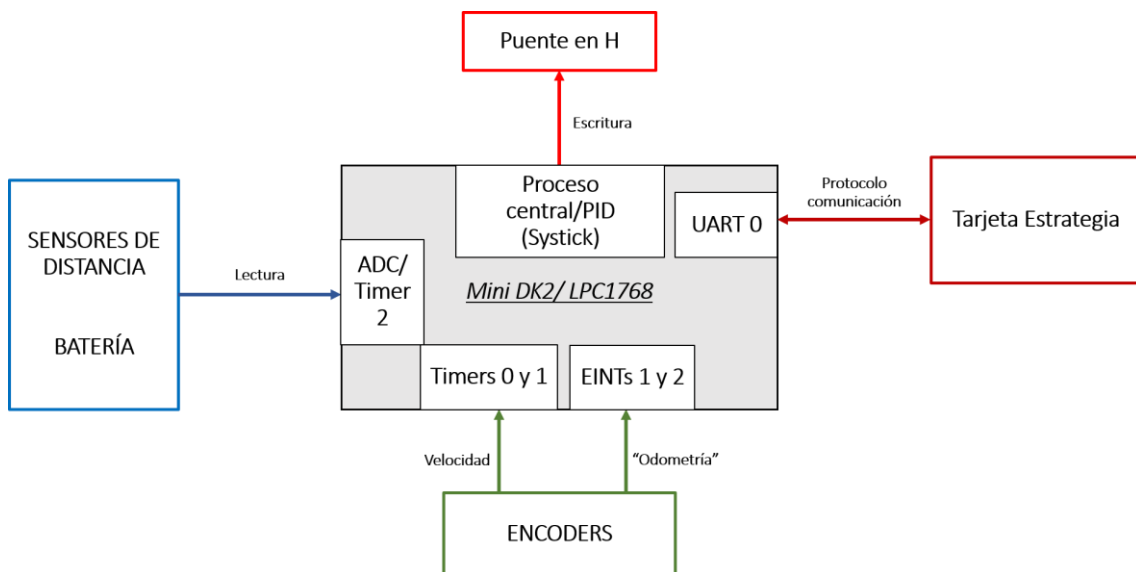


Figura 70. Esquema general de la relación entre las partes del software

Finalmente, este trabajo concluye con la exposición de los planos de las piezas diseñadas y de la electrónica diseñada, del presupuesto total, y del pliego de condiciones y manual de usuario para el correcto uso de este proyecto.

5. Memoria

5.1 Introducción

5.1.1 Presentación

En este trabajo de fin de grado se desarrolla una plataforma base móvil para un robot autónomo, orientado a participar en la competición “*Road Assistance*” del “*Robotic Day*” en Praga [1]. Se recoge así toda la evolución de este proyecto, con el objetivo de que futuros estudiantes que se embarquen en un proyecto de índole similar puedan aprender de los éxitos y fracasos de este trabajo.

5.1.2 La competición y objetivos

“*Robotic Day*” es un encuentro a nivel europeo para los amantes de la robótica, donde se celebran multitud de actividades y competiciones. Este encuentro tiene lugar en la *Charles University* de Praga, República Checa. El evento dura dos días, siendo el primero solo para participantes y el segundo además para todo el que quiera asistir como público a las competiciones y disfrutar del ambiente.

Este año 2017 en “*Robotic Day*” tuvieron lugar multitud de competiciones y a distintos niveles. Este proyecto está centrado en la competición “*Roadside Assistance*”. Esta competición simulaba un accidente de tráfico a pequeña escala, donde cada equipo participante debía ser capaz de crear un robot que, de forma autónoma, brindara asistencia como si de una carretera real se tratara. Había tres cometidos principales en esta competición en los cuales el robot debía demostrar sus habilidades. Los cometidos eran: recoger y depositar unas baterías en unos coches, retirar de la carretera un camión accidentado y señalizar la zona con unas señales de tráfico, y recoger la carga que transportaba y dejarla adecuadamente en otro camión.

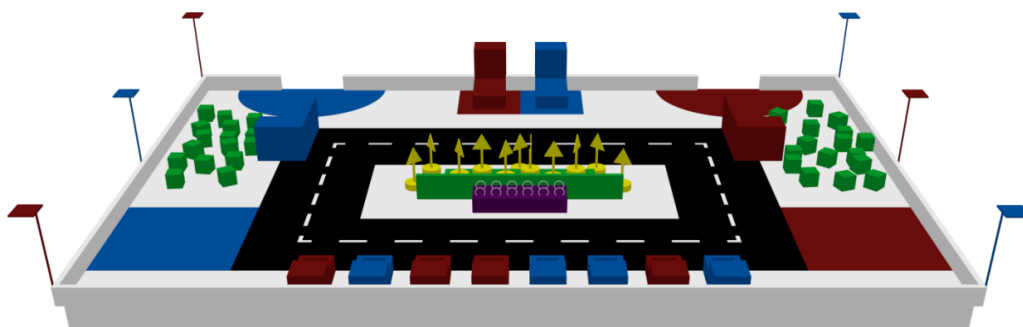


Figura 3. Campo de juego de “*Road Assistance*”

El propósito principal de este proyecto ha sido el de participar en la competición “*Roadside Assistance*” junto a unos compañeros (Figura 3), formando así el equipo Utopian Robotics. Aunque cada uno de nosotros desarrollamos una parte distinta del robot a todos nos unía la ambición por crear desde cero este robot, movidos por la

superación personal que requería tanto a nivel grupal como individual. Así, uno de los objetivos fue el de aprender a trabajar en un entorno de grupo de trabajo.

A nivel técnico, participar en la competición suponía el objetivo de pasar la fase de homologación, cumpliendo los criterios de seguridad, dimensionado y funcionamiento [1]. Estos se podrían resumir en tres: el robot debía tener un sistema de parada de emergencia, no colisionar con el robot contrincante y al menos puntuar en alguna de las tareas planteadas.



Figura 4. Robot Alan en competición "Road Assistance". A la izquierda Gonzalo, a la derecha Mario

Este proyecto ha tenido siempre la finalidad de diseñar una plataforma robusta y versátil. Así pues, los objetivos del proyecto son los siguientes:

- Crear una estructura robusta y modular que permitiera al resto del equipo desarrollar su parte convenientemente.
- Construir un sistema motriz con la fuerza necesaria para mover todo el conjunto del robot.
- Desarrollar un controlador para los motores y proporcionar un movimiento estable al robot.
- Desarrollar un sistema de odometría y transmitir la información de la posición a la estrategia del robot.

5.1.3 Organización del libro

Este trabajo consta de 2 partes diferenciadas: memoria, y sus complementos (presupuesto, pliego de condiciones, manual, planos y anexos) fundamentándola. A su vez, la memoria está dividida en 3 apartados:

- ***Diseño Mecánico***

Este apartado describe el proceso seguido del cálculo de los parámetros dinámicos, la elección de los motores y el diseño de todos los elementos mecánicos del proyecto. Uno de los parámetros dinámicos más importante es la aceleración máxima, condicionando las prestaciones de los motores. Dentro del diseño de los elementos mecánicos se hace una focalización especial en los encoders ya que son parte fundamental en el proyecto y robot total, construyéndose los *Encoders v.1* y más adelante adquiriendo los *Encoders v.2*.

- ***Diseño Electrónico***

Dentro de este apartado se encuentra descrita toda la electrónica, tanto la diseñada por el autor de este trabajo como la adquirida, así como sus modificaciones. Se explican todos los cálculos y consideraciones que son necesarias para comprender la elección de la electrónica expuesta. Al final del apartado se muestra un esquema de todas las conexiones de transmisión de datos entre los elementos del proyecto.

- ***Diseño Software***

En este apartado se describe todo el diseño a nivel software y las consideraciones tenidas en cuenta para éste. Se comienza detallando toda la lógica de programación de la tarjeta de control central (Mini-DK2) y como interactúa con los distintos elementos. Seguidamente se describe el proceso de modelización de los motores y su control. Finalmente se muestra el protocolo de comunicación con la tarjeta de estrategia y el método de cálculo desarrollado de la odometría.

5.2 Diseño Mecánico

5.2.1 Modelado dinámico y cinemático

La dinámica de un robot es un aspecto esencial en su diseño mecánico y aporta ciertos parámetros clave que determinan los límites en los movimientos del robot. En el estudio de la dinámica se ha de considerar el movimiento más restrictivo y más común que llevará a cabo el robot, siendo éste un movimiento de desplazamiento lineal. Este movimiento constará de una fase de aceleración, una fase de velocidad constante y otra fase de frenado. Todo este estudio se ha realizado siguiendo la metodología indicada por Javier Baliñas [2] que a su vez fue obtenida de [3].

En este proyecto, el robot presenta una estructura de dos ruedas de tracción diferencial y cuatro apoyos pasivos. Dichos elementos se encuentran distribuidos de forma simétrica respecto a los planos centrales y paralelos a los laterales del robot (Figura 5).

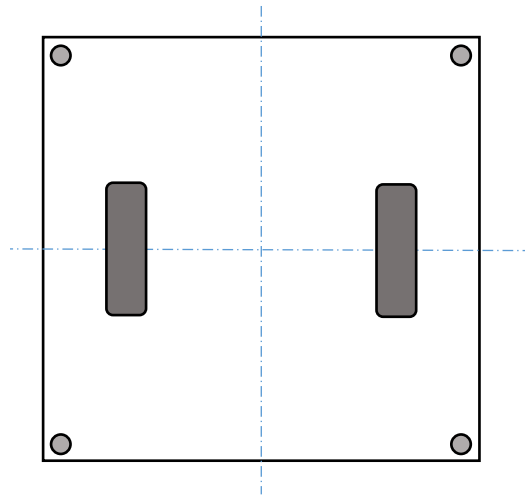


Figura 5. Planta del robot. Distribución de ruedas y apoyos

Es por ello que para el análisis dinámico se pueden estudiar las fuerzas respecto al perfil del robot (Figura 6). La fuerza “F1” representa la fuerza normal de la rueda contra el suelo, “ F_x ” representa la fuerza realizada por cada rueda, “F2” las fuerzas de reacción de los apoyos (roll-on), “ F_R ” la fuerza de rozamiento de los apoyos contra el suelo (representada solo en un apoyo a modo explicativo, siendo para el resto de apoyos igual), “G” representa la situación del centro de masas (descentrado ya que la disposición real de los elementos internos no será simétrica), “P” representa el peso de todo el conjunto y “v” la dirección de la velocidad.

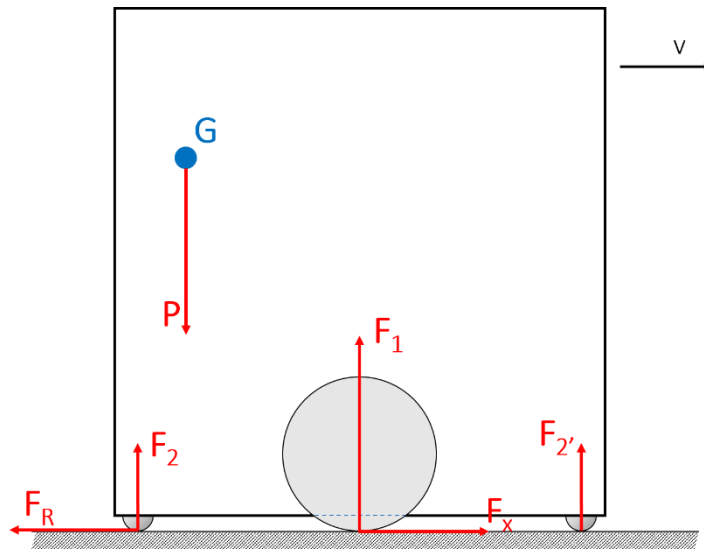


Figura 6. Esquema de fuerzas del robot

Como resultado de este análisis se buscan las aceleraciones límites tanto en la fase de aceleración como en la fase de frenado. Una consideración importante es que aunque el robot cuente con dos puntos de apoyo (vista de perfil), en cada caso a analizar solo apoya uno ya que los tres puntos de apoyo no están situados exactamente a la misma altura para mantener siempre el contacto de la rueda de tracción en caso de que hubiera pequeñas irregularidades en el campo. Además, si se consideraran los dos *roll-on* apoyados (las tres ruedas a la misma altura) se llega a un caso sin resolución posible, ya que si tanto el apoyo delantero como el trasero apoyan la rueda deslizaría.

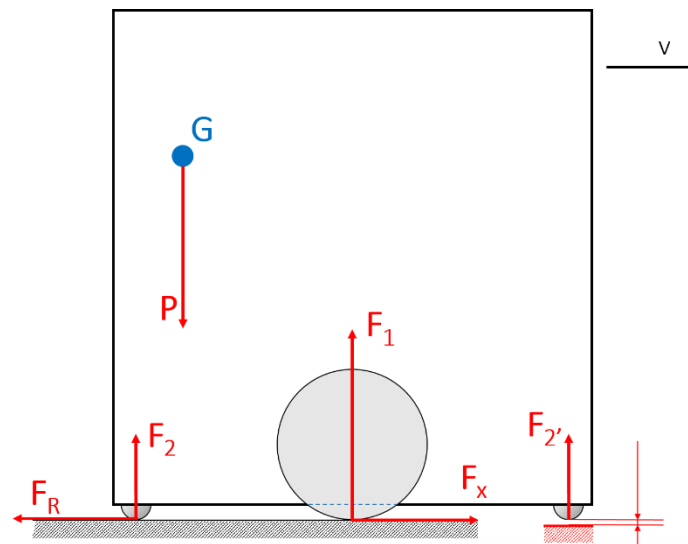
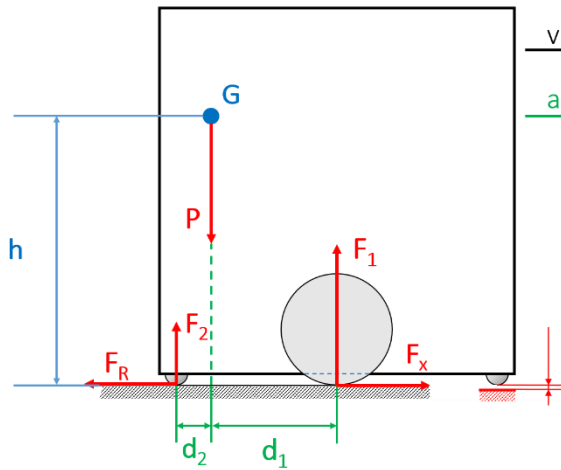


Figura 7. Esquema de fuerzas del robot. Consideración un solo apoyo

5.2.1.1 Aceleración máxima en las fases de aceleración y frenado

En primer lugar se analizan las distintas fuerzas para cada caso individual y una vez halladas las ecuaciones de las fuerzas que actúan se pasa a calcular la aceleración máxima atendiendo a la condición de no deslizamiento de las ruedas.

- Caso 1: Fase de aceleración hacia adelante



- P : peso
- h : altura al centro de masas
- F_1' : fuerza de reacción de las dos ruedas
- F_2' : fuerza de reacción de los dos apoyos
- F_1 : fuerza de reacción de una rueda
- F_2 : fuerza de reacción de un apoyo
- F_R : fuerza de rozamiento de un apoyo
- F_x : fuerza ejercida por una rueda
- G : centro de masas
- d_2 : distancia desde apoyo a "G"
- d_1 : distancia de "G" a rueda
- m : masa del conjunto
- a : aceleración del movimiento
- g : aceleración de la gravedad
- μ : coef. de rozamiento del apoyo

Figura 8. Caso 1: Aceleración hacia adelante

Suma de fuerzas verticales: $P = F_1' + F_2' = 2 \cdot F_1 + 2 \cdot F_2$ (1)

Suma de fuerzas horizontales ("v" cte.): $F_R \cdot 2 = F_x \cdot 2$ (2)

Suma de pares aplicados al punto G: $F_1 \cdot d_1 + F_x \cdot h - F_2 \cdot d_2 - F_R \cdot h = 0$ (3)

Desarrollando el sistema de ecuaciones

$$\left. \begin{aligned} P = F_1' + F_2' = 2 \cdot F_1 + 2 \cdot F_2 \\ F_1 \cdot d_1 + F_x \cdot h - F_2 \cdot d_2 - F_R \cdot h = 0 \end{aligned} \right\} \quad (4)$$

y sabiendo que

$$P = m \cdot g \quad (5)$$

$$F_x = m \cdot a \quad (6)$$

$$F_R = |F_2| \cdot \mu \quad (7)$$

obtenemos:

$$F_2 = \frac{m}{d_1 + d_2 + \mu \cdot h} \cdot \left(g \cdot \frac{d_1}{2} + a \cdot h \right) \quad (8)$$

$$F_1 = \frac{m}{d_1 + d_2 + \mu \cdot h} \cdot \left(\frac{g(d_2 + \mu \cdot h)}{2} - a \cdot h \right) \quad (9)$$

- Caso 2: Fase de frenado hacia adelante

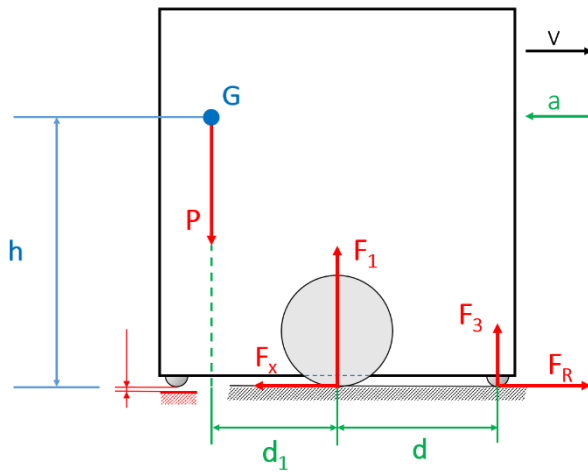


Figura 9. Caso 2: Frenado hacia adelante

- P : peso
- h : altura al centro de masas
- F_1' : fuerza de reacción de las dos ruedas
- F_3' : fuerza de reacción de los dos apoyos
- F_1 : fuerza de reacción de una rueda
- F_3 : fuerza de reacción de un apoyo
- F_R : fuerza contraria de reacción
- F_x : fuerza ejercida por una rueda
- G : centro de masas
- d_1 : distancia desde "G" a rueda
- d : distancia de rueda a apoyo
- m : masa del conjunto
- a : aceleración del movimiento
- g : aceleración de la gravedad
- μ : coef. de rozamiento del apoyo

Suma de fuerzas verticales: $P = F_1' + F_3' = 2 \cdot F_1 + 2 \cdot F_3$ (10)

Suma de fuerzas horizontales ("v" cte.): $F_R \cdot 2 = F_x \cdot 2$ (11)

Suma pares aplicados al punto G: $F_1 \cdot d_1 - F_x \cdot h + F_3 \cdot (d_1 + d) + F_R \cdot h = 0$ (12)

Desarrollando el sistema de ecuaciones

$$\left. \begin{aligned} P = F_1' + F_3' = 2 \cdot F_1 + 2 \cdot F_3 \\ F_1 \cdot d_1 - F_x \cdot h + F_3 \cdot (d_1 + d) + F_R \cdot h = 0 \end{aligned} \right\} \quad (13)$$

y sabiendo que

$$P = m \cdot g \quad (5)$$

$$F_x = m \cdot a \quad (6)$$

$$F_R = |F_3| \cdot \mu \quad (14)$$

obtenemos:

$$F_3 = \frac{m}{d + \mu \cdot h} \cdot \left(a \cdot h - g \cdot \frac{d_1}{2} \right) \quad (15)$$

$$F_1 = \frac{m}{d + \mu \cdot h} \cdot \left(g \cdot \frac{(d_1 + d + \mu \cdot h)}{2} - a \cdot h \right) \quad (16)$$

- Caso 3: Fase de aceleración hacia atrás

El estudio de las fuerzas es idéntico al caso 2, ya que tanto los apoyos como las componentes vectoriales son idénticas. Solo varía el sentido de la velocidad y el significado que le damos a las ecuaciones resultantes.

- Caso 4: Fase de frenado hacia atrás
En este caso se encuentra otra similitud. El estudio del caso 4 es idéntico al caso 1.
- Cálculo de las aceleraciones máximas: condición de no deslizamiento

Una vez analizada la dinámica del robot y las fuerzas que toman partido en los distintos movimientos, se procede a aplicar la condición de no deslizamiento de las ruedas. Ésta consiste en que la fuerza de empuje o retención ejercida por la rueda nunca sea mayor a la fuerza normal de reacción en la rueda, multiplicada por un factor de adherencia K_a [2].

$$F_x < K_a \cdot F_1 \quad (17)$$

Así se obtiene de forma teórica que en los casos 1 y 4:

$$a_{máx 1} = \frac{K_a \cdot g \cdot (d_2 + \mu \cdot h)}{2 \cdot (d_1 + d_2 + \mu \cdot h + K_a \cdot h)} \quad (18)$$

De igual modo en los casos 2 y 3 se obtiene que:

$$a_{máx 2} = \frac{K_a \cdot g \cdot (d_1 + d + \mu \cdot h)}{2 \cdot (d + \mu \cdot h + K_a \cdot h)} \quad (19)$$

Ahora se debe cuantificar estas aceleraciones. Para ello se debe suponer un caso optimista de la posición del centro de masas. Este punto es crucial y puede ser variado con facilidad en la fase de diseño. Aunque no pudiera ser modificado con facilidad, al suponer un caso optimista (centro de masas bajo y centrado) obtendremos unas aceleraciones límites mayores. De esta forma si el cálculo de los motores lo basáramos en estas aceleraciones no perderíamos prestaciones a la hora de dimensionar los motores.

Más adelante, en la fase de pruebas se determinará el límite real de las aceleraciones si fuera necesario, en función del centro de masas real.

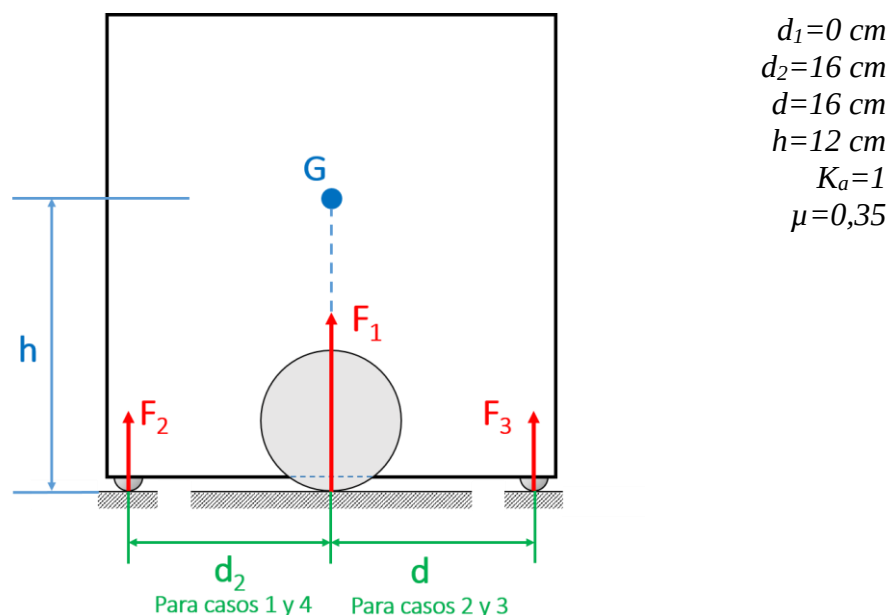


Figura 10. Caso optimista. Representación genérica, no todas las fuerzas

Al hacer esta suposición se encuentra que, al estar el centro de masas en el centro geométrico, todos los casos se aúnan en uno solo y obtenemos así la misma aceleración tanto en la fase de aceleración como en la de frenado.

El factor de adherencia ha sido elegido con ese valor ($K_a=1$) porque el material con el que se fabricarán las ruedas ya se ha encontrado en otros trabajos muy adherente en la madera [2]. El coeficiente de rozamiento se elige como 0,35 ya que aunque el apoyo *roll-on* disminuye este coeficiente, se supone el peor caso en el que pudiera estar atascado siendo un rozamiento directo entre acero y madera [4].

Obtenemos así el valor de la aceleración máxima:

$$a_{m\acute{a}x} = \frac{K_a \cdot g \cdot (d_2 + \mu \cdot h)}{2 \cdot (d_1 + d_2 + \mu \cdot h + K_a \cdot h)} \quad (20)$$

$$a_{m\acute{a}x} = 3,1 \frac{m}{s^2}$$

➤ Condición de elevación

Con el fin de determinar la estabilidad total del robot, se analiza la condición de elevación que consiste en saber si una aceleración elevada y repentina podría levantar las ruedas motrices robot. Para ello se supone que la fuerza normal de reacción en las ruedas debe ser estrictamente mayor que cero, garantizando que las ruedas siempre están apoyadas y por tanto que el robot no se sostiene solamente en los apoyos delanteros y traseros.

$$F_1 > 0 \quad (21)$$

Casos 1 y 4:

$$F_1 = \frac{m}{d_1 + d_2 + \mu \cdot h} \cdot \left(\frac{g(d_2 + \mu \cdot h)}{2} - a \cdot h \right) \quad (22)$$

Casos 2 y 3:

$$F_1 = \frac{m}{d + \mu \cdot h} \cdot \left(g \cdot \frac{(d_1 + d + \mu \cdot h)}{2} - a \cdot h \right) \quad (23)$$

Se toman los mismos valores de las variables que en la consideración para calcular la aceleración máxima ($d_1=0$ cm; $d_2=16$ cm; $d=16$ cm; $h=12$ cm; $K_a=1$; $\mu=0,35$).

Se puede ver que en todos los casos la primera fracción de las ecuaciones es mayor que cero, ya que todas las variables son mayores que cero (excepto $d_1=0$ cm). Entonces la única posibilidad de que F_1 sea cero o menor es cuando:

$$g \cdot \frac{(d_1 + d + \mu \cdot h)}{2} < a \cdot h \quad (24)$$

Por lo que la aceleración mínima de levantamiento de la rueda motriz es:

$$a_{\text{mín}} = 8,26 \frac{m}{s^2}$$

Esta aceleración garantiza que las ruedas motrices estén siempre pegadas al suelo y por tanto el robot no vuelque. Al ser mayor que la aceleración máxima de deslizamiento ($3,1 \frac{m}{s^2}$) el robot nunca podrá alcanzar la aceleración de levantamiento ya que antes deslizaría, dejando de acelerar consecuentemente. Finalmente se puede concluir que ya que las ruedas motrices no se podrán elevar del suelo tampoco existirá la posibilidad de que el robot pueda volcar por sí solo.

5.2.2 Cálculo de motores

Elegir los motores de la parte motriz es sin duda una tarea fundamental y crucial para el correcto funcionamiento de todo el proyecto. Se deben tener distintas consideraciones, como la potencia que será necesaria desarrollar por los motores, velocidad máxima, espacio disponible o incluso el presupuesto del que se dispone. Unos motores con poca potencia podrían hacer al robot demasiado lento o incluso podrían sufrir por sobrecalentamiento. Por otra parte, si se eligen unos motores muy potentes se puede poner en compromiso al resto de componentes, ya que suele implicar más espacio.

Una buena elección pasa por tener en cuenta todos estos aspectos. A continuación se exponen de forma detallada los diferentes pasos seguidos en este proyecto para la elección de los motores.

I. Estimación del sistema

Este robot realizará dos tipos de movimientos: movimientos lineales y giros sobre sí mismo (eje central). Los parámetros más restrictivos son la velocidad y la aceleración desarrollada en el movimiento lineal.

La aceleración máxima del robot nunca podrá superar la aceleración máxima de deslizamiento calculada de forma teórica en el apartado anterior ($a_{\text{máx}}=3,1 \frac{m}{s^2}$). La velocidad y la masa se estiman según el estudio de otros robots realizados para propósitos similares [5]. Así, los cálculos de este trabajo tendrán en cuenta los siguientes parámetros:

- Velocidad máxima = $0,5 \frac{m}{s}$
- Aceleración máxima = $0,5 \frac{m}{s^2}$
- Masa máxima del conjunto = 10 Kg
- Diámetro de la rueda motriz = 100 mm

Se elige una aceleración de $0,5 \frac{m}{s^2}$ ya que en un estudio previo se observa que para el propósito de este proyecto no es necesario alcanzar el límite teórico. Además, unos motores capaces de proporcionar la aceleración máxima teórica (teniendo en cuenta el resto de parámetros del robot) supondrían un coste fuera del presupuesto.

II. Determinación del par de carga

Todos los movimientos desarrollados seguirán un perfil trapezoidal de la velocidad (Figura 11). Este perfil presenta tres fases: aceleración, velocidad constante y frenada.

La aceleración y la frenada están limitadas por la inercia y el rozamiento, mientras que la fase de velocidad constante solo está limitada por el rozamiento.

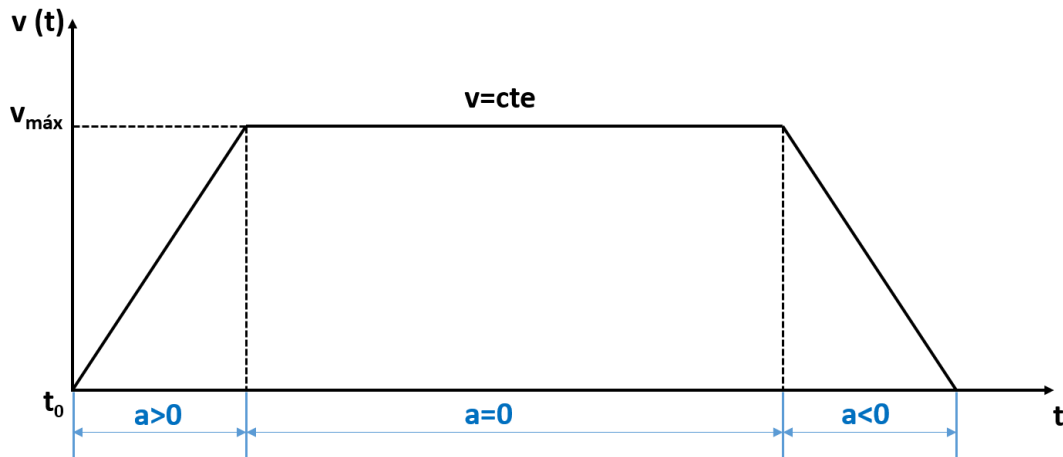


Figura 11. Perfil trapezoidal de la velocidad

La zona de aceleración es la más desfavorable, aplicándose el mayor par tratando de acelerar la masa hasta la velocidad requerida.

Los motores deben acelerar el conjunto en un tiempo t_0 hasta una velocidad lineal v . Conocemos la aceleración a y la masa m a la que queremos que el robot se desplace. También se tiene en cuenta la fuerza de rozamiento F_R consecuencia de los cuatro apoyos pasivos del robot. El coeficiente de rozamiento ($\mu=0,35$) utilizado es una aproximación genérica [4] entre madera y acero, ya que desconocen los materiales finales de construcción (tanto del campo de juego como de los apoyos de bola (*roll-on*)). Este coeficiente de rozamiento es el máximo que se podría alcanzar, en la suposición de que los apoyos de bola no giraran. En condiciones normales de funcionamiento será inferior gracias a estos.

Además, para calcular la fuerza de rozamiento se considera el peor de los casos: todo el peso en los apoyos pasivos. En condiciones normales de funcionamiento se tratará de colocar el centro de masas en la vertical de las ruedas, consiguiendo que la mayor parte del peso recaiga en ellas, aliviando los apoyos.

Se calcula así la fuerza F_M que deben desarrollar los motores:

F_M : Fuerza total a desarrollar

F : Fuerza eficaz requerida

F_R : Fuerza de rozamiento

μ : Coeficiente de rozamiento

N : Fuerza normal

P : Peso

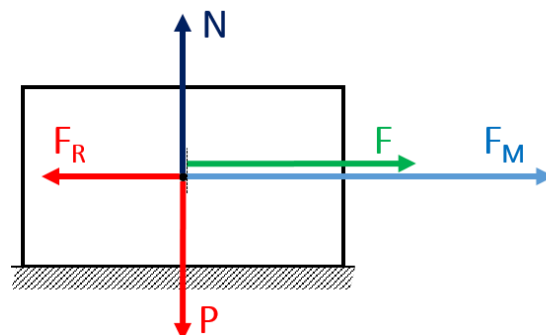


Figura 12. Esquema simplificado de fuerzas

$$F = m \cdot a \quad (25)$$

$$F = 10 \text{ Kg} \cdot 0,5 \frac{\text{m}}{\text{s}^2} = 5 \text{ N} \quad (26)$$

$$F_R = \mu \cdot N \quad (27)$$

$$|N| = |P| = m \cdot g \quad (28)$$

$$F_R = 0,35 \cdot 10 \text{ Kg} \cdot 9,81 \frac{\text{m}}{\text{s}^2} = 34,34 \text{ N} \quad (29)$$

$$F_M = F + F_R \quad (30)$$

$$F_M = 5 \text{ N} + 34,34 \text{ N} = 39,34 \text{ N} \quad (31)$$

Conociendo el radio r de la rueda motriz se calcula el par T como sigue:

$$T = F_M \cdot r \quad (32)$$

$$T = 39,34 \text{ N} \cdot \frac{0,1 \text{ m}}{2} = 1,97 \text{ N} \cdot \text{m} = 197 \text{ N} \cdot \text{cm}$$

Al utilizarse dos motores, el par mínimo que debe desarrollar cada uno es:

$$T_1 = T_2 = \frac{T}{2} = 98,5 \text{ N} \cdot \text{cm} \quad (33)$$

III. Determinación de la potencia mecánica

La potencia mecánica máxima P que deben otorgar los motores se calcula de la siguiente forma:

$$P = F_M \cdot v \quad (34)$$

$$P = 39,34 \text{ N} \cdot 0,5 \frac{\text{m}}{\text{s}} = 19,67 \text{ W}$$

Al utilizar dos motores, la potencia que debe ejercer cada uno es:

$$P_1 = P_2 = \frac{P}{2} = 9,84 \text{ W} \quad (35)$$

IV. Elección de los motores

Los motores finalmente utilizados son los Bosch FPC 12 V 0-130-822-019 (Figura 13). Este motor se consiguió en un desguace. Los parámetros del motor se detallan en la Tabla 1. Sus curvas de funcionamiento no han sido encontradas ya que no es un motor que actualmente se encuentre en el mercado. Todos los motores Bosch FPC presentan la misma forma en sus curvas de funcionamiento. Para estudiar su comportamiento hay que basarse en las de otro motor tipo FPC (Figura 14).

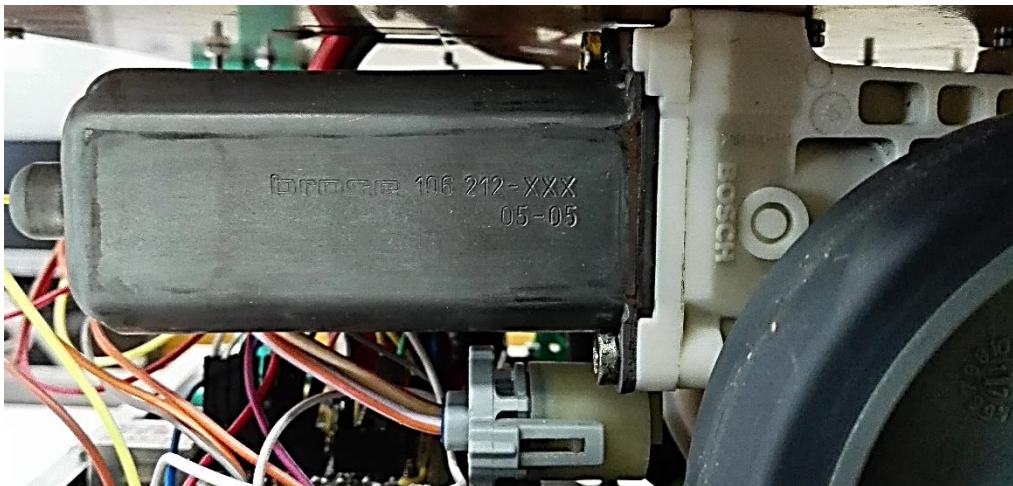


Figura 13. Motor Bosch FPC 12V 0-130-822-019

Motor	Bosch FPC
Tensión	12 V
Potencia nominal	9 W
Corriente nominal	7 A
Corriente de arranque	27 A
Máxima velocidad	82 RPM
Par continuo	1,0 Nm
Par máximo	11 Nm
Reducción	72:1
Peso	0,5 Kg

Tabla 1. Características motor Bosch FPC 12V 0-130-822-019

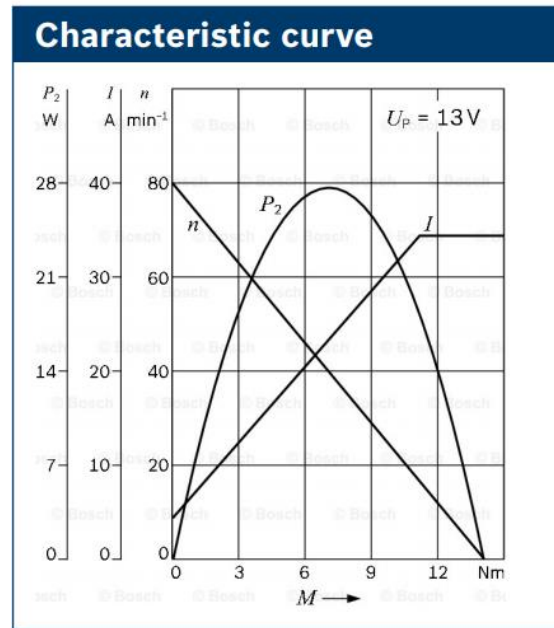


Figura 14. Curva característica de motores Bosch FPC

La velocidad máxima que pueden desarrollar es de 82 rpm, lo que equivale a una velocidad angular de $8,59 \frac{rad}{s}$ (ecuación 37). A su vez, con esta velocidad angular y conociendo el diámetro de las ruedas motrices (100 mm), podemos calcular la velocidad lineal máxima que pueden desarrollar. Dicha velocidad lineal será de 0,249 m/s.

$$1rpm = \frac{1r}{min} = \frac{2\pi rad}{60 s} \quad (36)$$

$$Velocidad\ angular_{m\acute{a}x}(\omega) = \frac{82 rpm \cdot 2\pi rad}{60 s} = 8,59 \frac{rad}{s} \quad (37)$$

$$\begin{aligned} Velocidad\ lineal_{m\acute{a}x} &= \omega \cdot \frac{D}{2} = 8,59 \frac{rad}{s} \cdot \frac{100 mm}{2} = 429 \frac{mm}{s} \\ &= 0,429 \frac{m}{s} \end{aligned} \quad (38)$$

Como se observa, la velocidad máxima de estos motores es inferior a la estimada en un 14,2%. La potencia también es inferior a la estimada, aunque en solo un 8,5%. Se ve así como estos motores están ligeramente por debajo de la estimación inicial, pero se decide utilizarlos ya que tienen unas prestaciones muy similares y además, el par que desarrollan es superior al estimado inicialmente.

5.2.3 Encoders

Los encoders son unos dispositivos electromecánicos encargados de la medición del espacio recorrido por un sistema, en nuestro caso un robot. Los encoders utilizados en este proyecto son ópticos y su gran mayoría funcionan mediante la lectura de la velocidad de un disco provisto de ranuras equitativamente distribuidas. La lectura se realiza a través de un sensor de luz, en la mayoría de los casos infrarrojo. También existen otros encoders de tipo magnético, pero en este proyecto no se verán.

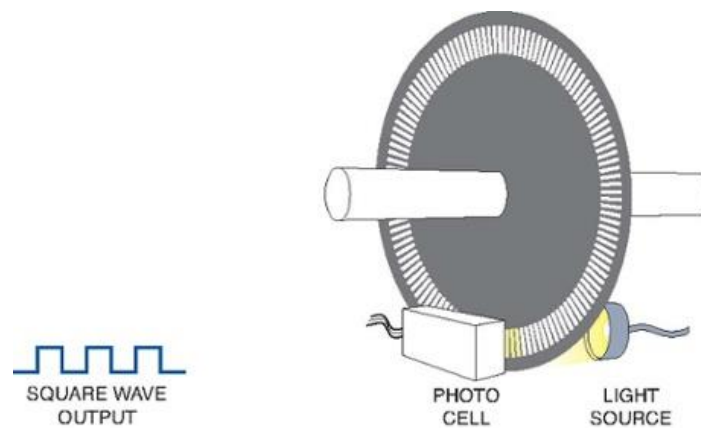


Figura 15. Lectura de encoder óptico [6]

Estos sensores ópticos disponen de un transmisor de luz y de un receptor de la misma, siendo en la mayoría de los casos una luz en el espectro infrarrojo. La forma de funcionar es simple: se hace pasar la luz emitida por la zona ranurada del disco, situado entre el transmisor y el receptor, captando así las luces y sombras producidas por el paso del ranurado entre el haz de luz. Obtenemos así una señal cuadrada de fácil digitalización.

Por lo tanto, conociendo las dimensiones del ranurado y el tiempo entre la detección de una ranura y otra, se puede calcular la velocidad a la que gira el disco. Colocando éste en el eje de un motor o en el eje de una rueda, podemos calcular la velocidad de un sistema móvil terrestre.

Además, los encoders suelen tener dos fases en cuadratura, es decir, dos señales idénticas pero desfasadas $\frac{\pi}{2}$ (90°). Con esto se consigue conocer el sentido de giro. La cuadratura se obtiene gracias a sensores dobles cuya separación esté en relación con el tamaño de las ranuras, proporcionando el desfase deseado.

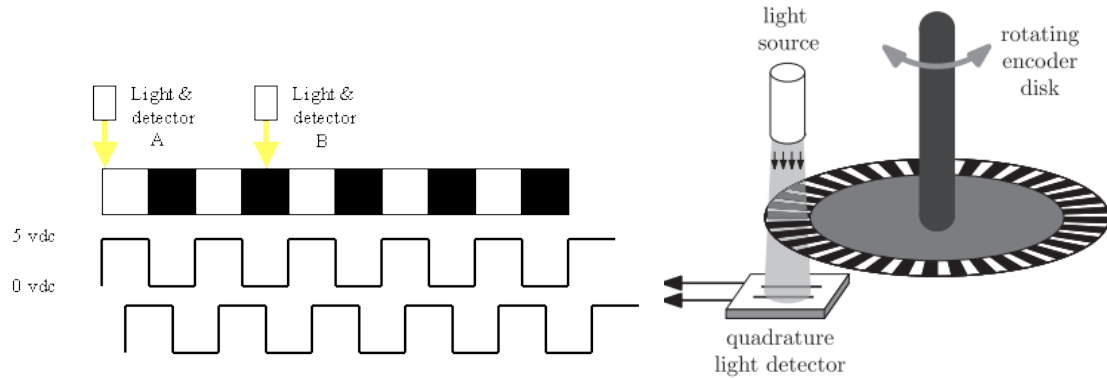


Figura 16. Lectura en cuadratura de encoder óptico [7]

En este proyecto se diseñan unos encoders (*Encoder v.1*) con el fin de demostrar que es posible la autofabricación e impresión de encoders con impresoras 3D. Más tarde se determina que para poder construir un robot competitivo es necesario incorporar a éste unos encoders comerciales (*Encoder v.2*), que garantizan una precisión profesional. Ambos son pasivos, es decir, desligados del sistema motriz para evitar falsas medidas producidas por el deslizamiento de las ruedas motrices u otros sucesos. Estos encoders miden el movimiento del robot de un modo directo.

5.2.3.1.1 *Encoder v.1*

El Encoder v.1 es la primera versión funcional de encoder que se hace. Estos encoders están diseñados en su totalidad por el autor de este proyecto, a excepción de los engranajes, que son adquiridos. El diseño abarca tanto a la parte mecánica como la electrónica. En lo concerniente a su fabricación las piezas fueron impresas por impresoras 3D Prusa i3 de la Universidad de Alcalá.

Con esta primera versión se consigue crear un prototipo funcional y demostrar así la posibilidad de autofabricación de encoders mediante impresión 3D.

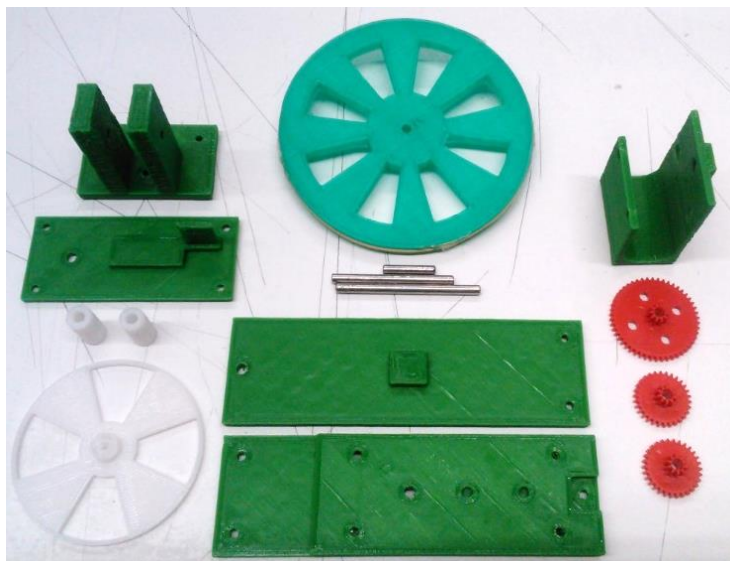


Figura 17. Piezas de Encoder v.1

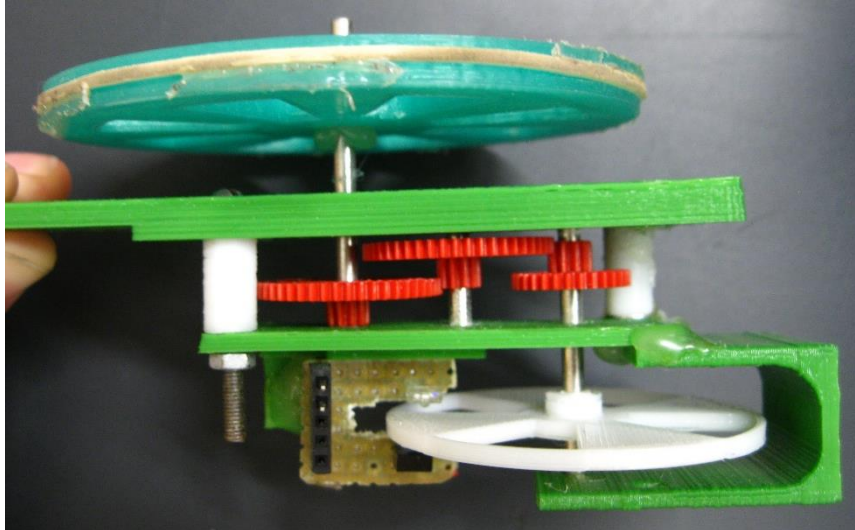


Figura 18. Encoder v.1. Vista superior

El diseño mecánico del encoder se puede separar en tres apartados principales: rueda y aceleradora, soporte y estructura móvil, y disco ranurado de lectura. Finalmente se estudia la resolución.

- **Rueda y aceleradora**

La rueda “motriz” del encoder es la encargada de recoger el movimiento del robot y transmitirlo a la aceleradora como movimiento angular. Por su parte, la aceleradora tiene como misión aumentar la resolución, incrementando la relación entre radianes desarrollados y espacio lineal. El conjunto rueda-aceleradora es así el transductor del encoder, pues transforma velocidad lineal en velocidad angular.

Para realizar una medición poco invasiva habrá que tener en cuenta dos factores:

- Perímetro de la rueda. Cuanto más grande es el perímetro de la rueda más par de fuerza se consigue, facilitando la transmisión de movimiento a la aceleradora.
- Factor de multiplicación de la aceleradora. Cuanto mayor es éste más resolución se obtiene, pero se pierde par de fuerza a razón de este factor. Esto puede repercutir en mayor o menor medida según la inercia del disco ranurado de lectura, por lo que habrá que guardar un compromiso entre este factor y el tamaño-peso del disco.

El buen engrane de los engranajes de la aceleradora también será determinante, puesto que el rozamiento entre estos disminuye el par de fuerza.

Finalmente se fabrica la rueda con un diámetro de 80 mm, con un rebaje en todo su perímetro para colocar una goma elástica que hará las veces de neumático. El factor de multiplicación de la aceleradora se fija en 25.

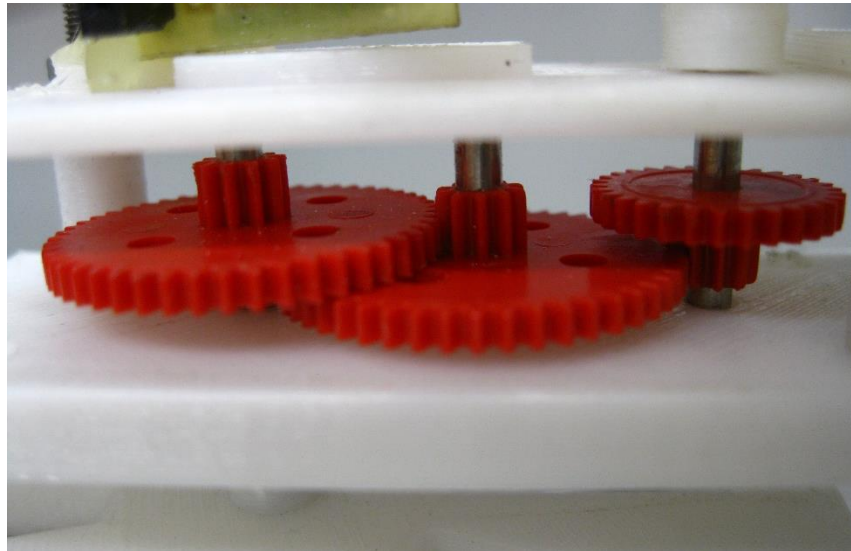


Figura 19. Aceleradora de Encoder v.1

- **Soporte y estructura móvil**

La estructura móvil es la que sustenta todo el conjunto y permite su movilidad necesaria. Esta movilidad es fundamental ya que con ella conseguimos evitar falsas medidas (adherencia constante al suelo) o mantener la integridad del propio encoder consecuencia del levantamiento inesperado de la rueda motriz opuesta.

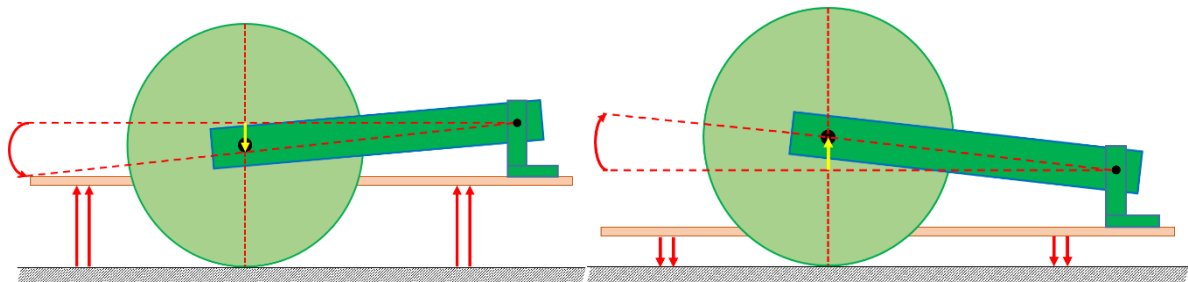


Figura 20. Adaptación al movimiento de la estructura móvil

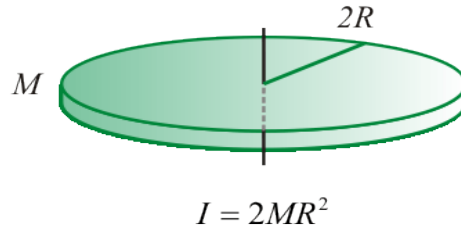
El soporte sustenta y une la estructura al robot, y permite la movilidad de todo el encoder. Este soporte está unido a la estructura por uno de sus extremos por un eje común de libre giro. Así se consigue el movimiento vertical que se detallaba anteriormente. Como este movimiento se espera pequeño, puede despreciarse el cambio de posición del eje de la rueda respecto la vertical producido por el radio de giro soporte-estructura.

- **Disco ranurado de lectura**

El disco ranurado de lectura es el encargado de transmitir la información de la velocidad angular al sensor óptico. Este disco tiene a lo largo de su perímetro agujeros o ranuras distribuidas uniformemente y siendo el espacio entre ellas de su mismo tamaño. Así, el sensor dispone del doble de cambios entre luz-sombra o flancos. Para realizar una buena medición habrá que tener en cuenta dos consideraciones:

- Momento de inercia del disco

Como se veía antes, si la inercia del disco es demasiado elevada puede llegar a dificultar el correcto funcionamiento del encoder, producir falsas medidas, e incluso producir una medición invasiva y por tanto errónea. La inercia de un disco depende de dos factores: su masa y su diámetro (Figura 21).



$$I = 2MR^2$$

Figura 21. Inercia de un disco

Al ser un disco ranurado también influye el radio en el que están situados dichos agujeros. En nuestro caso nos favorece que estén en el perímetro, pues eliminamos masa en un radio mayor y consecuentemente eliminamos más momento de inercia. Finalmente se decide fabricar el disco con un grosor de 2 mm y un diámetro de 50 mm. Se imprime en 3D ya que el plástico tiene una densidad inferior a otros materiales de construcción a la vez que presenta la suficiente rigidez.

- Tolerancia de fabricación

Es lógico pensar que cuantas más ranuras haya mayor será la resolución. De forma teórica esto es así, pero al ser un disco impreso por una impresora 3D hay que tener en cuenta la tolerancia de impresión y por tanto el valor absoluto del error cometido en relación con el tamaño de cada ranura. Este hecho se comprobó en el primer prototipo, donde se dotó de muchas ranuras al disco.

El disco disponía de 60 ranuras, lo que significa 120 flancos. Conociendo el diámetro medio en el que se encontraban las (\varnothing_{ran}), donde el sensor realiza la media, se obtiene la distancia flanco-flanco teórica.

$$\varnothing_{ran} = 45 \text{ mm}$$

$$\text{Circunf. ranurada} = \pi \cdot \varnothing_{ran} = 141,37 \text{ mm} \quad (39)$$

$$\text{Distancia}_{flanco-flanco} = \frac{141,37 \text{ mm}}{120} = 1,178 \text{ mm} \quad (40)$$

Se realizan entonces mediciones de la anchura real de los flancos. Así se halla la tolerancia y el error cometido en la medida.

Rango de las medidas:

$$1 \text{ mm} - 1,50 \text{ mm} \rightarrow \text{Tolerancia: } (-0,18 \text{ mm} < \text{Tol.} < +0,32 \text{ mm})$$

Error:

$$E_r(\%)^- = \frac{0,18 \text{ mm}}{1,18 \text{ mm}} \cdot 100 = \mathbf{10\%} \quad (41)$$

$$E_r(\%)^+ = \frac{0,32 \text{ mm}}{1,18 \text{ mm}} \cdot 100 = \mathbf{27,3\%} \quad (42)$$

Se puede comprobar que el error que se comete en la medida con 120 flancos es inasumible. Y ahora aparece la siguiente pregunta: para obtener un error admisible, ¿de cuántos flancos se pueden disponer?

Error (%) asumible: 2%

$$Error = \frac{\text{máx}(Tolerancia)}{Distancia_{flanco-flanco}} \quad (43)$$

$$Distancia_{flanco-flanco} = \frac{0,32 \text{ mm}}{0,02} = 16 \text{ mm} \quad (44)$$

Número de flancos:

$$n^{\circ} flancos = \frac{\pi \cdot \varnothing_{ran}}{Distancia_{flanco-flanco}} = \frac{\pi \cdot 45 \text{ mm}}{16 \text{ mm}} = \mathbf{8,84 \text{ flancos}} \quad (45)$$

Como es imposible tener un número decimal de flancos se decide fabricar un disco con 8 flancos, ya que 9 supondría superar el error admisible. Así se obtiene un error final de:

$$n^{\circ} flancos=4 \rightarrow Distancia_{flanco-flanco}=17,67 \text{ mm} \rightarrow \mathbf{Error (\%)=1,81\%}$$

- **Resolución**

La resolución es la distancia mínima recorrida que el encoder puede leer, es decir, la distancia recorrida por la rueda “motriz” por cada paso por el sensor de cada flanco. Finalmente, con un disco ranurado de 4 flancos la resolución se calcula como sigue:

$$Perímetro_{motriz} = \pi \cdot \varnothing_{motriz} = \pi \cdot 80 \text{ mm} = 251,33 \text{ mm} \quad (46)$$

$$Resolución = \frac{Perímetro_{motriz}}{Factor_{aceleradora} \cdot n^{\circ} flancos} = \frac{251,33 \text{ mm}}{25 \cdot 4} = \mathbf{2,51 \text{ mm}} \quad (47)$$

5.2.3.1.2 Encoder v.2

El Encoder v.2 es el encoder comercial H38S600B (Figura 22). Este encoder se obtiene a través de Amazon [8], la cual proporciona las características del dispositivo, mostradas en la Tabla 2.



Figura 22. Encoder H38S600B

Specifications	
Power Supply	DC 5-24 V
Pulses per revolution	600 p/r (two phase – 1200 p/r)
Output	AB two phase output rectangular ortogonal pulse circuit. The output for the NPN open collector output type.
Speed	More tan 5000 rpm
Shaft	Diameter: 6 mm / Length: 13 mm
Size	Diameter: 39 mm / Length: 36,5 mm
Connection	
Red	A phase
Green	B phase
White	Vcc power +
Black	Gnd power -

Tabla 2. Especificaciones del encoder H38S600B

Dentro de la parte de diseño mecánico en este caso, y por comparación al Encoder v.1, se puede simplificar a dos apartados: rueda, y soporte y estructura móvil.

La rueda es muy similar a la de la primera versión, solamente con modificaciones en el agujero del eje (ya que se acopla directamente al eje del encoder) y en el perímetro, dado que se usa una junta tórica en vez de una goma elástica como en la primera versión.

Esta junta tórica proporciona una robustez y versatilidad que no se conseguía anteriormente.

El soporte utilizado para la estructura móvil es exactamente la misma al Encoder v.1, solamente diseñando una nueva pieza para acoplar el nuevo encoder a la estructura móvil, reutilizando también una pieza de la primera versión.

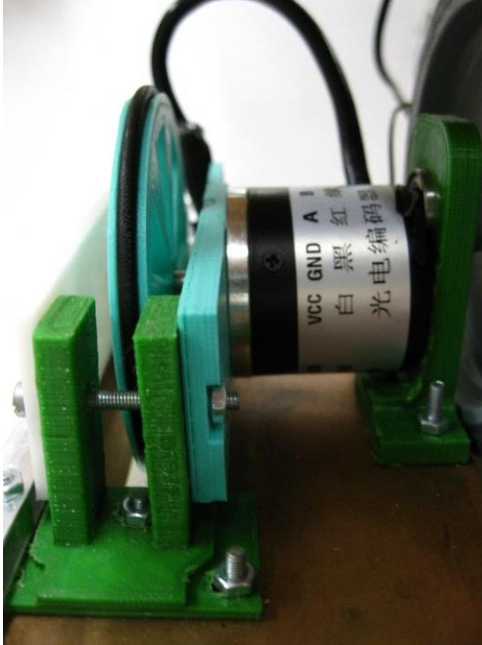


Figura 24. Encoder v.2. Vista 1

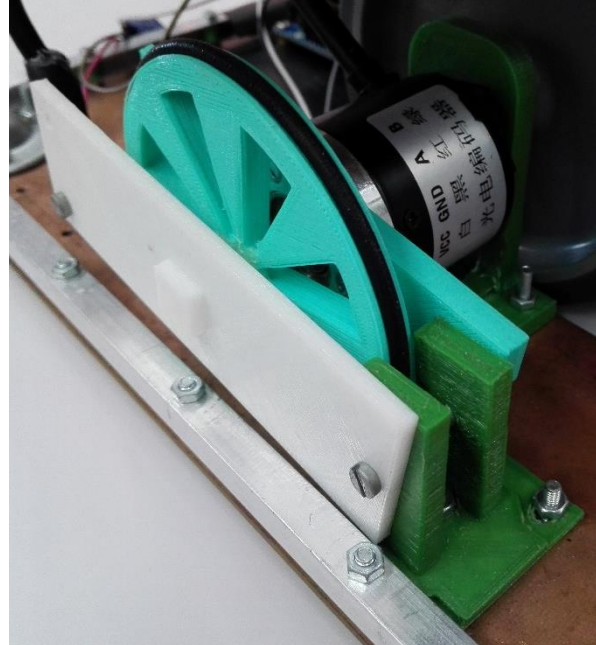


Figura 23. Encoder v.2. Vista 2

5.2.4 Ruedas motrices

Las ruedas motrices o de tracción utilizadas son las usadas en muchas carretillas industriales. Estas son adquiridas en la tienda almacén Brico Depot [9]. Al ser un tipo de ruedas comerciales e industriales aseguran tener una resistencia y unos materiales de construcción óptimos. También la goma de agarre asegura la condición de deslizamiento.



Figura 25. Rueda del robot

Estas ruedas están acopladas por un lado al motor mediante un sistema de piezas diseñadas e impresas en 3D (Figura 27y Figura 26). Estas piezas aprovechan el acople dentado del motor para conseguir una tracción perfecta.

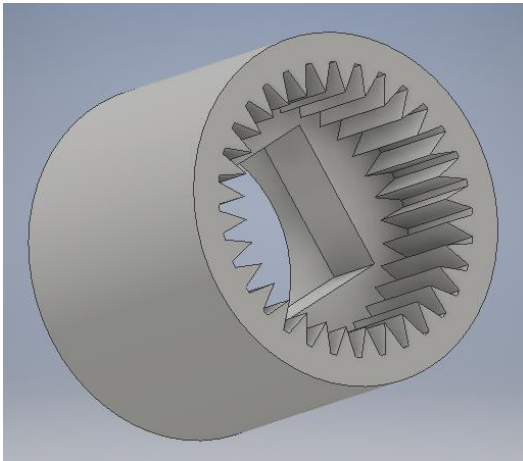


Figura 27. Acoplador rueda-motor. Vista 3D Inventor

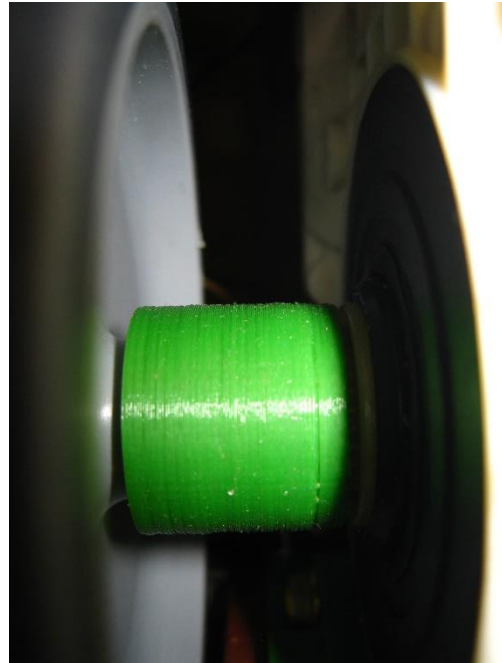


Figura 26. Acoplamiento entre rueda y motor

Para la sujeción del eje a esta pieza, se fabrica en dos piezas: una hueca con un extremo cuadrado (Figura 29) para el acople en la pieza y otra que actúa como relleno macizo de la anterior (Figura 28).

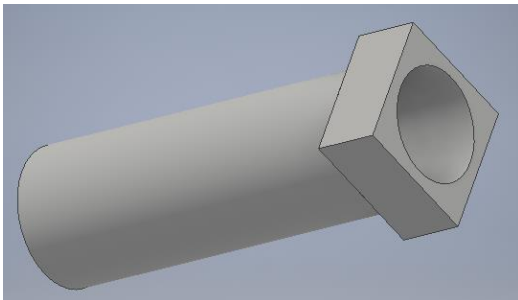


Figura 29. Eje hueco. Vista 3D Inventor

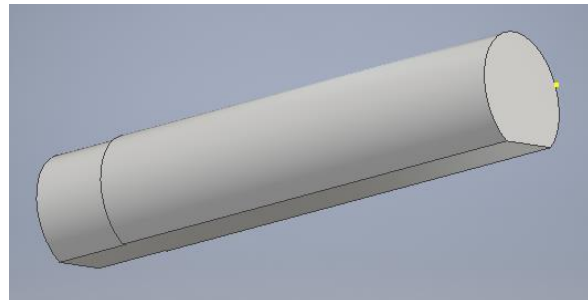


Figura 28. Eje macizo. Vista 3D Inventor

Por el lado opuesto, las ruedas se acoplan a unos rodamientos mediante un eje pasante. Estos rodamientos se encuentran alojados en un soporte que los levanta a la altura requerida (Figuraxx). Tanto los soportes como los ejes están diseñados por el autor de este proyecto e impresos en 3D.



Figura 31. Sujeción de rodamiento

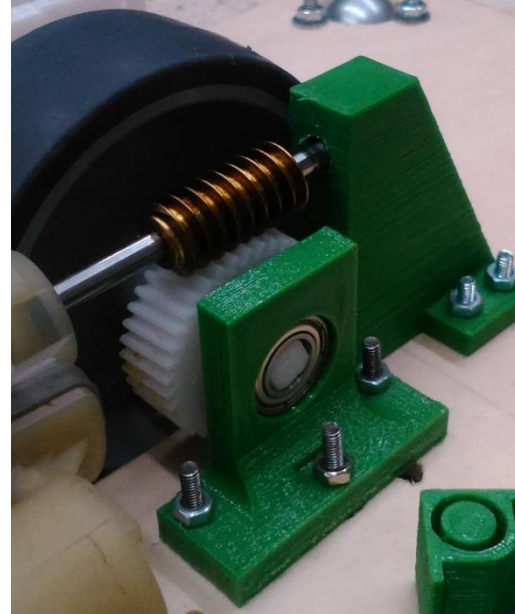


Figura 30. Sujeción de rodamientos.

5.2.5 Estructura

La estructura externa es de planta rectangular con 320 mm de ancho y 330 mm de largo. La normativa del “*Robotic Day*” de Praga [1] limita el perímetro máximo del robot en 1500 mm. Esta consideración es tomada en cuenta con el resto del equipo, pues influye también a los elementos móviles del robot que puedan salir y entrar de este. En lo que respecta a la estructura esta restricción se cumple, siendo el perímetro de 1300 mm.

La estructura principal está dividida en distintos pisos. Cada uno está formado por una plancha de fibra de vidrio de 1,65 mm de grosor con recubrimiento de cobre, en la cual se sustentan los componentes internos del robot. La plancha del primer piso que alberga al proyecto que aquí se expone, mide 320 mm de ancho y 330 mm de longitud. (Figura 32).

El resto de pisos están unidos a través de un sistema de columnas ancladas mediante piezas impresas en 3D y atornilladas a las planchas. Estas columnas son de 8 mm de diámetro y 95 mm de altura (Figura 33).

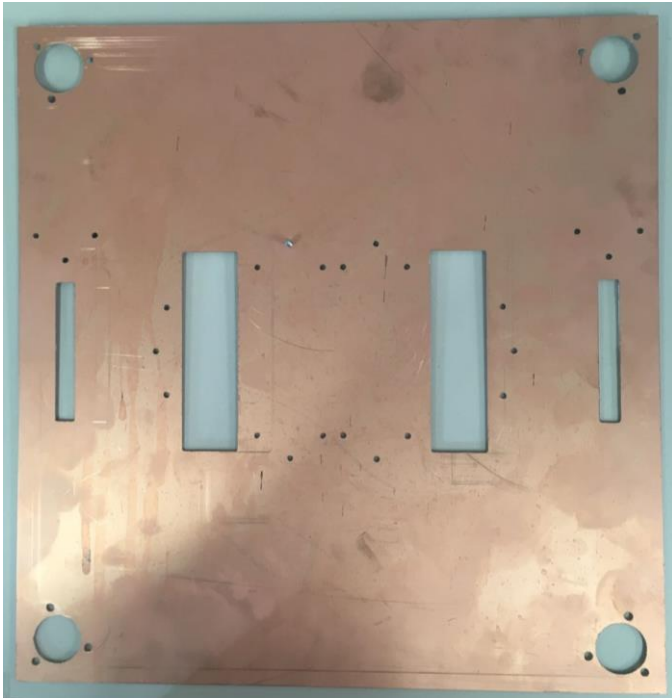


Figura 32. Base del robot

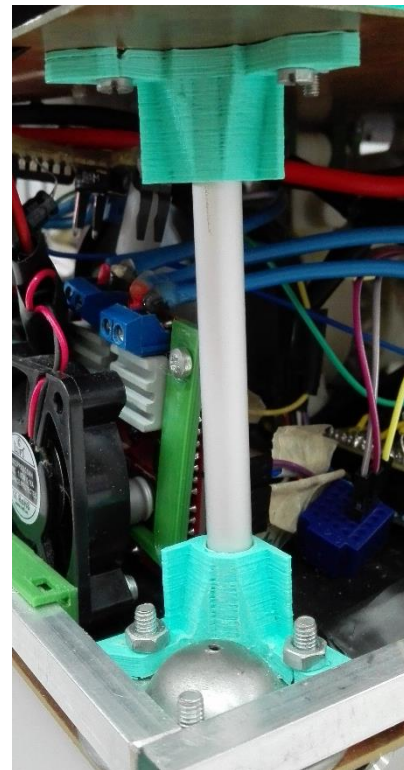


Figura 33. Columna de sujeción entre distintos niveles del robot

Los principales componentes anclados a esta plancha son los motores y los puntos de apoyo (roll-on), siendo estos últimos un apoyo en la distribución de los pesos y en la estabilidad del robot (Figura 35 y Figura 34). Los motores están sujetos a la plancha mediante unas piezas diseñadas a medida e impresas en 3D (Figura 36). Todos estos atornillados a la plancha.



Figura 35. Apoyo "roll-on". Vista superior



Figura 34. Apoyo "roll-on". Vista inferior

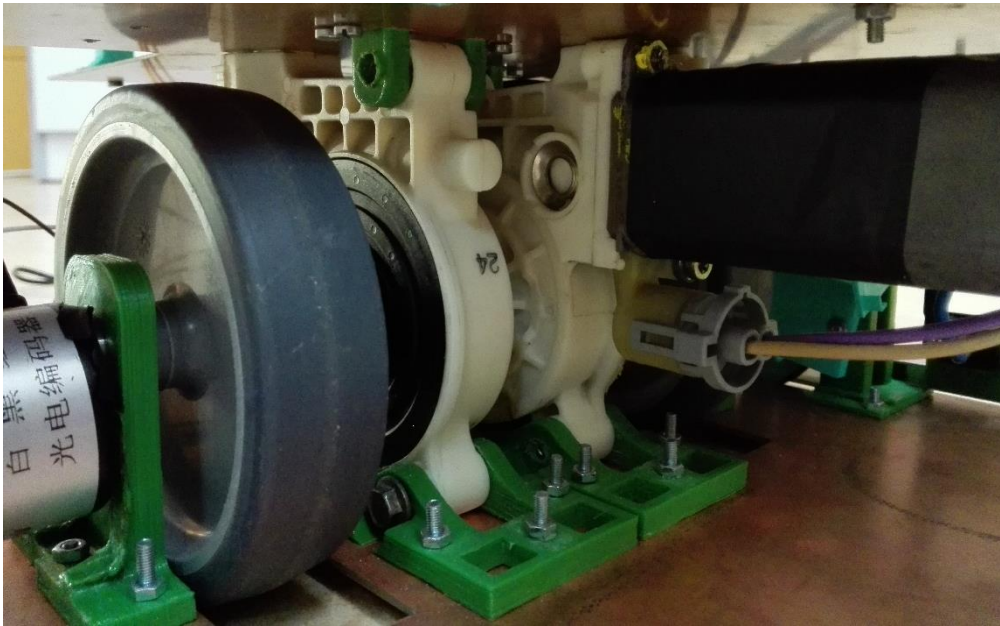


Figura 36. Sujeción de los motores a la base del robot

Para dar rigidez a todo el conjunto y que la plancha no pandee se colocan a lo largo de todo el perímetro de la plancha unos perfiles en "U" (8 mm x 8 mm x 1 mm), atornillados a la plancha en varios puntos y con elementos auxiliares para mantener la rigidez del mismo perfil (Figura 37). Estos elementos auxiliares se encuentran tanto debajo del perfil como dentro del propio perfil, dándole una rigidez extra.

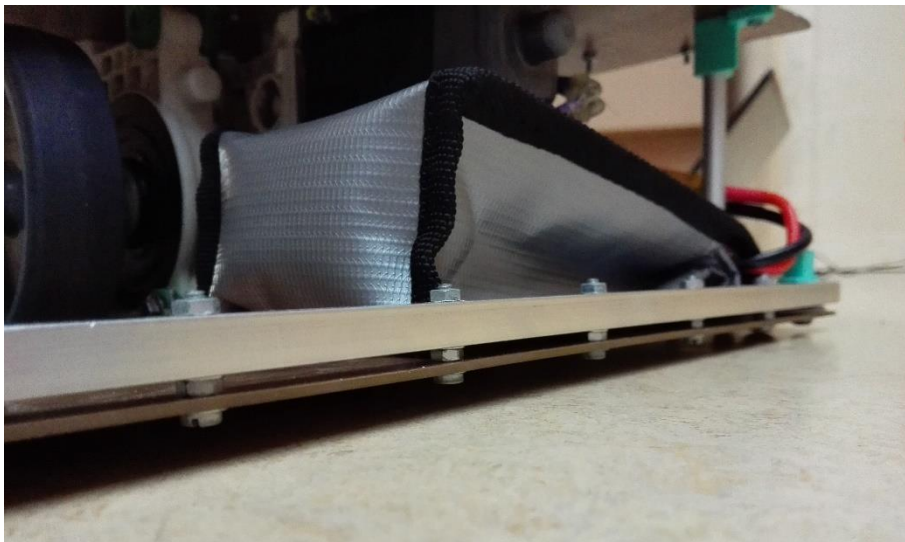


Figura 37. Sujeción del perfil en "U" a la base del robot

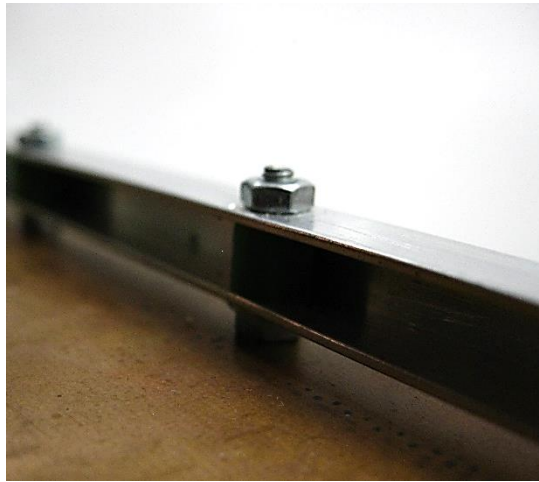


Figura 38. Detalle de sujeción del perfil en "U" a la base del robot

Todos los agujeros y aberturas de las planchas de los distintos pisos son realizados con una máquina CNC como fresadora (Figura 39), propiedad de la Universidad de Alcalá.

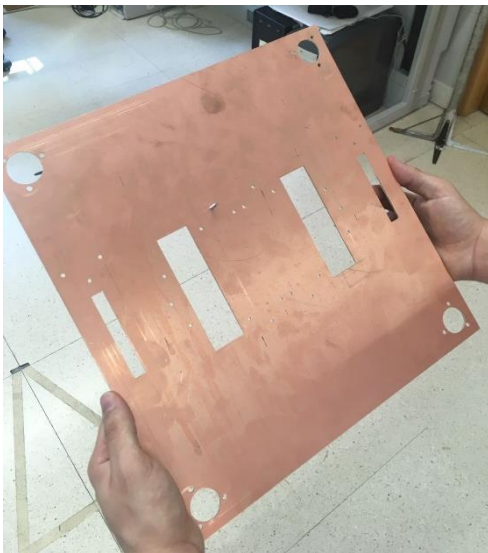


Figura 40. Base del robot sin elementos

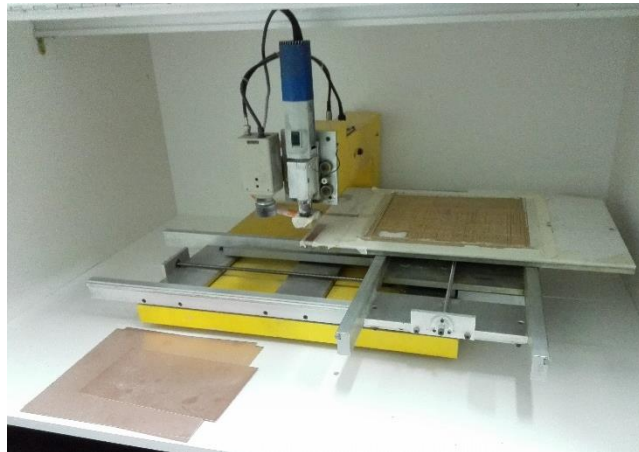


Figura 39. Máquina CNC

- Sensores de distancia

Los sensores de distancia GP2Y (Figura 41) están situados en el piso superior respecto al resto de componentes de este proyecto. Ancladas a la plancha del segundo piso se encuentran unas piezas auxiliares que sustentan un perfil en "U" (Figura 43). En este perfil se sujetan los sensores GP2Y mediante unas piezas impresas en 3D (Figura 42). Dichas piezas se diseñan para que la sujeción al perfil sea por presión, eliminando cualquier elemento fijo. Así se da la oportunidad de colocar los sensores a lo largo del perfil y poder variar su posición en función de la estrategia.

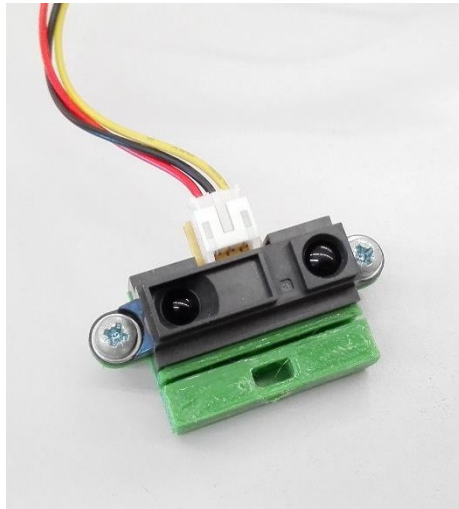


Figura 41. Sensor de distancia con pieza de sujeción

La altura efectiva de los sensores respecto al suelo es de 200 mm, salvando así la mayor altura de un elemento fijo del campo de juego [1] y por tanto no afectará a la estrategia dando falsos positivos.

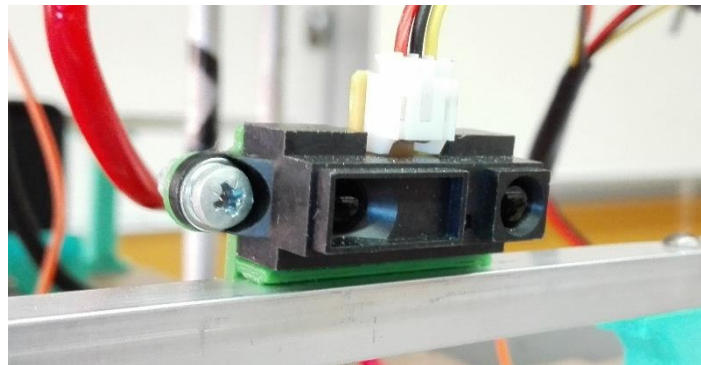


Figura 42. Sensor de distancia colocado en el perfil en "U"

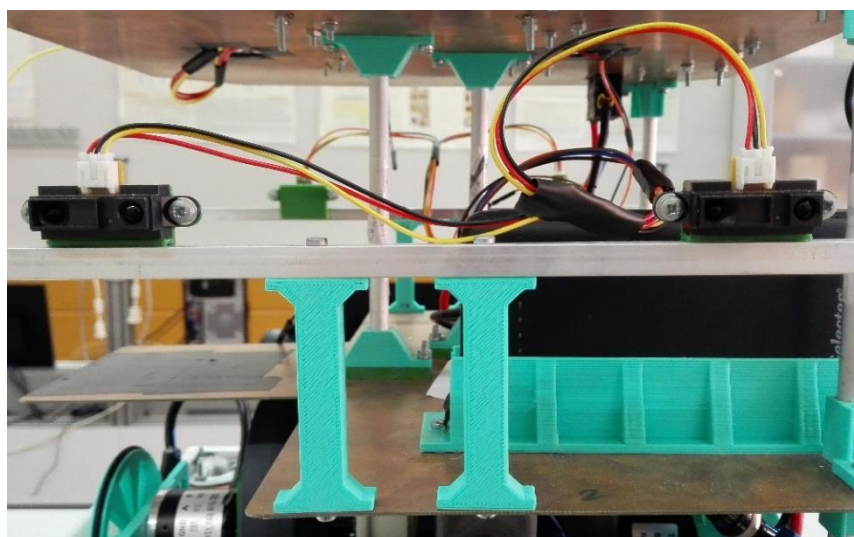


Figura 43. Sensores de distancia. Sujeción del perfil en "U"

- Ventilador y puente en H

Al igual que los sensores de distancia GP2Y, el ventilador y el puente en H están unidos a la estructura principal mediante el mismo sistema de la pieza por presión (Figura 44). Estos están amarrados al perfil en “U” del primer piso, dentro de la zona correspondiente a este proyecto. El ventilador se encuentra montado junto al puente en H mediante una pequeña pieza impresa en 3D.



Figura 44. Ventilador con tarjeta de potencia

- Interruptor y seta de seguridad

El interruptor está situado en el primer piso, próximo a la batería de potencia. Se elige esa localización para ahorrar espacio y dinero en el cableado, ya que debe estar entre la batería y el circuito de protección (eléctricamente hablando).

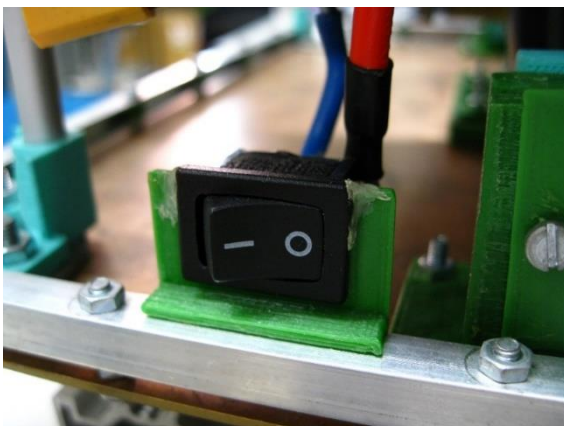


Figura 45. Interruptor de encendido de la electrónica de potencia

La seta de seguridad, por el contrario, debe encontrarse en lo alto del robot en una zona accesible. Se instala en la plancha del tercer piso como se muestra en las siguientes figuras:

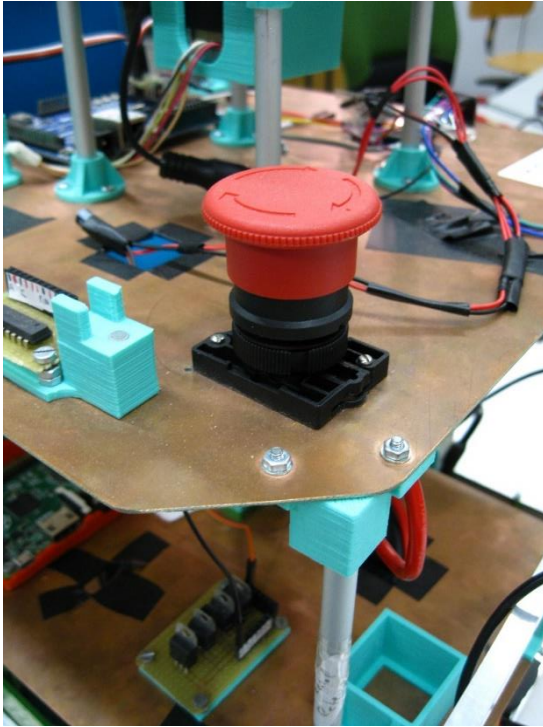


Figura 47. Seta de seguridad. Vista superior

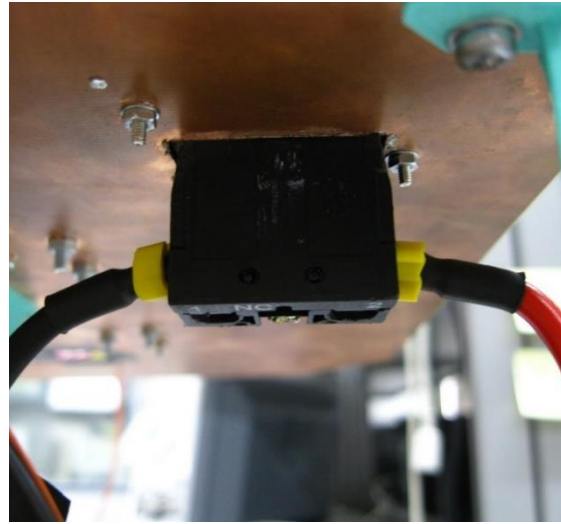


Figura 46. Seta de seguridad. Vista inferior

5.3 Diseño Electrónico

En este apartado se estudian los dispositivos electrónicos y circuitos desarrollados en este trabajo. La alimentación es un factor clave a la hora del desarrollo de la electrónica y condiciona su fabricación (e incluso su ubicación dentro del robot). En la siguiente Figura 48 se muestra un esquema de bloques para comprender mejor la relación entre los distintos componentes electrónicos del sistema. En rojo se muestra el flujo de la alimentación, en verde el flujo de datos y en negro una conexión clave entre las masas de las dos alimentaciones del sistema.

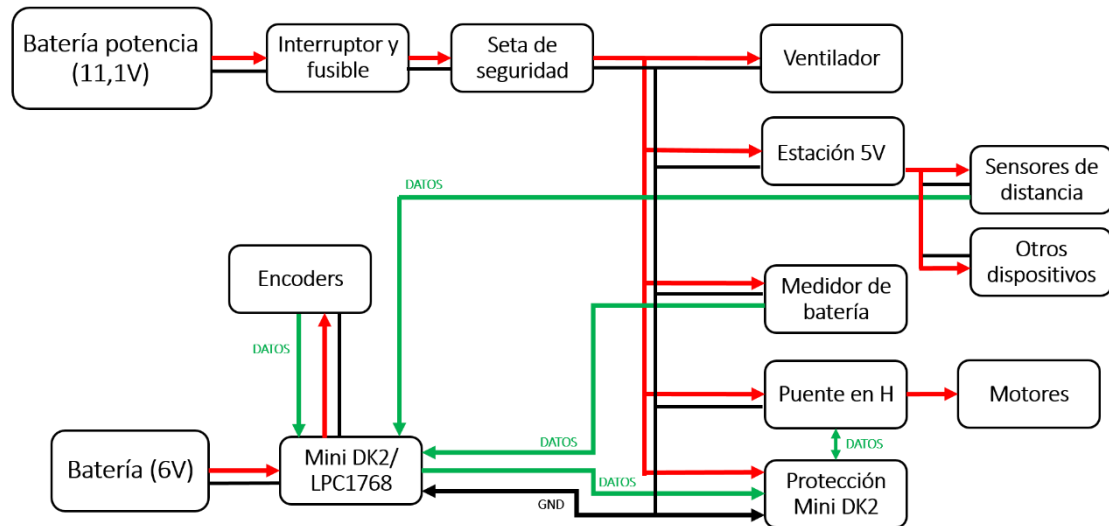


Figura 48. Esquema general de la electrónica

5.3.1 Mini DK2: LPC 1768

La Mini DK2 (Figura 49) es una tarjeta basada en el microcontrolador LPC1768, Cortex-M3 de ARM. Con ella se realiza toda la implementación software de este trabajo, abarcando desde todas las mediciones del hardware hasta el control de todo el sistema. La CPU trabaja a 100MHz y sus periféricos a 25MHz, suficiente velocidad para el correcto funcionamiento de este proyecto [10].

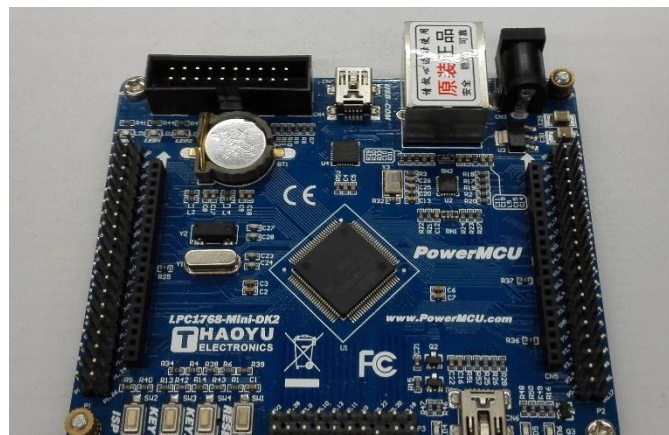


Figura 49. Mini-DK2 (LPC1768)

Unas de sus características relevantes para este proyecto son una interfaz USB configurable como UART o CAN, 6 salidas PWM de propósito general, un ADC de 12 bit y 8 canales, un DAC de 10 bit, 4 timers de propósito general y 70 pines I/O de propósito general del procesador (aunque efectivos son un número menor en la tarjeta Mini-DK2).

Algunos inconvenientes que se encuentran al trabajar con esta tarjeta son la baja protección de los pines I/O ante sobretensiones (razón por la que se desarrolla una tarjeta de protección) y el ruido en las mediciones del ADC. Este último no es condicionante puesto que no se necesita una resolución crítica en ninguna de las mediciones que se realizan.

5.3.2 Tarjetas de potencia: Puente en H y protección de Mini-DK2

Las tarjetas de potencia engloban la tarjeta Monster Moto Shield – Dual VNH2SP30 (Figura 50) y la tarjeta de protección creada para la Mini-DK2 (Figura 51).

- **Puente en H**

La tarjeta Monster Moto Shield – Dual VNH2SP30 se adquiere ya fabricada, puesto que en una estimación inicial se considera que construir una propia con las mismas prestaciones requiere más dinero y tiempo. Esta tarjeta está diseñada para ser usada sobre un Arduino [11] lo que nos asegura que los niveles lógicos para usarla son los adecuados ya que la Mini-DK2 trabaja con los mismos niveles lógicos [10].

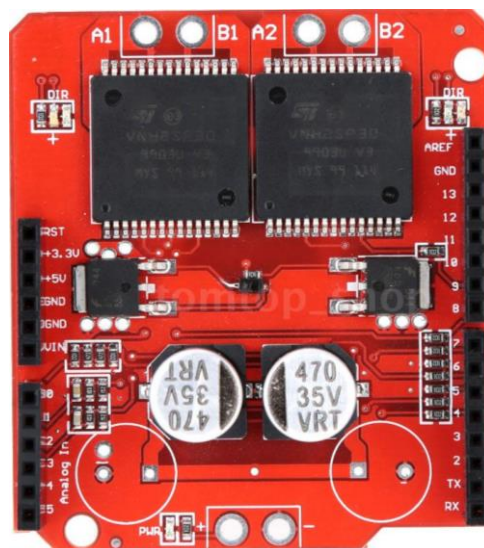


Figura 50. Monster Moto Shield – Dual VNH2SP30

La tarjeta se basa en los integrados VNH2SP30, los cuales son puentes en H de potencia. Lo que ofrece la tarjeta es el acondicionamiento de estos integrados para poder usarlos directamente con el microcontrolador, sin electrónica adicional. Su esquemático se muestra en el Anexo 11.1 y características son las mostradas en la Tabla 3.

Características	Valor
Power supply range	5,5V – 16V
Maximum current rating (per each motor)	30A
Practical continuous current (per each motor)	14A
Current sense output proportional to motor current	-
MOSFET on-resistance	19mΩ
Maximum PWM frequency	20 kHz
Thermal shutdown	-
Undervoltage and Overvoltage shutdown	-

Tabla 3. Características de la Monster Moto Shield – Dual VNH2SP30

Nº Pin	Description
Gnd	Ground
A0	Enable for motor 1
A1	Enable for motor 2
A2	Current sensor for motor 1
A3	Current sensor for motor 2
D7	Clockwise for motor 1 (A1)
D8	Counterclockwise for motor 1 (B1)
D4	Clockwise for motor 2 (A2)
D9	Counterclockwise for motor 2 (B2)
D5	PWM for motor 1
D6	PWM for motor 2

Tabla 4. Conexión de la Monster Moto Shield – Dual VNH2SP30

En primer lugar se observa que el rango del voltaje de alimentación es el apropiado, ya que la batería que se escoge es de 11,1V dado que los motores a usar son de 12V.

En lo concerniente a la corriente se observa que la máxima continua es superior a la nominal de cada motor (7A), y la máxima alcanzable (30A ya que se colocan disipadores para poder alcanzarla) es superior a la máxima de cada motor (27A). Además se dota a cada integrado con un disipador para ayudar en la obtención de corrientes altas.

En definitiva, la potencia que puede proporcionar el puente en H es superior a la potencia de los motores (concretamente el doble):

$$P_{nominal\ puente_H} = 2 \cdot (11,1V \cdot 14A) = \mathbf{310,8W} \quad (48)$$

$$P_{nominal\ motores} = 2 \cdot (11,1V \cdot 7A) = \mathbf{155,4W} \quad (49)$$

Los integrados VNH2SP30 disponen de autoprotección ante sobre temperaturas, normalmente causadas por corrientes elevadas, quedando así protegida la electrónica de potencia.

Aun con todas estas prestaciones, experimentalmente se detectan una serie de problemas:

- Problema con lectura de corriente. La lectura es inexistente indicando siempre que la corriente es nula. Se comprueba que el circuito de acondicionamiento de la tarjeta es correcto y se determina que el error proviene del integrado.
- Pines de Enable demandan corriente. Se observa que los pines de la Mini-DK2 configurados como GPIO de salida digital son incapaces de suministrar la

corriente mínima de activación para el Enable. Se decide utilizar la salida del DAC ya que suministra más corriente.

- Sobretensiones en el encendido. Cada vez que se enciende todo el sistema de potencia se producen sobretensiones en los pines de entrada a la tarjeta de control de los motores. Los pines de la Mini-DK2 al estar configurados como salidas sus resistencia son bajas encontrándose desprotegidos. Este hecho causa la destrucción de los puertos GPIO conectados.

- **Protección de Mini-DK2**

Los problemas vistos de la tarjeta Monster Moto Shield desembocan en la decisión de crear una tarjeta de protección para el microcontrolador (Figura 51).

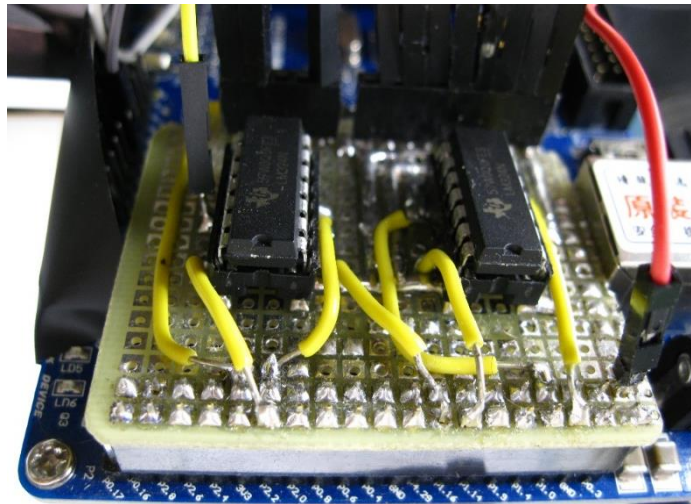


Figura 51. Tarjeta de protección de Mini-DK2

Esta tarjeta consiste en una serie de seguidores que harán de puente protector entre los puertos de la Mini-DK2 y la tarjeta Monster Motor Shield. Los seguidores proporcionan:

- Protección a sobretensiones.
- Corriente de salida superior (útil para el Enable).
- No variación del voltaje de la señal, es decir, no modificación de la información.

Los integrados utilizados para esta tarjeta son los LM324-N. Estos integrados cuentan con 4 amplificadores operacionales en su interior. Las características principales que se buscan se exponen en la Tabla 5.

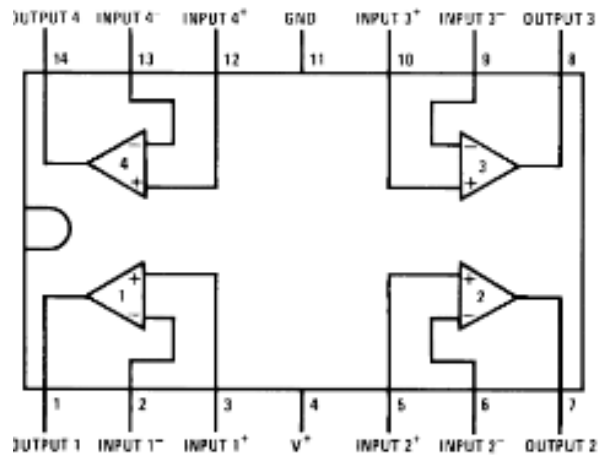


Figura 52. Esquema interno del integrado LM324-N

Parameter	Value
Supply Voltage, V^+	0V – 32V
Output Current ($V^+=15V$)	20mA
Slew Rate (Follower Pulse Response – $V=3V$)	0,25V/ μ s

Tabla 5. Parámetros del integrado LM324-N

La alimentación de los integrados se puede extraer de la batería (11,1V). Por otra parte la corriente de salida es suficiente para activar el Enable del puente en H. Concluyendo el $SR= 0,25V/\mu s$ nos asegura que la señal PWM no se ve afectada a efectos prácticos.

Para la transmisión de las señales las masas de la alimentación de la Mini DK2 y de la batería se unen en esta tarjeta.

Finalmente se fabrica la tarjeta de protección siendo su esquemático el mostrado en la Figura 53.

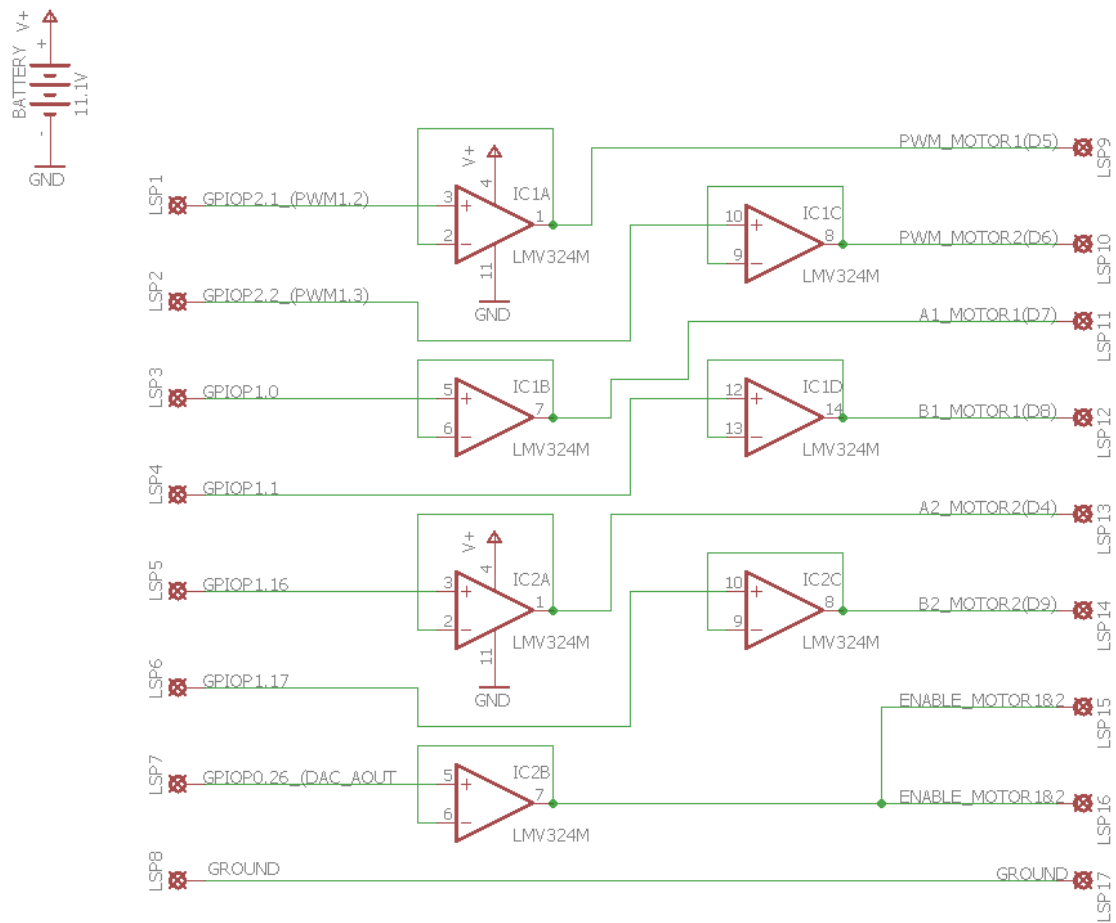


Figura 53. Esquemático tarjeta de protección de Mini-DK2

5.3.3 Lectura y acondicionamiento de los pulsos de los encoders

La lectura del Encoder v.1 se realiza mediante un sensor de luz infrarroja. Este sensor consiste en un emisor LED y un fototransistor doble como receptor. Cuando el disco ranurado del encoder atraviesa el haz de luz el transistor genera una señal que habrá que acondicionar para conseguir la señal digital adecuada. Ambos dispositivos (emisor y fototransistor) necesitan ser polarizados para su correcto funcionamiento.

La señal que genera el fototransistor doble es digital, ya que la información que contiene es digital (dos niveles), pero no puede ser leída directamente por el microprocesador. Es por ello por lo que hay que realizar una electrónica de acondicionamiento y ajustar así ciertos parámetros de esta señal.

- **Lectura**

El sensor usado es el de un ratón antiguo de ordenador. Estos utilizaban una bola para mover dos pequeños encoders, uno por cada eje cartesiano. El sensor está formado por un emisor LED de luz infrarroja y un fototransistor como receptor. En el Encoder v.1 es usado solo el sensor, no la electrónica de polarización, por lo que se crea en este proyecto.

A la hora de diseñar esta electrónica se tiene una consideración general: la corriente consumida, ya sea para no destruir el LED o para ajustar los niveles de tensión proporcionados por el fototransistor.

- Polarización del emisor LED

La tensión de conducción directa de cualquier LED es imprescindible para su correcta polarización, ya que con ella conseguimos limitar la corriente que lo atraviesa y así evitar su destrucción.

Este LED infrarrojo no dispone de un número de identificación, por lo que su tensión de conducción directa se halla de forma experimental con la ayuda de un osciloscopio y una fuente de alimentación. Además, observando características de LEDs similares, se halla la corriente máxima típica de estos dispositivos.

Parameter	Value
Forward Voltage (V_f)	1,1 V
Maximum Continuous Forward Current ($I_{f \text{ máx}}$)	50 mA

Tabla 6. Parámetros del LED

Tras conocer estos parámetros se determina que con una $I_f=5\text{mA}$ y una distancia de emisor-receptor de 6mm el fototransistor detecta correctamente la luz emitida por el LED. Utilizando un voltaje de alimentación de 5V se obtiene que la resistencia de polarización del LED debe ser:

$$R_{LED} = \frac{V_{CC} - V_f}{I_f} = \frac{5V - 1,1V}{5 \text{ mA}} = 780 \Omega \rightarrow 1k\Omega \quad (50)$$

- Polarización del fototransistor

El fototransistor utilizado es un Dual Fototransistor PT5529B/L2-F. Este consta de dos fototransistores en su interior (Figura 54), idénticos entre ellos. Sus parámetros característicos son los siguientes:

Parameter	Value
Collector-Emitter Saturation Voltage ($V_{CE(sat)}$)	0,4 V
Maximum Collector Current ($I_{C \text{ máx}}$)	20 mA

Tabla 7. Parámetros del fototransistor PT5529B/L2-F



Figura 54. Esquema interno del fototransistor PT5529B/L2-F

Ambos fototransistores actúan como generadores de corriente en función de la luz infrarroja recogida por su base. Experimentalmente se comprueba que a una distancia LED/fototransistor de 6mm, con la polarización anterior del LED, la corriente de colector

de cada fototransistor es de 0,17mA (muy por debajo de su corriente máxima). Se asegura así su correcto funcionamiento.

Para recoger la información del estado del fototransistor se le coloca a cada uno en serie una resistencia de 10k para conseguir unos niveles altos (“1”) de unos 1,7V. Más adelante se consigue reducir la distancia al LED consiguiendo casi el doble de corriente (0,3mA) y por lo tanto el doble de voltaje a la salida.

Finalmente el esquemático completo del circuito de lectura es el que sigue:

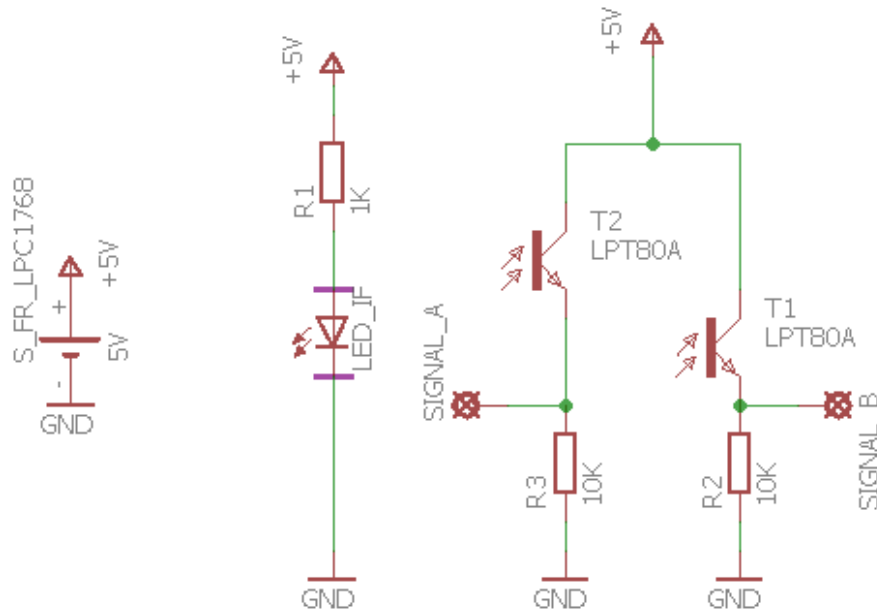


Figura 55. Esquemático circuito de lectura de Encoder v.1

- **Acondicionamiento**

Una razón por la que hace falta de un acondicionamiento es que estas señales tienen tiempos de subida y de bajada entre el nivel alto y bajo demasiado grandes (Figura 56). Otra razón es que la diferencia entre los niveles alto y bajo es pequeña, y en cuanto a voltaje se refiere, los dos niveles se encuentran en el mismo rango que el microcontrolador entiende por un “0” lógico (Figura 56).

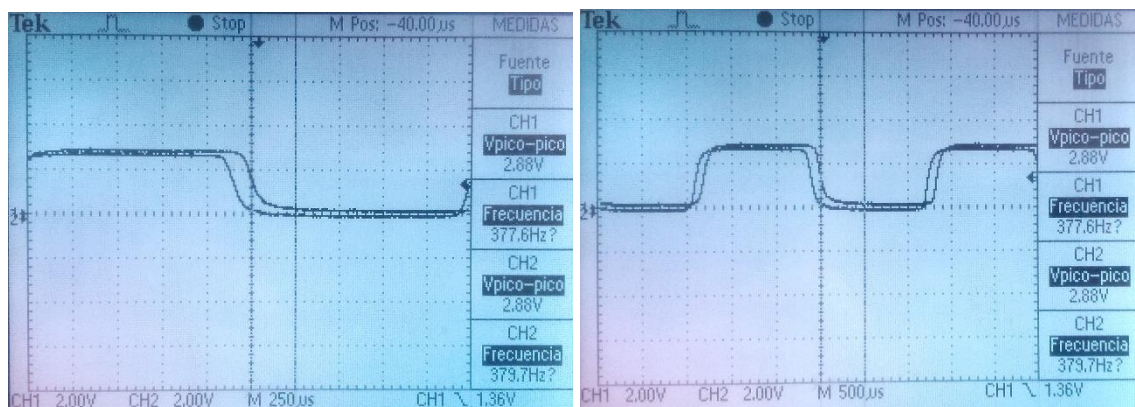


Figura 56. Señal en cuadratura Encoder v.1. Tiempos de subida/bajada de 250 µs

Todo esto hace que el procesador tenga problemas para recoger la información de los encoders. Es por ello que se debe adaptar la señal, para que la entienda como plenamente digital.

Los dos aspectos principales en el acondicionamiento son:

- Flancos rápidos y adaptación de los niveles al fondo de escala.

El procesador necesita recibir flancos (variaciones rápidas) para poder detectar la información. Para este cometido se diseña un circuito electrónico que actuará como comparador. Se trata de un amplificador operacional por cada señal, conectándose cada una directamente al terminal (+). Así se consigue que a su salida la variación sea la alimentación (fondo de escala).

Se utiliza el integrado LM324 anteriormente visto en la tarjeta de protección de la Mini DK2. Su SR (*Slew Rate*) es óptimo para este propósito también ($SR = 0,25V/\mu s$).

- Ajuste del punto medio.

El comparador necesita determinar el punto medio en el que cambia su salida entre los niveles alto y bajo. Esto se consigue con un divisor resistivo variable mediante un potenciómetro cuya salida se conecta al terminal (-) del amplificador operacional. Así se ajusta al valor medio de la señal de entrada (Figura 57). El valor del potenciómetro será de $100K\Omega$, para reducir el consumo de corriente.

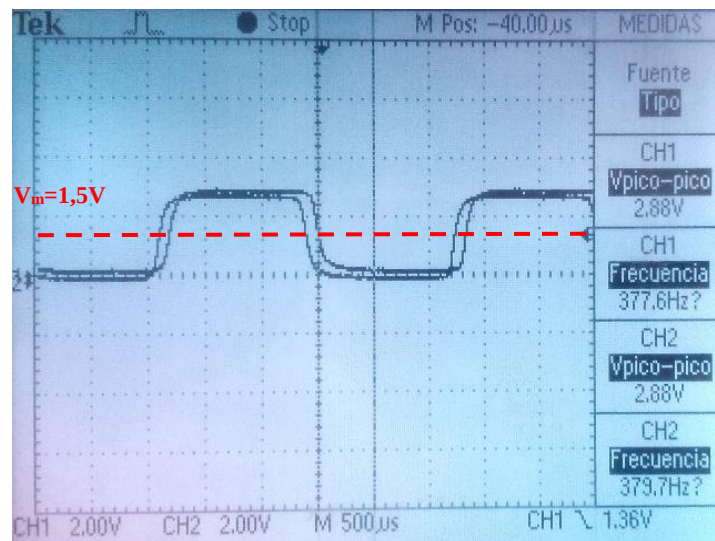


Figura 57. Señal sin acondicionar del Encoder v.1. Valor medio de la señal

Se encuentra interesante este ajuste también a la hora de elegir el material del disco ranurado del encoder. Se realizaron pruebas con discos impresos en 3D muy finos y de plástico blanco. A distancias LED - Receptor cortas se encontraba que el nivel bajo podía tomar valores que el procesador entendería como niveles altos. Esto era debido a que el disco era translúcido. Gracias al ajuste discos finos y ligeros pueden ser utilizados.

Finalmente se obtiene la señal acondicionada:

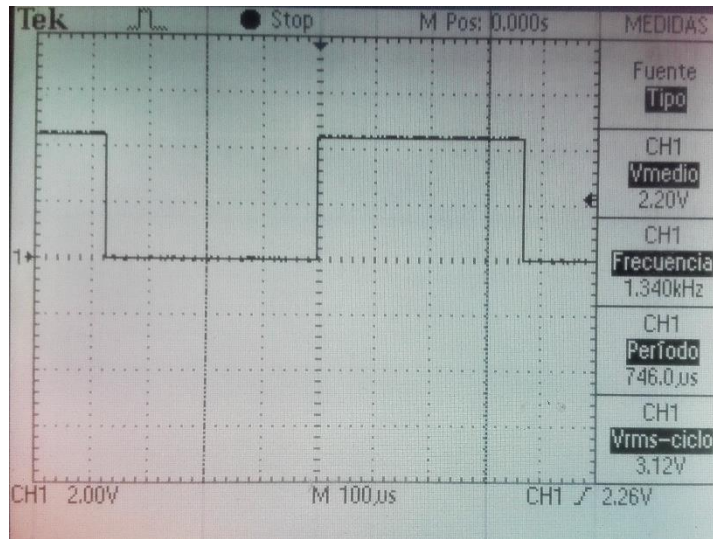


Figura 58. Señal acondicionada del Encoder v.1

El circuito de acondicionamiento por cada señal del encoder es el que sigue:

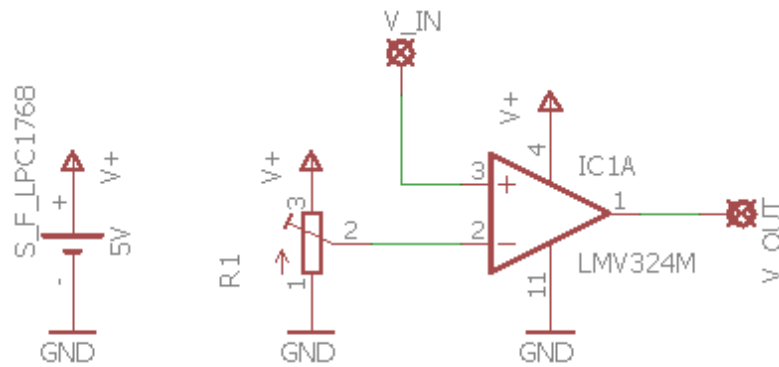


Figura 59. Esquemático acondicionamiento de las señales del Encoder v.1

5.3.4 Circuito de alimentación

5.3.4.1 Batería

La batería utilizada es una Li-Po de 3 celdas en serie y de tensión nominal 11,1V (Figura 60). Sus características se muestran en la siguiente tabla:



Figura 60. Batería Li-Po Zippy 3S 2700mAh

Características	Valor
Capacidad	2700 mA/h
Voltaje	3 S1P / 3 Cell / 11,1 V
Descarga	40C Constant / 50C Burst
Dimensiones	235 g (incluyendo cable y enchufe)
Dimensiones	137x45x20 mm
Enchufe de balanceo	JST-XH
Enchufe de descarga	XT60

Tabla 8. Características batería Li-Po Zippy 3S 2700mAh

Se elige esta batería entonces por diversas razones:

- Su tensión nominal (11,1V). Se encuentra cerca de los 12V nominales de los motores, por lo que su funcionamiento será el correcto.
- Su reducido peso (235g). Este es debido a que es utilizada comúnmente en aeromodelismo. Para el robot este es un factor crucial ya que sus prestaciones aumentan cuanto menor es su peso.
- Su capacidad (2700mA/h). Esta es suficiente para varios encuentros de la competición RoadAssistance. La corriente nominal de cada motor es de 7A (14A los dos motores), por lo que con una capacidad de 2700mA/h podremos disponer de un funcionamiento continuo aproximado de:

$$\text{Tiempo funcionamiento} = \frac{2,7A \cdot 60\text{min}}{14A} = \mathbf{11,57 \text{ min}} \quad (51)$$

Como complementos a la batería se adquieren un cargador y una bolsa ignífuga. El cargador es un iMax-B6 (Figura 61) y es especial para este tipo de baterías, pudiendo cargar la batería a una corriente controlada. La bolsa ignífuga servirá como medida de

seguridad en caso de que la batería arda (Figura 62). Es requerimiento también en la competición “RoadAssistance” [1].

También se adquiere una batería extra de repuesto para intercambiar en los encuentros.



Figura 61. Cargador iMax-B6

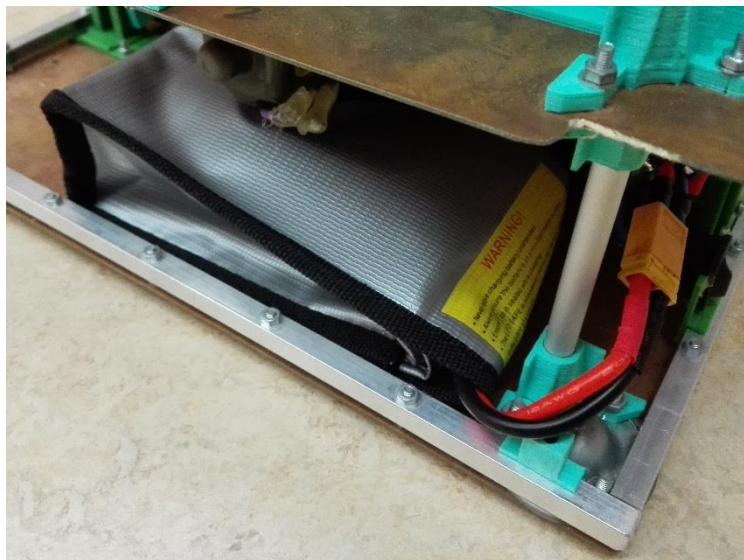


Figura 62. Batería en su posición dentro del proyecto

5.3.4.2 Circuito de encendido y seguridad

El circuito de encendido y seguridad se encuentra junto a la batería. Lo conforman un interruptor, un fusible y una seta de seguridad. Interesa que estos componentes estén físicamente y eléctricamente cerca de la batería, ya que gobiernan la alimentación de potencia del robot y su seguridad.

La seta de seguridad es la excepción en cuanto a cercanía ya que interesa su posición accesible en la parte alta del robot. Es por ello por lo que se requiere de un cableado extra para realizar el conexionado.

El fusible se encuentra situado en un portafusible cerca del interruptor. Este portafusibles facilita la tarea de reemplazo del fusible en caso de su rotura. La corriente máxima que este soporta es de 16A, menor que la corriente máxima demandada por los motores (en situación nominal 14A). En pruebas realizadas se comprueba que tampoco en los arranques del robot se consume más de la corriente nominal. Esto es debido a los arranques suaves que realiza.

El esquemático del circuito completo es el que sigue:

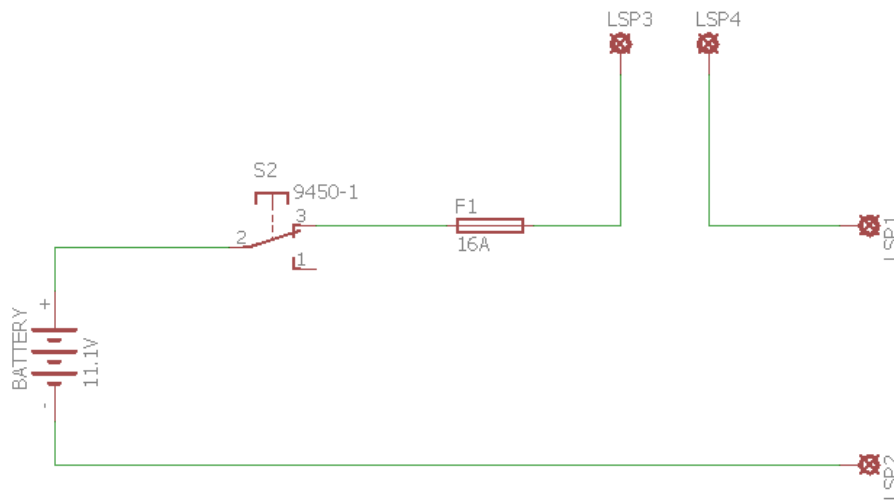


Figura 63. Esquemático del circuito de alimentación. LSP3 y LSP4 representan los terminales de la seta de seguridad

5.3.4.3 Estación de 5V

Esta estación de 5V nace de la necesidad de suministrar alimentación a diversos dispositivos del resto del robot y de este trabajo. La estación cuenta con 4 reguladores de tensión LM7805 cuyas salidas se duplican (Figura 64).

Se comienza con la necesidad de separar la alimentación de una electrónica de control y unos servomotores. Se decide suministrar la alimentación de los servos desde la batería, regulando el voltaje a 5V.

Más adelante, en la fase de pruebas de los sensores de distancia se detectan unas series de problemas al alimentarlos desde la Mini DK2. Esta tarjeta comienza a producir ruido mientras alimenta a los sensores. Se halla finalmente que esto es debido al funcionamiento de los sensores, los cuales emiten pulsos de luz infrarroja de “alta” intensidad en comparación a la consumida por la Mini-DK2. Así, trabajando en un régimen no continuo, evitan un consumo excesivo de corriente.

Esta demanda de picos de corriente a la Mini DK2 provoca un ruido no deseable, ya que puede acarrear problemas futuros. Es por ello por lo que los sensores de distancia se pasan a alimentar con la estación de 5V.

Finalmente, vista la gran practicidad de la estación, otros dispositivos del robot son conectados a ella.

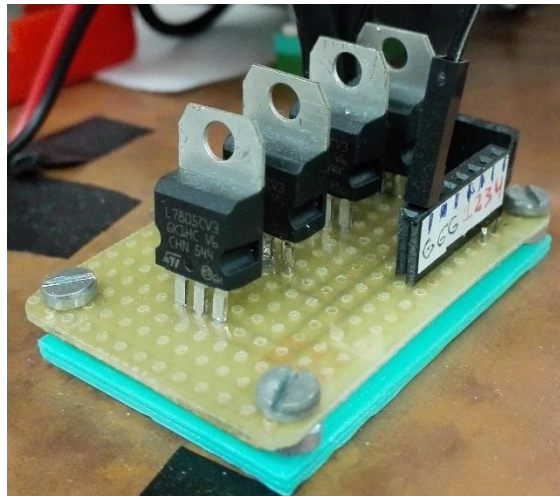


Figura 64. Estación de 5V

En lo que respecta al circuito el fabricante recomienda colocar para un correcto funcionamiento un condensador de 0,33 µF por cada regulador si este no se encuentra pegado a la fuente de tensión (en este caso la batería). Es por ello que se coloca un condensador de 1,32 µF a la entrada de la alimentación de la estación (4 condensadores por cada regulador en paralelo).

El esquemático del circuito de la estación es el que sigue:

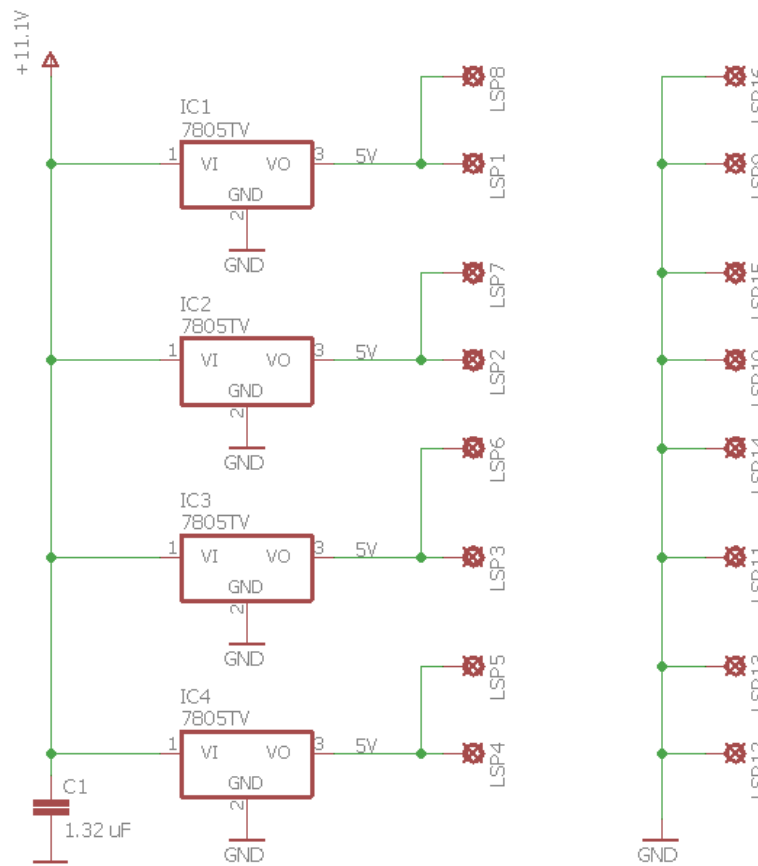


Figura 65. Esquemático de la estación de 5V

5.3.5 Medidor del nivel de la batería

El medidor del voltaje de la batería tiene dos usos principales: la adaptación del controlador software y la protección de la batería. Este medidor toma el valor de voltaje de la batería y mediante un seguidor se envía al ADC del microcontrolador.

Pero existe un inconveniente, hay que ajustar el fondo de escala del voltaje de la batería al fondo de escala del ADC. Esto se consigue mediante un divisor resistivo, atenuando el voltaje de la batería por un factor de división variable. Se usa para este propósito un potenciómetro de 100K Ω , reduciendo así la corriente consumida.

Se añade un condensador electrolítico próximo al voltaje recogido por el potenciómetro para evitar ruidos en la medida. Al no esperar variaciones bruscas no afectará negativamente en la medida.

Para el seguidor se utiliza el amplificador operacional LM324 visto anteriormente. El esquemático del circuito es el que sigue:

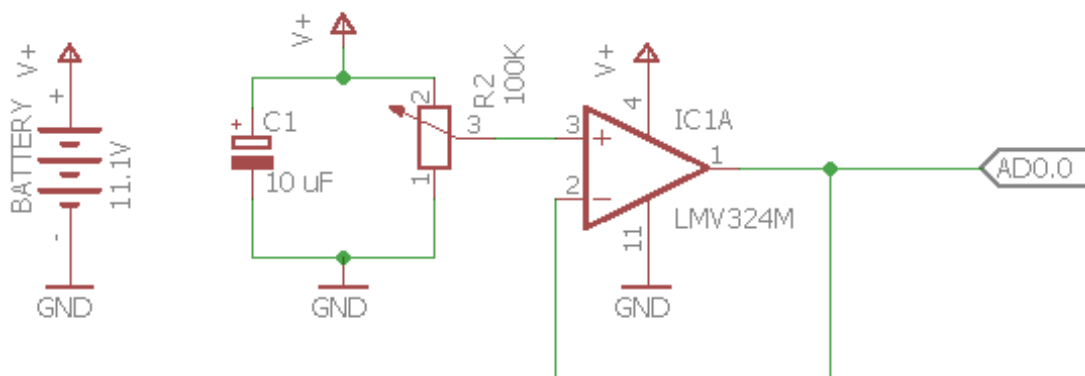


Figura 66. Esquemático del medidor de nivel de la batería

5.3.6 Sensores de distancia

Los sensores de distancia utilizados son los SHARP GP2Y0A21YK0F (Figura 67). Estos dispositivos constan de una electrónica de acondicionamiento interna por lo que no será necesario fabricarla. Su alimentación es de 5V, suministrada por la estación de 5V anteriormente comentada.

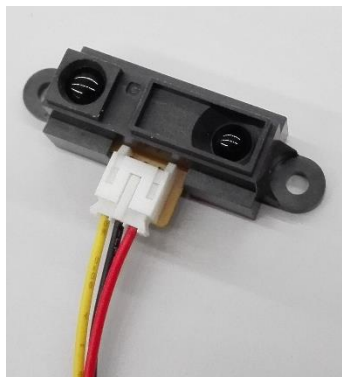


Figura 67. Sensor de distancia SHARP GP2Y0A21YK0F

La información de la distancia la facilita de manera analógica mediante la variación de una salida en tensión. La manera en la que obtiene la distancia a un objeto se realiza mediante triangulación. Emite pulsos de luz infrarroja de “alta” intensidad con un cierto ángulo respecto a la normal de medición. En función del ángulo de la luz reflejada en el objeto cercano, y recogido por un sensor PSD, se calcula la distancia. El diagrama de bloques de su electrónica interna es el que sigue:

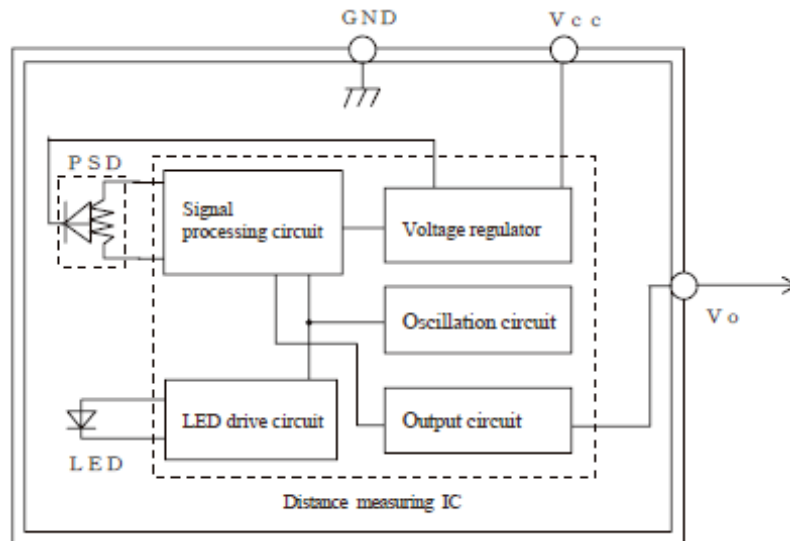


Figura 68. Esquema de componentes internos de los sensores SHARP GP2Y0A21YK0F

El comportamiento de la señal de salida se analiza más adelante en su correspondiente apartado dentro del *Diseño Software*.

5.3.7 Esquema general de conexiones de Mini-DK2

En la Figura 69 se muestra las conexiones de datos entre los distintos elementos del proyecto.

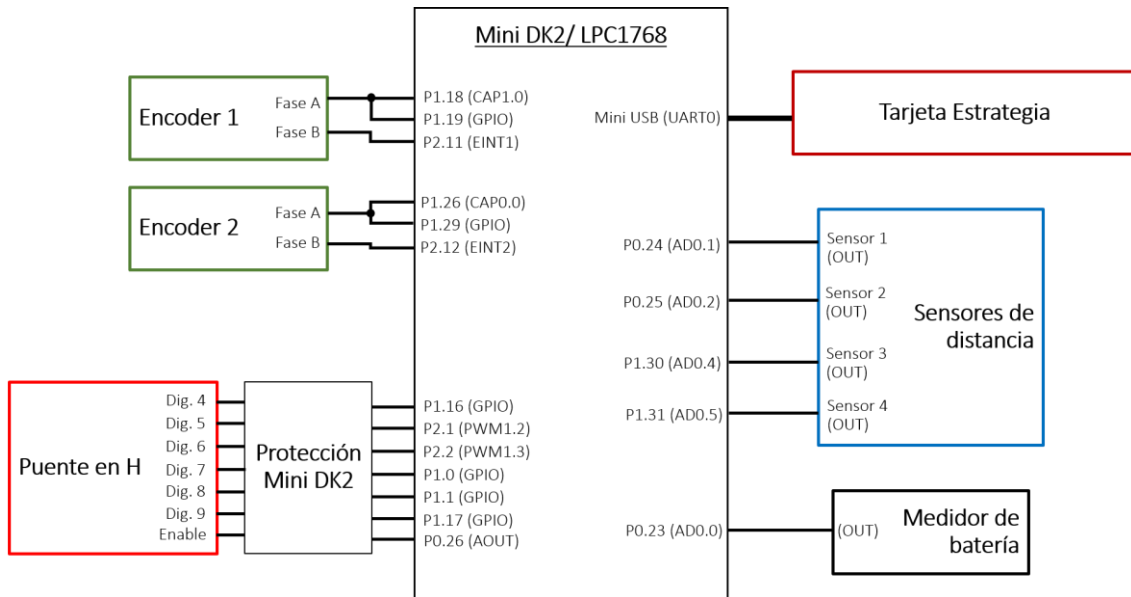


Figura 69. Esquema del conexionado de los elementos electrónicos a la Mini-DK2

5.4 Diseño Software

5.4.1 Software de Control Central

El software de control central será el encargado de comandar y organizar todos los procesos del controlador y sus complementos. Así se conseguirá la mayor eficiencia y rapidez posible en el control del movimiento sin que distintos procesos internos entren en un conflicto crítico para el sistema. Para este propósito se utiliza el microcontrolador LPC_1768 (Cortex-M3).

En el siguiente esquema se muestran los distintos procesos y la relación que mantienen entre ellos.

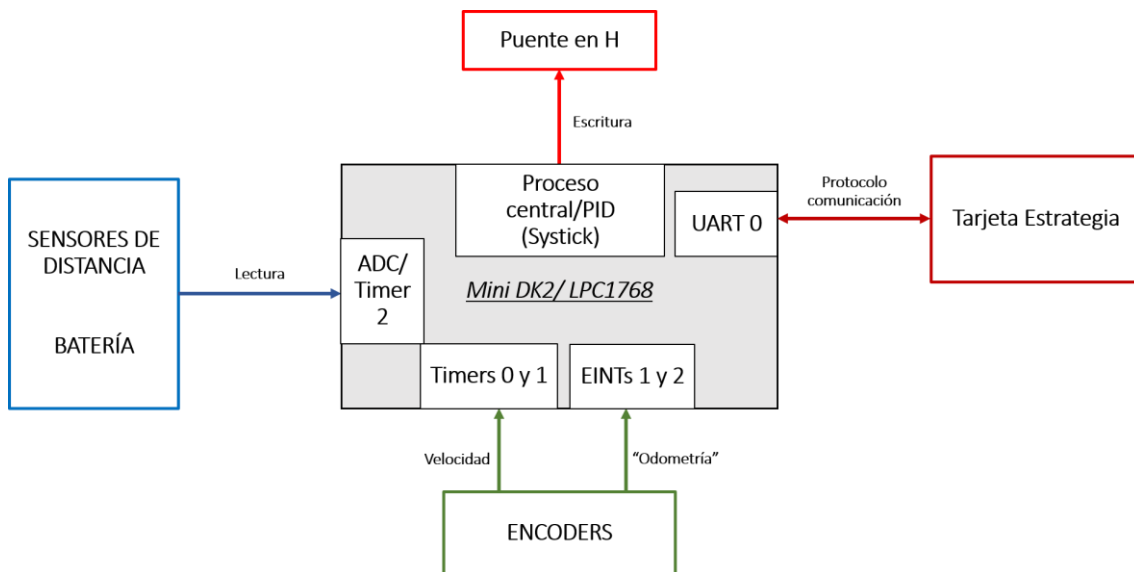


Figura 70. Esquema general de la relación entre las partes del software

5.4.1.1 Comportamiento Sensores de distancia

Los sensores de distancia utilizados en este proyecto, como se veía anteriormente, son los GP2Y0A21YK0F. En este apartado se analiza el comportamiento de la información que aporta y su consecuente tratamiento a nivel software, ya que informa de los posibles obstáculos en el entorno próximo al robot.

La información aportada por estos sensores es de carácter analógico, por lo que se utiliza el módulo ADC de la LPC_1768 para leer y convertir dicha información. Serán necesarios 4 canales, ya que es el número usado de estos sensores. Anteriormente se analizaba el valor máximo del voltaje proporcionado por estos, ya que se debía realizar una adecuada electrónica de acondicionamiento de acuerdo con el voltaje máximo admitido por el ADC. Este valor máximo es de aproximadamente 3,15V por lo que el ADC no tendrá problema para leerlos, cuyo valor límite es de 3,3V.

Se pasa ahora a analizar una gráfica proporcionada por el fabricante, donde se muestra la relación entre la distancia medida y la salida proporcionada por el sensor:

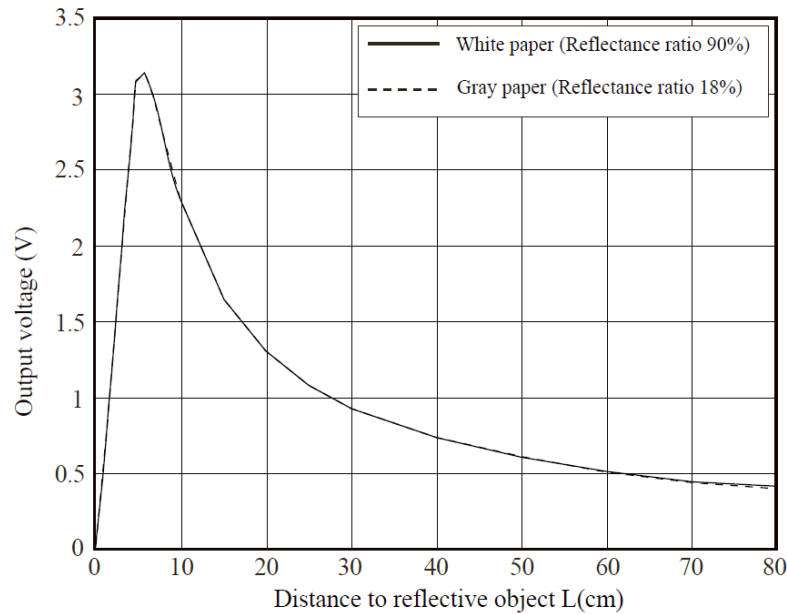


Figura 71. Salida del sensor GP2Y en función de la distancia

Se observa que en la zona de interés para el proyecto ($L > 5\text{cm}$) la salida no sigue una relación lineal respecto a la distancia. Sin embargo, esto no es un problema ya que la utilidad de estos sensores es la de determinar una zona límite donde el robot parará, evitando así una colisión. Al no ser la de realizar mediciones exactas, no hará falta determinar el modelo matemático de la relación.

Se procede entonces a realizar medidas experimentales y determinar así la siguiente gráfica:

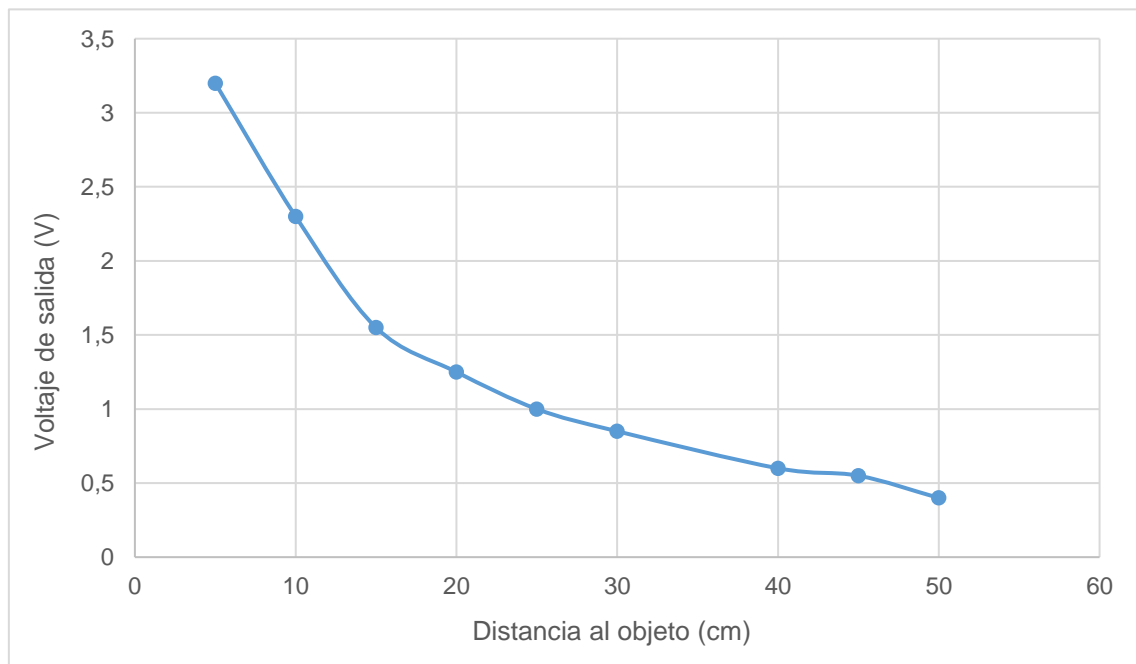


Figura 72. Gráfica sensores GP2Y medidas experimentales

Las conclusiones que se obtienen son que efectivamente la gráfica proporcionada por el fabricante se ajusta a la realidad y además se tienen los valores reales de los sensores a utilizar.

- **Distancia de parada**

El tiempo de parada hallado de forma experimental es 330ms (ver apartado *Modelado de la planta del robot y su controlador*). Posteriormente se verán las medidas experimentales de la planta del robot y en particular el tiempo de parada yendo a velocidad máxima 400 mm/s (0,4 m/s). Se calcula la distancia recorrida en la frenada, siendo 0,197 m (ecuaciones (52) y (53)). Experimentalmente se comprueba que dicha distancia es correcta. Se determina así que el robot debe frenar a 20 cm (0,2m) de un obstáculo, por lo que se asegura que el robot no entrará en la zona $L < 5\text{cm}$, donde podrían producirse falsas medidas dada la relación anteriormente mostrada y además se guarda una distancia de seguridad suficiente respecto al obstáculo.

v : velocidad

v_0 : velocidad inicial

x : espacio

x_0 : espacio inicial

a : aceleración

t : tiempo

(52)

$$v = v_0 + a \cdot t$$

(53)

$$x = x_0 + v_0 \cdot t + \frac{a}{2} \cdot t^2$$

- **Tratamiento de la información**

Como se comentaba anteriormente, la señal proporcionada por los sensores es leída mediante el ADC. Dentro del programa se trata la información como un voltaje y se determina una condición por la cual una variable asignada a cada sensor se actualiza en función de si ese valor es mayor o menor a un límite determinado. Este límite será 1.3V correspondiente a $L=20\text{ cm}$ (distancia de parada ya determinada), hallado tanto de forma teórica como experimental anteriormente. El valor límite considerado es el mínimo hallado experimentalmente de los distintos sensores, por lo que el error que se comete es conservador (en la zona $L > 5\text{cm}$ un mayor voltaje determina una distancia menor).

Además de estas consideraciones nos encontramos con un problema, el ruido del ADC. En unas condiciones controladas de prueba (sin el resto de elementos del robot en funcionamiento) se comprueba que cada medida realizada por el ADC se puede considerar como válida y real (a pesar de algunas variaciones producidas por el modo de muestreo Burst del ADC a una frecuencia de 97 kHz). Sin embargo, al poner bajo test al sistema de medida junto el resto de elementos del robot (cables sin apantallar y diversos dispositivos como los motores) se producen falsas medidas. Experimentalmente se observa y ratifica que nunca se producen más de 5 falsas medidas de forma consecutiva, por lo que se procede a realizar un filtrado digital. Dicho filtrado analiza las medidas de cada sensor y solo activa la variable limitadora correspondiente si se producen más de 5 medidas positivas consecutivas.

Esto lleva a analizar de nuevo el tiempo de respuesta, ya que la lectura de los sensores se produce a 10 Hz, es decir, cada 100 ms. Esto implica que un resultado positivo más allá del límite de seguridad será notificado con 500 ms de retraso en el peor de los casos. Consecuentemente debe aumentarse la distancia límite, para que cuando se notifique al sistema, este actúe y frene a una distancia de $L=20\text{ cm}$. Siendo la velocidad máxima del robot 400 mm/s se tiene que:

$$x = \text{Vel}_{\text{máx}} \cdot 500 \text{ ms} = 400 \frac{\text{mm}}{\text{s}} \cdot 500 \text{ mm} = 200 \text{ mm} = 20 \text{ cm}$$

Siendo $x=20$ cm la distancia a añadir para asegurar la parada requerida, quedando finalmente que la distancia límite es $L=40$ cm y un voltaje límite asociado de 600 mV (Figura 72).

5.4.1.2 Librería de control del Puente en H

Para este proyecto se crea una librería de control para la electrónica de potencia, en concreto para los puentes en H. La razón de esto es aliviar la carga de programación tanto al autor de este trabajo como al usuario que desee modificar el programa de este proyecto. Se crea así una librería de alto nivel para controlar por ende el movimiento de las ruedas.

La electrónica de potencia utilizada la Monster Moto Shield – Dual VNH2SP30 como ya se veía en apartados anteriores. Esta tarjeta está compuesta por dos puentes en H VNH2SP30, cuya tabla de verdad de control proporcionada por el fabricante es la siguiente:

IN _A	IN _B	DIAG _A /EN _A	DIAG _B /EN _B	OUT _A	OUT _B	CS	Operating mode
1	1	1	1	H	H	High Imp.	Brake to V _{CC}
1	0	1	1	H	L	I _{SENSE} =I _{OUT} /K	Clockwise (CW)
0	1	1	1	L	H	I _{SENSE} =I _{OUT} /K	Counterclockwise (CCW)
0	0	1	1	L	L	High Imp.	Brake to GND

Tabla 9. Tabla de verdad del integrado VNH2SP30

Cada puente en H cuenta con una señal de habilitación de cara al usuario (cada integrado cuenta con dos pines de habilitación, pero unidos en la tarjeta) y otras dos para el control de la dirección del movimiento del motor o su parada. A efectos de control, la señal de habilitación se impone como común a los dos puentes, activándose o desactivándose simultáneamente. Por otro lado, a cada señal de control de dirección se le asigna una salida GPIO de la LPC_1768 y a la señal de habilitación se le asigna una salida del DAC, ya que como se veía en apartados anteriores esa señal requería de una cierta corriente y el DAC puede proporcionarla. De esta forma, la librería se conforma de las siguientes funciones:

- `void stop_motor1 (void);`
- `void stop_motor2 (void)`
- `void enable_puente_H (void);`
- `void disable_puente_H (void);`
- `void write_motor_1 (int, uint8_t);`
- `void write_motor_2 (int, uint8_t);`

En lo concerniente a la habilitación y deshabilitación del puente en H, la salida del DAC proporciona su nivel máximo (3.3V) para el nivel lógico 1 y 0V para el nivel lógico 0. En cuanto a las dos funciones de escritura de los motores, en ellas se determina el ciclo PWM y la dirección de cada motor ($FORWARD=1$, $REVERSE=0$).

5.4.1.3 Comportamiento secuencia inicial

El comportamiento de la secuencia inicial engloba todos aquellos procesos que se realizan cada vez que se lanza el sistema (los procesos concernientes a la función *main* del programa). La siguiente secuencia muestra dichos procesos y su orden de ejecución:



Primero se configuran todas las salidas GPIO concernientes al control del puente en H, así como los registros del módulo PWM. Lo propio también con los pines asociados a los encoders y los registros de los timers en modo captura.

Seguidamente se configuran los registros del ADC encargado de la lectura de los periféricos. De igual modo se inicia el módulo de comunicación UART.

Hasta este momento ninguna interrupción se encontraba activa, pero todas preparadas en su configuración. Se procede entonces a la asignación de prioridades e iniciación de las interrupciones.

Esto marca el inicio del funcionamiento del sistema y por ello se efectúa una parada inicial de los motores, para evitar cualquier imprevisto y movimientos indeseados.

Tras esto se deja un tiempo para que los valores de los sensores sean recogidos adecuadamente, ya que tras el encendido del sistema hay un periodo transitorio. Dejado este tiempo se recoge la lectura del nivel de batería y se procede a limitar el ciclo máximo PWM. En el apartado de “*Adaptación al nivel de batería*” se expondrá en detalle esta limitación.

Con todos los procesos activos y la información necesaria, se inicia el control central que con el cual comienza el funcionamiento real y organizado del sistema. Se habilita entonces la electrónica de potencia ya que ahora si se tiene un control seguro sobre ella.

Figura 73. Secuencia inicial

5.4.1.4 Comportamiento periódico del control central

El comportamiento periódico del control central engloba todas aquellas acciones que controlan de forma periódica el sistema. Para este fin ya se ha visto que se usa el timer SysTick. Dentro de este proceso periódico se engloban desde el protocolo de comunicación con la estrategia hasta el controlador PI del movimiento. En este apartado se obtiene una visión general del comportamiento. Cada subproceso se detalla en profundidad más adelante.

Respecto a la comunicación destacar que el envío de la odometría y el estado de la batería se realiza una de cada diez veces, hecho que se detalla más adelante en el apartado “Comunicación con la tarjeta de estrategia”. La siguiente secuencia muestra todas las acciones y su orden de ejecución:

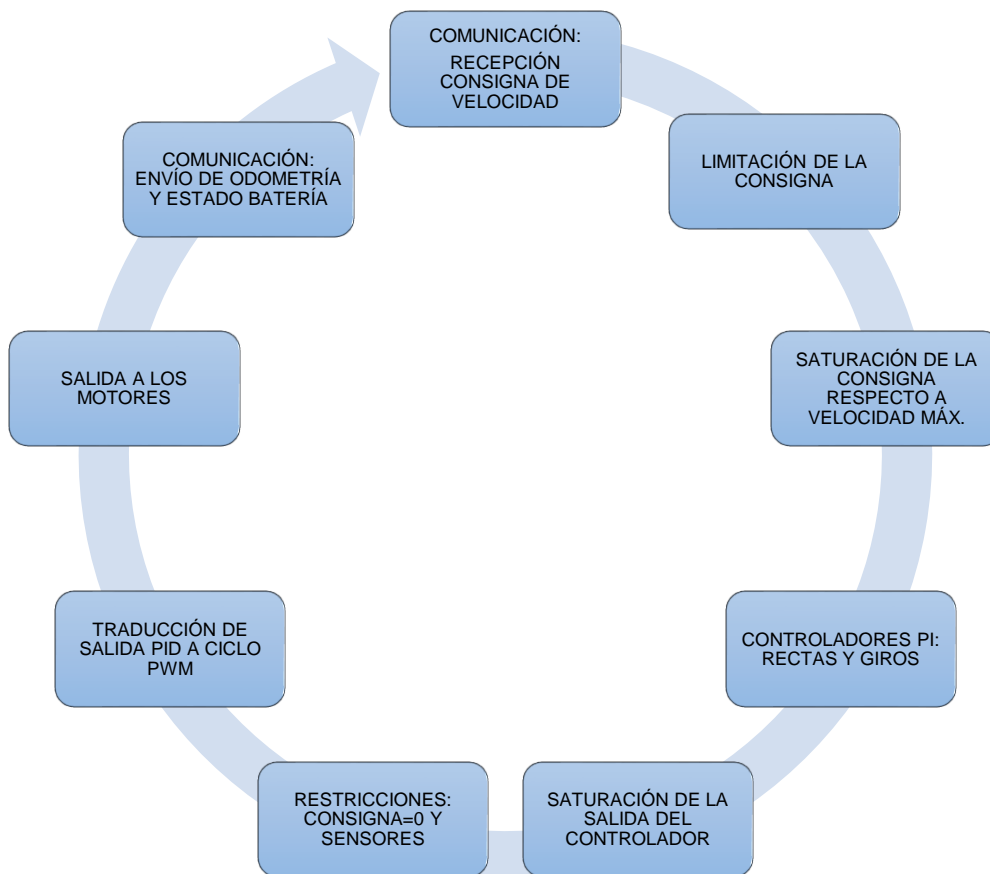


Figura 74. Secuencia de acciones del proceso de control central

5.4.1.5 Software de Control PI

El software de control PI abarca el controlador matemático del movimiento del robot, pero también alguna funcionalidad añadida, como la limitación de consigna por aspectos de seguridad o la adaptación de la salida del controlador al nivel de batería. Este control se aloja, como ya se ha visto, dentro del proceso periódico central.

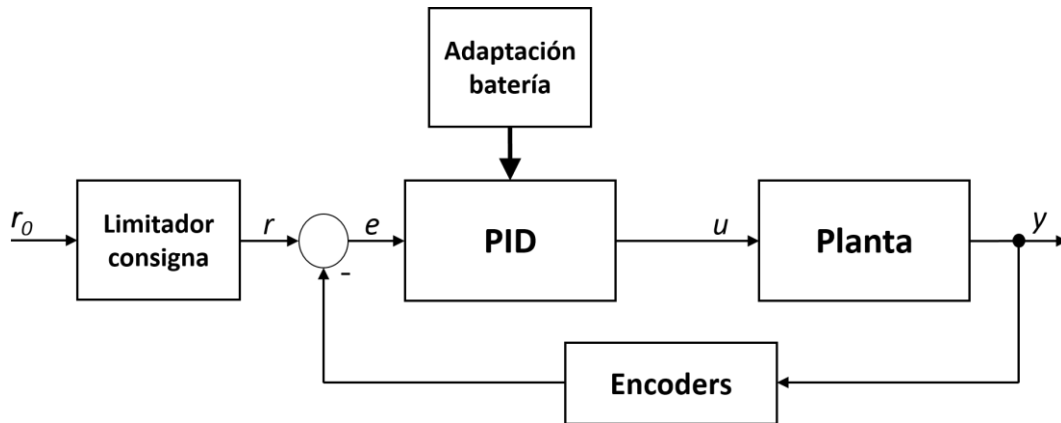


Figura 75. Esquema del controlador total desarrollado

5.4.1.5.1 Lectura de los encoders: cálculo de la velocidad

En la lectura de los pulsos de los encoders se encuentran dos tipos de información: la velocidad y la odometría. Para el control del movimiento solo interesa la velocidad, dejando la odometría para más adelante.

Para el cálculo de la velocidad se utiliza la información contenida en el tiempo entre los pulsos recibidos por una de las líneas del encoder. Estos pulsos son leídos por un pin de un timer de la LPC_1768 configurado en modo capture, uno por encoder. Cada vez que cada pin asociado detecte un flanco de subida se carga el valor del contador del timer, se hace la diferencia con el último valor guardado y se actualiza dicho valor al actual para realizar la misma acción al siguiente flanco de subida. La diferencia entre el número anterior y actual de cuentas nos dará el tiempo transcurrido entre flancos ya que conocemos la frecuencia a la que trabaja el timer, 25 MHz (frecuencia de los periféricos en la LPC_1768, 4 veces menor a la frecuencia de la CPU, 100 MHz).

Ahora bien, para calcular la velocidad se debe saber la distancia espacial asociada entre cada pulso que produce el encoder. Para esto es necesario saber el número de pulsos que suministra por cada vuelta completa. En el código del programa se tiene una constante con el valor previamente calculado de ese espacio entre pulsos, mostrándose en la siguiente ecuación:

$$\text{Distancia entre pulsos} = \frac{\text{Perímetro rueda encoder}}{\text{Número pulsos por vuelta}} = \frac{\varnothing \text{ rueda} \cdot \pi}{N} \quad (54)$$

Así, en cada interacción de la rutina asociada a cada interrupción, se tiene los siguientes cálculos:

$$1. \text{ Número cuentas entre pulsos} = C.\text{timer}_{\text{actual}} - C.\text{timer}_{\text{anterior}} \quad (55)$$

$$2. \text{Tiempo entre pulsos} = \frac{\text{Número cuentas entre pulsos}}{\text{Frecuencia timer}} \quad (56)$$

$$3. \text{Velocidad} = \frac{\text{Distancia entre pulsos}}{\text{Tiempo entre pulsos}} \quad (57)$$

5.4.1.5.2 Controlador PI

El controlador PID es tradicionalmente usado para el control del movimiento en sistemas móviles dada su versatilidad, y su sencillez matemática y computacional. Este tipo de controlador compara las medidas de los valores de un proceso o planta y un valor de referencia externo al controlador. La diferencia o error entre estos es procesado para calcular la nueva entrada de la planta que controla. El PID tratará de ajustar la planta hasta que los valores medidos de ésta coincidan con el valor de referencia deseado.

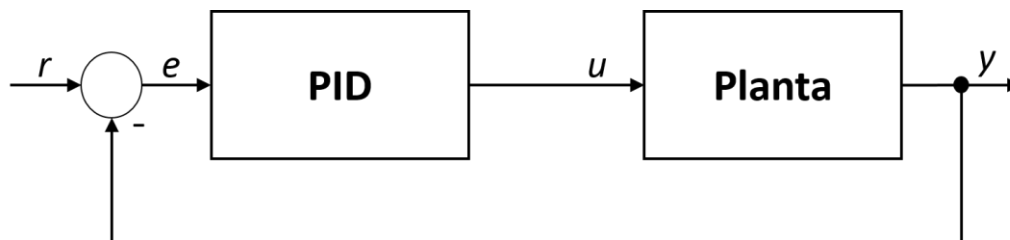


Figura 76. Esquema de controlador en lazo cerrado

Su funcionamiento se basa en la lectura mediante sensores del estado a controlar de una planta. El error entre la referencia y dicho estado será procesado por el PID de tres maneras para:

1. Manejar el presente, mediante el término proporcional (P).
2. Recuperar desde el pasado, mediante el término integral (I).
3. Anticipar el futuro, mediante el término derivativo (D).

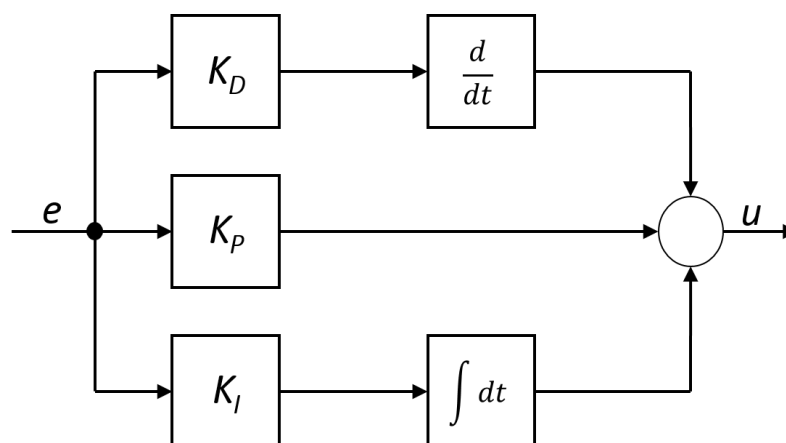


Figura 77. Esquema de PID detallado

En este proyecto se estudió la posibilidad de aplicar un controlador PID discreto Backward Euler [12], pero más tarde se comprobó que el cálculo matemático era excesivo y los resultados no diferían al PID tradicional.

En un principio se aplicó el controlador PID discreto tradicional, pero en la etapa de ajuste, tanto simulada como experimental, se encontraron inestabilidades en la salida. Finalmente se simplificó a un PI ya que aportaba mejores resultados al movimiento.

La razón fue que el término derivativo volvía inestable al movimiento. Esto es así ya que la planta se encontraba con perturbaciones periódicas, debidas a las tolerancias en la construcción de la transmisión motor-rueda. La planta tiene una respuesta lenta ante la salida del controlador debido a su peso, por lo que el término derivativo no era capaz de tener efecto con la rapidez necesaria en la planta para corregir dichas perturbaciones. Cuando la planta respondía a esta corrección las perturbaciones presentaban ya signo contrario o eran inexistentes. Por lo tanto, el error del controlador aumentaba descontroladamente, volviéndose el término derivativo contraproducente, al contrario de lo que podría esperarse.

Estas perturbaciones periódicas son asumibles y se decide eliminar el término derivativo del PID. Esto se expone con resultados experimentales más adelante, en el apartado de modelado y simulación.

Existe así un controlador por cada motor y por cada tipo de movimiento (rectas o giros). El momento de inercia no es el mismo en una recta y en un giro en el sitio (respecto al eje central), implicando comportamientos distintos de la planta y por ende un control distinto. Se dispone entonces de 4 controladores (Figura 78), compartiendo todos ellos las mismas ecuaciones pero con distintas constantes P e I. Las ecuaciones discretas utilizadas en el código del programa son las mostradas abajo.

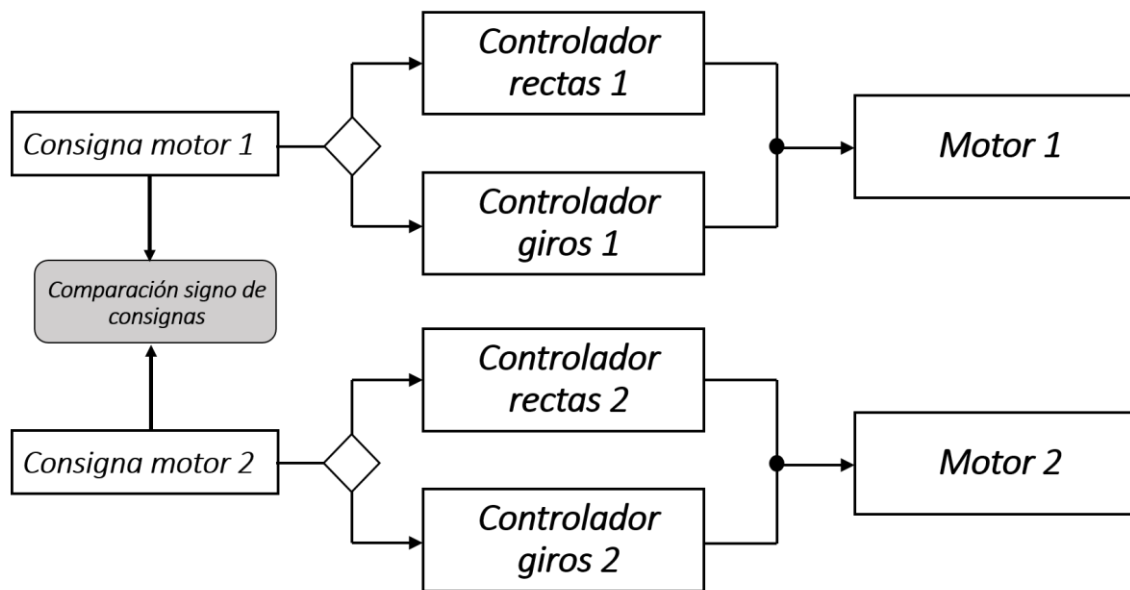


Figura 78. Lógica de los 4 controladores PI

- (58)
- (59)
- (60)

$$e_0 = r - y$$

$$e_1 = e_1 + e_0$$

$$u = k_p \cdot e_0 + k_i \cdot e_1$$

r: consigna de referencia
u: entrada a la planta
y: salida de la planta
e₀: error actual
e₁: error acumulado
k_p: término proporcional
k_i: término integral

donde k_p y k_i son:

$$(61) \quad k_p = C_p \quad C_p: \text{constante proporcional}$$

$$(62) \quad k_i = C_i \cdot T_s \quad C_i: \text{constante integral}$$

La saturación de la consigna se realiza mediante unas sentencias *if/else*, evitando así que la salida matemática del controlador entre en discordancia con la velocidad real desarrollable.

Para finalizar, la salida del controlador (en velocidad) debe ser traducida a número de cuentas del registro del módulo PWM correspondiente, variando así el ciclo de trabajo de la señal PWM dirigida hacia los motores. Esto se lleva a cabo mediante el siguiente cálculo, siendo u la salida del controlador:

$$N^{\circ} \text{ cuentas} = u \cdot \frac{N^{\circ} \text{ cuentas}_{m\acute{a}x}}{\text{Velocidad}_{m\acute{a}x}} \quad (63)$$

La relación lineal entre la velocidad y el voltaje es solamente válida en régimen permanente (velocidad constante). Esto es así ya que la aceleración es nula y la corriente es siempre la misma (pues depende del par). A velocidad constante la única fuerza que hay que vencer es la del rozamiento, que no depende de la velocidad, por lo que el par es constante.

Entonces, se consigue un control muy ajustado a la realidad. En régimen permanente se alcanza la velocidad exacta y en el transitorio el controlador funciona bien puesto que, aunque no existe una relación lineal velocidad-voltaje, un mayor voltaje sigue significando un aumento de la velocidad.

➤ Frecuencia de muestreo

La frecuencia de muestreo del PI y por ende del proceso de control central se razona siguiendo el teorema de Nyquist:

$$\text{Frec}_{\text{muestreo}} > 2 \cdot \text{Frec}_{m\acute{a}x} \quad (64)$$

Como se ve más adelante, el tiempo de aceleración máxima del robot es de 0,46 segundos. Si el robot acelerara y frenara seguidamente (a una velocidad 4 veces inferior a la máxima, por lo que el tiempo de aceleración sería 0,115 segundos) se obtendría una frecuencia de:

$$\text{Frec}_{m\acute{a}x} = \frac{1}{T} = \frac{1}{0,115 \text{ s}} = 8,7 \text{ Hz} \approx 10 \text{ Hz} \quad (65)$$

Según el teorema de Nyquist valdría con imponer una frecuencia de muestreo de 20 Hz, pero para que el control sea más fiable se impone una frecuencia de muestreo 10 veces mayor a la máxima [13]:

$$\text{Frec}_{\text{muestreo}} = 10 \cdot \text{Frec}_{m\acute{a}x} = 100 \text{ Hz} \quad (66)$$

5.4.1.5.3 Limitador de la consigna

Dentro del control del movimiento se incluye un limitador de la consigna de velocidad por motivos de seguridad. La planta a controlar (los motores DC junto con el robot) requiere aceleraciones y frenadas progresivas, especialmente las aceleraciones. Por lo tanto, para filtrar posibles errores de la estrategia y descargarle de tal responsabilidad, se fija una aceleración máxima (tanto positiva como negativa) y cualquier variación mayor en términos absolutos que se produzca en la entrada de la consigna será suavizada acorde a esta aceleración máxima fijada.

La forma de funcionar de este limitador es la siguiente:

- Cada vez que se recoge una nueva consigna de velocidad se analiza si dicha consigna está incrementándose positiva o negativamente, comparándose con la anterior (previamente guardada).
- Se realiza la diferencia entre ambas y si la diferencia es mayor de lo fijado por la aceleración máxima, la consigna actual se modifica recortándose de tal forma que la variación respecto a la anterior no sea mayor a la variación máxima fijada.
- Finalmente la consigna actual se guarda para repetir el proceso en el siguiente ciclo del control (si se modifica la consigna, ya sea por el limitador o por otro elemento del control, se guardará la consigna modificada ya que es la aplicada).

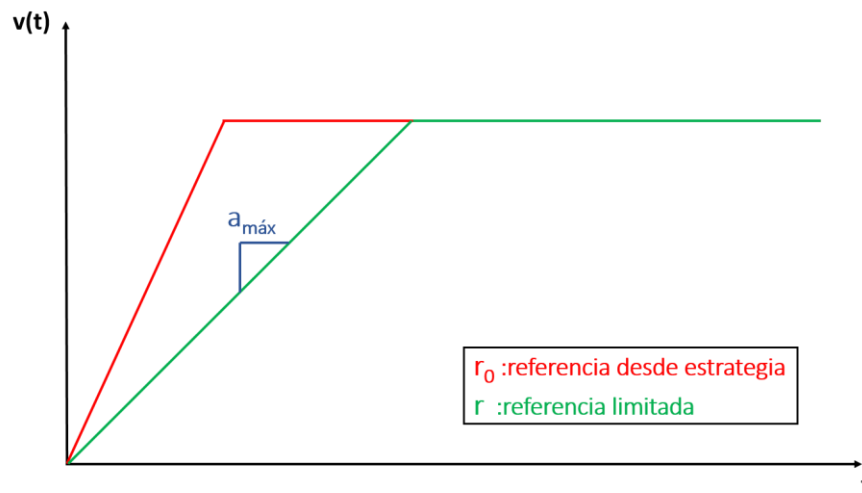


Figura 79. Limitación de la aceleración máxima

La lógica matemática del limitador reside en que un aumento lineal de la velocidad se extrapola en una aceleración espacial del movimiento. Para este proyecto la aceleración máxima fijada será de 300 mm/s^2 , demostración se ve en el apartado de modelado y simulación. Al ser la frecuencia del control de 100 Hz, se tiene que cada interacción se realiza cada 10 ms, por lo que la variación máxima en términos de velocidad es $3 \text{ mm/(s} \cdot 10\text{ms)}$ o lo que es lo mismo $3(\text{mm/s})/\text{ciclo}$.

$$\text{Aceleración máxima} = 300 \frac{\text{mm}}{\text{s}^2} \quad (67)$$

$$\text{Variación máxima de velocidad} = 300 \frac{\text{mm/s}}{\text{s}} = 3 \frac{\text{mm/s}}{10\text{ms}} = 3 \frac{\text{mm/s}}{\text{ciclo}} \quad (68)$$

5.4.1.5.4 Adaptación al nivel de batería

Este apartado surge de la necesidad de aprovechar al máximo la carga de la batería y de la optimización del controlador, creando así una lógica de adaptación software. Además, se encuentra también la necesidad de adaptar cualquier tipo de batería (y su propio voltaje variable) a los motores utilizados en el proyecto. Aparece así la posibilidad de cambiar los motores sin realizar ningún cambio significativo en la lógica del programa.

Durante la realización de este proyecto se usaron diferentes tipos de baterías (incluyendo las finalmente utilizadas), todas ellas con voltajes nominales distintos. Sin embargo, los motores utilizados siempre tuvieron el mismo valor nominal de voltaje. En la fase de desarrollo se encontraron anomalías en el movimiento, debidas en parte a que el controlador no disponía de los datos reales del nivel de la batería. Es por ello por lo que se plantea realizar esta lógica de adaptación.

- **Batería a máxima carga**

Si el voltaje de la batería a máxima carga es superior al voltaje máximo admitido por los motores puede encontrarse con un problema, aunque no crítico ya que es sabido que los motores DC admiten sobretensiones durante ciertos periodos de tiempo. Aun con esta consideración, hace falta hacer la adaptación a un nivel superior de voltaje para obtener un preciso control del movimiento.

- **Batería por debajo de su valor nominal**

Tras un ciclo de descarga de la batería se encuentra que su capacidad comienza a bajar y su voltaje lo hace con ella también. Este hecho no supone un problema a nivel físico para los motores, pero sí a un nivel lógico para el controlador, ya que si el nivel baja demasiado, el controlador podría volverse inestable tratando de establecer una velocidad no alcanzable. En la etapa de aceleración del movimiento esto podría no apreciarse, ya que acelera hasta la velocidad máxima posible y se mantiene, pero el problema aparece en la etapa de deceleración. El error matemático acumulado por el controlador (error del término integral) aumenta descontroladamente, produciéndose un efecto rebote cuando la consigna disminuye a valores de velocidad alcanzables.

Por el contrario, si se quiere alcanzar una velocidad menor que la máxima alcanzable, estaríamos empeorando las prestaciones del controlador. Si éste se diseña con la premisa de una tensión constante, cuando la tensión sea menor el controlador “creará” que sigue teniendo la misma capacidad en la respuesta transitoria, desembocando en un empeoramiento del régimen transitorio y por ende en posibles inestabilidades.

- **Lógica de la adaptación**

La adaptación se compone de tres partes: una de lectura y establecimiento de valores máximos, otra de saturación de consigna y otra de adaptación de la salida a los niveles máximos.

- Lectura y valores máximos.

Esta parte se encuentra, dentro del programa, junto con la lectura de los sensores de distancia y puntualmente en la secuencia inicial de arranque. En ella se mide el voltaje

de la batería y se calcula el ciclo de trabajo PWM máximo, estableciéndose menor al ciclo total de la PWM si la tensión es mayor a la nominal de los motores y manteniéndose igual al ciclo total si la tensión es menor. Con esto se consigue limitar en todo momento la salida hacia los motores a un máximo de su valor nominal para evitar sobrecalentamientos en régimen permanente.

$$\text{Ciclo trabajo PWM máx.} = \frac{\text{Ciclo PWM} \cdot V_{\text{nominal}}}{V_{\text{real}}} \quad (69)$$

También se debe calcular la velocidad máxima alcanzable asociada al nivel de tensión de la batería, necesaria para los cálculos matemáticos del controlador. Para ello se debe calcular previamente y de forma experimental la velocidad máxima desarrollada por el sistema en condiciones nominales, guardándose como una constante en el programa. Posteriormente se aplica una relación lineal entre tensión y velocidad, con la lógica vista en el apartado anterior “Controlador PI”.

$$\text{Velocidad}_{\text{máx}} = V_{\text{real}} \cdot \frac{\text{Velocidad}_{\text{máx. teórica}}}{V_{\text{nominal}}} \quad (70)$$

Añadir que en la medición y cálculo del voltaje de la batería se aplica un factor de corrección, ya que como se veía en el apartado de *Diseño Electrónico*, para medir el voltaje se debía adaptar a la entrada del ADC con un factor de división o atenuación determinado (C_{aten}).

$$V_{\text{real}} = V_{\text{medido}} \cdot C_{\text{aten}} \quad (71)$$

- Saturación de consigna.

Esta consiste en limitar la consigna de velocidad enviada por la estrategia a la velocidad máxima alcanzable (calculada anteriormente). Tiene por objetivo eliminar la acumulación de error (error del término integral del controlador) como anteriormente se exponía.

- Adaptación de la salida.

En esta etapa se consigue la optimización del controlador al voltaje de la batería y por ende a la velocidad máxima desarrollable. Obtenemos así que las prestaciones del controlador, sobre todo en el régimen transitorio, no empeoren y se mantengan las características de diseño sea cual sea la tensión de la batería. Esto es así ya que el diseño se realiza suponiendo un máximo y un mínimo, y al adaptar la salida del controlador a los límites reales, sus características a nivel teórico y matemático se conservan. De lo contrario, los límites reales con los teóricos utilizados por el controlador entrarían en discordancia, empeorando el control.

La adaptación de la salida está comprendida en el Controlador PI visto anteriormente (ecuación (69)). Éste usa la velocidad máxima determinada por la lógica de adaptación.

• **Protección de la batería**

En adición a toda la lógica de adaptación enfocada al control de los motores se incorpora una pequeña funcionalidad de protección de la batería.

Las baterías del tipo Li-Po no soportan las sobre descargas, concluyendo estas en la destrucción de las mismas. Para proteger de este hecho a la batería se crea una variable

que indica si el voltaje de la batería se encuentra por encima o por debajo de un nivel de seguridad de 10,4V (Figura 80). Si la batería se encuentra por debajo de este nivel el microcontrolador establece una PWM del 0% de ciclo, anulando así la alimentación de los motores y evitando que la batería se descargue. Otros dispositivos del robot pueden seguir alimentándose de la batería, pero frente a los motores se considera un gasto despreciable.

Esta variable de seguridad es enviada continuamente por el puerto serie a la tarjeta de inteligencia central, para que pueda actuar también en consecuencia ya que afecta a todo el robot.

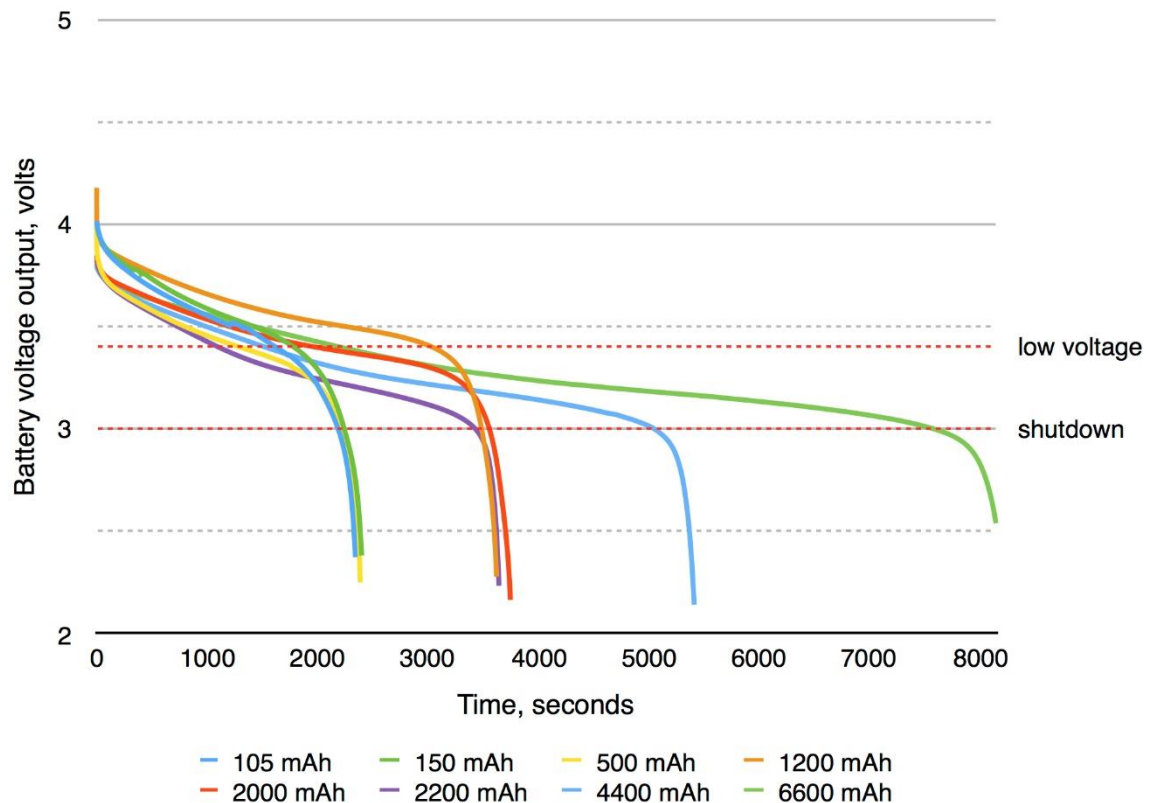


Figura 80. Descarga de baterías Li-Po y su límite de descarga. Se muestra la tensión por celda individual

5.4.2 Modelado y simulación de la planta del robot y su controlador

5.4.2.1 Recogida de datos y su procesamiento

La recogida de datos es el primer paso de todo un proceso para modelar y controlar eficientemente a una planta basándonos en su simulación. En este proyecto los datos que se recogen son de la velocidad del robot respecto al tiempo y de la odometría. En este apartado se utilizan velocidades, y con la ayuda de Matlab se logra determinar el modelo de la planta de cada motor del robot junto a sus no linealidades.

- **Recogida de datos**

Los datos que se pretenden recoger se almacenan de forma temporal en variables globales. Estas son enviadas en cada ciclo del proceso central de control (SysTick) mediante el puerto serie (UART). Los valores de estas variables son enviados separados por comas (“,”) preparando su posterior procesamiento en Matlab.

La recogida de los datos se realiza mediante CoolTerm, una aplicación simple de terminales de puertos serie. Se abre así un puerto serie para la recepción de datos (Figura 81), tomando precaución en seleccionar el mismo baudrate al que se reciben los datos. Este programa da la opción de grabar todos los datos que entran por el puerto y guardarlos en un archivo de texto (Figura 82).

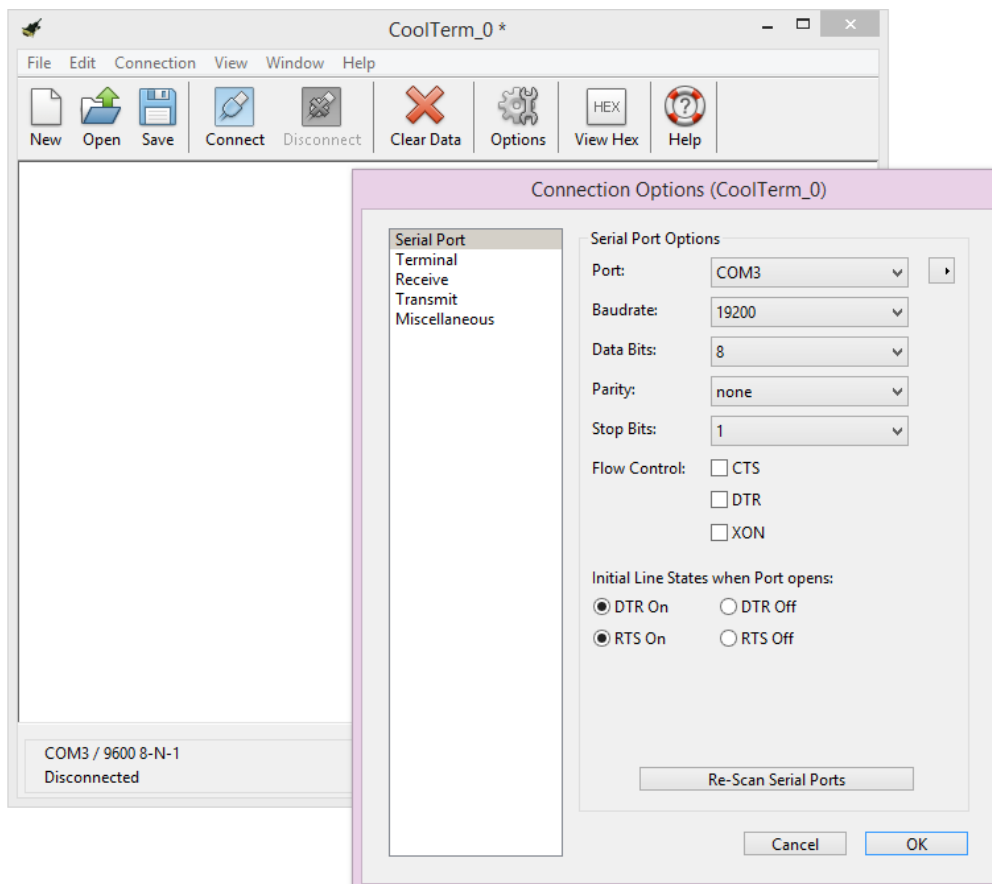


Figura 81. Vista del programa "CoolTerm". Configuración

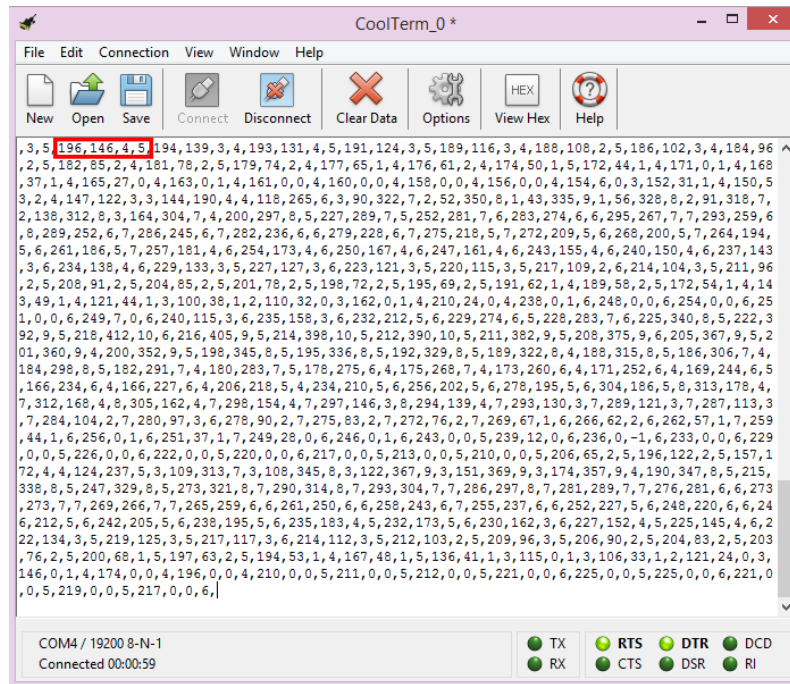


Figura 82. Vista del programa "CoolTerm" recibiendo datos. En rojo la cuádrupla con la información de un ciclo de muestreo

- **Procesamiento**

El procesamiento de los datos se realiza mediante un script en Matlab (“*calculo_motor.m*” del proyecto), teniendo distintas fases.

1. Lectura del archivo de texto

El primer paso es recoger los datos desde el archivo de texto creado por CoolTerm. Para ello se utiliza la función “*csvread*” la cual lee y detecta las “,” separando y guardando los valores numéricos en un array en el Workspace de Matlab. Cada valor numérico ocupará una posición en el array, eliminándose las “,”.

2. Separación de los datos

Una vez recogidos en un solo array se procede a separarlos según su procedencia, es decir, crear 4 arrays distintos: velocidad de la rueda 1, velocidad de la rueda 2, pulsos del encoder 1 y pulsos del encoder 2.

Se veía antes (Figura 82) que la disposición de los datos enviados es:

1º- vel. rueda 1	2º- vel. rueda 2	3º- odomet. 1	4º- odomet. 2
------------------	------------------	---------------	---------------

Tabla 10. Disposición de los daos enviados al ordenador

Se separan los datos dentro de una sentencia *for* de tal forma que en cada array se vayan guardando consecutivamente la posición correspondiente de los datos. Tras esto se puede trabajar ya por separado cada tipo de información.

Hay que tener en cuenta el número de valores recogidos, dependientes estos del tiempo de la prueba realizada, para determinar el tamaño del array. En el ejemplo que se analiza en el paso 4 la duración de la prueba es de 5 s y recogiendo cada cuádrupla de valores cada 10 ms (frecuencia del proceso central de control) se obtienen 20.000 valores en total, es decir, 500 valores por cada tipo de dato.

3. Creación del array referencia

La referencia o consigna aplicada a los motores no es recogida por lo que se debe crear en previsión de usarla a la hora de hallar el modelo de cada motor. Además su representación nos ayuda en el análisis de los datos.

En el ejemplo que se describe en el siguiente paso la excitación es un escalón de valor 140 mm/s y de 4 s de duración, aunque se debe tener en cuenta que la duración total es de 5 s, valiendo 0 mm/s el segundo restante. Ayudándose de un bucle *for* se crea un array del mismo tamaño que el de cada tipo de dato (500 valores en este ejemplo).

Finalmente, una consideración a tener en cuenta es que se debe asignar al primer elemento del array el valor 0 mm/s. Esto es debido a que si directamente el escalón empieza en 140 mm/s la función de Matlab encargada de generar el modelo matemático no tiene información del régimen “transitorio” de ese escalón y no es capaz de generar un modelo correcto.

4. Procesamiento del array velocidad

Es necesario modificar la información recogida de la velocidad ya que esta contiene perturbaciones que no forman parte del modelo de los motores. Estas perturbaciones deben ser eliminadas antes de modelar matemáticamente la planta.

Las figuras de este paso 4 muestran los datos de una de las ruedas, viéndose experimentalmente que se comportan igual.

Se encuentran dos tipos de perturbaciones en la velocidad: una pequeña excentricidad del eje de la rueda que genera frenadas periódicas y un rebote del Encoder v.1 en la frenada final de los movimientos. Ambos son tratados por separado.

Además se puede determinar la aceleración máxima que puede desarrollar el robot:

$$\text{Aceleración} = \frac{\text{Vel}_{ref}}{\text{Tiempo}_{subida}} = \frac{140 \text{ mm/s}}{460 \text{ ms}} = 300 \text{ mm/s}^2 \quad (72)$$

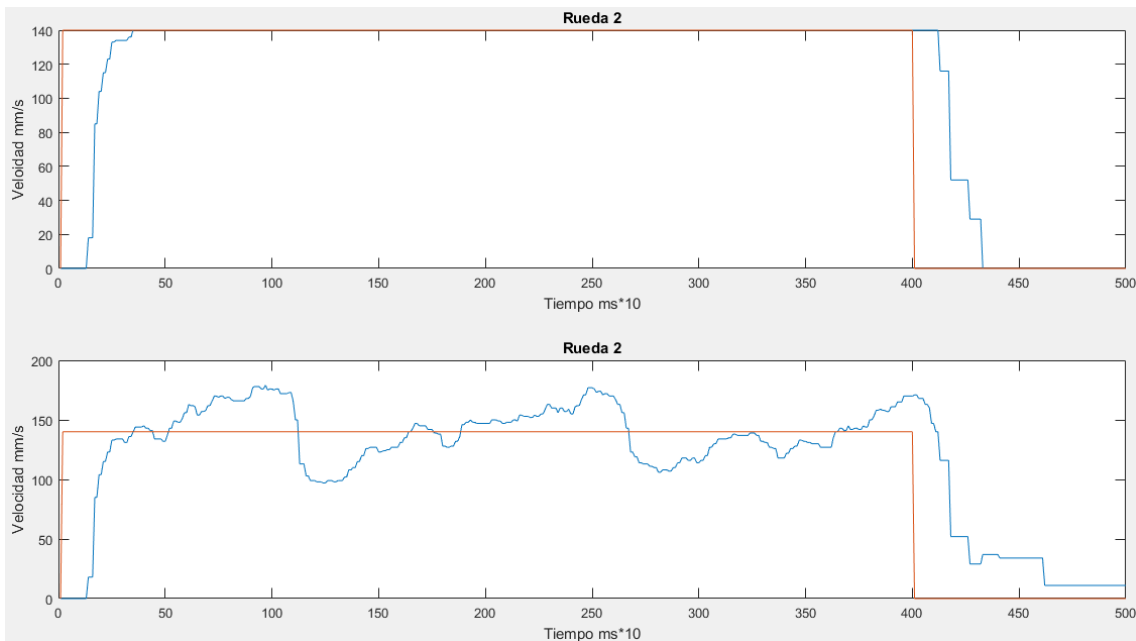


Figura 83. Arriba-Respuesta del motor sin perturbación / Abajo-Respuesta real del motor

- Recorte del freno periódico

Se puede ver en este ejemplo como la rueda realiza 3 giros completos. La perturbación se repite claramente durante el régimen permanente del movimiento, observándose que el valor medio es aproximadamente el valor de la excitación de referencia (140 mm/s).

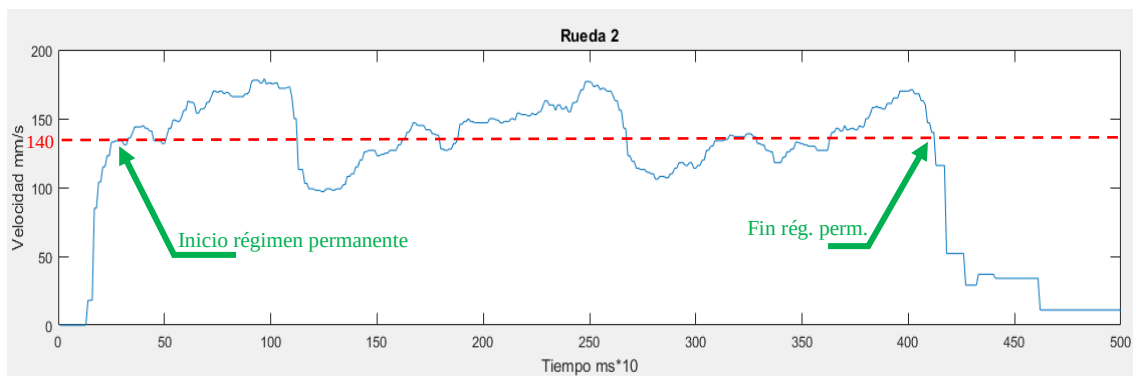


Figura 84. Análisis del régimen permanente de la respuesta del motor

Si la perturbación no existiera la velocidad en régimen permanente debería ser la de la referencia. Es por ello que en todo este régimen se establece esa velocidad (cambiando los datos correspondientes del array), quedando como se muestra en Figura 85.

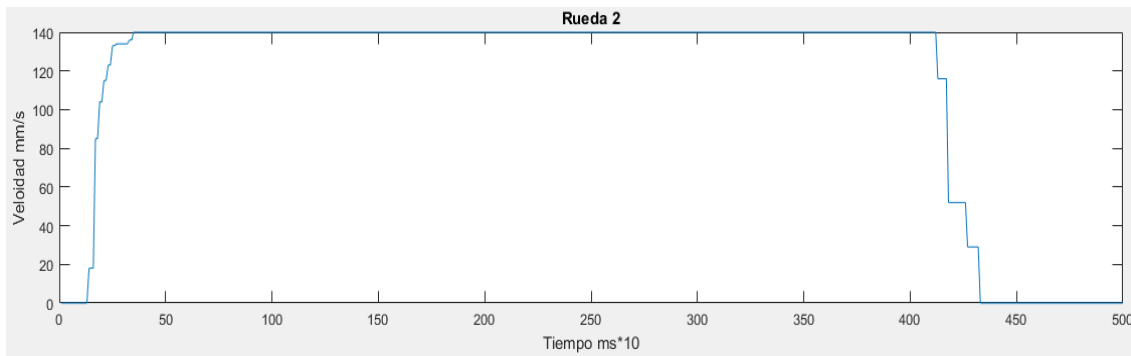


Figura 85. Respuesta en velocidad del motor sin perturbación

- Eliminación del rebote del Encoder v.1

Este rebote de la rueda dentada del Encoder v.1 es debido a que los engranajes de la aceleradora del encoder tienen una pequeña holgura. Esta perturbación ni siquiera forma parte del movimiento real del robot, por lo que hay que eliminarla.

Se detecta el momento en el que en pleno régimen transitorio de bajada la velocidad vuelve a subir (Figura 86). En ese instante el array se modifica estableciéndose a 0 mm/s la velocidad, ya que es la progresión aproximada de la forma de la frenada.

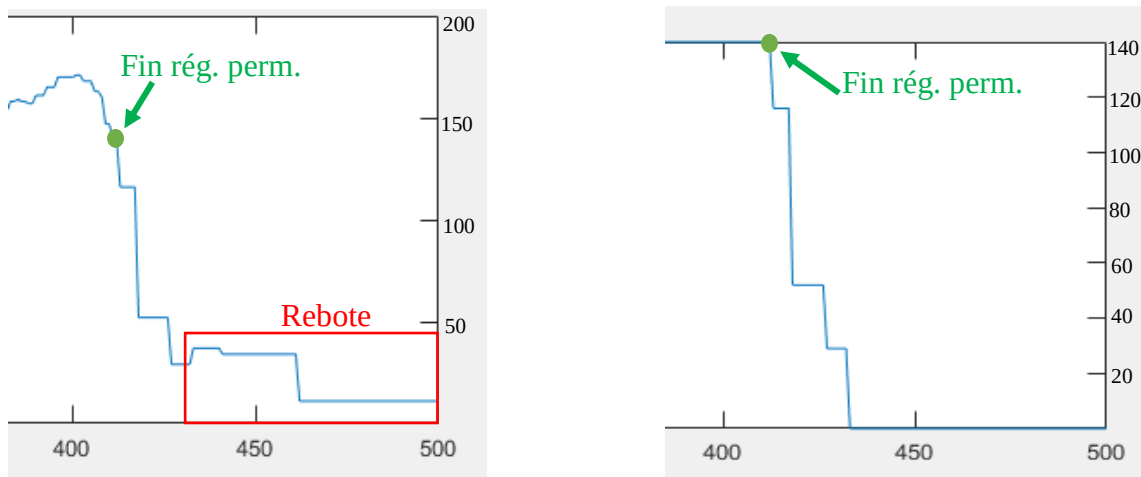


Figura 86. Estudio del rebote de los encoders y su procesamiento

5.4.2.2 Modelado de planta y simulación

El modelado de una planta consiste en hallar la función matemática capaz de describir su comportamiento. En este proyecto se busca modelar la respuesta de cada motor con la masa de todo el robot.

Este modelado lo llevamos a cabo con la ayuda de Matlab, el cual proporciona una Toolbox de Modelación de Procesos de Sistemas Dinámicos. Ésta proporciona un modelo muy acertado de la planta bajo análisis, pero ciertas no linealidades de la planta deben ser añadidas posteriormente. El modelo final se integra en Simulink.

Finalmente, y a modo de comparación, se crea en Simulink una simulación de una situación real.

- **Modelado**

El modelado se realiza mediante la Toolbox de Matlab de Modelación de Procesos de Sistemas Dinámicos. A modo de ejemplo se muestra la realización del modelo del motor “derecho” del robot (Rueda 2). El manejo y funcionamiento de la Toolbox consta de las siguientes fases:

1. Obtención de primer modelo: Toolbox

En primer lugar debemos obtener el modelo aproximado de la planta mediante Matlab. Para ello se utiliza la Toolbox que proporciona para dicho cometido. La iniciamos buscando en la sección de “APPS” en las pestañas superiores y eligiendo “System Identification” (Figura 87).

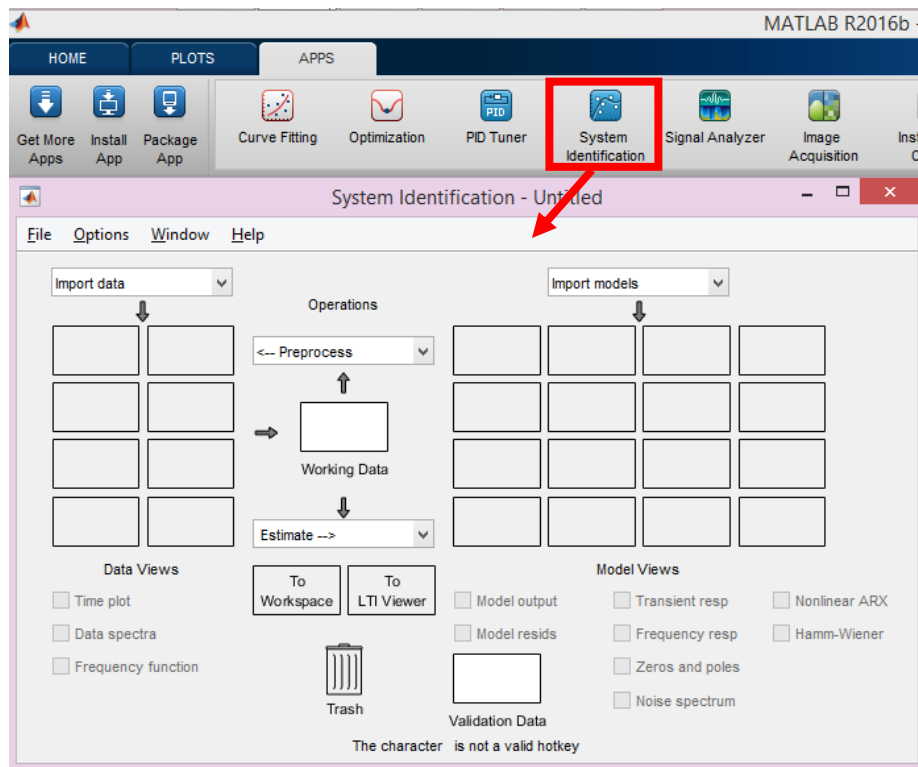


Figura 87. "System Identification"

Esta aplicación permite utilizar la Toolbox de forma gráfica, facilitando el trabajo. Dentro de ella se encuentran dos partes diferenciadas: a la izquierda se ponen los datos de entrada a analizar y a la derecha se irán mostrando los resultados obtenidos.

Se deben importar los datos abriendo el desplegable “Import data” y eligiendo la opción “Time domain data...” según se muestra en la Figura 88.

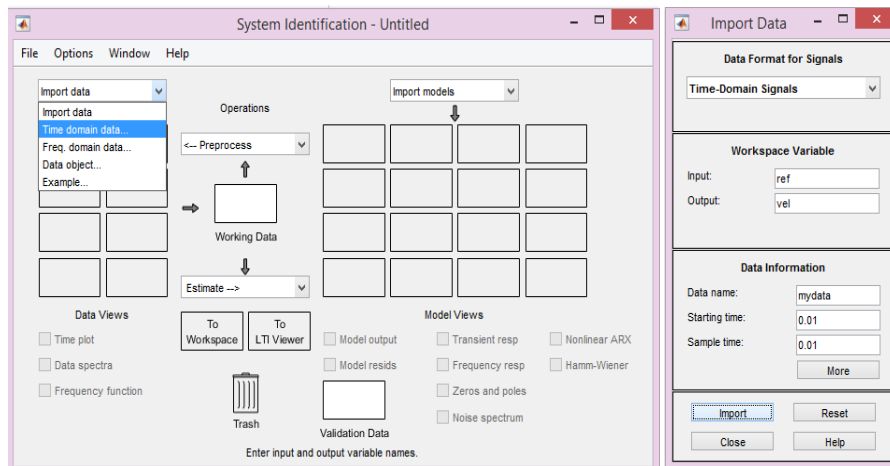


Figura 88. Importación de los datos recogidos a "System Identification"

Acto seguido se abre una pequeña ventana donde se introducen los datos propiamente dichos (Figura 88). En "Input" escribimos el nombre del array donde hayamos guardado los datos de la referencia y en "Output" lo propio con el de la velocidad recogida. El "Starting time" y el "Sample time" deben coincidir y su valor es de 0,01 s ya que el periodo de muestreo de los datos fue ese (frecuencia del proceso central de control $\rightarrow 100\text{Hz}$).

Para finalizar se pulsa "Import". En ese momento pregunta si queremos transformar los arrays fila por arrays columna, a lo que aceptamos ya que es la forma de trabajar de la Toolbox.

A la izquierda de la aplicación ya aparece la representación de los datos. Seguidamente debemos abrir el desplegable "Estimate - - >" (Figura 89) donde elegimos la opción "Transfer Function Models..." ya que es la función de transferencia lo que buscamos. Se abre una ventana en la que debemos indicar el número de polos y ceros esperados de nuestra planta. Es necesario pues hacer una estimación inicial del sistema, hallando que para un motor de corriente continua de imanes permanentes se tienen 3 polos y 0 ceros [14].

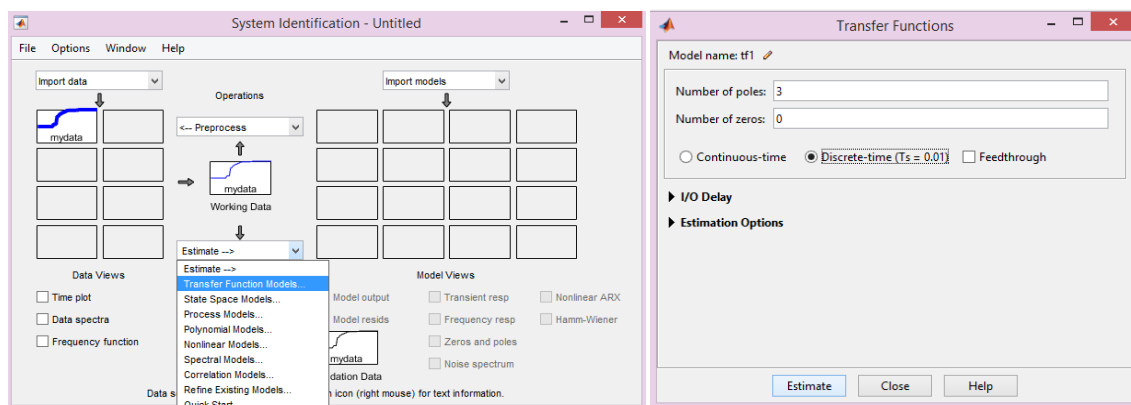


Figura 89. Modo de estimación y configuración de polos y ceros

La errónea indicación de los polos y ceros desemboca en un modelo de la planta totalmente erróneo. También se debe elegir la opción “Discrete-time (Ts=0.01)” ya que los datos han sido recogidos de forma discreta y el modelo matemático debe estar en el dominio de “z” [13]. Pulsamos el botón “Estimate” y la Toolbox finalmente nos calcula el modelo matemático, guardándose a la derecha de la aplicación. Para poder trabajar con él debemos introducirlo en el Workspace (Figura 90).

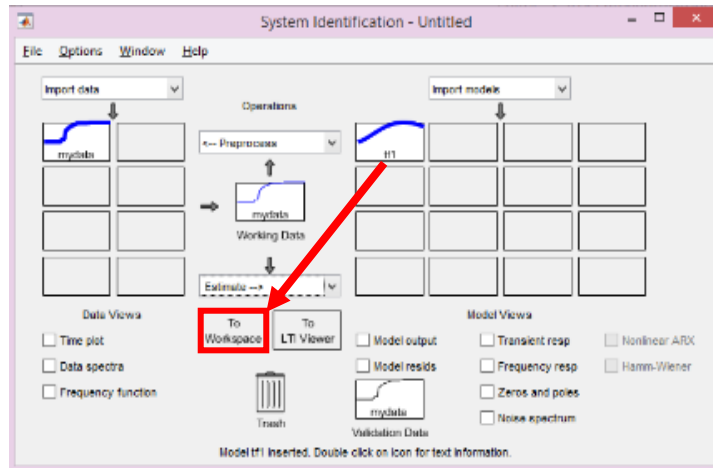


Figura 90. Exportación del modelo matemático al "Workspace"

Para realizar una comprobación rápida de que el modelo se ha creado correctamente se ejecuta la función “step(tf1)” incluyendo “tf1”, la planta del motor analizado. Esta función genera la respuesta de una función de transferencia frente a una entrada escalón.

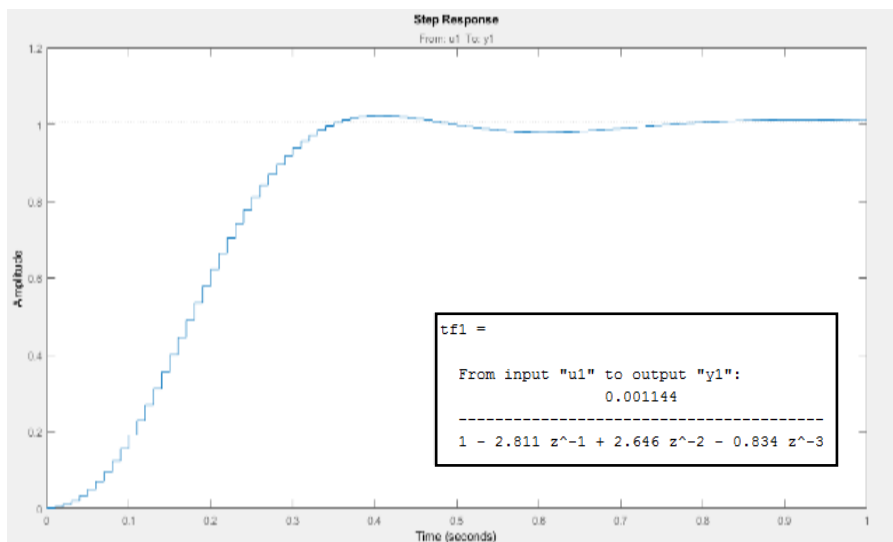


Figura 91. Comprobación de la respuesta del modelo matemático de la planta

2. Modelo en Simulink: no-linealidades

Una vez obtenida la función de transferencia de nuestra planta se debe completar el modelo incluyendo no-linealidades. Para ello utilizamos el entorno Simulink. En primer lugar se crea un bloque con nuestra función de transferencia (en el dominio “z”). A la entrada y salida de la función se introducen las dos no-linealidades de nuestro modelo:

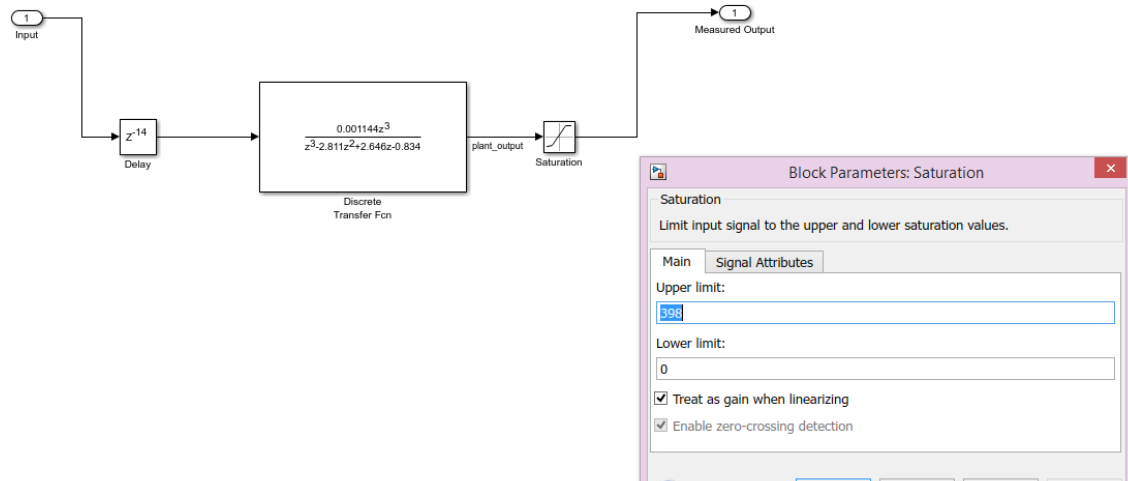


Figura 92. Modelo en Simulink de la planta de un motor

- Delay de reacción

Este parámetro simula el retardo en la reacción del motor ante una excitación del mismo. Se puede observar en la Figura 93 que el retardo es de 14 ciclos de muestreo ($T=0.01s \rightarrow 140$ ms). En el dominio de “z” esto se traduce en multiplicar a nuestra función de transferencia por z^{-14} (Figura 92).

Este retardo es prácticamente el mismo cuando se retira la excitación al motor (la excitación se retira en el ciclo 400, reaccionando en el 412 (Figura 94)). Se deduce así que el retardo es aplicable en cualquier momento, por ello el bloque “Delay” en Simulink se coloca a la entrada de la función de transferencia, afectando siempre.

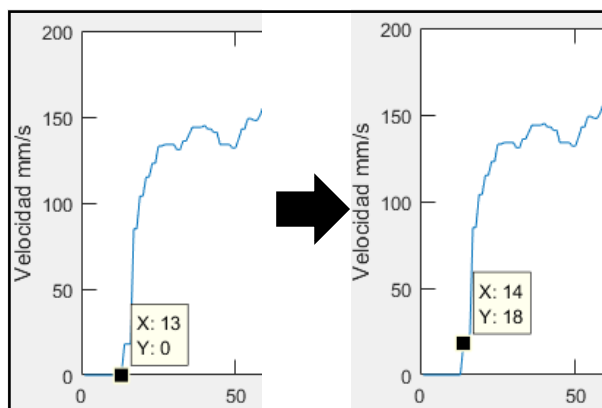


Figura 93. Estudio del delay de reacción en la aceleración

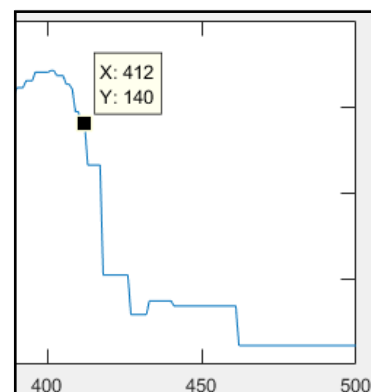


Figura 94. Estudio del delay de reacción al quitar excitación

- Saturación de la salida

La saturación sirve para modelar la máxima velocidad desarrollable por el motor a la tensión máxima. Para hallar esta velocidad se realiza una prueba con la batería

cargada al máximo y enviando como consigna directamente el máximo ciclo PWM. Obtenemos así de la lectura de los encoders que la velocidad máxima desarrollable es **398 mm/s**.

Para modelar en Simulink este límite se introduce el bloque “Saturation” con el valor de la velocidad hallada (Figura 92).

- **Simulación**

La simulación se realiza en el entorno de Simulink donde se incluye el modelo de la planta del motor como un bloque “Plant1”. Esta simulación se compone de los siguientes elementos:

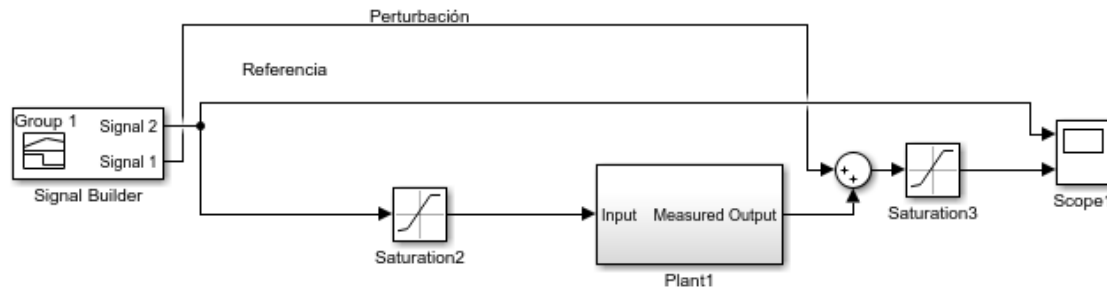


Figura 95. Modelo en Simulink de la excitación de la planta con un escalón y perturbación

- Generador de señales

Su bloque correspondiente es el “Signal Builder” y con el generamos la señal de referencia y también la perturbación producida por la excentricidad del eje (Figura 96). La perturbación se estima de forma aproximada y se “suma” a la salida de la planta del motor. Realmente la perturbación es de signo negativo ya que frena al motor.

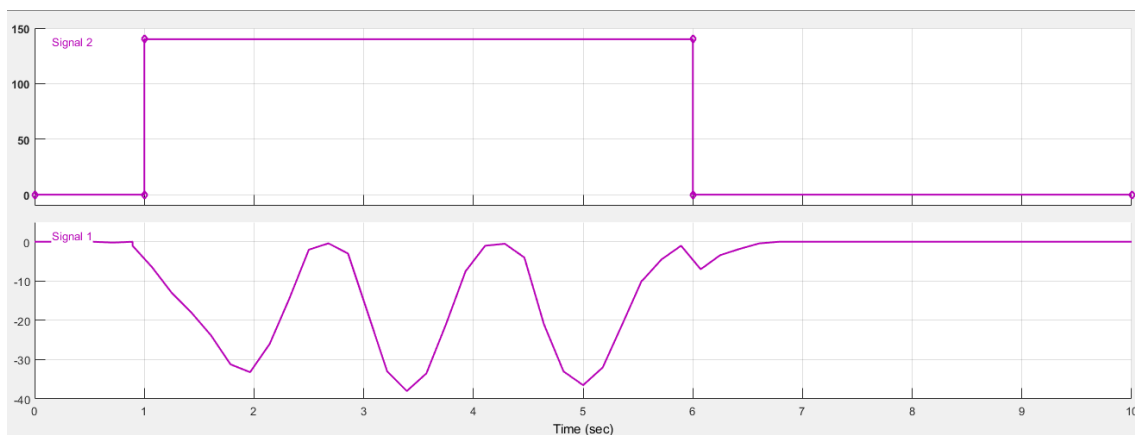


Figura 96. Bloque "Signal Builder". Signal 2 – Referencia / Signal1 – Perturbación

- Saturación a la entrada de la planta

En esta simulación actúa simplemente como medida de “protección” ya que limita el valor máximo de referencia a 398 mm/s (velocidad máxima, que corresponde con el voltaje máximo de la batería).

- Saturación a la salida

Este bloque se introduce, aun teniendo saturación la propia planta, porque la perturbación podría variar la salida fuera de los límites reales (tanto por arriba como por

abajo). Esta saturación tiene como límite inferior 0 mm/s (la perturbación solo frena) y como límite superior 398 mm/s.

- “Scope”

Con el podemos observar las señales que introducimos en él. Así conseguimos el siguiente resultado:

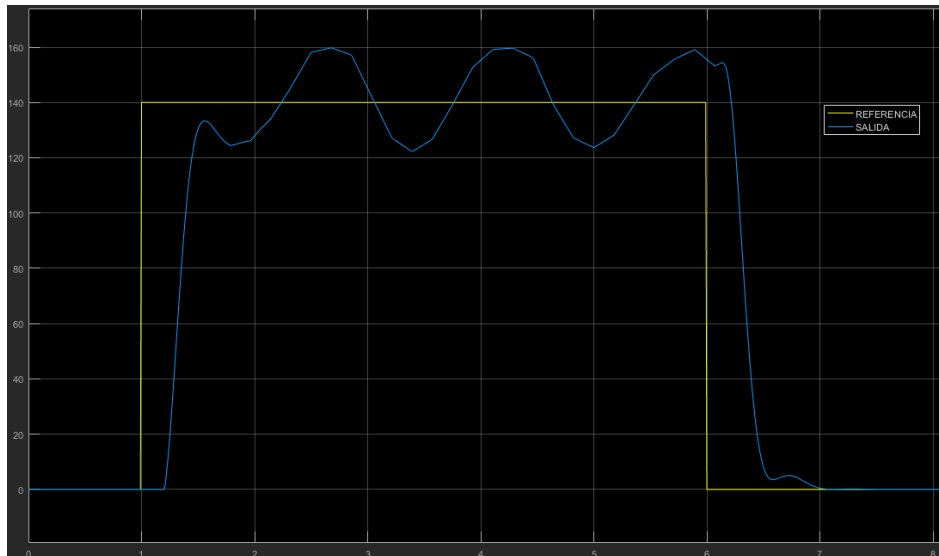


Figura 97. Simulación de la respuesta de la planta ante excitación escalón

Finalmente hacemos la comparativa con los resultados obtenidos de forma experimental anteriormente (Figura 98), comprobando que el modelo creado para el motor “derecho” se ajusta a la realidad de forma aproximada.

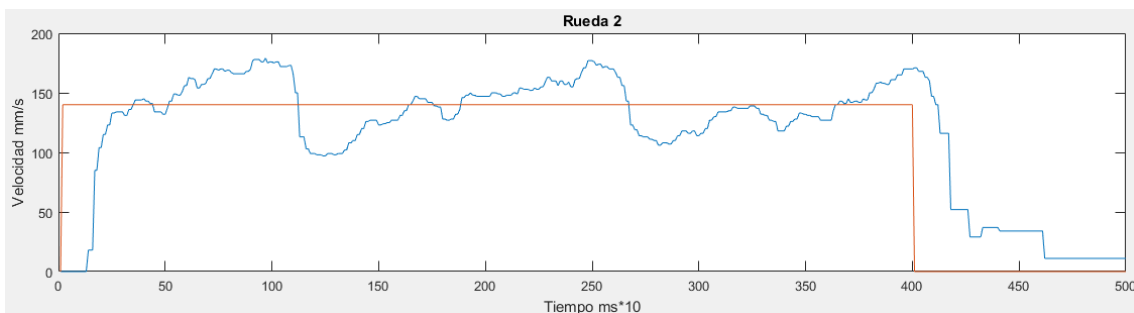


Figura 98. Simulación de la respuesta de la planta ante excitación escalón

5.4.2.3 Modelado, ajuste y simulación del controlador

Una vez modelada la planta de cada motor, se pasa a modelar en simulación el control que actuará sobre cada uno. El controlador es un PI tal y como se explicaba en el apartado *Controlador PI*, demostrándose aquí gráficamente por qué no se implementa un PID completo.

Aprovechando la simulación anterior de verificación de los modelos, se modifica añadiendo los bloques que conforman el controlador y realimentando la salida para crear el bucle de control, restándosela a la referencia.

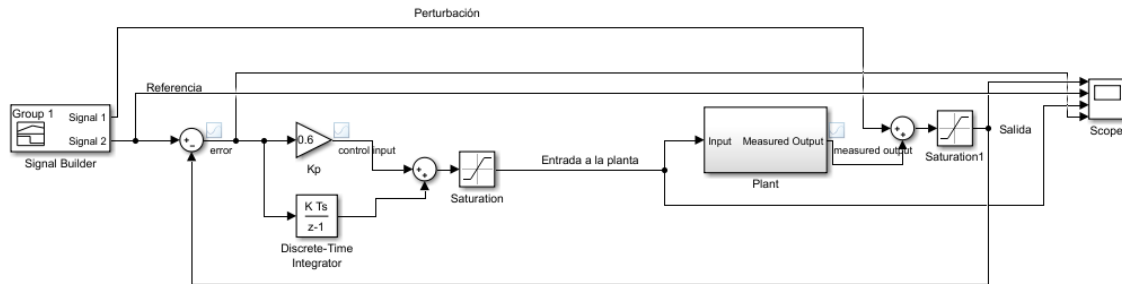


Figura 99. Modelo en Simulink de controlador PI

Los bloques correspondientes al controlador son:

- K_p (“Gain”)
Este bloque representa al término proporcional P, que es directamente la constante proporcional. Es un bloque de ganancia que multiplica al error (referencia-salida).
- K_i (“Discrete – Time Integrator”)
Este bloque representa al término integral I. Este realiza una acumulación del error o integración de él, y lo multiplica por la constante integral y el periodo de muestreo. El bloque es para un control discreto en el dominio de “z”, teniendo que indicar el periodo de muestreo:

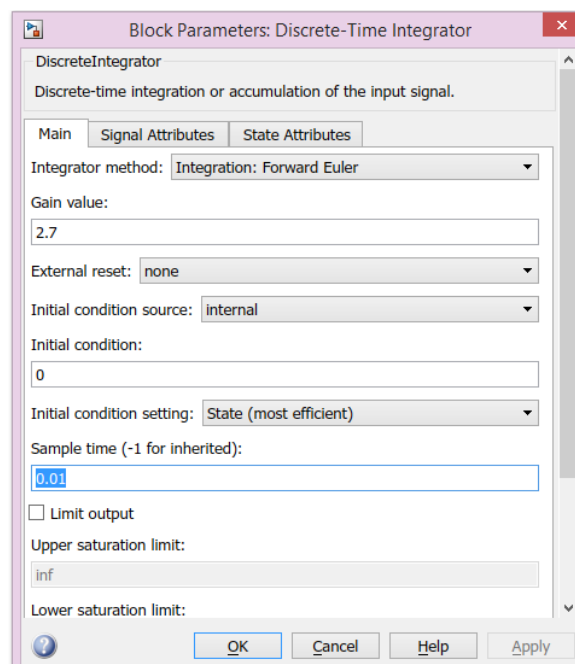


Figura 100. Bloque “Discrete-Time Integrador”
($T_s=0,01$)

Las salidas de los dos bloques del controlador se suman como se veía en las ecuaciones (58), (59) y (60) del apartado *Controlador PI*.

Para realizar el ajuste de las constantes del controlador se comienza por un ajuste aproximado siguiendo el método que sigue Javier Baliñas en su trabajo [2]. Seguidamente

se realiza un ajuste fino siguiendo el método prueba-error hasta obtener el resultado buscado. Se encuentra que para cada motor la constante proporcional se encuentra entre 0,3 y 0,6 y la constante integral entre 2 y 5.

Con la simulación preparada y el controlador ajustado se obtienen los siguientes resultados:

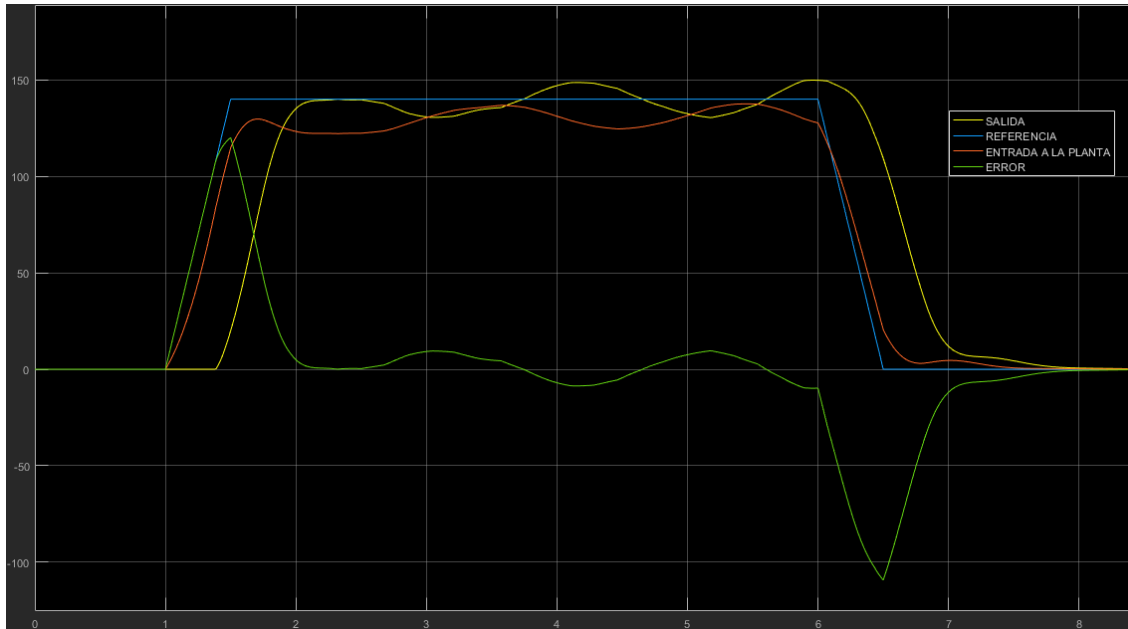


Figura 101. Simulación de PI: $K_p=0,4$ y $K_i=2$

Se puede observar como la salida del sistema (en amarillo) se ajusta aproximadamente a la referencia, salvando en gran medida la perturbación aplicada. En los regímenes transitorios el error (en verde) aumenta mucho, a consecuencia del retardo en la reacción de la planta. Durante el transcurso del régimen permanente el error fluctúa entorno a cero, lo que demuestra la buena actuación del controlador.

Finalmente, en naranja, se muestra la salida del controlador (o entrada a la planta), viéndose que no se descontrola y que realiza un control suave.

- **Verificación**

Una vez simulado el controlador se pasa a implementarlo en la Mini-DK2. Las constantes del controlador deben ser reajustadas, puesto que el modelo simulado de la planta de cada motor no es completamente exacto. Finalmente las constantes establecidas son las que siguen:

	Constante P	Constante I
Motor izquierdo (Rueda 1)	0,37	5
Motor derecho (Rueda 2)	0,322	4,5

Tabla 11. Constantes PI implementadas

Otra razón por la que difieren las constantes finales a las de la simulación es porque el robot sufrió cambios hasta el último momento, añadiéndose cosas a él y aumentando su masa. No se consideró necesaria la re-obtención del modelo puesto que se trataban de reajustes finos.

Finalmente los resultados obtenidos con el robot final son los que siguen:

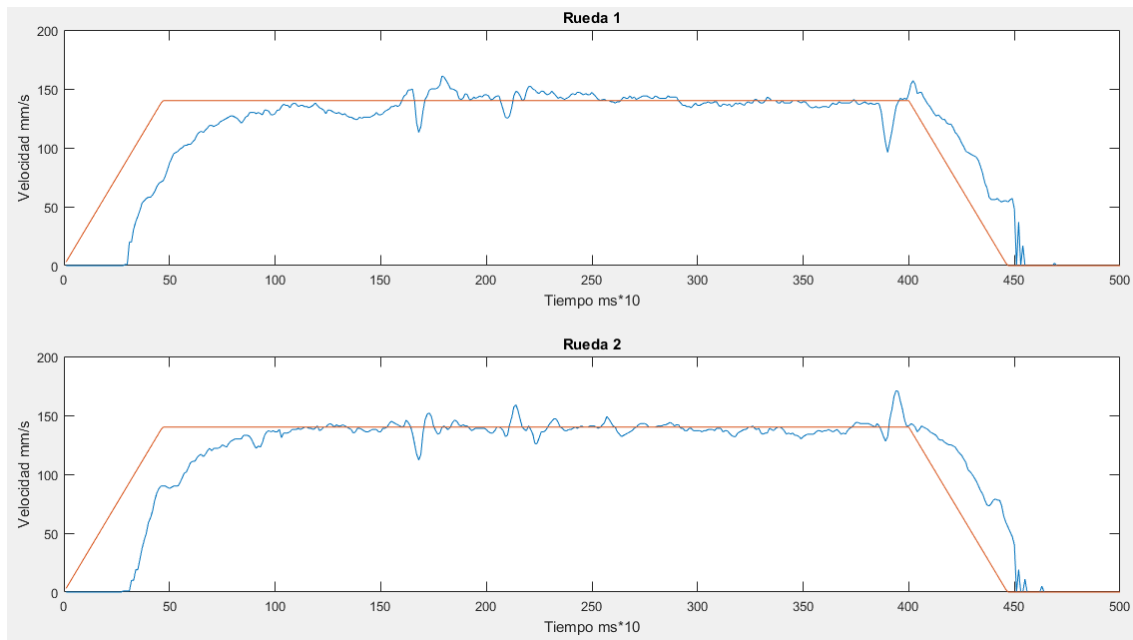


Figura 102. Perfil de velocidades de las ruedas en línea recta y referencia de 140 mm/s

Se observa que la salida (en azul) se ajusta muy bien a la referencia (140 mm/s, al igual que la simulación), incluso mejor que en la simulación. Esto puede ser debido a que en la simulación se sobre-dimensionó la perturbación, siendo la real mucho más pequeña. También se observa un comportamiento amortiguado como en el caso de la simulación. Este hecho es favorable ya que da estabilidad al movimiento.

Se consigue así controlar los motores satisfactoriamente, pasando de un robot con un movimiento perturbado (Figura 102) a un robot estable.

- **Por qué no un controlador PI+D**

El término derivativo se descartó rápidamente ya que incluía inestabilidad en el control. Esto es debido al retardo en la reacción o respuesta de los motores, por lo que el término derivativo no tenía efecto con la rapidez suficiente desembocando en una inestabilidad del sistema. Este hecho se explica de forma detallada en el apartado *Controlador PI*.

En la simulación mostrada a continuación se observa que por pequeña que fuera la constante del término derivativo solo contribuía a un empeoramiento de la salida del sistema o si era muy pequeña directamente no contribuía. En la simulación la constante del término derivativo es 0,17 (Figura 104).

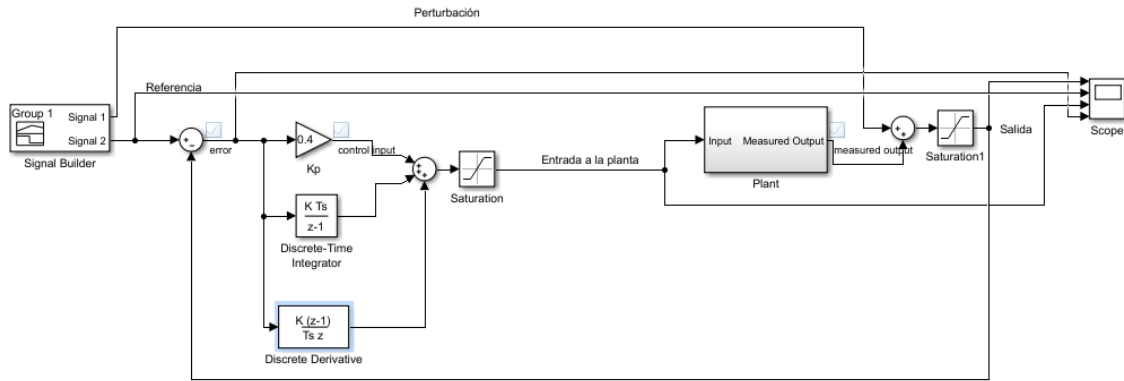


Figura 103. Modelo en Simulink de PI+D

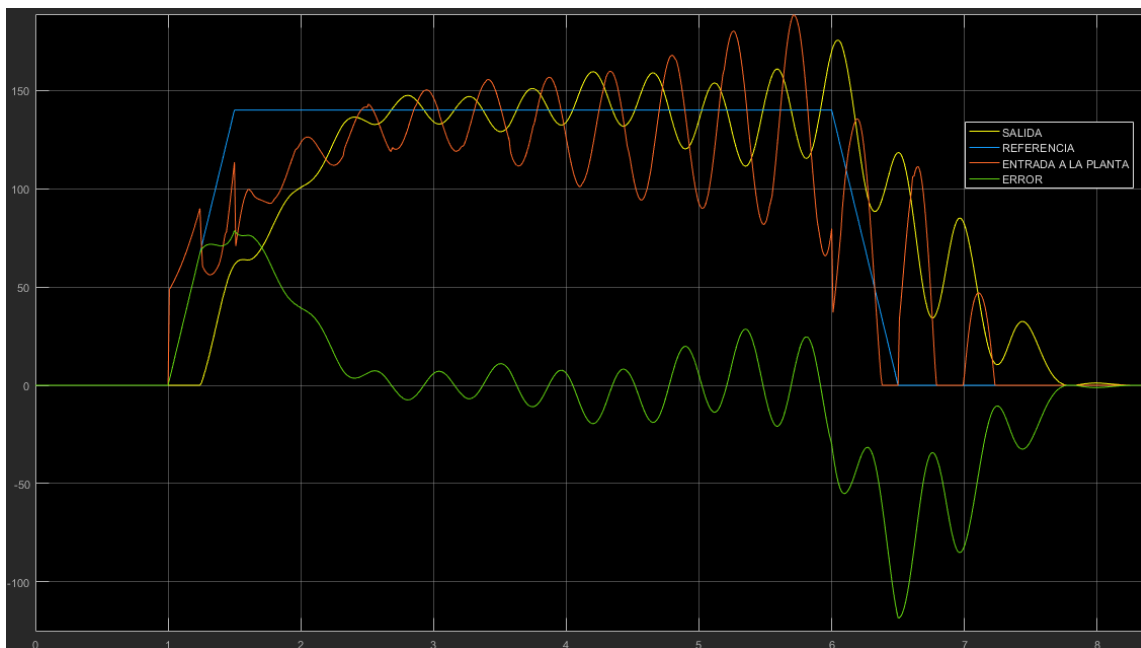


Figura 104. Simulación de PI+D

5.4.3 Comunicación con la tarjeta de estrategia

El proyecto se crea con carácter modular para que sea reutilizable en otros robots o trabajos de misma índole. La conexión con la estrategia se realiza para mantener la comunicación imprescindible con el resto del robot y como forma de elevar ciertas funcionalidades a niveles superiores. La finalidad es la de descargar a la parte motriz del robot de un sobre-procesamiento en el cálculo de trayectorias y la estrategia asociada a ellas. Así pues, la estrategia es montada sobre ROS utilizando una RaspBerry Pi.



Figura 105. Tarjeta de estrategia. RaspBerry Pi

En definitiva el proyecto tiene carácter modular, orientado más a la parte hardware y su control que a altos niveles, para que una estrategia pueda usarlo como herramienta. Se crea así un protocolo para interactuar con el resto del robot. En esta comunicación se reciben las consignas de velocidad y se envían los datos de la odometría recogidos por los encoders. La comunicación se realiza a través del bus serie (UART0).

5.4.3.1 Protocolo y tiempos

Para una correcta comunicación entre la tarjeta de control de este proyecto (Mini-DK2) y la tarjeta de la estrategia debe establecerse un protocolo. Los tiempos y la forma de estructurar la información es clave para su buen funcionamiento.

- **Protocolo**

El protocolo creado consiste en un “peloteo” continuo de paquetes de datos. Por cada paquete enviado se obtiene otro por respuesta antes de volver a enviar el siguiente. Gracias a las librerías para el manejo de la UART de la LPC1768 se consigue enviar y almacenar los datos mediante un buffer, trabajando así de forma paralela al hilo principal. La siguiente figura muestra la lógica desde el punto de vista de este proyecto:

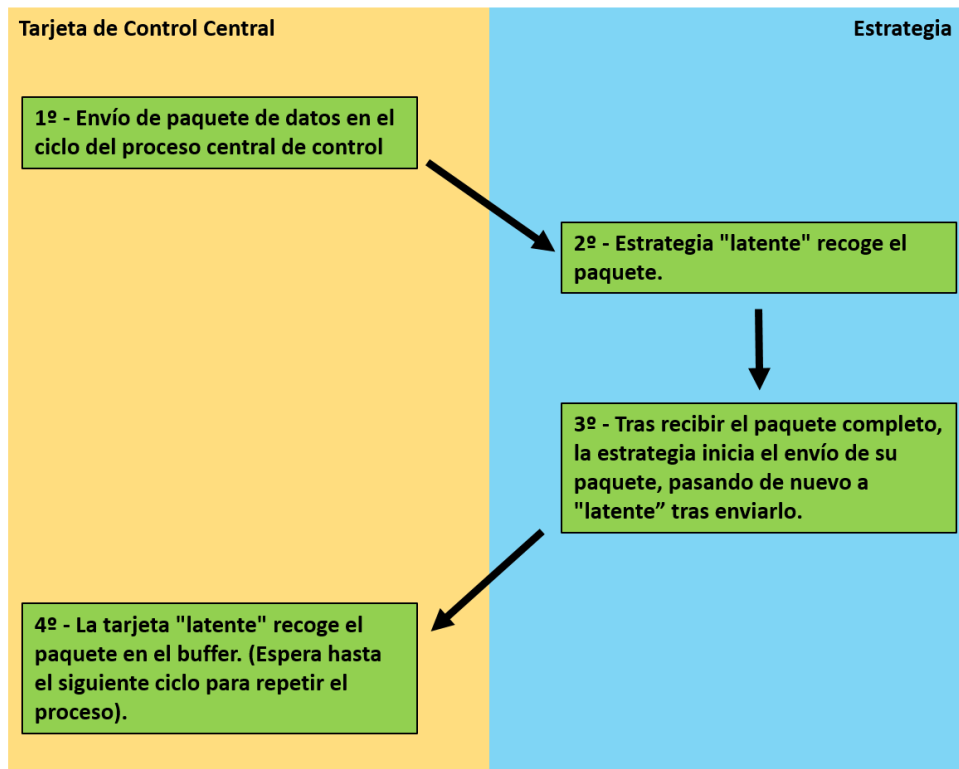


Figura 106. Protocolo de comunicación

- **Formato de la información**

El formato de la información o paquete es distinto para el caso de enviar desde la Mini-DK2 o el de recibido desde la RaspBerry Pi, pero nunca cambia. El formato para cada caso y desde el punto de vista de este proyecto es el siguiente:

Envío hacia la tarjeta de estrategia					
1º	2º	3º	4º	5º	6º
Pulsos encoder 1	Carácter “,”	Pulsos encoder 2	Carácter “,”	Nivel de batería de seguridad	Fin de paquete “\n” (salto de línea)

Tabla 12. Paquete de envío hacia la tarjeta de estrategia

Recepción de la tarjeta de estrategia								
1º	2º	3º	4º	5º	6º	7º	8º	9º
Signo de consigna velocidad rueda 1 (“+” o “-“)	Consigna velocidad rueda 1: centenas	Consigna velocidad rueda 1: decenas	Consigna velocidad rueda 1: unidades	Signo de consigna velocidad rueda 2 (“+” o “-“)	Consigna velocidad rueda 2: centenas	Consigna velocidad rueda 2: decenas	Consigna velocidad rueda 2: unidades	Fin de paquete “\r” (retorno de carro)

Tabla 13. Paquete de recepción de la tarjeta de estrategia

- Interpretación de la consigna de velocidad

Como se muestra en la Tabla 13 la consigna de velocidad de cada rueda se recibe dividida en centenas, decenas y unidades. Esto es interpretado correctamente al principio de cada ciclo del proceso central de control y traducido a la referencia de velocidad para el controlador de cada motor.

- Fin de paquete

Un aspecto importante en los dos casos es qué carácter indica el fin de cada paquete. Aunque cada paquete tiene siempre la misma estructura es necesario establecer y conocer esos caracteres (aunque no sea el mismo) para poder trabajar con las librerías de la UART de la LPC1768 y de ROS.

En las primeras pruebas de comunicación un error fue que cada parte enviaba como fin de paquete un carácter ASCII distinto, esperando que el del otro fuera igual. En el caso de la LPC1768 para cambiar el carácter esperado como fin de paquete recibido hay que modificar la función “void UART0_IRQHandler(void)” (Figura 107), que se encuentra dentro de “uart.c”. En este caso se establece a “13” que es el “retorno de carro” (en lenguaje C: “\r”).

```
void UART0_IRQHandler(void) {
    switch(LPC_UART0->IIR&0x0E) {
    case 0x04: /* RBR, Receiver Buffer Ready */
        *ptr_rx=LPC_UART0->RBR; /* lee el dato recibido y lo almacena */
        if (*ptr_rx==13) /* Caracter return --> Cadena completa
            {
                *ptr_rx=0; /* Añadimos el caracter null para tratar los datos recibidos como una cadena*/
                rx_completa = 1; /* rx completa */
                ptr_rx=buffer; /* puntero al inicio del buffer para nueva recepción */
            }
        break;

    case 0x02: /* THRE, Transmit Holding Register empty */
        if (*ptr_tx!=0) LPC_UART0->THR=*ptr_tx++; /* carga un nuevo dato para ser transmitido */
        else tx_completa=1;
        break;
    }
}
```

Figura 107. Modificación del fin del paquete a recibir

Por el contrario, las librerías de ROS utilizadas por la estrategia esperan el carácter ASCII “10” como fin de paquete que es el “salto de línea” (en lenguaje C: “\n”). Es por ello que se envía al final de cada paquete dirigido a la estrategia.

- **Tiempos**

El envío realizado por parte de la Mini-DK2 se lanza (de forma paralela) cada ciclo del proceso central de control, configurado a una frecuencia de 100 Hz (T=0,01 s.). Experimentalmente se comprueba que la estrategia puede asimilar perfectamente esta frecuencia de envío, dándole tiempo a calcular la posición actualizada del robot con los datos de odometría enviados en cada ciclo.

La Mini-DK2 realiza los envíos de forma concurrente al resto de procesos, siendo el tiempo que la CPU está ocupada en el envío despreciable frente al tiempo que tarda en enviar el paquete (pues es el envío es por interrupción). La recepción del paquete por parte de la Mini-DK2 se realiza de forma paralela almacenándose en un buffer, por lo que no interfiere en el funcionamiento normal del sistema.

La velocidad de envío de datos o baudrate se establece en 19200 baudios porque el envío de un máximo de 6 bytes a esa velocidad tarda 3,1 ms (siendo el período del ciclo del proceso central de control 10 ms), es decir, tarda un 31% del ciclo. Además, la estrategia devuelve un paquete de un máximo de 9 bytes tardando 4,65 ms a esa velocidad. En total, el tiempo empleado en la comunicación por cada ciclo son 7,71 ms,

el 77% del ciclo. Es por ello por lo que se elige esta velocidad, ya que si se estableciera otra menor (9600 baudios) la comunicación podría tardar más de un ciclo del proceso central.

5.4.4 Estructuración y prioridades de las interrupciones

La estructuración de toda la lógica del control central pasa por organizar todo el código tanto a nivel software como a nivel estético, facilitando la comprensión del mismo a cualquiera que desee trabajar con él. La organización software es crucial para el correcto funcionamiento de todo el sistema, centrándonos en las prioridades que se les asigna a las distintas interrupciones, y por ende, a los procesos que estas desencadenan. En este proyecto no se utilizan subprioridades (*PriorityGrouping=2*; 32 niveles de prioridad preemptive), por lo que procesos con prioridad mayor podrán desalojar procesos con menor prioridad en ejecución.

- **Estructuración**

Antes de asignar las prioridades a cada interrupción debemos analizarlas en su conjunto. Se tienen los siguientes procesos, por orden alfabético:

Proceso	Interrupción LPC_1768
Comunicación Serie (UART)	UART 0
Encoder 1 – Lectura odometría	EINT 1
Encoder 2 – Lectura odometría	EINT 2
Encoder 1 - Lectura velocidad	TIMER 1
Encoder 2 - Lectura velocidad	TIMER 0
Lectura de sensores y batería	TIMER 2
Proceso periódico central (PID, toma de decisiones y comunicación con estrategia)	SysTick

Tabla 14. Estructuración de los procesos /interrupciones

Observando estos procesos podemos comprender que la lectura de odometría tiene una importancia especial, ya que es crucial para el buen desarrollo de la estrategia, la cual debe ser lo más exacta posible y eliminar posibles fuentes de error. A su vez, es imprescindible que cuando esta lectura requiera de ser ejecutada no se vea cohibida por ningún otro proceso, ya que podría perderse información.

Por otra parte, se tiene la lectura de velocidad y el proceso periódico central, teniendo estos un gran peso dentro del proyecto, ya que de ellos depende que el robot desarrolle un correcto movimiento. En lo que al controlador respecta, se debe buscar que trabaje a tiempo real, tratando de que ningún otro proceso lo pise, pero a la vez interesa que utilice el valor más actual de la velocidad y de la información de odometría.

Finalmente se encuentra la comunicación serie (referida al proceso interno de la LPC_1768) y la lectura de sensores y batería, siendo este último un proceso que admite no trabajar a tiempo real pero si con una cierta periodicidad.

- **Estudio de tiempos**

Para dar una visión concreta de cuan rápidos o lentos son los distintos procesos entre ellos, debe estudiarse su duración (C), su periodo mínimo de ejecución (T) y el plazo límite de la finalización del proceso (D). En la siguiente tabla se detallan estos tiempos:

Proceso	C	T	D
Comunicación Serie – Interrupción (UART)	0,40 μ s	1,11 ms	1,11 ms
Encoder 1 – Lectura odometría (EINT 1)	0,27 μ s	0,65 ms	100 μ s
Encoder 2 – Lectura odometría (EINT 2)	0,27 μ s	0,65 ms	100 μ s
Encoder 1 - Lectura velocidad (TIMER 1)	3,86 μ s	0,65 ms	0,65 ms
Encoder 2 - Lectura velocidad (TIMER 0)	3,86 μ s	0,65 ms	0,65 ms
Lectura de sensores y batería (TIMER 2)	11,9 μ s	100 ms	10 ms
Proceso periódico central (Systick)	17,5 μ s	10 ms	10 ms

Tabla 15. Estudio de tiempos de las interrupciones

Todos los procesos expuestos, excepto el periódico central, son cortos y de frecuencia mayor, por lo que es muy difícil que se molesten entre ellos o al proceso periódico central. Así, con esta observación global, pasamos a determinar las prioridades finales de cada proceso.

- **Prioridades**

Visto que la lectura de la odometría es el proceso más importante se le da la mayor prioridad. Esto es así ya que la pérdida de información prolongada en el tiempo en la lectura de la odometría podría afectar a la autonomía del robot y a su estrategia, resultado de un error acumulado.

El carácter prioritario de esta lectura es debido al modo en el que se realiza. Cada encoder genera dos señales de lectura: una utilizada para disparar el proceso de lectura de la odometría y la otra es la propiamente leída. En función del estado de la segunda (nivel alto “1” o nivel bajo “0”) se obtiene la información del sentido de giro de la rueda del robot. Pero estos niveles cambian constantemente, siendo válida la información solo durante un corto período de tiempo.

Para el caso del Encoder v.1 este hecho es de gran relevancia, ya que el desfase entre sus dos señales de lectura es pequeño y por tanto el tiempo en el que la información válida está disponible (Figura 108). Para el Encoder v.2 este tiempo es mayor, ya que sus señales en cuadratura ponen a disposición esta información durante la mitad del período de la señal de lectura (Figura 108).

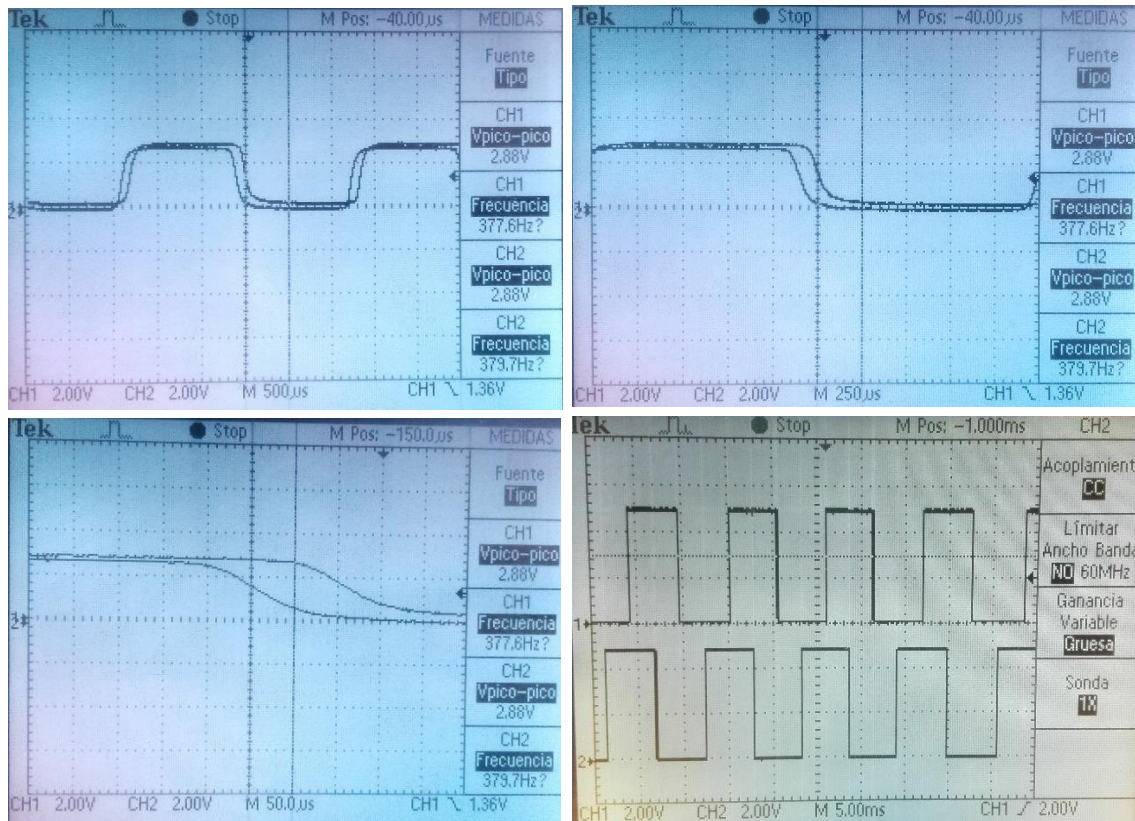


Figura 108. Arriba/Izquierda-Señal en cuadratura Encoder v.1. Arriba/derecha y abajo/izquierda-Detalles de cuadratura Encoder v.1 (100 µs entre señales). Abajo/derecha-Señal en cuadratura Encoder v.2

Después se encuentra el proceso periódico central y la lectura de velocidad, con importancia similar. La diferencia entre ellos se encuentra en la duración de cada uno y con qué frecuencia se ejecutan. El proceso central se ejecuta a una frecuencia de 100 Hz y con una duración de 17,5 µs (Figura 109), mientras que la lectura de velocidad de un encoder dura muy poco en comparación e interrumpe con una frecuencia máxima de 1,13 kHz para el Encoder v.1 y 1,54 kHz para el Encoder v.2 (ecuación (73)).

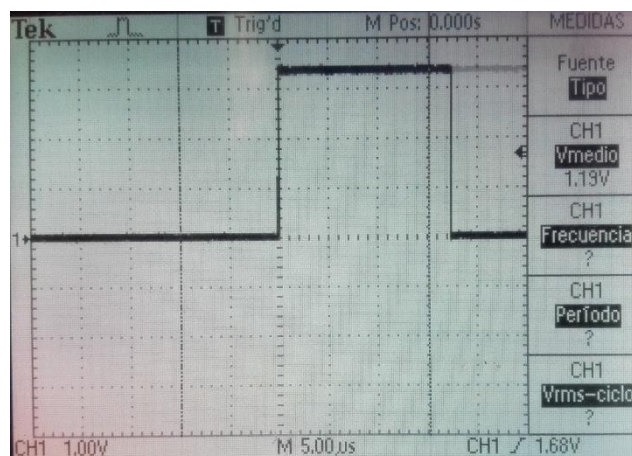


Figura 109. Duración de 17,5 µs del Systick

$$Frec_{m\acute{a}x} = \frac{1}{T_{m\acute{i}n}} = \frac{2 \cdot Vel_{m\acute{a}x} \cdot n}{Per\acute{i}m.} \quad (73)$$

donde:

Perím. = *perímetro del disco de lectura*

Vel_{máx} = *velocidad máxima desarrollable (ver Modelado de planta y simulación)*

n = *número de flancos del disco de lectura multiplicado por la aceleradora*

Se decide darles la misma prioridad, ya que y poniéndonos en el peor caso, si la lectura de velocidad fuera llamada mientras el proceso periódico central acaba de comenzar, solamente se perdería la última lectura de velocidad (dados los tiempos y frecuencias anteriormente descritas). Además, podríamos tener disponible esa lectura “perdida” en el siguiente ciclo de control del movimiento, si no es refrescada antes por otra más nueva, ya que se ejecutaría al acabar el proceso periódico central.

Por el contrario, si la lectura de velocidad está en ejecución y el proceso central requiere de ejecutarse, la lectura dura tan poco que podríamos seguir hablando de tiempo real para el proceso central ya que no tendría que esperar básicamente nada. Este mismo hecho se puede aplicar también a la lectura de la odometría.

Para finalizar, con las menores prioridades se tiene a la comunicación serie (UART) y la lectura de sensores y batería. La comunicación serie al no manejar grandes cantidades de datos y al ser el proceso central el que lee el buffer de esta y envía nuevos datos a la estrategia, dispone de 10 ms para realizar toda la comunicación, siendo más que suficientes. La lectura de sensores y batería, con la menor prioridad de todas, no necesita ejecutarse a frecuencias necesariamente altas pudiendo ser desalojada si otra lo requiriera. Lo más crítico de este proceso sería la lectura de los sensores de proximidad GP2Y, pero la velocidad de movimiento del robot es significativamente más lenta que la del procesamiento de la LPC_1768, por lo que desde el punto de vista del juego de la competición esta lectura se realiza instantáneamente.

Así obtenemos la organización software de todas las interrupciones:

Proceso	Interrupción LPC_1768	Prioridad
Encoder 1 - Lectura odom.	EINT 1	0
Encoder 2 - Lectura odom.	EINT 2	0
Encoder 1 - Lectura vel.	TIMER 1	1
Encoder 2 - Lectura vel.	TIMER 0	1
Proceso central	SysTick	1
Comunicación serie (UART)	UART 0	2
Lectura sensores y batería	TIMER 2	3

Tabla 16. Prioridades de las interrupciones

5.4.5 Trayectorias y odometría

En este proyecto se estudia la forma de las trayectorias, aunque como se comentaba anteriormente, su desarrollo e implementación se lleva a cabo por la tarjeta de estrategia a modo de elevar esas funcionalidades a niveles superiores.

Por otro lado, la odometría es estudiada y sí desarrollada en este proyecto, pero su implementación también se eleva al nivel superior de la estrategia, aliviando a la Mini-DK2 del cálculo de la odometría. Por tanto es la única parte software desarrollada que no se implementa en la Mini-DK2, si no en la RaspBerry Pi utilizando ROS. Es usada así por la estrategia como una herramienta, utilizando la información de los encoders enviada desde la Mini-DK2.

5.4.5.1 Lógica de las trayectorias

Las trayectorias, como ya se veía en el apartado *Desarrollo Mecánico*, siguen una forma trapezoidal (Figura 110). Estas trayectorias son enviadas como consigna de velocidad desde la estrategia. Estas consignas están discretizadas, siendo el periodo de muestreo el periodo del proceso central de control ($T=0,01$ s.).

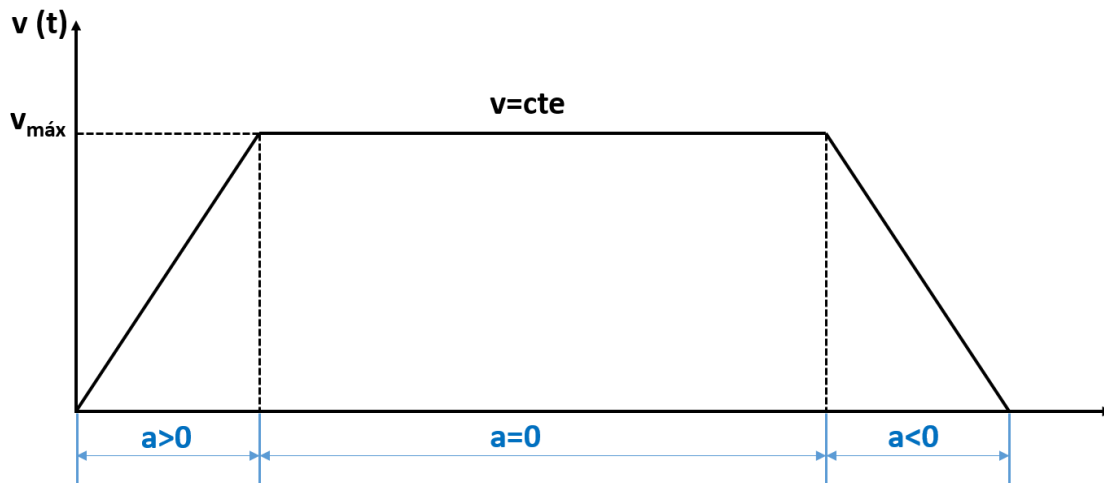


Figura 110. Lógica de trayectorias. Perfil trapezoidal

Las trayectorias que lleva a cabo el robot son de dos tipos:

- Líneas rectas

Las líneas rectas siguen el perfil trapezoidal tal como es: un tiempo inicial de aceleración, otro de velocidad constante y otro de frenada. Las consignas de cada rueda son iguales y de mismo sentido.

En estas líneas rectas el robot puede alcanzar una velocidad “rápida” y por lo tanto adquirir una cierta inercia. Es por ello que la frenada se realiza de forma progresiva.

- Giros en el sitio

En los giros en el sitio se busca la precisión, por lo que la velocidad es menor que en el caso anterior. Las consignas de cada rueda son iguales en velocidad pero contrarias en sentido.

Al contrario que en las líneas rectas, estos giros no siguen totalmente el perfil trapezoidal, eliminando la frenada progresiva. Esto es así ya que la velocidad es baja (y por tanto la inercia también), siendo admisible una parada en seco.

La razón a este hecho es la precisión buscada, pues llegar al ángulo de giro deseado adquiere una importancia especial. Experimentalmente se comprueba que en el tiempo de frenada “brusca” a una velocidad de 50 mm/s el robot gira alrededor de 5°, por

lo que la estrategia (siempre realimentando el ángulo actual desde la odometría) manda la orden de parada 5° grados antes de llegar al objetivo. En caso de que no se consiga el ángulo deseado, la estrategia recalcula otro giro en el sitio de los grados de diferencia al objetivo.

5.4.5.2 Odometría

La odometría es el estudio de la posición y orientación en el espacio de un sistema móvil con ruedas respecto a un origen (x_0, y_0, θ_0) . Para realizar este estudio se utiliza la información de los encoders pasivos; unas ruedas con sensores para medir la rotación de las mismas. El eje de estos encoders se encuentra horizontalmente alineado con el de las ruedas motrices.

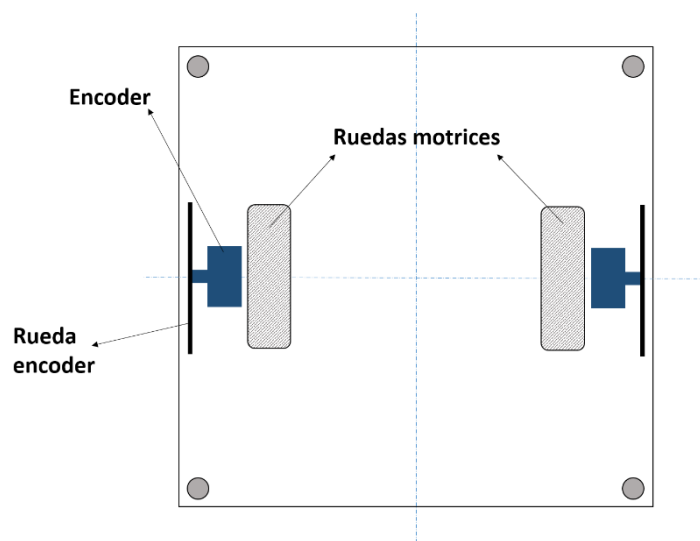


Figura 111. Planta robot. Encoders y ruedas

En este proyecto se crea un modelo matemático propio para hacer los cálculos de la odometría. Este modelo es un conjunto de ecuaciones que analizan cada situación del movimiento seguido por el robot. Así, se categoriza cada movimiento en función de su naturaleza (línea recta, giro en el sitio, giro con radio de curvatura,...) aplicándose unas ecuaciones u otras en consecuencia. Con esto se intenta evitar al máximo el error por aproximación, aplicando en cada situación las ecuaciones que más se ajustan a la realidad.

- **Estudio inicial**

El estudio inicial se realiza sobre el trabajo desarrollado por Javier Baliñas [2], ya que desarrolla la odometría para un robot de tracción diferencial, siendo este el caso. En él se encuentra una buena base por donde comenzar a hacer un estudio propio, pero también se encuentran muchas cosas a mejorar.

- Base

Como base por la que comenzar se encuentran dos aspectos principales: la obtención del ángulo de orientación en cada interacción y la acumulación progresiva de posición y ángulo.

El ángulo de orientación desarrollado en cada interacción del ciclo de muestreo está directamente relacionado con la diferencia entre la distancia recorrida por cada encoder en esa interacción:

$$\theta_k = (N_D - N_I) \cdot C_\theta \quad (74)$$

donde:

θ_k = ángulo desarrollado

N_D = pulsos del encoder derecho

N_I = pulsos del encoder izquierdo

C_θ = constante para convertir pulsos en ángulos

En el apartado *Desarrollo Propio* se desarrolla esta constante de conversión, ya que difiere de la de este estudio previo.

En lo concerniente a la acumulación de la posición del robot en el espacio se encuentra interesante el planteamiento que se realiza: calcular la distancia en línea recta desde el punto inicial al final (L), calcular el ángulo de ese recorrido recto ficticio (θ_{med}) y sumar a la posición total las proyecciones en x e y de la nueva interacción.

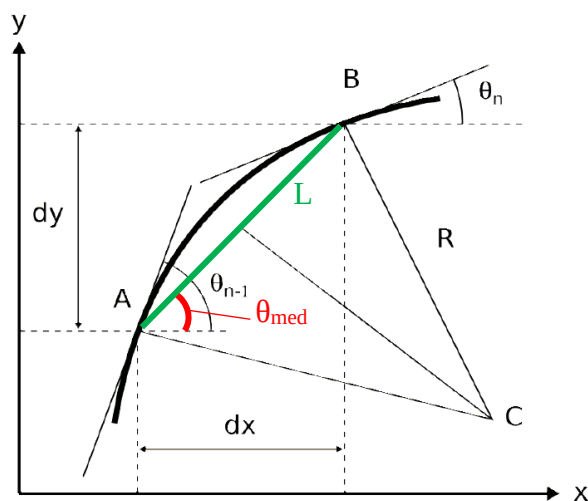


Figura 112. Cálculo de posición final método Javier Baliñas

- Aspectos a mejorar: aproximaciones

En el trabajo realizado por Javier Baliñas se expone que el cálculo de la distancia real recorrida por el robot se puede aproximar por movimiento rectos [2]. Esta aproximación está bien si se tiene en cuenta un muestreo muy rápido, pero aun así se comete un cierto error en movimientos curvos.

Es por ello que se decide calcular la distancia real recorrida en cada caso, ya que se dispone en todo momento de la información necesaria y la complejidad de cálculo no difiere de la aproximación.

$$d = \frac{N_D + N_I}{2} \quad (74)$$

donde:

d = distancia recorrida por el robot

N_D = pulsos del encoder derecho

N_I = pulsos del encoder izquierdo

- **Desarrollo propio**

El desarrollo propio parte de la base anterior como se comentaba. En primera instancia se debe analizar la distribución de los elementos en el robot y los parámetros clave para la odometría del mismo.

El robot que se desarrolla en este proyecto tiene tracción diferencial (dos ruedas) las cuales están centradas respecto a la planta del robot. Los encoders, también dos ruedas pero pasivas, se encuentran igualmente centrados pero desplazados hacia el exterior del robot. Este hecho hace que la odometría utilice solo la información de los encoders ya que siguen el mismo movimiento que las ruedas motrices pero “aumentado”.

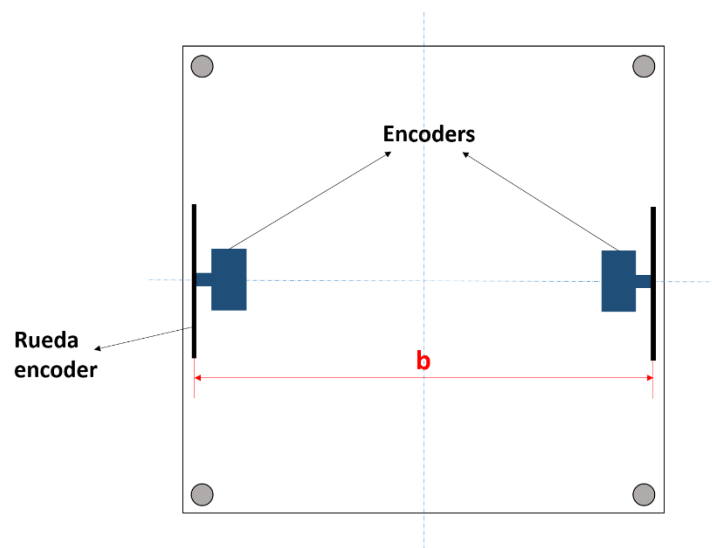


Figura 113. Planta robot. Encoders y distancia entre ellos

Ahora bien, los encoders recogen la información de la distancia recorrida en pulsos, provenientes de sus discos ranurados de lectura. Es por ello que la distancia recorrida por la rueda del encoder se puede expresar de la siguiente manera:

$$d_{encoder} = \frac{\text{Pulsos} \cdot \text{Perímetro}}{N} = \frac{\text{Pulsos} \cdot D \cdot \pi}{N} \quad (75)$$

donde:

Pulsos = pulsos recogidos por el encoder

N = número de pulsos por cada vuelta completa de la rueda del encoder

D = diámetro de la rueda del encoder

Se obtiene así la constante de conversión pulsos-distancia:

$$C_d = \frac{d_{encoder}}{\text{Pulsos}} = \frac{D \cdot \pi}{N} \quad (76)$$

Con esta conversión pulsos-distancia ya se puede comenzar con la lógica matemática de la odometría. Esta se subdivide en distintos casos según el tipo de movimiento realizado por el robot, a fin de reducir al mínimo posible las aproximaciones. Todos estos casos tienen como parte común la obtención del ángulo final desarrollado en cada interacción del ciclo de muestreo. Este ángulo se obtiene tal y como se mostraba en *Estudio Inicial*:

$$\theta_r = \frac{(N_D - N_I) \cdot C_d}{b} \quad (77)$$

donde:

θ_r = variación del ángulo de orientación

N_D = pulsos del encoder derecho

N_I = pulsos del encoder izquierdo

C_d = constante de conversión pulsos-distancia

b = distancia entre ruedas de encoder

Se obtiene así que la constante de conversión pulsos-ángulo es:

$$C_\theta = \frac{\theta_r}{(N_D - N_I)} = \frac{C_d}{b} = \frac{D \cdot \pi}{N \cdot b} \quad (78)$$

A modo de comprobar la veracidad de esta fórmula se realiza su demostración:

$$\boxed{\text{Arco}_{\text{circunf}} = \text{Ángulo} \cdot \text{Radio}_{\text{circunf}}}$$

$$\left. \begin{array}{l} L_1 = \theta_r \cdot r_1 \\ L_2 = \theta_r \cdot r_2 \end{array} \right\} L_2 - L_1 = \theta_r \cdot (r_2 - r_1)$$

$$\theta_r = \frac{(L_2 - L_1)}{(r_2 - r_1)} = \frac{C_d \cdot (N_2 - N_1)}{b}$$

$$\theta_r = C_\theta \cdot (N_2 - N_1)$$

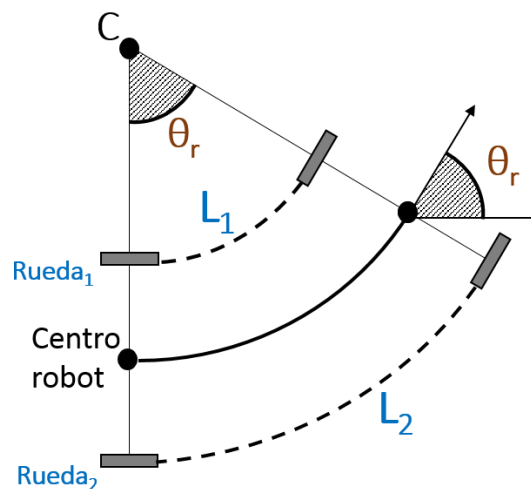


Figura 114. Demostración θ_r

donde:

L_1 = distancia recorrida por encoder 1

L_2 = distancia recorrida por encoder 2

r_1 = radio curvatura trayectoria encoder 1

r_2 = radio curvatura trayectoria encoder 2

N_1 = pulsos del encoder 1

N_2 = pulsos del encoder 2

θ_r = variación del ángulo de orientación

Tras la obtención del ángulo de orientación de cada interacción " θ_r " se pasa a calcular la variación total de la posición del robot, dividiéndose en los siguientes casos:

Movimiento recto ($N_D = N_I$).

En el movimiento recto la distancia recorrida por el centro del robot es la misma que la recorrida por ambas ruedas.

$$L = N_D \cdot C_d = N_I \cdot C_d \quad (79)$$

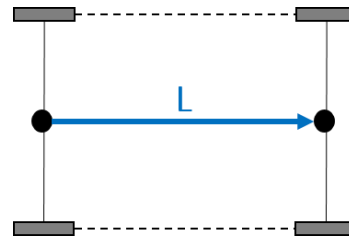


Figura 115. Movimiento recto

Giro con una rueda parada ($N_D = 0, N_I > 0$ / $N_D > 0, N_I = 0$).

En un giro con una de las ruedas paradas, la distancia recorrida L por el centro del robot es la cuerda del arco de circunferencia.

$$Cuerda_{arco} = 2 \cdot \text{sen} \left(\frac{\theta}{2} \right) \cdot \text{radio}$$

$$L = 2 \cdot \text{sen} \left(\frac{\theta_r}{2} \right) \cdot \frac{b}{2} \quad (80)$$

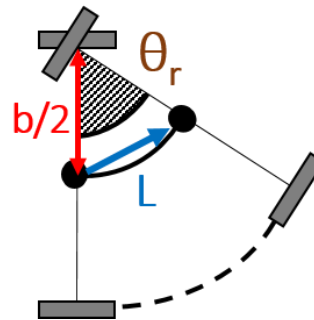


Figura 11699. Giro con una rueda parada. Variante ND > 0, NI=0

Giro aproximadamente en el sitio ($N_D < 0, N_I > 0$ / $N_D > 0, N_I < 0$).

En este caso se realiza una aproximación (la única en este cálculo odométrico). Se supone un muestreo rápido en la lectura de los encoders, pudiendo aproximar un giro casi en el sitio por muchos giros con una rueda parada. Se toma como esta rueda parada la que su N asociada tenga el menor valor absoluto. Por lo que:

$$L = 2 \cdot \text{sen} \left(\frac{\theta_r}{2} \right) \cdot \frac{b}{2} \quad (80)$$

Caso particular. Giro en el sitio ($N_D = -N_I$).

En este caso particular no hay aproximación. Al ser las dos N iguales pero de signo contrario, el centro del robot se encuentra situado exactamente en el mismo sitio. Tenemos así que:

$$L = 0 \quad (81)$$

Curva hacia delante ($N_D > 0, N_I > 0$).

Para el cálculo de la distancia en este caso se debe analizar que radio de curvatura entre r_1 y r_2 es el más pequeño. A este se le suma $\frac{b}{2}$, convirtiéndose otra vez en el cálculo de la cuerda del arco de circunferencia, pero en este caso con el radio aumentado.

$$L_1 = N_I \cdot C_d \quad (82)$$

$$L_2 = N_D \cdot C_d \quad (83)$$

- Si $N_I < N_D$ (giro "positivo"):

$$(84) \quad r_1 = \frac{L_1}{\theta_r}$$

$$(85) \quad L = 2 \cdot \text{sen} \left(\frac{\theta_r}{2} \right) \cdot \left(r_1 + \frac{b}{2} \right)$$

- Si $N_I > N_D$ (giro "negativo"):

$$(86) \quad r_2 = -\frac{L_2}{\theta_r}$$

$$(87) \quad L = 2 \cdot \text{sen} \left(\frac{\theta_r}{2} \right) \cdot \left(r_2 + \frac{b}{2} \right)$$

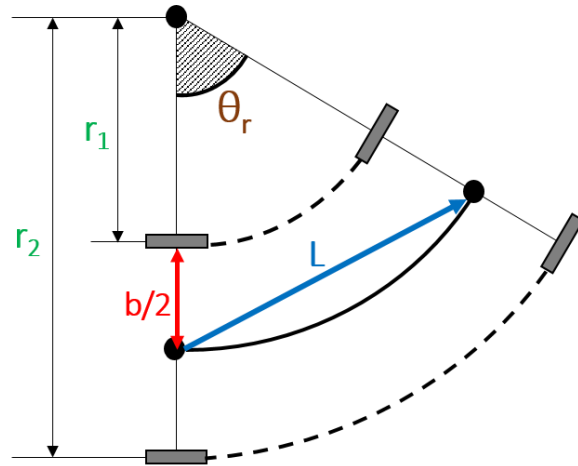


Figura 117. Curva hacia delante. Variante de $N_I < N_D$

Curva hacia atrás ($N_D < 0, N_I < 0$).

Este caso es prácticamente igual al anterior, con la salvedad de que al ir hacia atrás trabajamos con valores negativos de las N. Esto hace que para determinar bien si el giro es positivo o negativo, y qué radio es el menor de entre r_1 y r_2 , debemos analizar el valor absoluto de las N.

$$L_1 = N_I \cdot C_d \quad (82)$$

$$L_2 = N_D \cdot C_d \quad (83)$$

- Si $|N_I| < |N_D|$ (giro "negativo"):

$$(84) \quad r_1 = \frac{L_1}{\theta_r}$$

$$(85) \quad L = 2 \cdot \text{sen} \left(\frac{\theta_r}{2} \right) \cdot \left(r_1 + \frac{b}{2} \right)$$

- Si $|N_I| > |N_D|$ (giro "positivo"):

$$(86) \quad r_2 = -\frac{L_2}{\theta_r}$$

$$(87) \quad L = 2 \cdot \text{sen} \left(\frac{\theta_r}{2} \right) \cdot \left(r_2 + \frac{b}{2} \right)$$

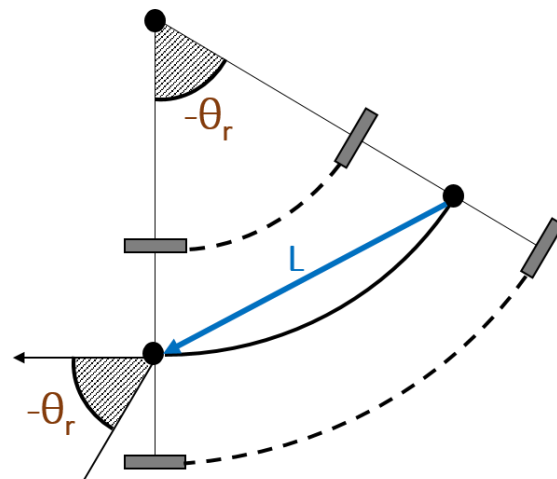


Figura 118. Curva hacia atrás. Variante de $|N_I| < |N_D|$ ángulo negativo

Tras el cálculo de uno de estos casos se obtiene la posición final combinando la distancia recta "ficticia" recorrida "L" y el ángulo " θ_{med} " (ecuación (90)), sumando las proyecciones en x e y a la posición anterior (como se veía en la Figura 112):

$$\left. \begin{aligned} d_x &= |L| \cdot \cos(\theta_{med}) \\ d_y &= |L| \cdot \sin(\theta_{med}) \end{aligned} \right\} \begin{aligned} x_k &= x_{k-1} + d_x \\ y_k &= y_{k-1} + d_y \end{aligned} \quad (88)$$

donde:

- x_k = coordenada en x de la posición final
- y_k = coordenada en y de la posición final
- x_{k-1} = coordenada en x de la posición anterior
- y_{k-1} = coordenada en y de la posición anterior
- d_x = proyección en x del avance en la interacción actual
- d_y = proyección en y del avance en la interacción actual

El ángulo θ_{med} se obtiene de la siguiente manera:

$$\theta_{med} = \frac{\theta_r}{2} + \theta_{k-1} \quad (90)$$

y su demostración es la que sigue:

$$\begin{aligned} 180^\circ - \theta &= 2\alpha \\ 180^\circ - \theta &= 2 \cdot (90^\circ - \beta) \\ 180^\circ - \theta &= 180^\circ - 2\beta \\ -\theta &= -2\beta \rightarrow \beta = \frac{\theta}{2} \end{aligned}$$

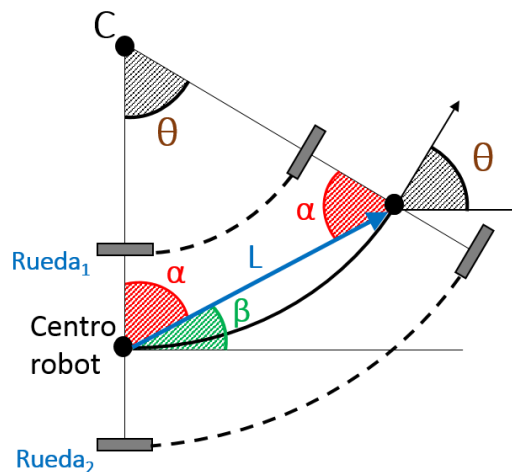


Figura 119. Demostración θ_{med}

En la ecuación (90) " θ_{k-1} " representa al ángulo de orientación del robot en la interacción anterior. Este se suma, puesto que el ángulo de cada interacción se acumula al total anterior.

Nótese que la ecuación (90) es válida para todos los casos, ya que es genérica. Para los movimientos rectos θ_{med} es directamente θ_{k-1} ya que $\theta_r = 0$, manteniéndose la validez del resto de ecuaciones.

Obtenemos así todas las coordenadas de la posición espacial del robot:

$$P_k = (x_k, y_k, (\theta_r + \theta_{k-1})) = (x_k, y_k, \theta_k) \quad (91)$$

• **Simulación**

La simulación de los cálculos de la odometría se realiza mediante Matlab, ya que podemos representar gráficamente el recorrido del robot. Se prepara un script el cual realiza los siguientes pasos:

- Dibuja el campo de juego y las zonas por las que el robot no puede moverse.
- Lee los datos de un (tanto velocidades como odometría) de una prueba realizada.

- Inicializa las variables para el cálculo de la odometría y se fijan los parámetros D (diámetro de la rueda del encoder), N (pulsos por vuelta de esta rueda) y b (distancia entre ruedas de encoders). También se fija la posición inicial y el ángulo de orientación inicial que en la competición es normalmente $\frac{\pi}{2}$ respecto a los ejes de referencia.
- Realiza todos los cálculos de la lógica desarrollada para la obtención de la trayectoria seguida. Calcula todos los puntos intermedios de cada interacción creando un array con todos ellos. Además crea un array separado con la información de la orientación en cada interacción.
- Se representa la trayectoria sobre el campo de juego inicialmente dibujado.
- A modo de estudio se crea una referencia virtual para representar los perfiles de velocidades de cada rueda. También se representa la variación de la orientación.

Prueba de línea recta – Encoder v.2

Esta prueba es la misma que la realizada para la verificación que se realizó anteriormente del modelado del controlador. El resultado obtenido es el que sigue:

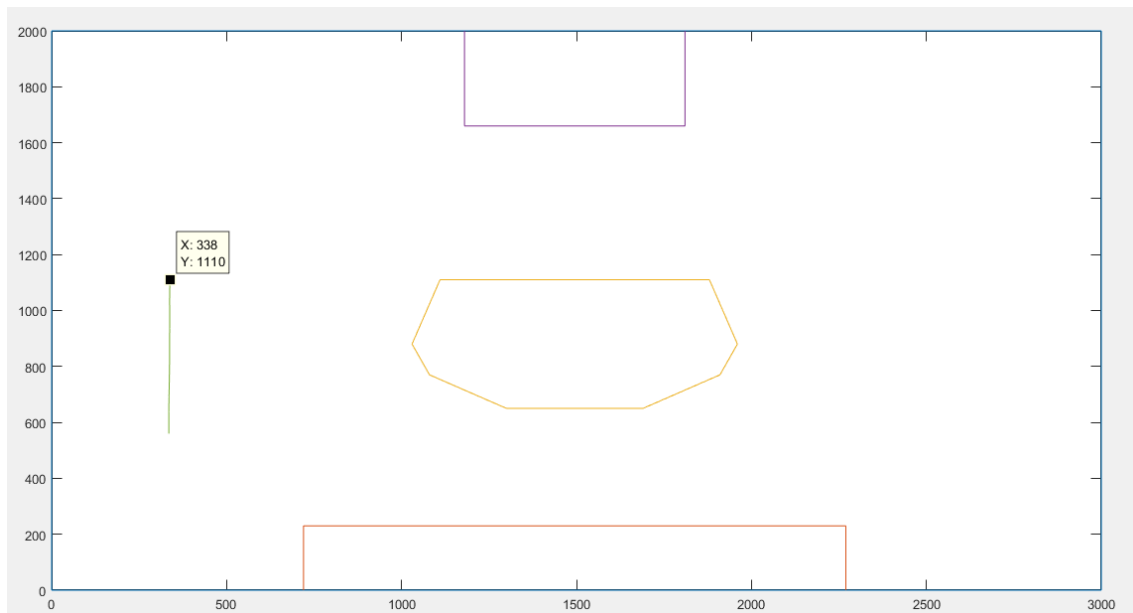


Figura 120. Trayectoria en prueba línea recta

En verde se puede ver la trayectoria seguida por el robot, mostrándose la posición final. La orientación inicial de esa recta fue de $\frac{\pi}{2}$ y la posición inicial (335mm, 560mm). Se observa que la desviación en x de la recta es de 3 mm, la cual es imperceptible a simple vista. Esto nos da idea de la precisión conseguida con todo el sistema encoders-odometría.

La longitud de la recta es de 550mm (1110-560), midiéndose in situ y comprobando que la medida real no llegaba a diferir en más de 1mm. Esto también nos da idea de su exactitud, ya que incluso la medida a mano de la distancia recorrida puede introducir error.

La variación en el ángulo de orientación nos da idea de la rectitud del movimiento y del trabajo desarrollado por el controlador de cada motor.

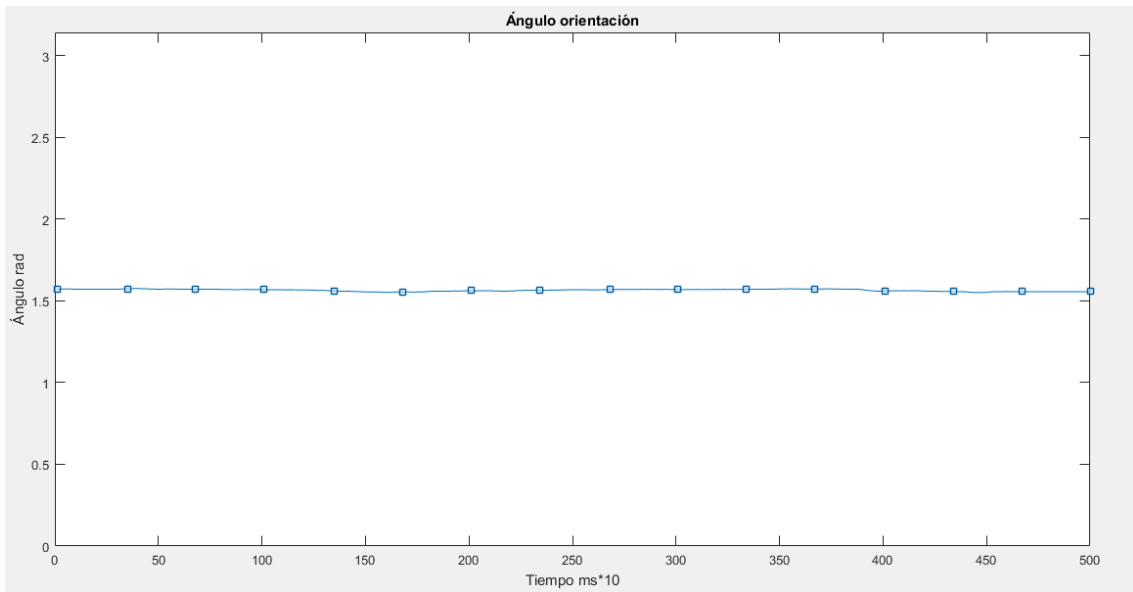


Figura 121. Perfil de orientación en prueba línea recta

Se representa la variación entre 0 y π , viéndose que prácticamente el ángulo no varía respecto al inicio ($\frac{\pi}{2} \approx 1.5708$). Finalmente el perfil de velocidad de cada rueda motriz es el que veíamos en el apartado *Modelado, ajuste y simulación del Controlador*.

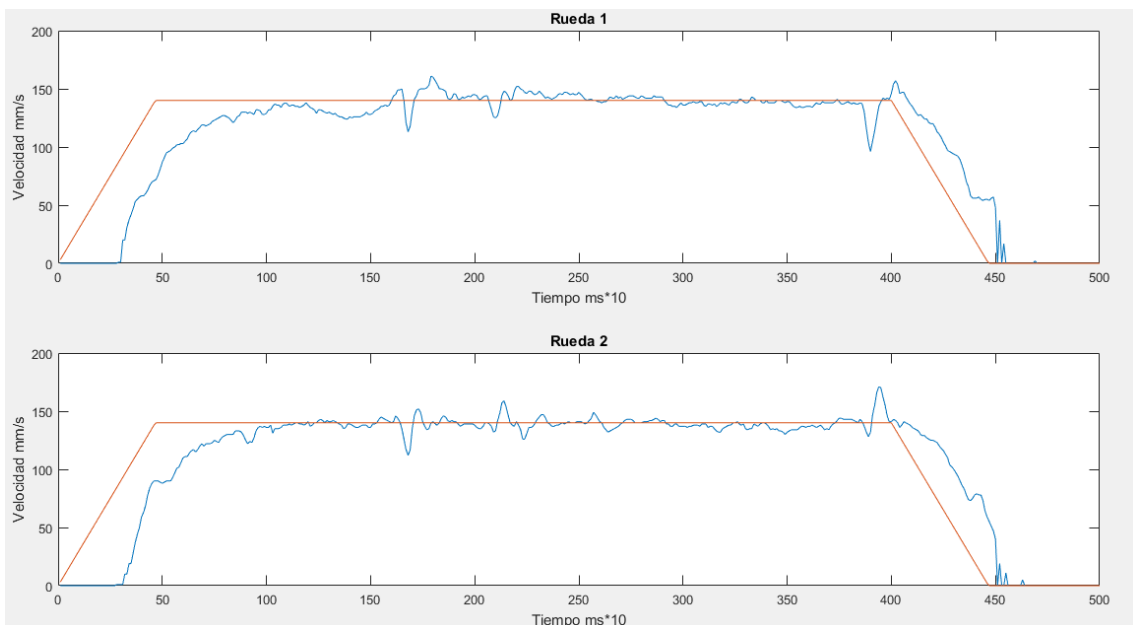


Figura 122. Perfil de velocidades y referencias en prueba línea recta

Prueba de giro libre – Encoder v.1

Esta prueba se realiza en etapas tempranas del proyecto, analizando el funcionamiento de los Encoder v.1 en el prototipo inicial. Sin controlador se envió una consigna igual para ambos motores, obteniéndose una curva por resultado.

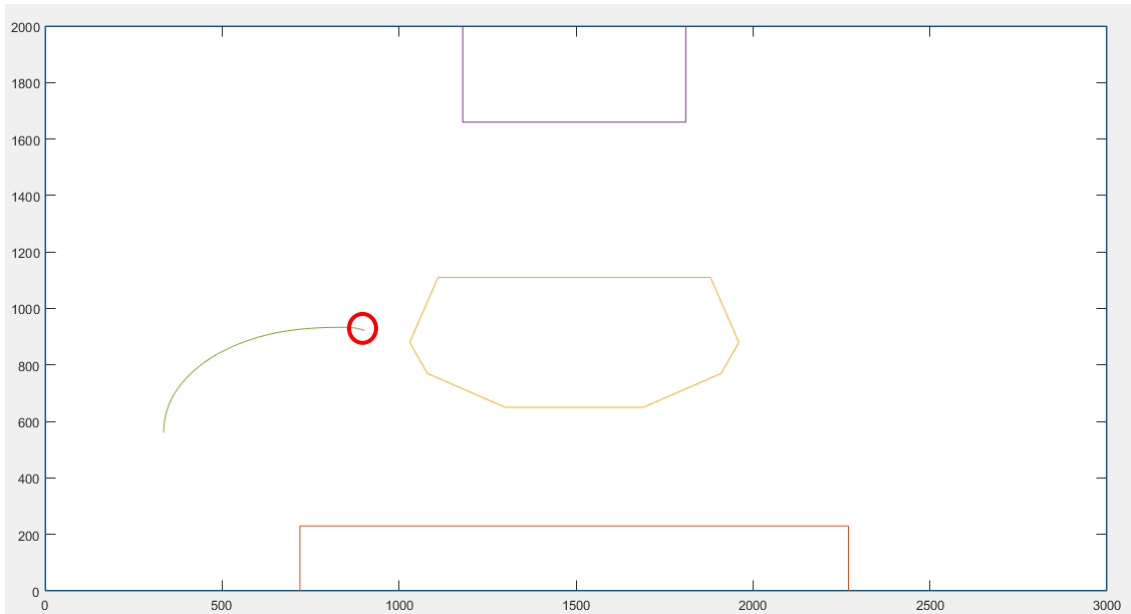


Figura 123. Trayectoria en giro libre

Se puede observar como al final del recorrido (círculo rojo) la orientación del robot toma valores negativos. En la realidad se comprueba que efectivamente en el giro el robot acaba mirando un poco hacia abajo. Se comprueba así que la lógica de la odometría trabaja correctamente con los ángulos negativos.

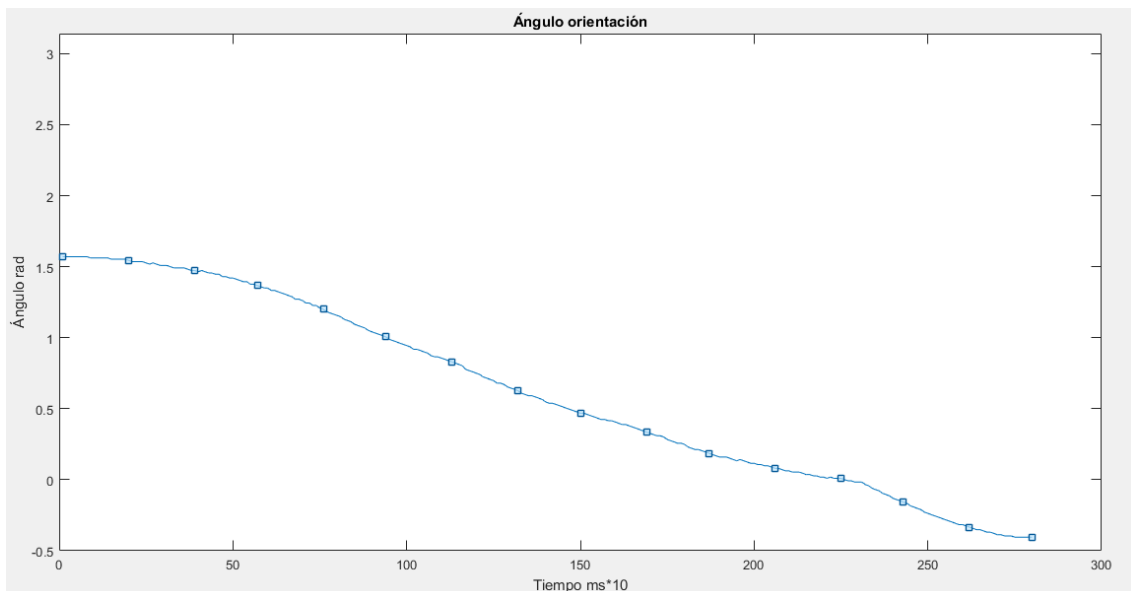


Figura 124. Perfil de la orientación en giro libre

En los perfiles de velocidades se observa como la rueda 2 (derecha) tiene velocidades menores y por tanto menor recorrido que la rueda 1, concordando con el giro a derechas de la trayectoria.

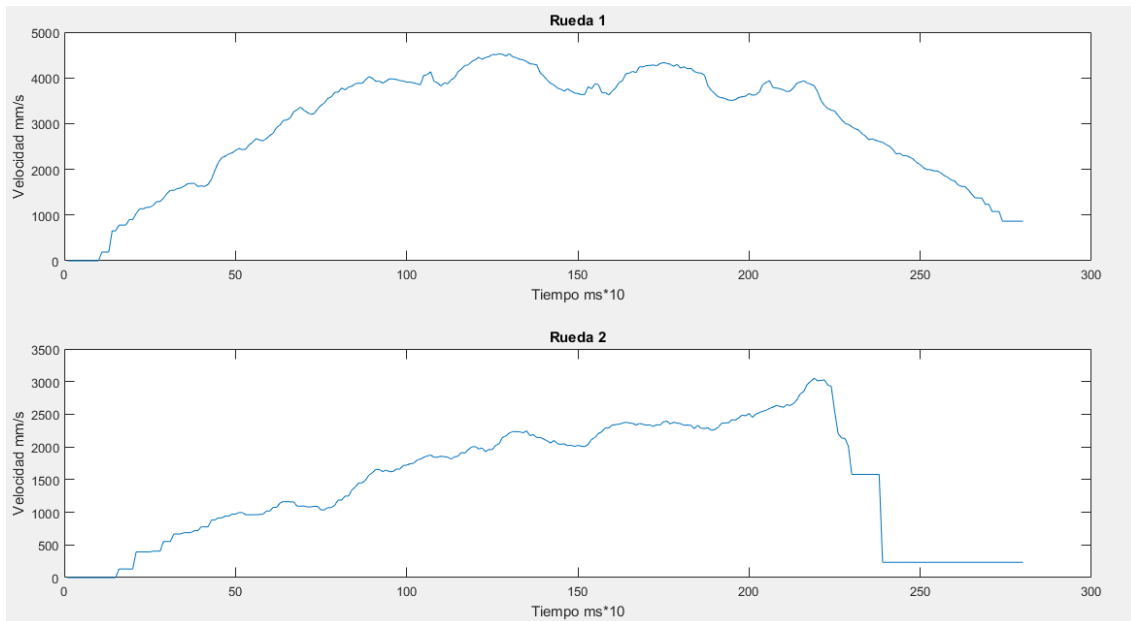


Figura 125. Perfil velocidades de ruedas en giro libre

Comparación de la odometría de Javier Baliñas

Se realiza una comparación a modo de ver las diferencias y mejoras de la lógica desarrollada contra la lógica desarrollada por Javier Baliñas. En esta comparación hay dos puntos clave:

- La constante de conversión a ángulos se halla con la mitad de b :

$$C_{\theta} = \frac{\theta_r}{(N_D - N_I)} = \frac{C_d}{\frac{b}{2}} = \frac{Cd \cdot 2}{b} \quad (92)$$

- La aproximación vista anteriormente de la distancia recorrida:

$$L = \frac{N_D + N_I}{2} \quad (93)$$

El resultado obtenido es el mostrado en la Figura 126.

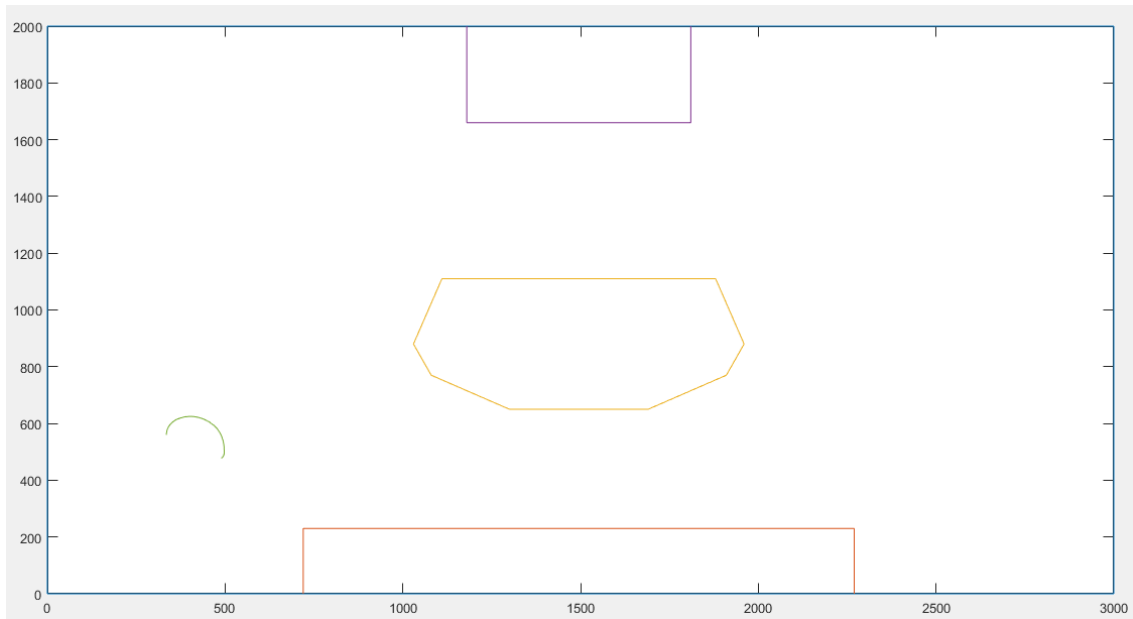


Figura 126. Comparación 1 de odometría Javier Baliñas

Se observa que la trayectoria representada difiere completamente de los resultados obtenidos anteriormente y por tanto de la realidad. Esto se achaca a una posible errata en la constante de conversión a ángulos, ya que no tiene sentido que se use $\frac{b}{2}$ en vez de b .

Se cambia esta constante para así comparar solamente la diferencia producida por la aproximación en la distancia recorrida en cada interacción. El resultado que se obtiene es el siguiente:

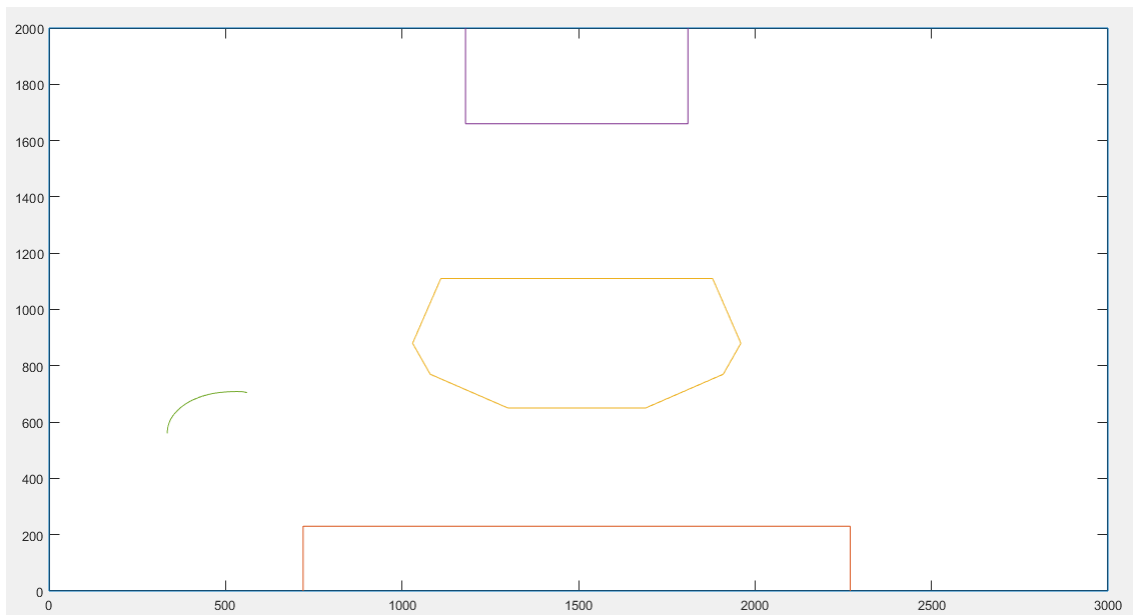


Figura 127. Comparación 2 de odometría Javier Baliñas

El resultado que se obtiene difiere de la realidad, aunque la trayectoria tiene la misma forma que la real pero de dimensiones reducidas. El error esperado por la aproximación que Baliñas realiza en su trabajo no debería tan grande, por lo que se concluye que puede tratarse de otra errata.

5.5 Conclusiones

En todo el recorrido de este proyecto, el autor de este trabajo ha experimentado lo que supone embarcarse en un proyecto conjunto; un proyecto con un objetivo final y un equipo para cumplirlo. Sin restar importancia al ámbito técnico, el aspecto humano ha primado desde inicio a fin, siendo un factor clave en el éxito y en el aprendizaje de lo que significa un proyecto real.

El autor de este trabajo se ha enfrentado a la realidad de tener que organizar (junto al resto del equipo) al propio equipo de trabajo. En un principio el equipo dividió el trabajo por partes, surgiendo así este proyecto. Esta idea inicialmente funcionó muy bien, pero según todo el proyecto del robot avanzó, esa idea necesitaba de flexibilidad. Al plantearse cada parte del robot como un TFG por separado hubo momentos en los que miembros del equipo no tenían carga de trabajo, mientras que otros la tenían en exceso. Hacia el final de todo el proyecto conjunto este hecho cambió, y la implicación directa de todos en todas las partes ayudó a acelerar considerablemente la construcción del robot.

En lo personal, el autor de este trabajo ha lidiado también con la organización del proyecto individual. En un inicio se le dio demasiado peso al cálculo teórico de distintas partes del proyecto, dejando a un lado la creación de prototipos y las pruebas en general. Esto a largo plazo hizo que el proyecto se frenara, ya que es más importante la creación de prototipos intermedios funcionales que el cálculo de parámetros finales. Muchos de esos cálculos no fueron usados a lo largo del proyecto, comprendiéndose que primero ha de tenerse una idea aproximada del proyecto (prototipos) y después afinarla para un resultado final (cálculos y construcción precisa).

En el ámbito técnico, este proyecto ha servido al autor del mismo a afianzar conocimientos adquiridos a lo largo del grado, a la vez que esos conocimientos han impulsado el proyecto en sí. Asignaturas como Control I, Control II y Control Electrónico (esta última en particular) han servido como base imprescindible para realizar el controlador de los motores y el análisis de datos recogidos. Se han asentado también muchos conceptos teóricos de control que solo se habían visto en el papel, comprendiendo el por qué e implicaciones de teoremas que antes parecían carecer de praxis. También, asignaturas como Física I, Sistemas Mecánicos o Resistencia de Materiales han ayudado a la hora del diseño mecánico y del cálculo que conlleva. Finalmente, la asignatura Sistemas Electrónicos Digitales ha sido un pilar fundamental para unificar todos los elementos del proyecto a través de la tarjeta Mini-DK2, al igual que se hizo en la práctica de esta asignatura.

Para concluir, este proyecto ha servido al autor para asentar todo el conocimiento adquirido durante los años de estudio en la universidad y para adentrarse en el mundo real, dando una visión muy necesaria del entorno laboral futuro, donde la forma de trabajar es en equipo. También este trabajo intenta hacer de guía y base para futuros ingenieros y sus proyectos en el ámbito de la robótica. Para que puedan aprender tanto de los éxitos como de los fracasos en el recorrido de este proyecto, pudiendo plantear los suyos propios de una forma más organizada y eficiente.

5.6 Trabajos Futuros

Como trabajos futuros se plantean varias ampliaciones y mejoras a lo ya existente:

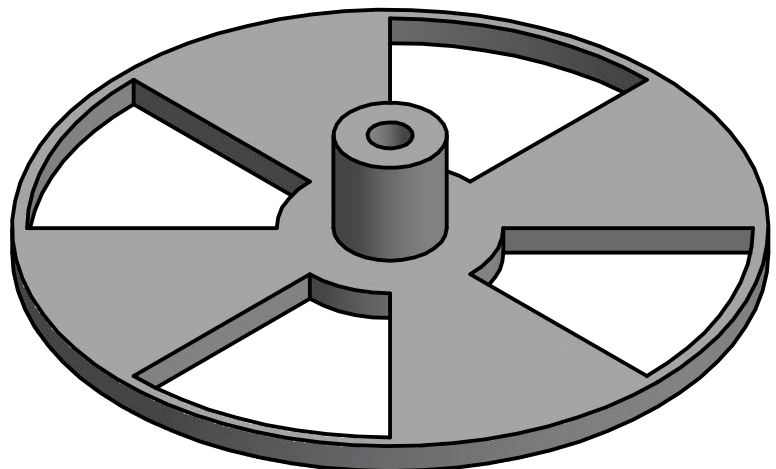
- Conexionado entre elementos
El conexionado del proyecto tanto en alimentación como en transmisión de datos tiene a veces un difícil acceso. Es por ello que se plantea una ampliación donde se unifiquen estas conexiones en buses en la medida de lo posible, creando puertos de fácil acceso donde conectar estos buses.
- Puente en H + Tarjeta Protección Mini-DK2
Otra ampliación que se plantea es la de unificar las dos tarjetas de potencia, reduciendo así el espacio ocupado y mejorando las conexiones. Esta tarjeta única se acoplaría a la Mini-DK2 como una shield.
- Reestructuración de los elementos
Esta modificación tendría como objetivo conseguir una plataforma base más compacta y robusta, dejando más espacio a los demás elementos del robot.
- Controlador PID discreto Backward Euler
Otra modificación sería cambiar el controlador PI tradicional por el PID Backward Euler. Este controlador se estudió si implementarse en el proyecto, pero finalmente se decidió utilizar el PI tradicional dados sus buenos resultados. En simulación se vio que el Backward Euler mejoraba la respuesta del sistema frente a perturbaciones, por lo que sería una modificación futura interesante.
- Protección de los sensores de distancia
Esta ampliación surge de la necesidad de proteger a los sensores de posibles sobrecargas en la alimentación. En los días de la competición “RoadAssistance” [1] un cortocircuito en la alimentación de la batería hizo que los sensores dieran falsas medidas, teniendo que realizar modificaciones software para solventar este hecho.

6. Planos

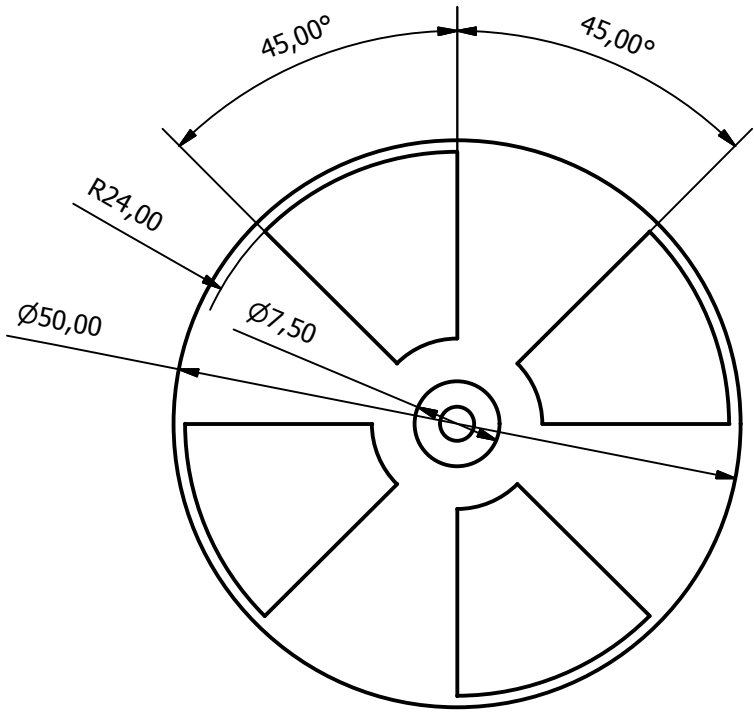
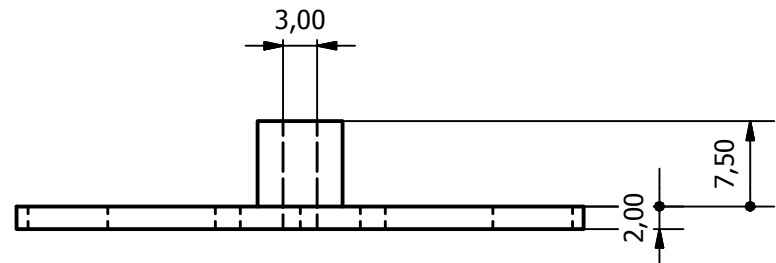
6.1 Planos mecánicos

En este apartado de planos mecánicos se muestran todas las piezas diseñadas para la realización de este proyecto. Piezas como la tornillería o perfiles de refuerzo estructural se omiten ya que no han sido diseñadas y sus dimensiones se especifican en la memoria.

Isométrica (2 : 1)

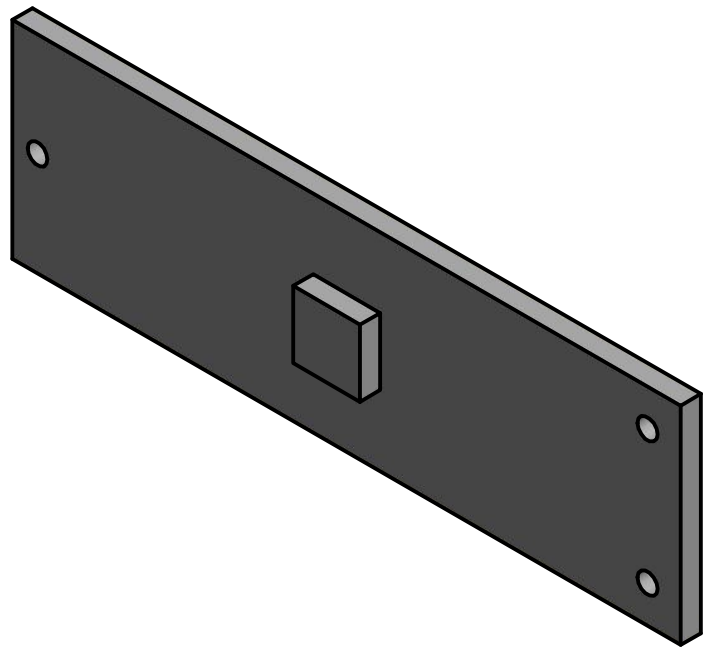


VISTA1 (1,5 : 1)

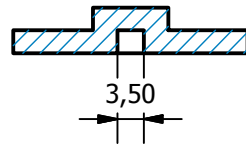


Diseño de Miguel Tradacete Á.	Revisado por Julio Pastor	Aprobado por Julio Pastor	Fecha	Fecha 17/01/2018	
UTOPIAN ROBOTICS - UNIVERSIDAD DE ALCALÁ			ENCODER V.1		
Disco de Lectura				Edición	Hoja 1 / 8

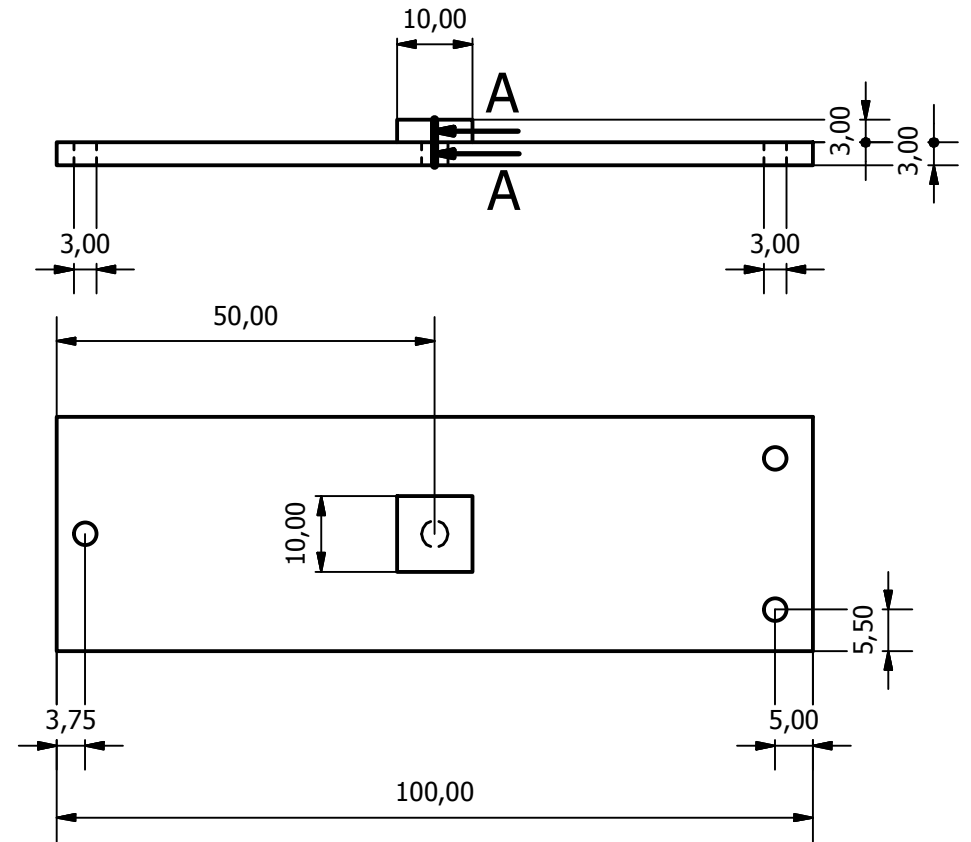
Isométrica (1,25 : 1)



A-A (1 : 1)



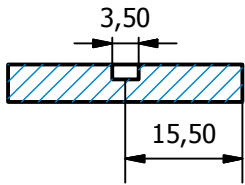
VISTA1 (1 : 1)



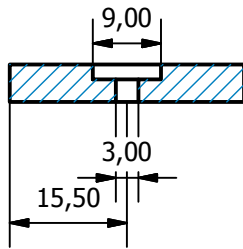
Diseño de Miguel Tradacete Á.	Revisado por Julio Pastor	Aprobado por Julio Pastor	Fecha	Fecha 15/01/2018
UTOPIAN ROBOTICS - UNIVERSIDAD DE ALCALÁ		ENCODER V.1		
Pared Externa			Edición	Hoja 2 / 8

VISTA1 (1 : 1)

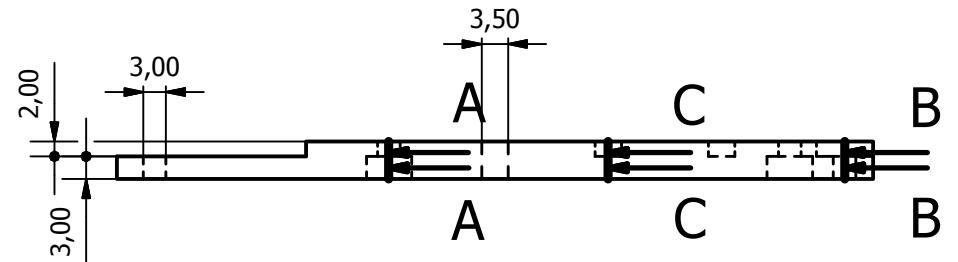
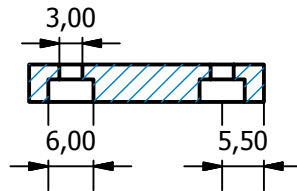
C-C (1 : 1)



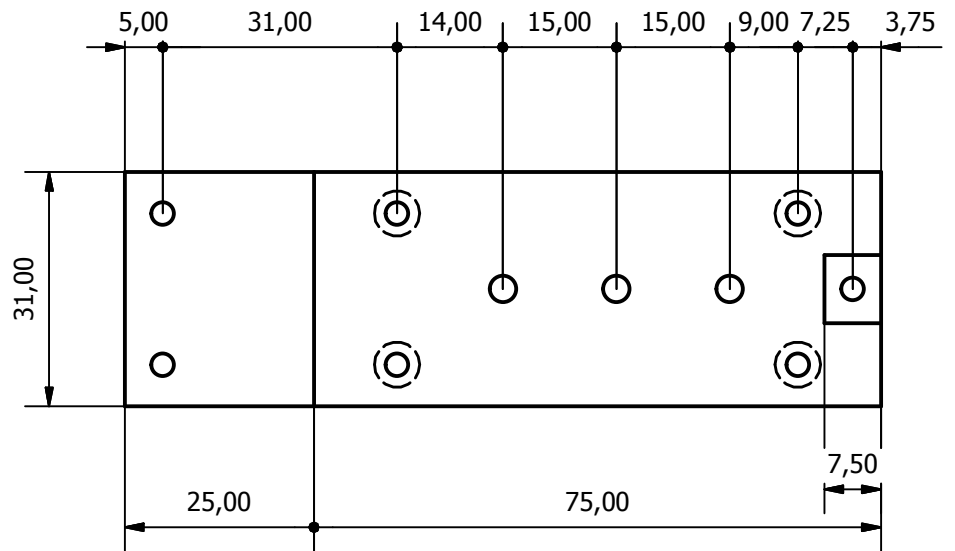
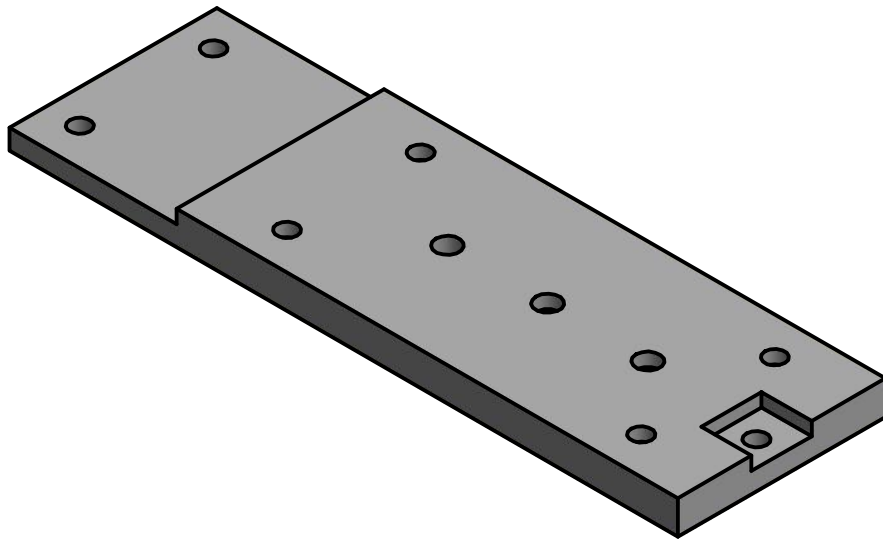
B-B (1 : 1)



A-A (1 : 1)



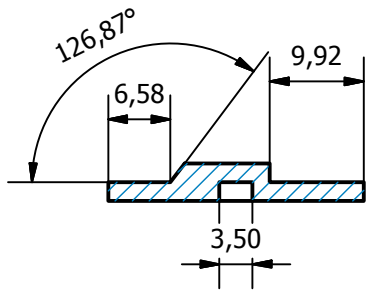
Isométrica (1,25 : 1)



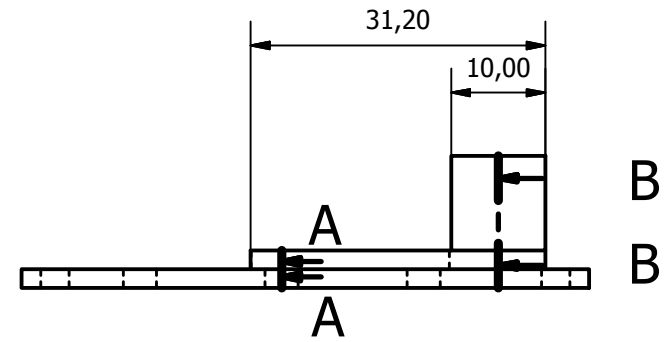
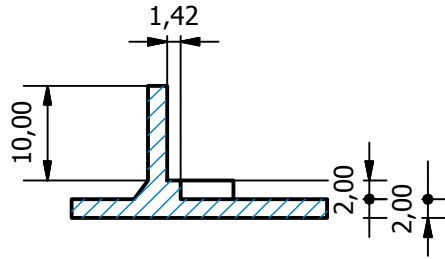
Diseño de Miguel Tradacete Á.	Revisado por Julio Pastor	Aprobado por Julio Pastor	Fecha	Fecha 15/01/2018	
UTOPIAN ROBOTICS - UNIVERSIDAD DE ALCALÁ			ENCODER V.1		
Pared Intermedia			Edición	Hoja 3 / 8	

VISTA1 (1,25 : 1)

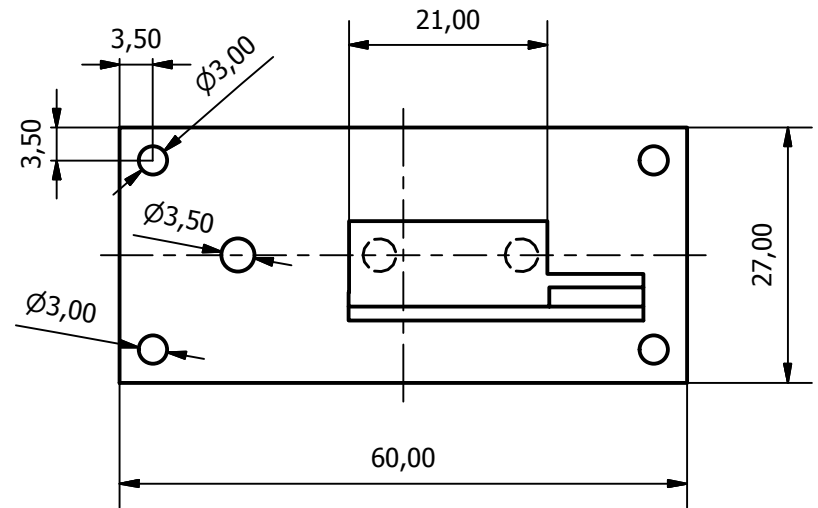
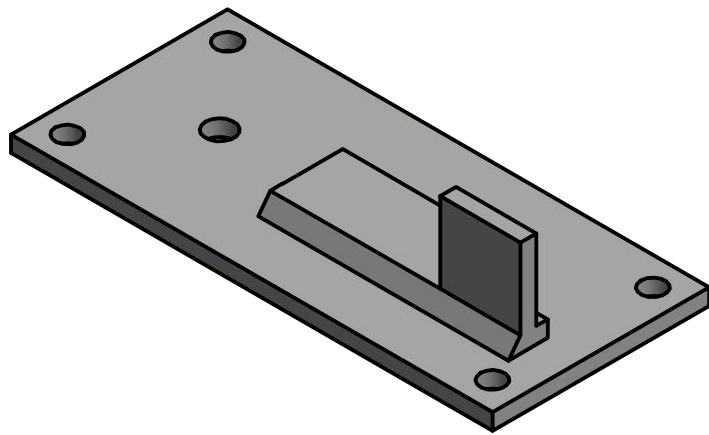
A-A (1,25 : 1)



B-B (1,25 : 1)

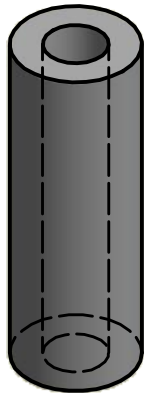


Isométrica (1,5 : 1)

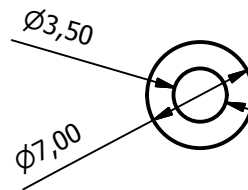


Diseño de Miguel Tradacete Á.	Revisado por Julio Pastor	Aprobado por Julio Pastor	Fecha	Fecha 17/01/2018	
UTOPIAN ROBOTICS - UNIVERSIDAD DE ALCALÁ			ENCODER V.1		
Pared Interior			Edición	Hoja 4 / 8	

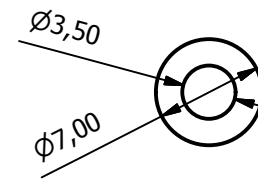
Isométrica 1 (2,5 : 1)



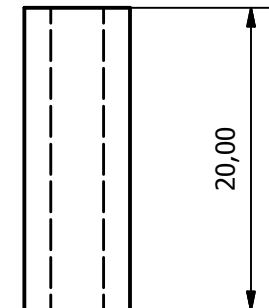
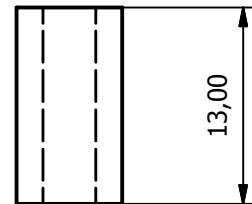
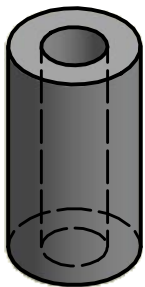
VISTA2 (2 : 1)



VISTA1 (2 : 1)

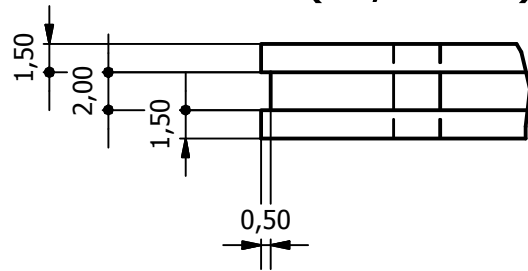


Isométrica 2 (2,5 : 1)

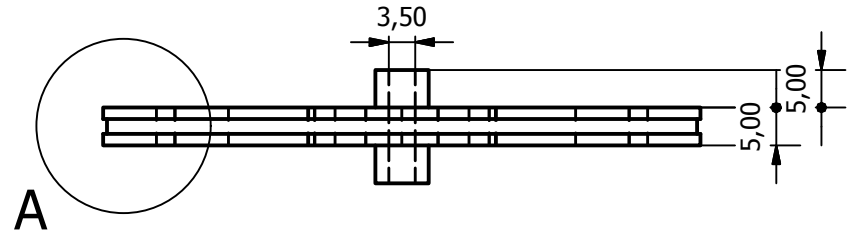


Diseño de Miguel Tradacete Á.	Revisado por Julio Pastor	Aprobado por Julio Pastor	Fecha	Fecha 17/01/2018	
UTOPIAN ROBOTICS - UNIVERSIDAD DE ALCALÁ			ENCODER V.1		
Piezas Auxiliares				Edición	Hoja 5 / 8

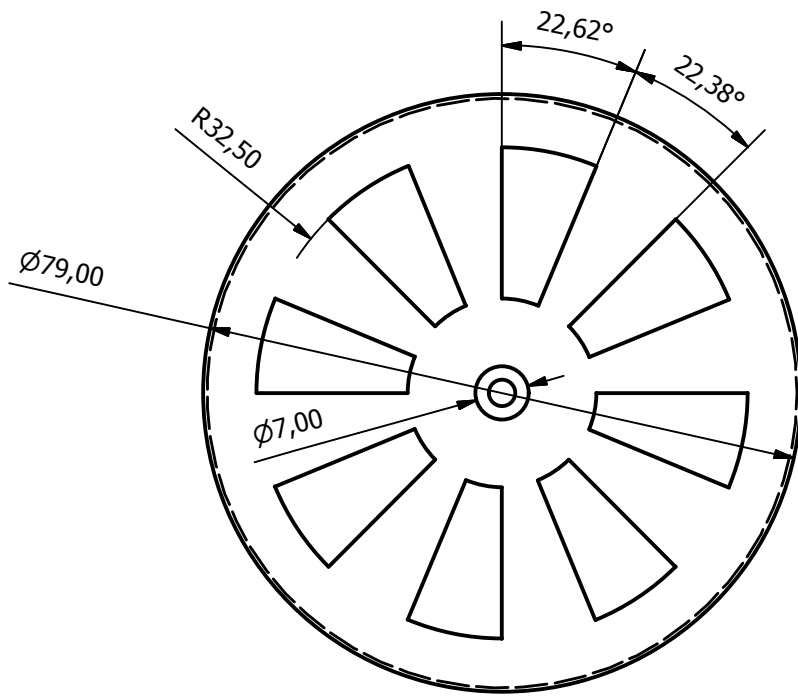
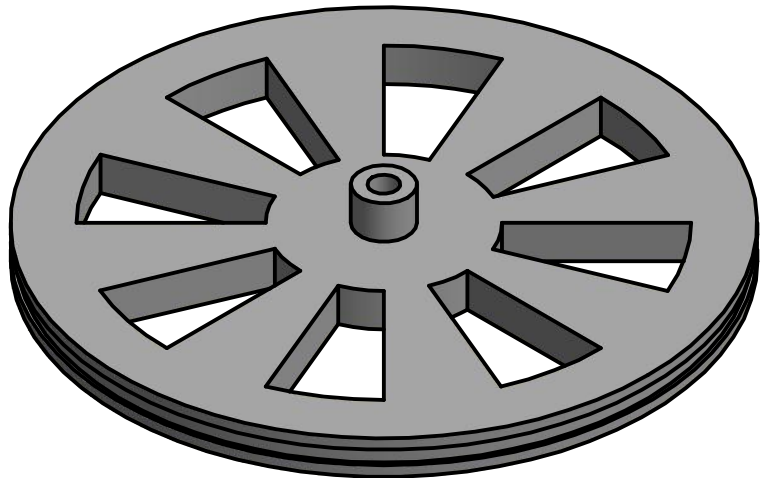
A (2,5 : 1)



VISTA1 (1 : 1)

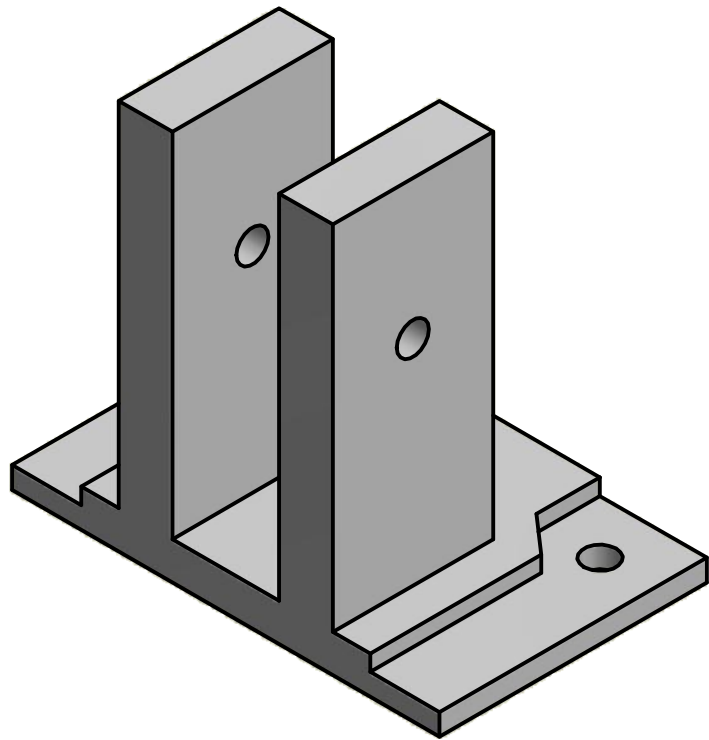


Isométrica (1,25 : 1)

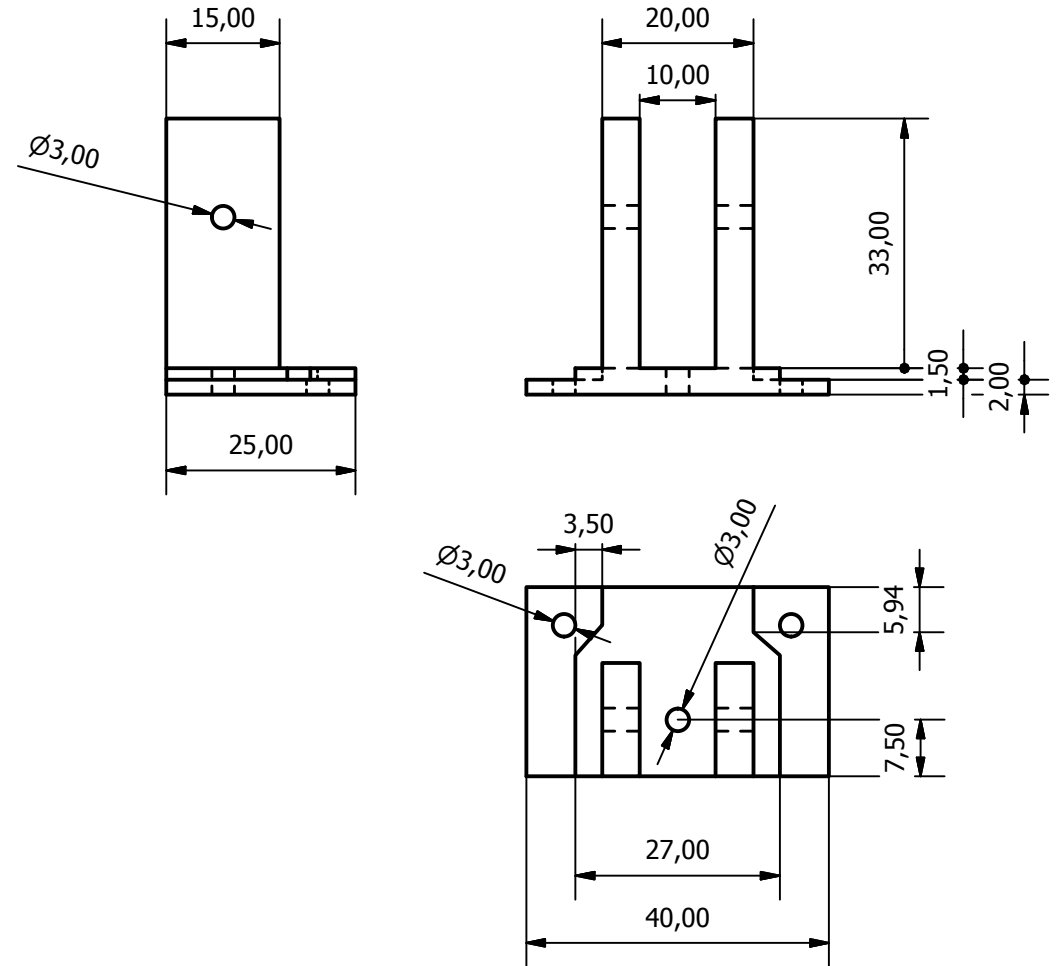


Diseño de Miguel Tradacete Á.	Revisado por Julio Pastor	Aprobado por Julio Pastor	Fecha	Fecha 17/01/2018	
UTOPIAN ROBOTICS - UNIVERSIDAD DE ALCALÁ			ENCODER V.1		
Rueda			Edición	Hoja 6 / 8	

Isométrica (2 : 1)

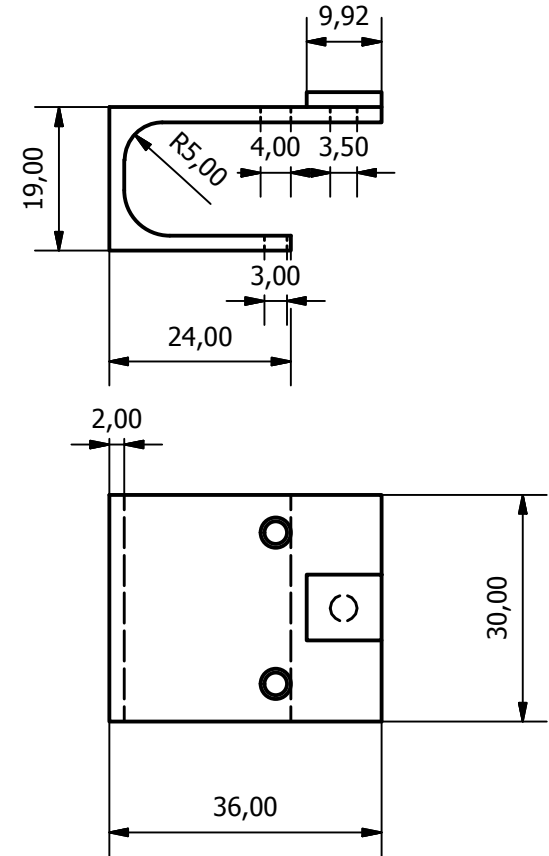
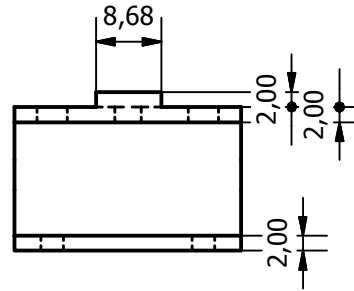
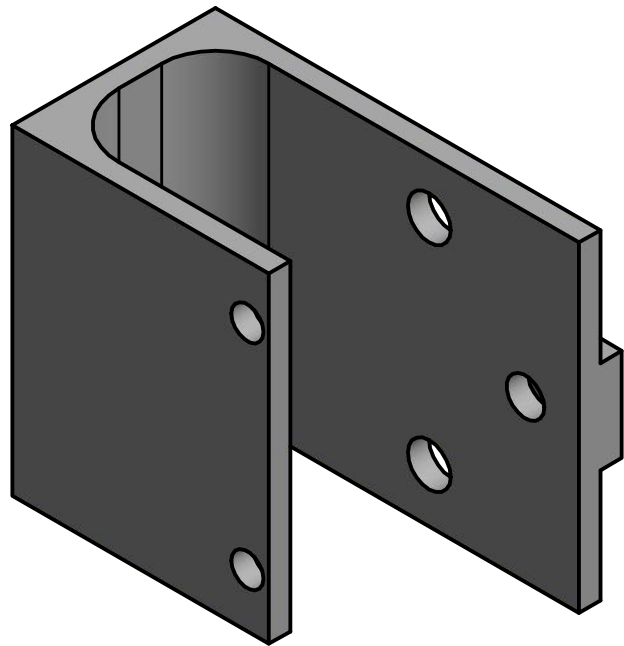


VISTA1 (1 : 1)



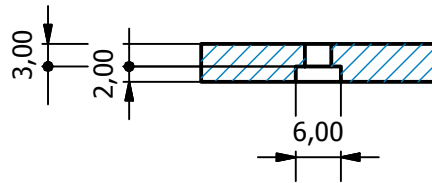
Diseño de Miguel Tradacete Á.	Revisado por Julio Pastor	Aprobado por Julio Pastor	Fecha	Fecha 15/01/2018
UTOPIAN ROBOTICS		ENCODER V.1		
UNIVERSIDAD DE ALCALÁ		Soporte	Edición	Hoja 7 / 8

Isométrica (2: 1)

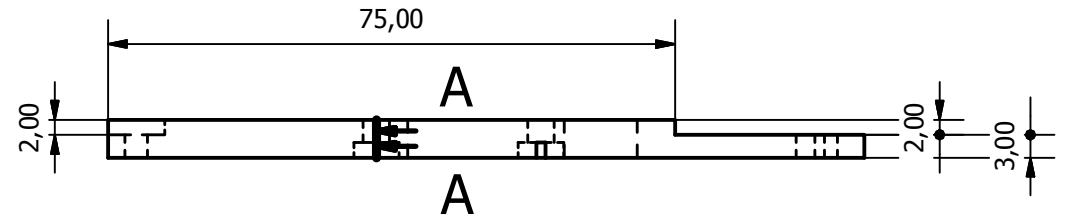


Diseño de Miguel Tradacete Á.	Revisado por Julio Pastor	Aprobado por Julio Pastor	Fecha	Fecha 17/01/2018	
UTOPIAN ROBOTICS - UNIVERSIDAD DE ALCALÁ			ENCODER V.1		
Sujección Disco Lectura				Edición	Hoja 8 / 8

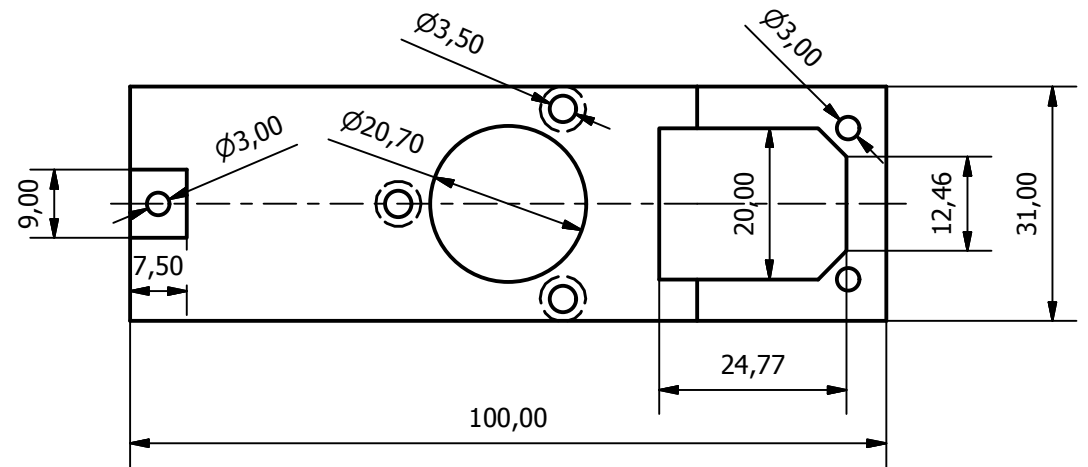
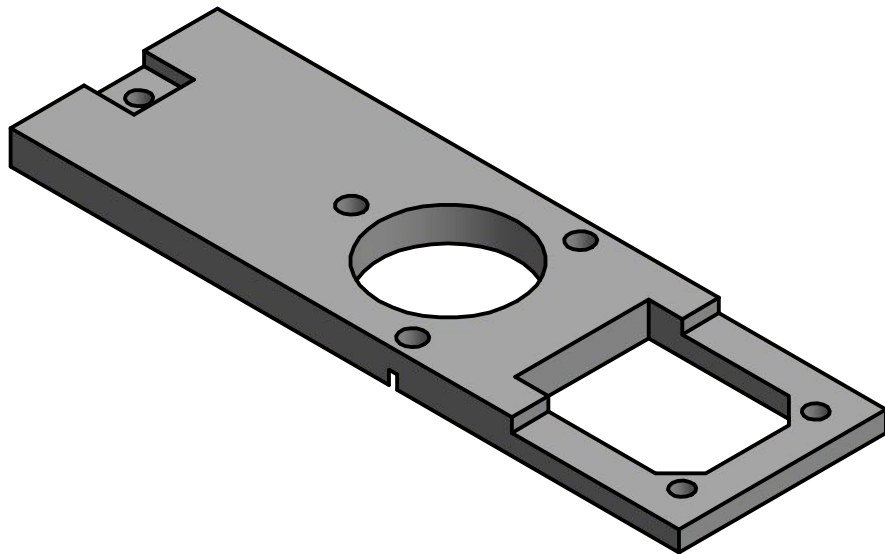
A-A (1 : 1)



VISTA1 (1 : 1)

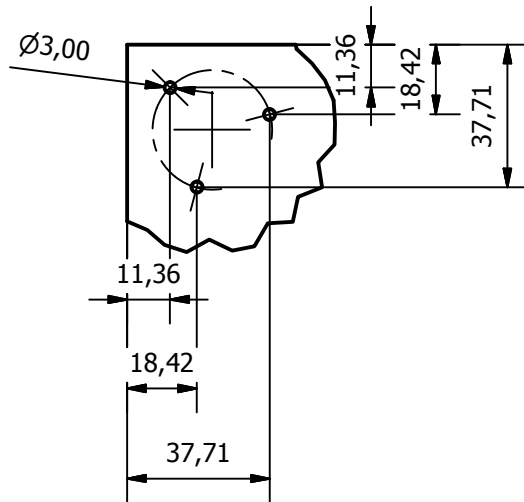


Isométrica (1,25 : 1)

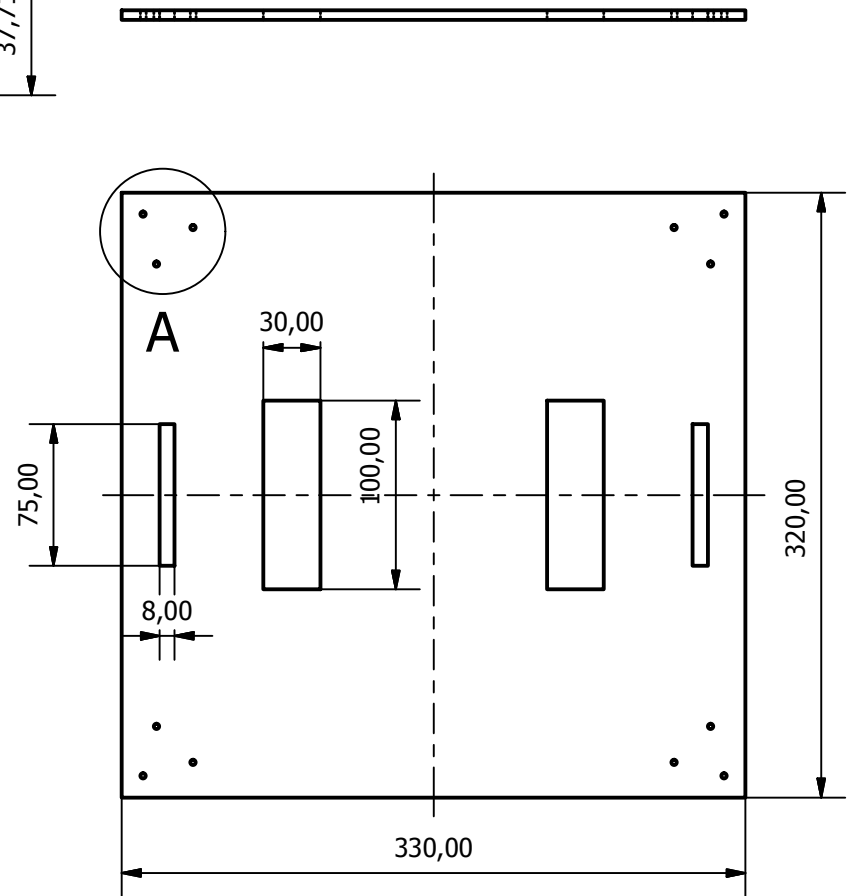


Diseño de Miguel Tradacete Á.	Revisado por Julio Pastor	Aprobado por Julio Pastor	Fecha	Fecha 17/01/2018	
UTOPIAN ROBOTICS - UNIVERSIDAD DE ALCALÁ			ENCODER V.2		
Pared Encoder_v.2				Edición	Hoja 1 / 1

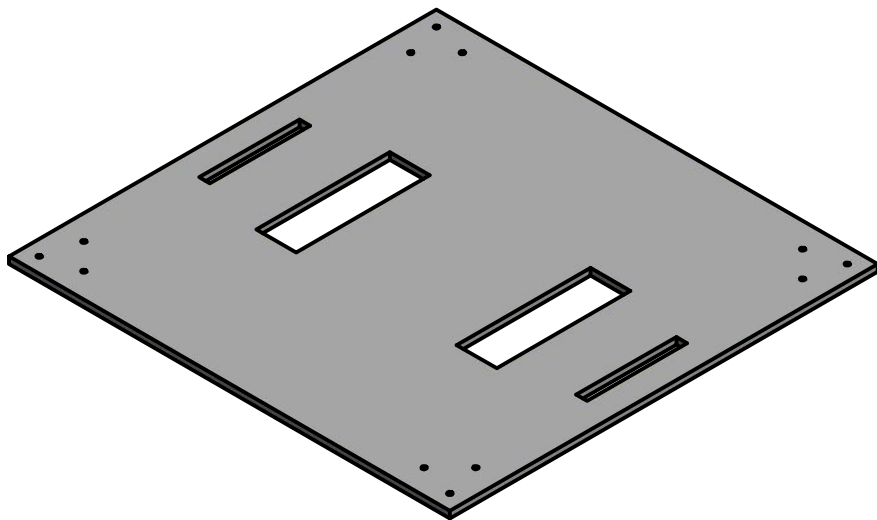
A (1 : 2)



VISTA1 (1 : 4)

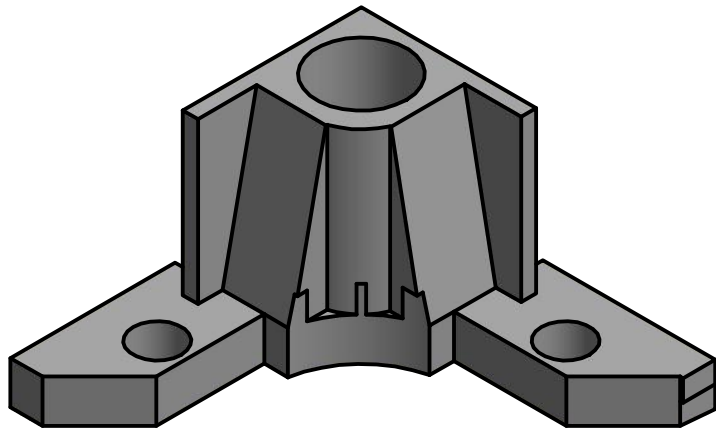


Isométrica (1 : 4)

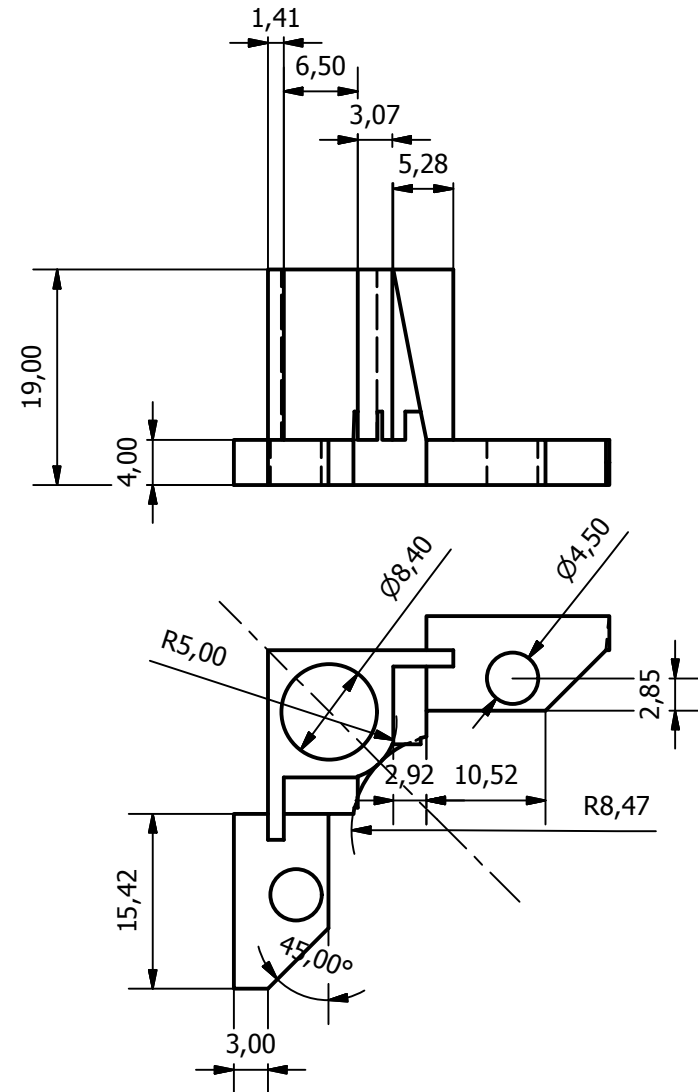


Diseño de Miguel Tradacete Á.	Revisado por Julio Pastor	Aprobado por Julio Pastor	Fecha	Fecha 18/01/2018	
UTOPIAN ROBOTICS - UNIVERSIDAD DE ALCALÁ			ESTRUCTURA EXTERNA		
Suelo Robot				Edición	Hoja 1 / 5

Isométrica (2 : 1)

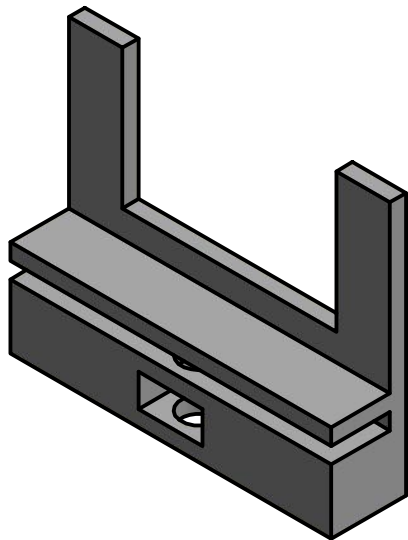


VISTA1 (1,5 : 1)

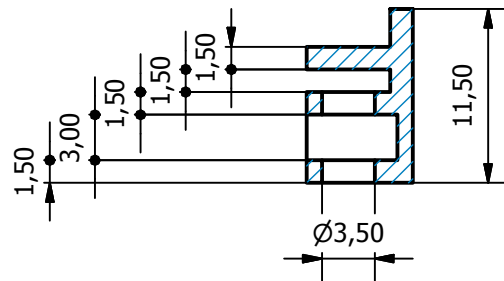


Diseño de Miguel Tradacete Á.	Revisado por Julio Pastor	Aprobado por Julio Pastor	Fecha	Fecha 18/01/2018	
UTOPIAN ROBOTICS - UNIVERSIDAD DE ALCALÁ			ESTRUCTURA EXTERNA		
			Sujeción Columna	Edición	Hoja 2 / 5

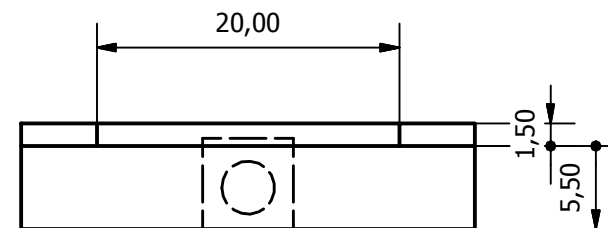
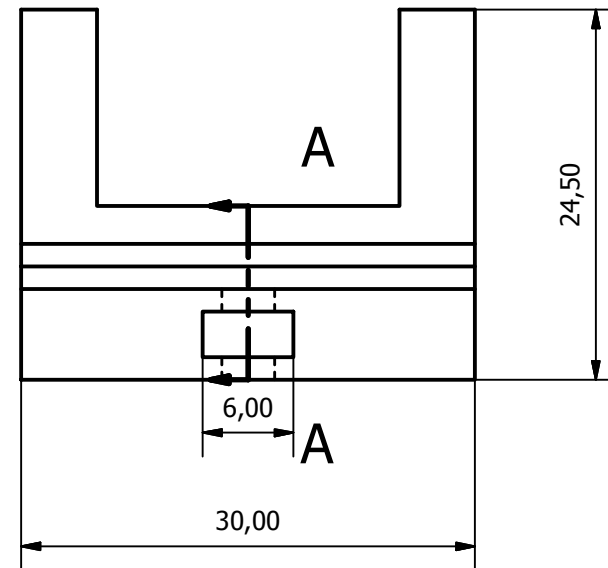
Isométrica (2 : 1)



A-A (2 : 1)

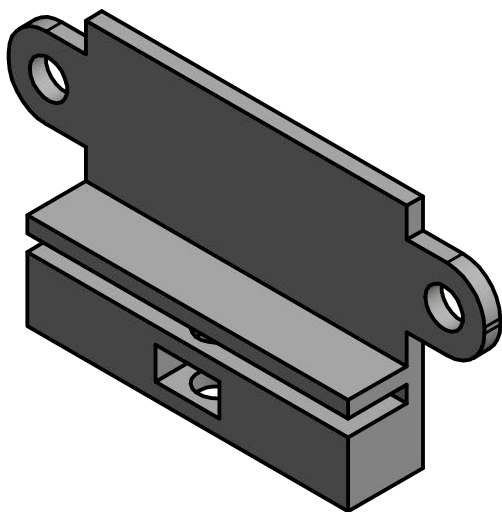


VISTA1 (2 : 1)

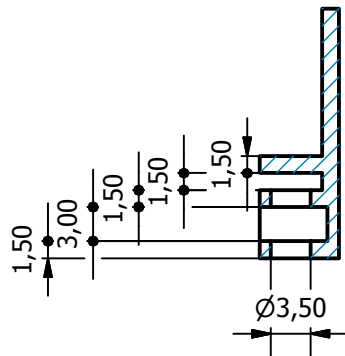


Diseño de Miguel Tradacete Á.	Revisado por Julio Pastor	Aprobado por Julio Pastor	Fecha	Fecha 18/01/2018	
UTOPIAN ROBOTICS - UNIVERSIDAD DE ALCALÁ			ESTRUCTURA EXTERNA		
Sujeción Interruptor				Edición	Hoja 3 / 5

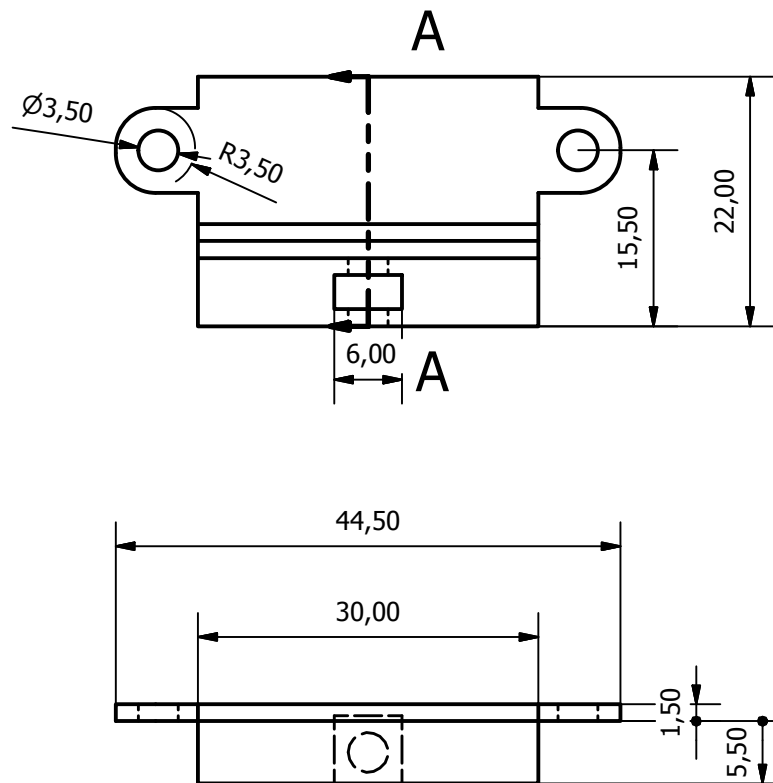
Isométrica (2 : 1)



A-A (1,5 : 1)

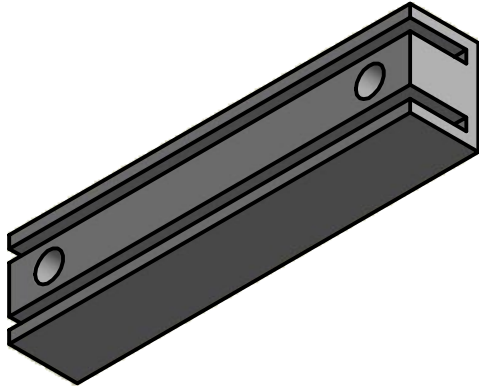


VISTA1 (1,5 : 1)

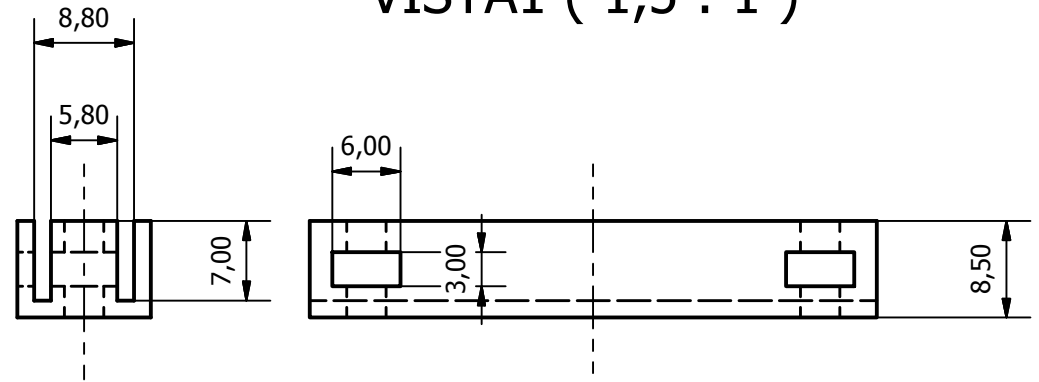


Diseño de Miguel Tradacete Á.	Revisado por Julio Pastor	Aprobado por Julio Pastor	Fecha	Fecha 03/02/2018	
UTOPIAN ROBOTICS - UNIVERSIDAD DE ALCALÁ			ESTRUCTURA EXTERNA		
Sujeción Sensor				Edición	Hoja 4 / 5

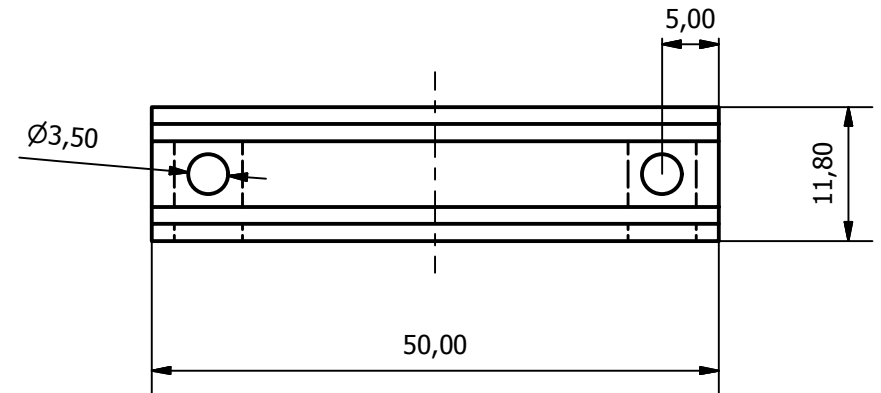
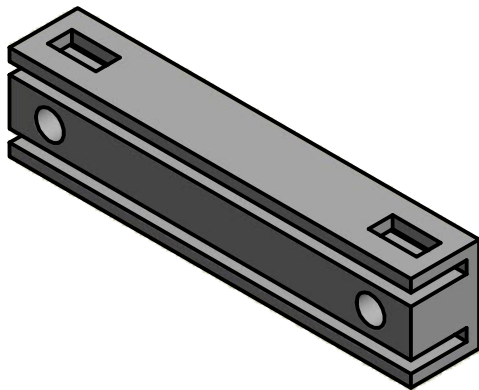
Isométrica 1 (1,5 : 1)



VISTA1 (1,5 : 1)

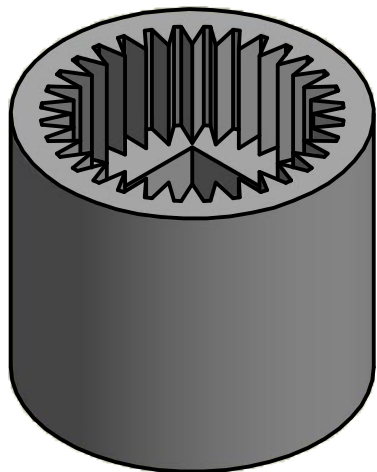


Isométrica 2 (1,5 : 1)

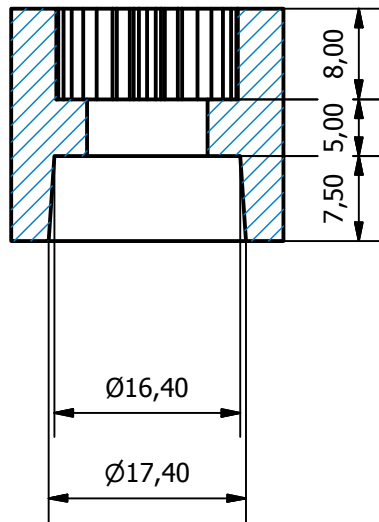


Diseño de Miguel Tradacete Á.	Revisado por Julio Pastor	Aprobado por Julio Pastor	Fecha	Fecha 18/01/2018	
UTOPIAN ROBOTICS - UNIVERSIDAD DE ALCALÁ			ESTRUCTURA EXTERNA		
Sujeción Ventilador				Edición	Hoja 5 / 5

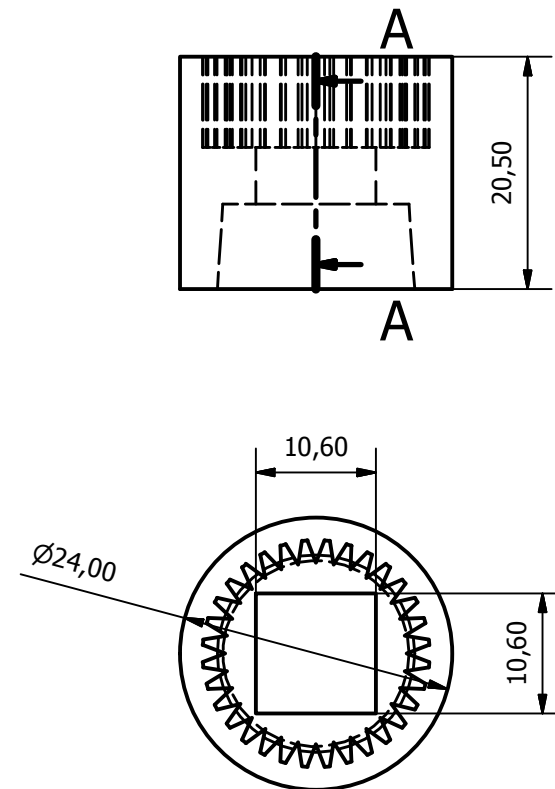
Isométrica (2 : 1)



A-A (1,5 : 1)

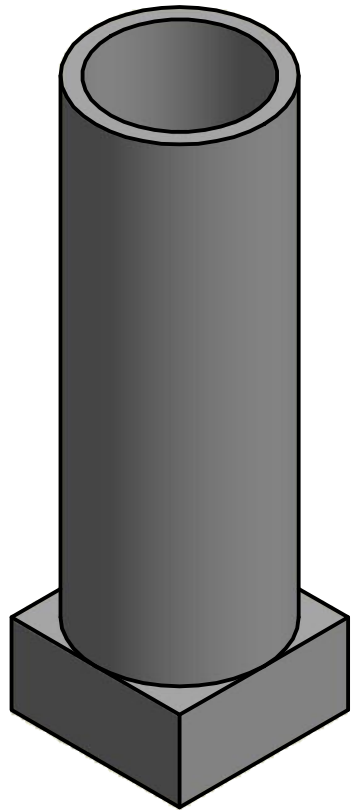


VISTA1 (1,5 : 1)

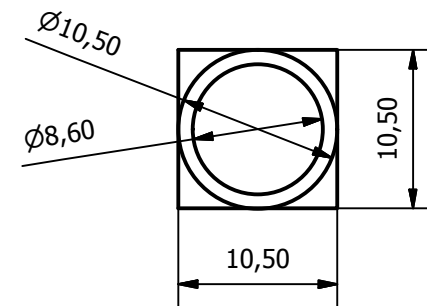
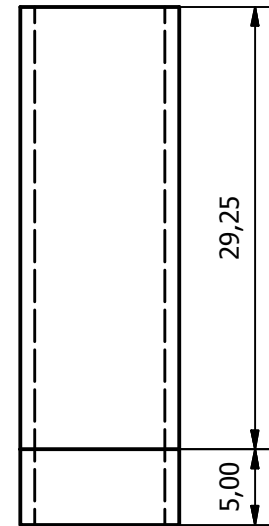


Diseño de Miguel Tradacete Á.	Revisado por Julio Pastor	Aprobado por Julio Pastor	Fecha	Fecha 18/01/2018	
UTOPIAN ROBOTICS - UNIVERSIDAD DE ALCALÁ			MECÁNICA DE TRACCIÓN		
			Acoplador Motor	Edición	Hoja 1 / 5

Isométrica (3 : 1)

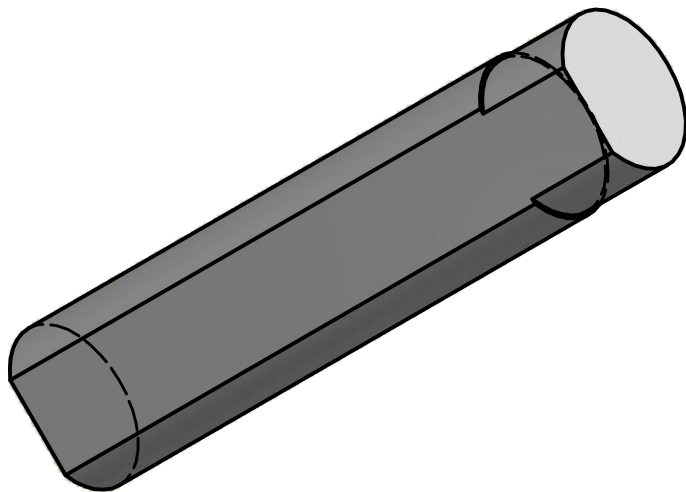


VISTA1 (2 : 1)

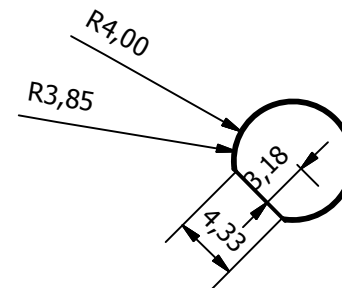
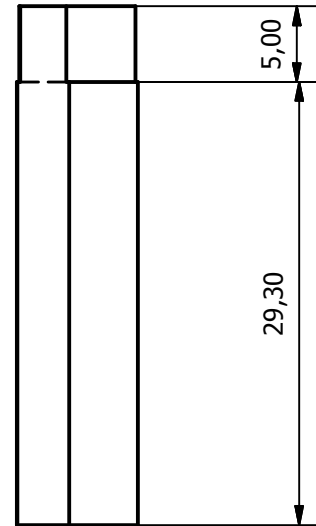


Diseño de Miguel Tradacete Á.	Revisado por Julio Pastor	Aprobado por Julio Pastor	Fecha	Fecha 17/01/2018	
UTOPIAN ROBOTICS - UNIVERSIDAD DE ALCALÁ			MECÁNICA DE TRACCIÓN		
			Eje Hueco Motor	Edición	Hoja 2 / 5

Isométrica (3 : 1)

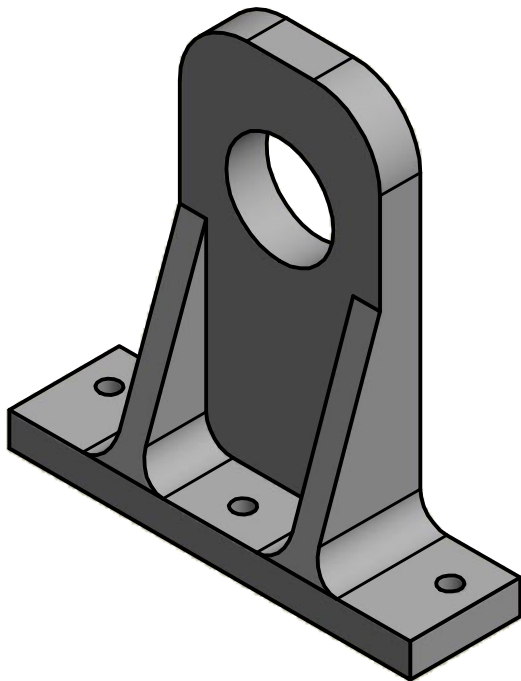


VISTA1 (2 : 1)

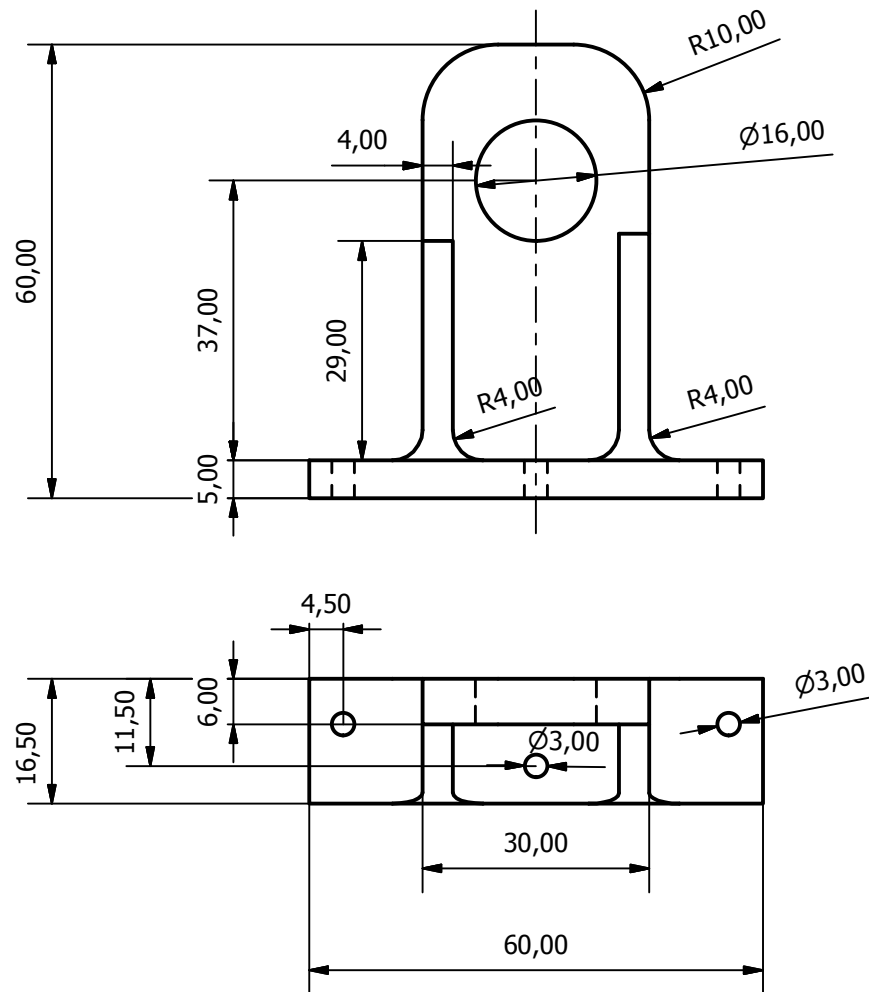


Diseño de Miguel Tradacete Á.	Revisado por Julio Pastor	Aprobado por Julio Pastor	Fecha	Fecha 17/01/2018	
UTOPIAN ROBOTICS - UNIVERSIDAD DE ALCALÁ			MECÁNICA DE TRACCIÓN		
Eje Macizo Motor				Edición	Hoja 3 / 5

Isométrica (1,25 : 1)

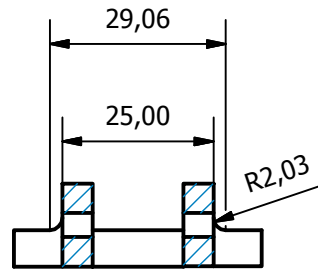


VISTA1 (1 : 1)

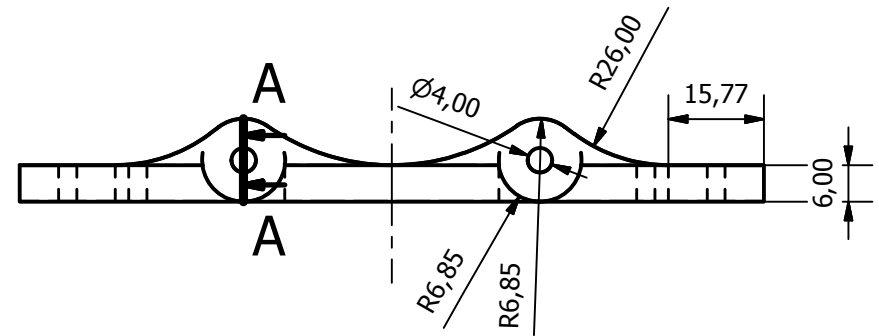


Diseño de Miguel Tradacte Á.	Revisado por Julio Pastor	Aprobado por Julio Pastor	Fecha	Fecha 17/01/2018	
UTOPIAN ROBOTICS - UNIVERSIDAD DE ALCALÁ			MECÁNICA DE TRACCIÓN		
Soporte Rodamiento			Edición	Hoja 4 / 5	

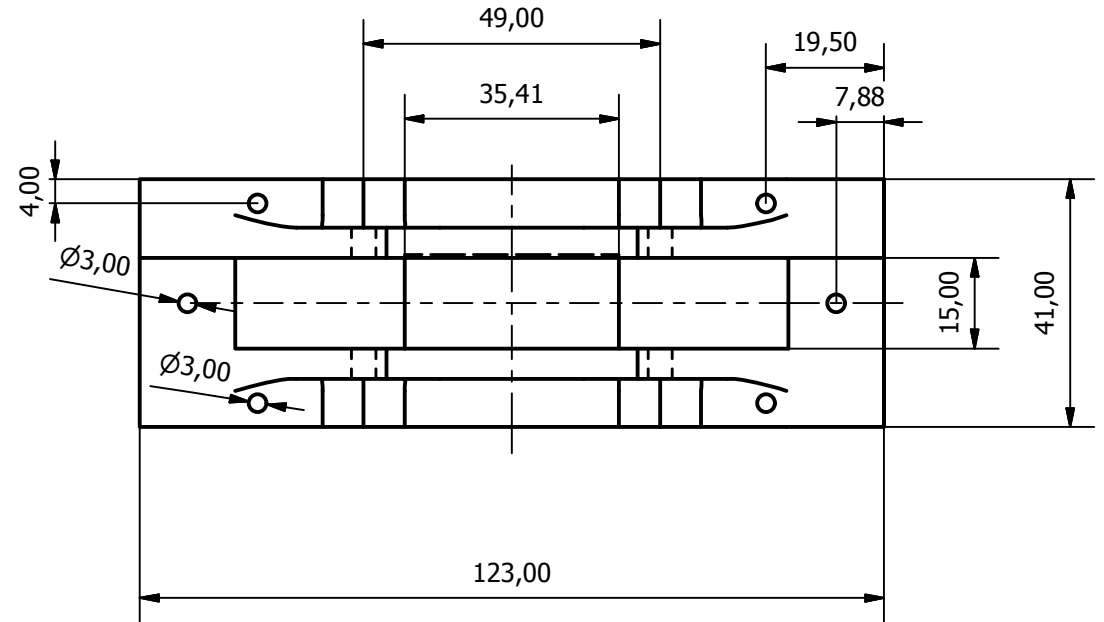
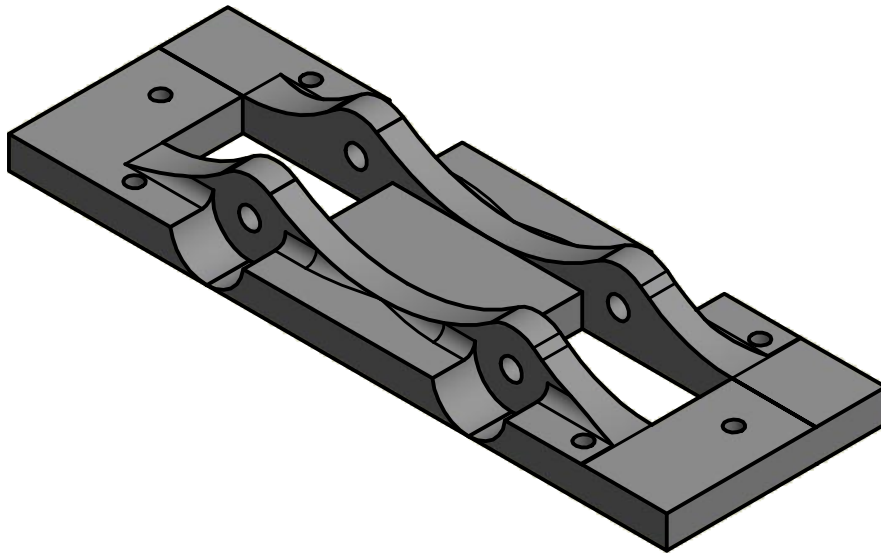
A-A (1 : 1,25)



VISTA1 (1 : 1,25)



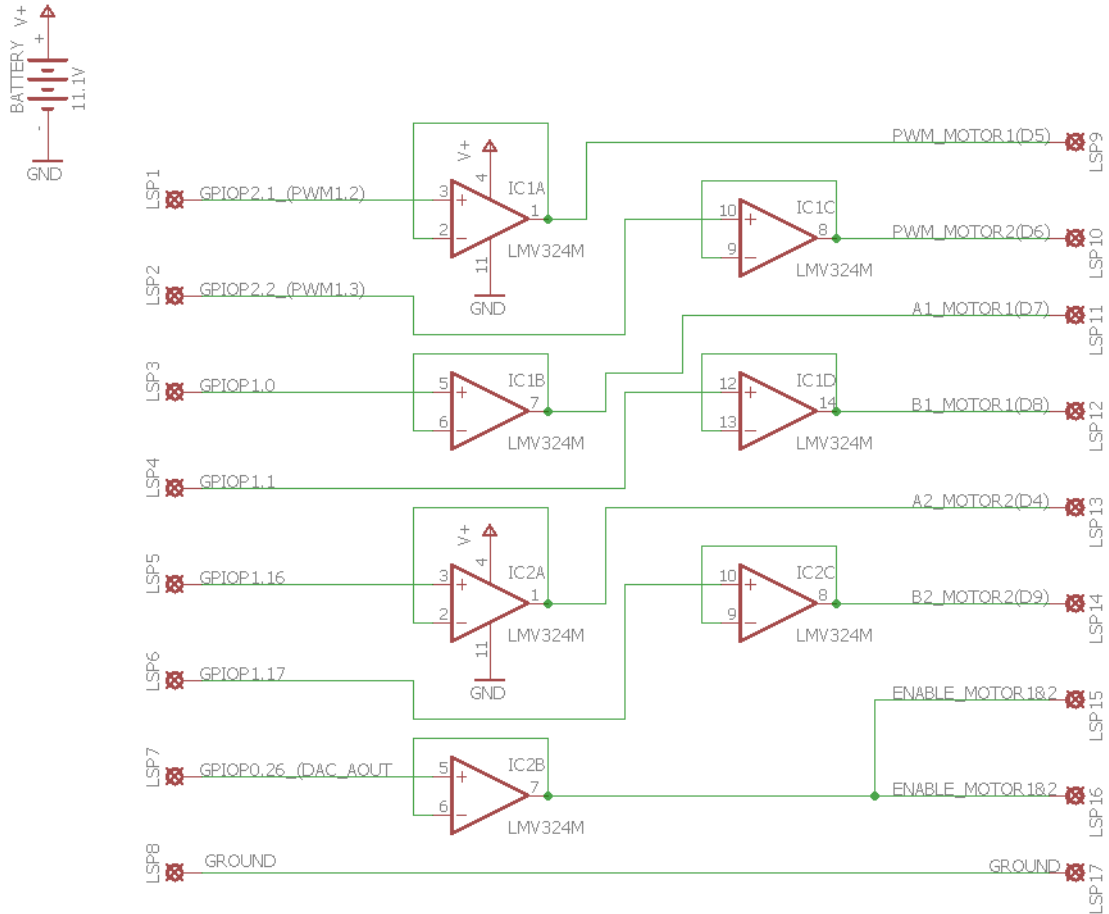
Isométrica (1 : 1)



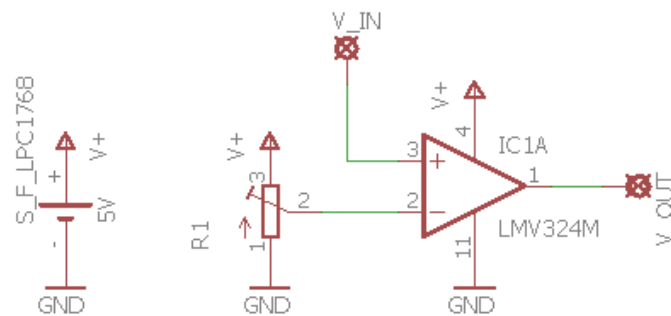
Diseño de Miguel Tradacete Á.	Revisado por Julio Pastor	Aprobado por Julio Pastor	Fecha	Fecha 22/01/2018	
UTOPIAN ROBOTICS - UNIVERSIDAD DE ALCALÁ			MECÁNICA DE TRACCIÓN		
Sujeción Motor				Edición	Hoja 5 / 5

6.2 Planos de la electrónica

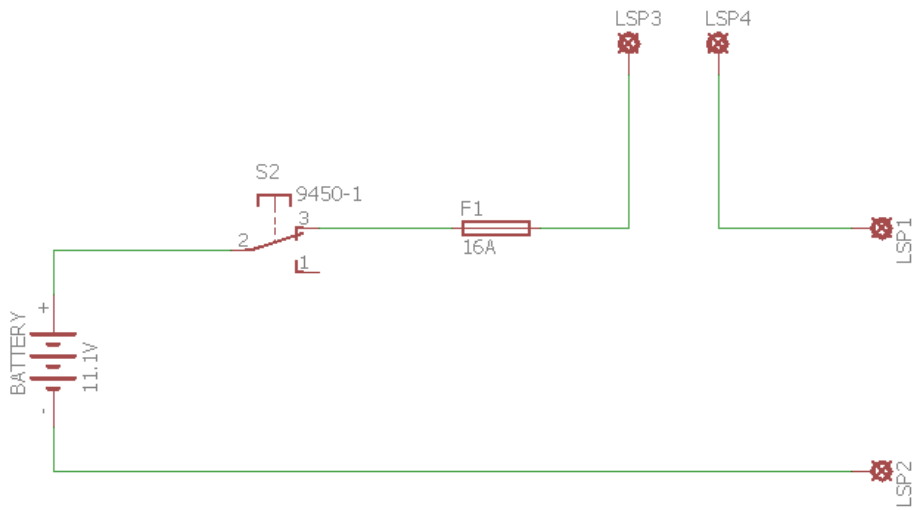
6.2.1 Protección de Mini-DK2



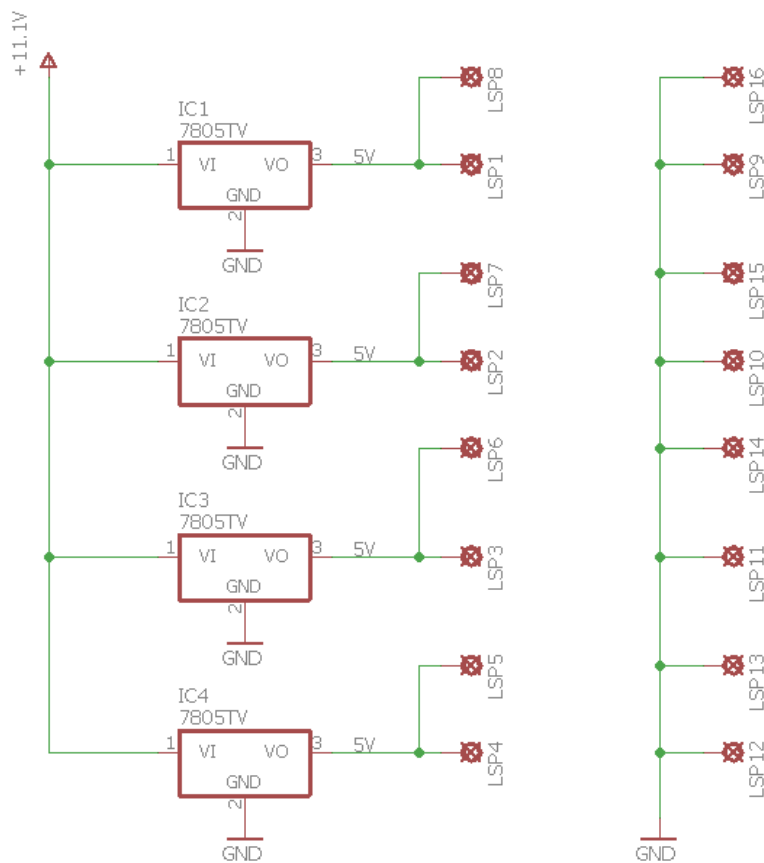
6.2.2 Lectura y acondicionamiento de Encoder v.1



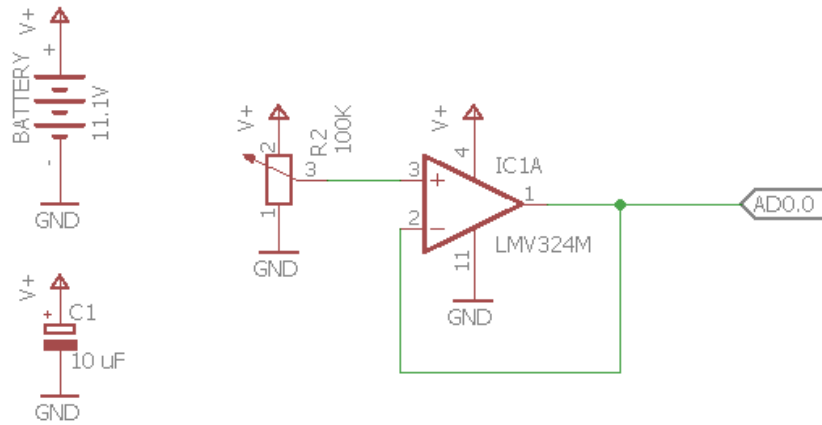
6.2.3 Circuito de alimentación



6.2.4 Estación de 5V



6.2.5 Medidor del nivel de batería



7. Pliego de condiciones

7.1 Condiciones de diseño

La totalidad de la responsabilidad del buen uso y correcto funcionamiento de este proyecto es asumida si se cumplen las especificaciones que siguen en este apartado.

7.2 Condiciones de materiales y equipos

7.2.1 Electrónica

Se deben cumplir las condiciones de diseño mostradas en los planos, concernientes al conexionado, alimentaciones y componentes a utilizar.

7.2.2 Elementos mecánicos y estructurales

La impresión 3D de las piezas a utilizar debe cumplir:

- Tolerancia igual o inferior de $\pm 0,15$ mm.
- Plástico tipo PLA.
- Relleno interno mínimo del 20% en modo Honeycomb (Panal de abeja).

Las ruedas deben disponer de una superficie con un factor de adherencia superior al 0,95. También deben ser ruedas de carretilla para uso industrial o similares capaces de soportar más de 10 Kg.

La plancha utilizada como suelo de los elementos internos debe ser de fibra de vidrio y del grosor indicado en la memoria (1,65 mm).

Se debe disponer de una seta de seguridad roja o de un color llamativo con diámetro de al menos 20 mm y debe estar situada en una zona visible y de fácil acceso.

Si se va a hacer uso de este proyecto para participar en competiciones tales como las del Robotic Day en Praga, Eurobot y similares deberán cumplirse las especificaciones de sus respectivas normativas además de las restricciones propias de este proyecto.

7.2.3 Baterías

Para la alimentación de todo el sistema de potencia (tracción y parte de la electrónica) es necesario contar con una batería tipo Li-Po con las siguientes características:

- Tensión nominal 11-12 V.
- Una capacidad mínima de 2700 mAh.
- Un peso máximo de 250 g.
- Una descarga de 40C mínimo.

El cargado de esta batería se debe realizar con un cargador específico para este tipo de baterías, con control de la corriente de carga y detección del estado de la batería. Además, la batería debe encontrarse siempre (se use o no) en el interior de una bolsa ignífuga protectora especial para este tipo de baterías.

La alimentación de la tarjeta de control no debe exceder los 7 V y en caso de utilizarse otra distinta a la expuesta en este proyecto no podrá excederse el límite marcado por el fabricante de la misma.

7.2.4 Cableado

Todo el cableado que une las baterías con la electrónica de potencia, la seta de seguridad y los motores debe tener una sección mínima 2,5 mm².

7.2.5 Equipos mecánicos

Es necesario contar con una impresora 3D capaz de proporcionar las condiciones de impresión expuestas anteriormente y una máquina CNC en modo de fresadora con la potencia suficiente para trabajar la fibra de vidrio.

7.2.6 Equipos electrónicos

Tanto en la fase de diseño como en la de montaje y pruebas es necesario un osciloscopio digital de dos canales con el que realizar medidas.

Es necesario también contar también con un soldador de al menos 11 W y una fuente de alimentación para la realización de pruebas y montaje de la electrónica.

7.2.7 Equipos informáticos

Para todo el diseño 3D de las piezas, la programación del software y la recogida de datos para la obtención del modelo de los motores es necesaria la utilización de los siguientes recursos:

7.2.7.1 Recursos hardware

- Ordenador con la capacidad de ejecutar Autodesk Inventor y MatlabR2016b en paralelo.
- Puerto Serie disponible o puerto USB.
- J-Link EDU Segger Microcontroller para cargar la programación en la tarjeta Mini DK2.

Los equipos descritos deben estar debidamente conectados a la red eléctrica y deben funcionar correctamente.

7.2.7.2 Recursos software

- Sistema operativo capaz de ejecutar Autodesk Inventor, MatlabR2016b, Keil y CoolTerm.
- Programa de diseño Autodesk Inventor.
- Programa MatlabR2016b.
- Entorno de programación Keil para la tarjeta Mini-DK2.
- Programa de creación de terminales serie CoolTerm.

Se debe disponer de las licencias pertinentes para cada uno de estos programas si lo requieren, así como de su instalación debidamente realizada.

7.2.8 Seguridad e Higiene

Siempre se deberá tener en cuenta los posibles accidentes que se pueden producir por la utilización de este proyecto o su traslado. Para evitarlos se debe disponer de:

- Guantes y gafas protectoras en el montaje mecánico, las pruebas, la utilización de la máquina CNC y en la manipulación de la batería.
- Una mascarilla debidamente colocada a la hora de imprimir piezas en 3D o utilizar la máquina CNC.
- Un puesto de trabajo debidamente acondicionado para no sufrir daños cervicales durante un largo periodo de trabajo.
- Luminosidad adecuada en el área de trabajo.

7.2.9 Personal

La persona al cargo del diseño hardware y software debe tener experiencia en el diseño 3D y en la programación de microcontroladores.

A su vez, la persona encargada del montaje, fabricación de piezas y pruebas debe tener experiencia con impresoras 3D, manejo de máquinas CNC y experiencia soldando componentes electrónicos.

7.2.10 Condiciones de ejecución

Para el correcto montaje, uso, manejo y funcionamiento de este proyecto es necesario haber leído y entendido correctamente todos los apartados de este trabajo.

Cualquier usuario que tome como punto de partida este proyecto para crear uno nuevo o haga uso de este proyecto deberá mencionar visiblemente a Utopian Robotics y a la Universidad de Alcalá tanto en su proyecto como en cualquier documento referente al mismo.

8. Presupuesto

8.1 Introducción

El presupuesto de este proyecto engloba todo el coste material e intangible para llevar a cabo el mismo. Se supone el diseño y desarrollo, construcción, y fase de pruebas y puesta en marcha de la plataforma base, control del movimiento y todo el conjunto de elementos del robot que interactúan con este proyecto.

Para la realización del presupuesto se supone que este trabajo es un proyecto I+D (Investigación y Desarrollo), por lo que carece del visado del colegio de ingenieros, no teniéndose en cuenta los costes de certificación. Tampoco se incluyen los honorarios por la redacción de este trabajo.

Los proveedores del material utilizado son mayoritariamente Brico Depot y Electrónica Alcalá. A modo de simplificación solo se indican (con una referencia a bibliografía) aquellos proveedores distintos.

8.2 Coste mano de obra

En toda la trayectoria de este proyecto se requiere de personal con distinta capacitación, distinguiéndose dos fases de trabajo: diseño y desarrollo, y construcción. De este modo las distintas fases se llevan a cabo por dos empleados distintos con remuneraciones distintas (Tabla 17). La estimación de las horas se realiza en el apartado “*Presupuesto material*”, dividiendo esta estimación en las distintas partes del proyecto.

Puesto	Sueldo/hora
<i>Ingeniero Técnico Industrial</i>	35 €/h
<i>Técnico Superior en Mecatrónica</i>	16 €/h

Tabla 17. Coste mano de obra

8.3 Presupuesto material

Resumen Presupuesto Material

<i>Sección</i>	<i>Descripción</i>	<i>Cantidad</i>	<i>Precio Unit.</i>	<i>Precio Total</i>
8.3.1	Elementos Mecánicos de Tracción	1	3.235,04 €	3.235,04 €
8.3.2	Encoder v.1	2	2.053,43 €	4106,86 €
8.3.3	Encoder v.2	2	427,50 €	855 €
8.3.4	Estructura Externa	1	543,21 €	543,21 €
8.3.5	Tarjeta de Control Central	1	5.887,46 €	5.887,46 €
8.3.6	Tarjetas de Potencia	1	957,83 €	957,83 €
8.3.7	Electrónica Encoder v.1	2	691,99 €	1383,98 €
8.3.8	Alimentación	1	628,07 €	628,07 €
8.3.9	Electrónica Medición de Batería	1	271,38 €	271,38 €
8.3.10	Sensores de Distancia	1	433,20 €	433,20 €
Total				18.302,03 €

Tabla 18. Resumen presupuesto material

Las piezas realizadas por impresión 3D utilizan como materia prima plástico PLA, en concreto rollos BQ Filamento PLA Easy Go 1,75mm [15]. Cada rollo de filamento pesa 1 Kg, por lo que conociendo su precio se puede hallar el coste de cada gramo de pieza impresa. Un rollo BQ Filamento PLA Easy Go 1,75mm [15] cuesta 20,2 €, por lo que:

$$\text{Precio pieza 3D} = \frac{20,2 \text{ €}}{1000 \text{ g}} = 0,02 \text{ €}$$

8.3.1 Elementos Mecánicos de Tracción

<i>Descripción</i>	<i>Cantidad</i>	<i>Precio Unit.</i>	<i>Precio Total</i>
Material			
Motor Bosch FPC 12V 0-130-822-019 [16]	2	10 €	20,00 €
Ruedas Polipropileno Ø 100	2	3,75 €	7,50 €
Rodamientos Ø 8 – Ø 16 [17]	2	2,17 €	4,34 €
Rueda Bola de Acero Ø 15 Pletina Ø 45	4	3,65 €	14,60 €
Tornillería M3, M4, M5	1	2,75 €	2,75 €
Cable H07V-K 1 x 1,5	0,8 m	0,31 €/m	0,25 €
Pieza Soporte Rodamiento (7 g)	2	0,02 €/g	0,28 €
Pieza Acoplador Motor (25 g)	2	0,02 €/g	1 €
Pieza Eje Hueco Motor (1 g)	2	0,02 €/g	0,04 €
Pieza Eje Macizo Motor (5 g)	2	0,02 €/g	0,2 €
Pieza Sujeción Motor (2 g)	2	0,02 €/g	0,08 €
Mano de obra			
Ingeniero Técnico Industrial	80	35 €/h	2800 €
Técnico Superior en Mecatrónica	24	16 €/h	384 €
Total			3.235,04 €

Tabla 19. Costes elementos mecánicos de tracción

8.3.2 Encoder v.1

<i>Descripción</i>	<i>Cantidad</i>	<i>Precio Unit.</i>	<i>Precio Total</i>
Material			
Pieza Soporte (6 g)	1	0,02 €/g	0,12 €
Pieza Pared Exterior (4 g)	1	0,02 €/g	0,08 €
Pieza Pared Intermedia (4 g)	1	0,02 €/g	0,08 €
Pieza Pared Interior (4 g)	1	0,02 €/g	0,08 €
Pieza Rueda (10 g)	1	0,02 €/g	0,2 €
Pieza Disco de Lectura (2 g)	1	0,02 €/g	0,04 €
Pieza Sujeción Disco Lectura (3 g)	1	0,02 €/g	0,06 €
Piezas Auxiliares Encoder (1g)	1	0,02 €/g	0,02 €
Juego Engranajes/Ejes	1	10 €	10,00 €
Tornillería M3	1	2,75 €	2,75 €
Mano de obra			
Ingeniero Técnico Industrial	40	35 €/h	1400 €
Técnico Superior en Mecatrónica	40	16 €/h	640 €
Total			2.053,43 €

Tabla 20. Costes Encoder v.1

8.3.3 Encoder v.2

<i>Descripción</i>	<i>Cantidad</i>	<i>Precio Unit.</i>	<i>Precio Total</i>
Material			
Encoder 600 P/R H38S600B [8]	1	16,11 €	16,11 €
Pieza Rueda (10 g)	1	0,02 €/g	0,2 €
Pieza Pared (4 g)	1	0,02 €/g	0,08 €
Tornillería M3	1	2,75 €	2,75 €
Cable conexionado	6	0,06 €	0,36 €
Mano de obra			
Ingeniero Técnico Industrial	8	35 €/h	280 €
Técnico Superior en Mecatrónica	8	16 €/h	128 €
Total			427,50 €

Tabla 21. Costes Encoder v.2

8.3.4 Estructura Externa

<i>Descripción</i>	<i>Cantidad</i>	<i>Precio Unit.</i>	<i>Precio Total</i>
Material			
Plancha Fibra Vidrio – Cobre	1	-	0 €
Perfil “U” 8 x 8 x 8 mm	1	2,95 €	2,95 €
Tubo Redondo Aluminio 8 x 1 mm	1	1,35 €	1,35 €
Pieza Sujeción Columna (2 g)	4	0,02 €/g	0,16 €
Tornillería M3	1	2,75 €	2,75 €

Mano de obra

Ingeniero Técnico Industrial	8	35 €/h	280 €
Técnico Superior en Mecatrónica	16	16 €/h	256 €
Total			543,21 €

Tabla 22. Costes estructura externa

8.3.5 Tarjeta de Control Central

<i>Descripción</i>	<i>Cantidad</i>	<i>Precio Unit.</i>	<i>Precio Total</i>
Material			
Mini DK2 / LPC1768	1	44 €	44,00 €
J-Link EDU Segger Microcontroller	1	111,72 €	111,72 €
Cable conexionado	20	0,06 €	1,20 €
Porta pilas AA (4)	1	1 €	1,00 €
Pilas AA	4	0,26 €	1,04 €
Interruptor	1	0,5 €	0,50 €
Mano de obra			
Ingeniero Técnico Industrial	160	35 €/h	5600 €
Técnico Superior en Mecatrónica	8	16 €/h	128 €
Total			5.887,46 €

Tabla 23. Costes tarjeta de control central

8.3.6 Tarjetas de Potencia

<i>Descripción</i>	<i>Cantidad</i>	<i>Precio Unit.</i>	<i>Precio Total</i>
Material			
Monster Moto Shield – Dual VNH2SP30 [18]	1	5,78 €	5,78 €
Ventilador 12 V 50 x 50 mm	1	1,75 €	1,75 €
Clemas	2	0,20 €	0,40 €
Disipador Aluminio	2	0,40 €	0,80 €
Pieza Sujeción Ventilador (3 g)	1	0,02 €/g	0,06 €
Pieza Unión Ventilador (2 g)	1	0,02 €/g	0,04 €
Placa de Soldadura 10 x 10 mm	1	2,5 €	2,50 €
Array Pines Hembra	1	0,10 €	0,10 €
Zócalo 16 pines	2	0,60 €	1,20 €
Integrado LM324-N	2	0,50 €	1,00 €
Cable prototipado	1 m	0,20 €/m	0,20 €
Mano de obra			
Ingeniero Técnico Industrial	16	35 €/h	560 €
Técnico Superior en Mecatrónica	24	16 €/h	384 €
Total			957,83 €

Tabla 24. Costes tarjetas de potencia

8.3.7 Electrónica Encoder v.1

<i>Descripción</i>	<i>Cantidad</i>	<i>Precio Unit.</i>	<i>Precio Total</i>
Material			
IR LED	1	0,20 €	0,20 €
Dual Phototransistor	1	0,15 €	0,15 €
Resistencia 1K	1	0,10 €	0,10 €
Resistencias 10K	2	0,10 €	0,20 €
Potenciómetro 100K	2	0,25 €	0,50 €
Integrado LM324-N	1	0,50 €	0,50 €
Mini-Protoboard	1	2 €	2,00 €
Cable prototipado	0,2 m	0,20 €/m	0,04 €
Cable conexionado	5	0,06 €	0,30 €
Mano de obra			
Ingeniero Técnico Industrial	16	35 €/h	560 €
Técnico Superior en Mecatrónica	8	16 €/h	128 €
Total			691,99 €

Tabla 25. Costes electrónica encoder v.1

8.3.8 Alimentación

<i>Descripción</i>	<i>Cantidad</i>	<i>Precio Unit.</i>	<i>Precio Total</i>
Material			
ZIPPY Compact 2700mAh 3s 40c Lipo Pack [19]	2	20,16 €	40,32 €
Fire Retardant LiPoly Battery Bag [8]	2	2,35 €	4,70 €
Cargador iMAX-B6 [8]	1	38,99 €	38,99 €
Fusible 16 A	2	0,20 €	0,40 €
Porta Fusible	1	1 €	1,00 €
Cable H07V-K 1 x 1,5	1 m	0,31 €/m	0,31 €
Interruptor	1	1,50 €	1,50 €
Pieza Sujeción Interruptor (2 g)	1	0,02 €/g	0,04 €
Seta de Seguridad [8]	1	4,81 €	4,81 €
Mano de obra			
Ingeniero Técnico Industrial	8	35 €/h	280 €
Técnico Superior en Mecatrónica	16	16 €/h	256 €
Total			628,07 €

Tabla 26. Costes alimentación

8.3.9 Electrónica Medición de Batería

<i>Descripción</i>	<i>Cantidad</i>	<i>Precio Unit.</i>	<i>Precio Total</i>
Material			
Condensador 10 μ F	1	0,10 €	0,10 €
Potenciómetro 100K	2	0,25 €	0,50 €
Integrado LM324-N	2	0,50 €	1,00 €
Clemas	1	0,20 €	0,20 €
Placa de soldadura 5 x 5 mm	1	1,5 €	1,50 €
Cable prototipado	0,10 m	0,20 €/m	0,02 €
Cable conexionado	1	0,06 €	0,06 €
Mano de obra			
Ingeniero Técnico Industrial	4	35 €/h	140 €
Técnico Superior en Mecatrónica	8	16 €/h	128 €
Total			271,38 €

Tabla 27. Costes electrónica medición de batería

8.3.10 Sensores de Distancia

<i>Descripción</i>	<i>Cantidad</i>	<i>Precio Unit.</i>	<i>Precio Total</i>
Sensor Distancia Sharp GP2Y0A21YK0F [18]	4	5 €	20,00 €
Pieza Sujeción Sensor Distancia (2 g)	4	0,02 €/g	0,16 €
Integrados LM7805	4	0,50 €	2,00 €
Placa de soldadura 5 x 5 mm	1	1,5 €	1,50 €
Array Pines Hembra	1	0,10 €	0,10 €
Cable conexionado	24	0,06 €	1,44 €
Mano de obra			
Ingeniero Técnico Industrial	8	35 €/h	280 €
Técnico Superior en Mecatrónica	8	16 €/h	128 €
Total			433,20 €

Tabla 28. Costes sensores de distancia

8.4 Coste de las amortizaciones

<i>Equipo</i>	<i>Precio</i>	<i>Duración</i>	<i>Uso</i>	<i>Precio Total</i>
Ordenador Asus	1200 €	5 años	6 meses	120 €
Impresora 3D Prusa i3	300 €	3 años	2 semanas	3,83 €
Osciloscopio Digital 2 Canales	3995 €	10 años	5 meses	166,46 €
Fuente de Alimentación	179,50 €	10 años	5 meses	7,48 €
Polímetro	25 €	10 años	5 meses	1,04 €
Soldador 11 W	21 €	5 años	2 semanas	0,16 €
Total				299 €

Tabla 29. Costes de amortizaciones: equipo

<i>Software</i>	<i>Precio</i>	<i>Duración</i>	<i>Uso</i>	<i>Precio Total</i>
Matlab 2016-b	-	-	3 meses	0 €
Autodesk Inventor 2016	-	-	3 meses	0 €
Keil	-	-	4 meses	0 €
Slic3r	Gratuito	-	2 semanas	0 €
Pronterface	Gratuito	-	2 semanas	0 €
Paquete Microsoft Office	-	-	8 meses	0 €
Microsoft Windows 8	-	-	16 meses	0 €
Total				0 €

Tabla 30. Costes de amortizaciones: software

8.5 Coste global

<i>Descripción</i>	<i>Coste</i>
Presupuesto material (con mano de obra)	18.302,03 €
Amortizaciones de equipos	299 €
Amortizaciones de software	0 €
Total	18.601,03 €

Tabla 31. Coste global

9. Manual de usuario

9.1 Medidas de seguridad

Durante el manejo o utilización de proyectos de este tipo siempre existen unos ciertos riesgos, pudiendo ser de descargas eléctricas, lesiones e incendio. Para reducir estos riesgos y poder trabajar con total seguridad se exponen a continuación las normas que se deben respetar:

1. Debe trabajar en un entorno adecuado: no se debe trabajar o dejar parte de los elementos del robot expuestos a la lluvia, viento o ambientes muy húmedos. De igual modo no se puede dejar al lado de fuentes de calor como radiadores u otros equipos, ya que existe el riesgo de incendio o explosión.
2. El proyecto y robot completo debe ser guardado en un lugar idóneo: cuando no se está haciendo uso de este debe ser guardado en un lugar seco y deben desconectarse todas las baterías.
3. Nunca debe arrastrarse o empujarse el robot ya que la reductora de los motores y sus elementos asociados podrían verse dañados.
4. Siempre que vaya a ser usado debe comprobarse el estado de las piezas. Cualquier pieza dañada o mal colocada puede ser peligrosa. Compruébese la buena alineación de los elementos giratorios.
5. El estado de las baterías debe ser comprobado antes de cada uso y solo usarlas si se encuentran correctamente. Las baterías Li-Po sobre-descargadas pueden hincharse, incendiarse y explotar.
6. Si se desconoce algún componente de este proyecto no debe ser manipulado. Existen componentes de alta potencia que pueden producir daños físicos.
7. Mantenga este proyecto fuera del alcance de los niños: tanto el robot como el entorno de trabajo puede ser peligroso para ellos.

Cualquier persona que manipule o utilice este proyecto sin tener en cuenta estas normas y las condiciones expuestas en el “*Pliego de condiciones*” de este trabajo se expone fuera de las garantías de seguridad personal o de los equipos.

Cualquier modificación que se realice en este proyecto fuera de las condiciones expuestas correrá bajo la responsabilidad del usuario autor de dichas modificaciones.

9.2 Manual de usuario

9.2.1 Encendido y conexionado

- **Encendido de la alimentación de potencia: batería**

Para dar energía a todo el sistema de tracción y su electrónica de potencia debe conectarse adecuadamente la batería (Figura 128) la cual solo tiene una posición posible de conexión, y accionar el interruptor que se muestra en la Figura 129. La posición “-“ es encendido y “O” es apagado.



Figura 128. Conexión de la batería

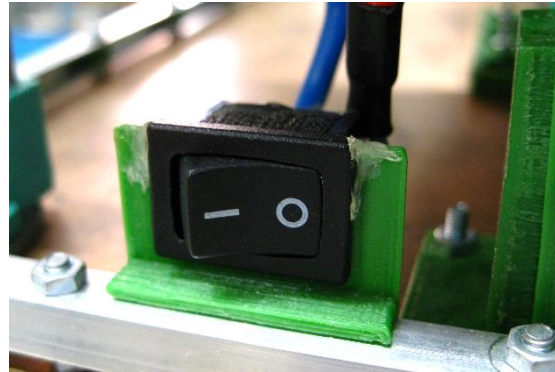


Figura 129. Interruptor de encendido de electrónica potencia

- **Encendido de la alimentación de la tarjeta de control y electrónica asociada**

La batería de esta tarjeta se conecta a ella mediante un conector Jack. Depende de la batería utilizada deberán respetarse las condiciones de uso de dicha batería.

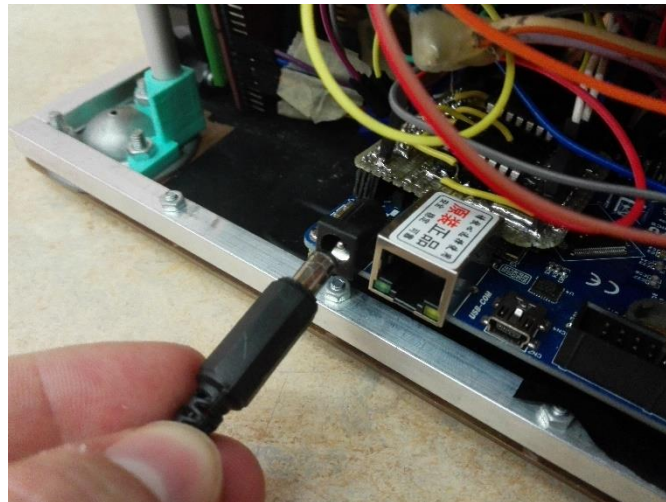


Figura 130. Conector jack de alimentación Mini-DK2

- **Conexión entre robot y ordenador**

La conexión debe realizarse mediante un cable mini USB – USB entre el puerto mini USB incorporado en la Mini-DK2 y un puerto USB del ordenador. La distancia mínima de este cable debe ser de 50 cm ya que de lo contrario podría entorpecer en las pruebas a realizar.

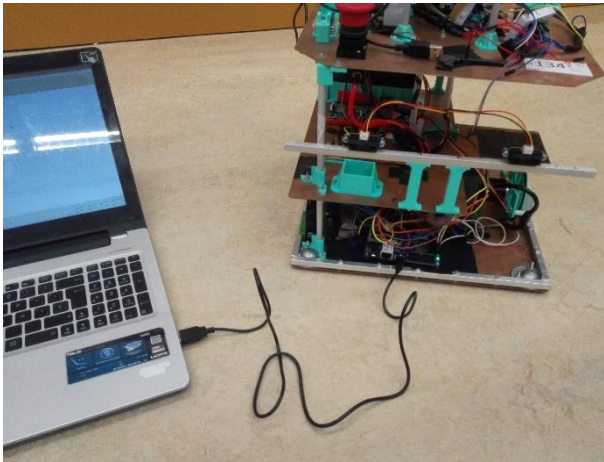


Figura 131. Conexión entre el ordenador y el proyecto

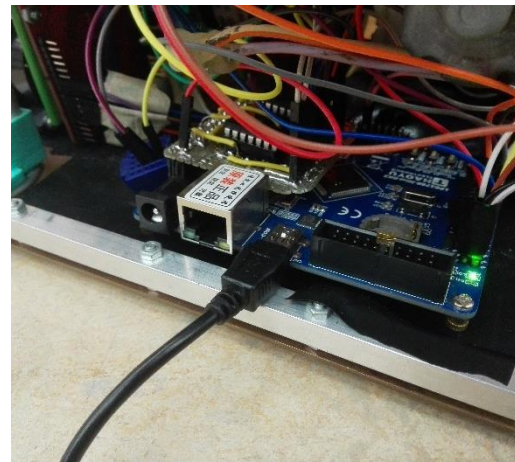


Figura 132. Conexión mini USB de Mini-DK2

- **Conexión entre elementos del robot**

Las conexiones de los elementos de baja potencia o de la electrónica asociada a la tarjeta de control se deben realizar con cable de conexionado de tipo “Arduino”.

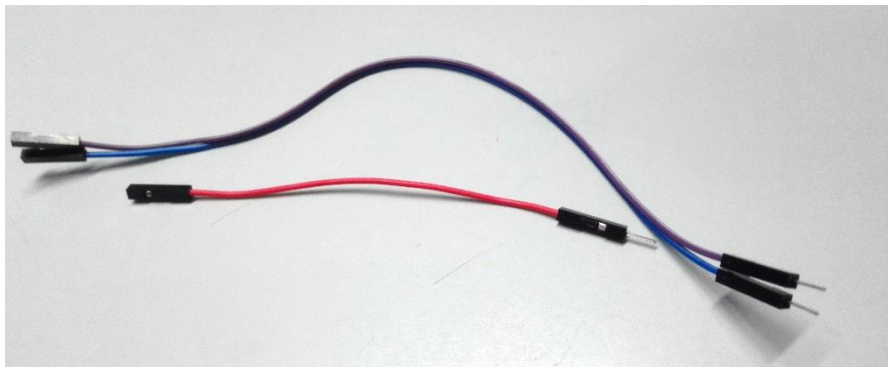
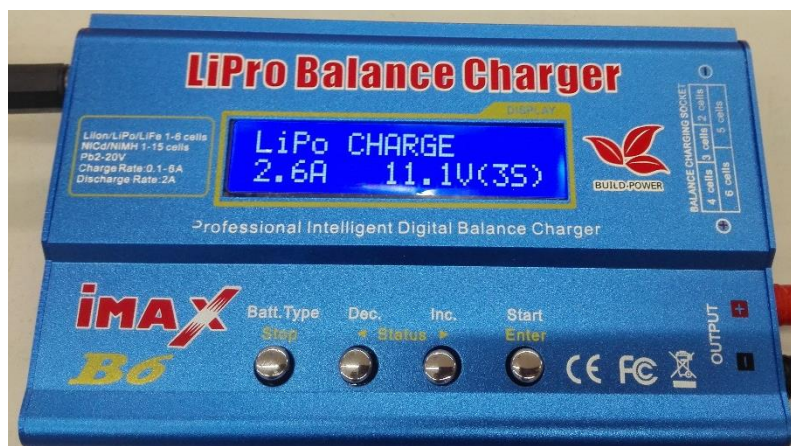


Figura 133. Cables de conexionado tipo "Arduino"

- **Carga de la batería**

La carga de la batería debe realizarse bajo las condiciones del cargador que se use. Este cargador debe ser específico para el tipo de batería que se esté usando, tal y como se detalla en el apartado “Pliego de condiciones”.



Figuraa 134. Configuración del cargador



Figura 135. Conexión para la carga de la batería

- **Apagado de emergencia**

El apagado de emergencia se realiza pulsando la seta de seguridad colocada en lo alto del robot (Figura 136). Una vez pulsada corta la alimentación de la batería de potencia y queda bloqueada. Para desbloquearla hay que girarla en sentido horario, recomendándose siempre que antes de hacerlo se ponga en posición de apagado el interruptor de potencia.

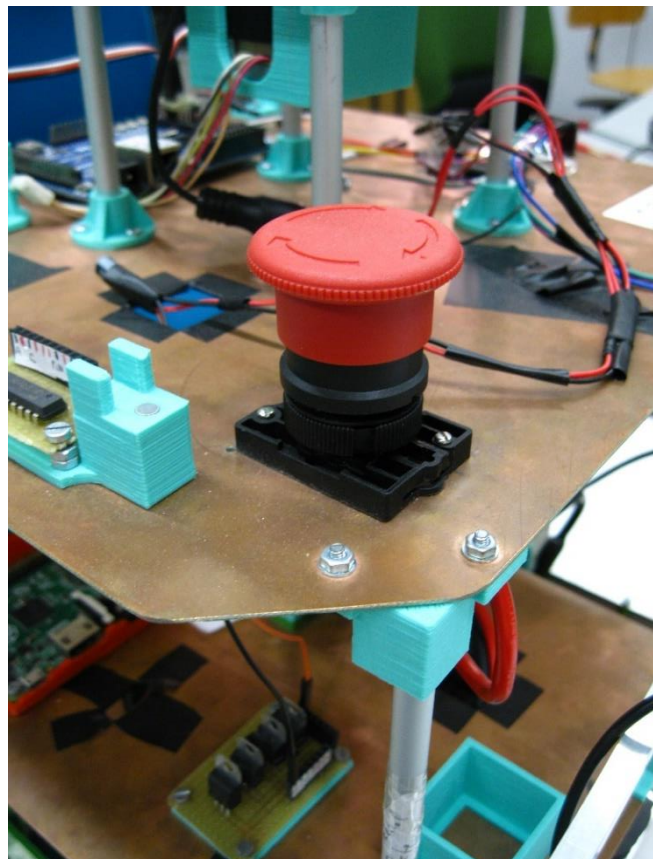


Figura 136. Seta de seguridad

9.2.2 Configuración del control PI

Este proyecto puede ser usado para distintos propósitos, por lo que los elementos del robot al que acompaña pueden cambiar. Esto significa que las condiciones en las que se diseñó el controlador pueden variar, debiéndolo hacer él también.

Para cambiar las constantes del controlador PI debemos disponer del entorno de programación Keil y al inicio del programa, donde se definen las constantes, modificar el valor de las constantes. Aparecen separadas como “PID – A” para el motor 1 y “PID – B” para el motor 2 (Figura 137).

```

39 //PID - A
40 const float Akp=0.37;
41 const float Aki=5*Ts;
42 const float Akd=6/Ts;
43 float Au0, Ay, Ae0, Ae1, Ar, Ar_aux, Arr;
44 uint8_t signo_A, sig_A_aux;
45
46 //PID - B
47 const float Bkp=0.322;
48 const float Bki=4.5*Ts;
49 const float Bkd=6/Ts;
50 float Bu0, By, Be0, Be1, Br, Br_aux, Brr;

```

Figura 137. Constantes PI dentro del programa

9.2.3 Protocolo de comunicación

Para realizar una comunicación con este proyecto de la forma descrita en el apartado “Diseño Software: Comunicación con la tarjeta de estrategia” a de disponerse de una tarjeta de estrategia con una salida de puerto serie.

El protocolo viene descrito en el apartado anteriormente mencionado. Las consideraciones que hay que tener en cuenta cuando se le envíen consignas de velocidad a este proyecto son las siguientes:

- A de enviarse la información como se muestra en la Tabla 13.

Recibo de la tarjeta de estrategia								
1º	2º	3º	4º	5º	6º	7º	8º	9º
Signo de consigna velocidad rueda 1 (“+” o “-“)	Consigna velocidad rueda 1: centenas	Consigna velocidad rueda 1: decenas	Consigna velocidad rueda 1: unidades	Signo de consigna velocidad rueda 2 (“+” o “-“)	Consigna velocidad rueda 2: centenas	Consigna velocidad rueda 2: decenas	Consigna velocidad rueda 2: unidades	Fin de paquete “\r” (retorno de carro)

Tabla 13. Paquete de recepción de la tarjeta de estrategia

- La información es procesada por este proyecto de la siguiente forma:

Consigna completa de velocidad rueda 1 → “1º dato” “2º dato 3º dato 4º dato” mm/s

Consigna completa de velocidad rueda 2 → “5º dato” “6º dato 7º dato 8º dato” mm/s

Ejemplo:

Paquete de datos enviado: “+050-050”

Consigna velocidad rueda 1 → + 050 mm/s = + 50 mm/s

Consigna velocidad rueda 2 → - 050 mm/s = - 50 mm/s

En este ejemplo el robot giraría en el sitio en sentido horario.

Además de las consignas que se le envían a este proyecto, también se recibe por parte de él un dato importante: el “nivel” de la batería. Es un dato de tipo bool que nos indica si la batería ha sobrepasado el límite de seguridad para su uso (ver detalladamente en el apartado “*Diseño Software: Software de Control PI: Adaptación al nivel de batería*”).

Este dato se encuentra en la posición 5º del paquete de datos que envía este proyecto a la estrategia:

Envío hacia la tarjeta de estrategia					
1º	2º	3º	4º	5º	6º
Pulsos encoder 1	Carácter “,”	Pulsos encoder 2	Carácter “,”	Nivel de batería de seguridad	Fin de paquete “\n” (salto de línea)

Tabla 1. Paquete de envío hacia la tarjeta de estrategia

En este paquete también se encuentra la información de los pulsos de cada encoder en cada ciclo de muestreo. Para la implementación correcta de la odometría usando esta información véase el apartado “*Diseño Software: Trayectorias y odometría*”.

10. Bibliografía

- [1] <http://roboticday.org/> (Última vez consultada en 20/02/2018).
- [2] Baliñas Santos, Javier (2016) (TFC) Manual de referencia para el desarrollo de robots de Eurobot.
- [3] http://www.rcva.fr/images/stories/site/cours/10ansdexperience/devoir_de_vacances.pdf (Última vez consultada en 16/07/2016).
- [4] <https://es.wikibooks.org/wiki/F%C3%ADsica/Est%C3%A1tica/Rozamiento> (Última vez consultada en 20/02/2018).
- [5] Rafael Salazar Arcucci, Marcelo (2006) (TFC) Caponata, un robot que juega al golf: control del movimiento.
- [6] <http://blog.nidec-avtron.com/encoders/how-optical-encoders-work> (Última vez consultada en 20/02/2018).
- [7] <https://www.sinotech.com/products/motors/optical-encoders-overview/> (Última vez consultada en 20/02/2018).
- [8] <https://www.amazon.es> Encoder H38S600B (Última vez consultada en 20/02/2018).
- [9] <https://www.bricodepot.es/> Ruedas Polipropileno Ø 100 (Última vez consultada en 20/02/2018).
- [10] LPC176x/5x User manual, Rev. 3 — 20 December 2013. NXP, 2010.
- [11] <http://www.instructables.com/id/Monster-Motor-Shield-VNH2SP30/> (Última vez consultada en 20/02/2018).
- [12] <http://controlsystemslab.com/QkQjZ/discrete-time-pid-controller-implementation/> (Última vez consultada en 05/02/2017).
- [13] Espinosa, F., Rodríguez, Fco. J., “Tema 1: Revisión de técnicas de control digital”, Asignatura Ingeniería de Control Electrónico. Universidad de Alcalá. Curso 2016-17.
- [14] Carbajo Martín, M., “Tema 3: Modelos”, Asignatura Ingeniería de Control I. Universidad de Alcalá. Curso 2015-16.
- [15] <https://www.pccomponentes.com> Filamento PLA (Última vez consultada en 21/02/2018).
- [16] <http://www.desguaceslatorre.es/> Motores Bosch 12V 0-130-822-190 (Última vez consultada en 21/02/2018).
- [17] <https://world-bearing.com/es/> Rodamientos Ø 8 – Ø 16 (Última vez consultada en 21/02/2018).

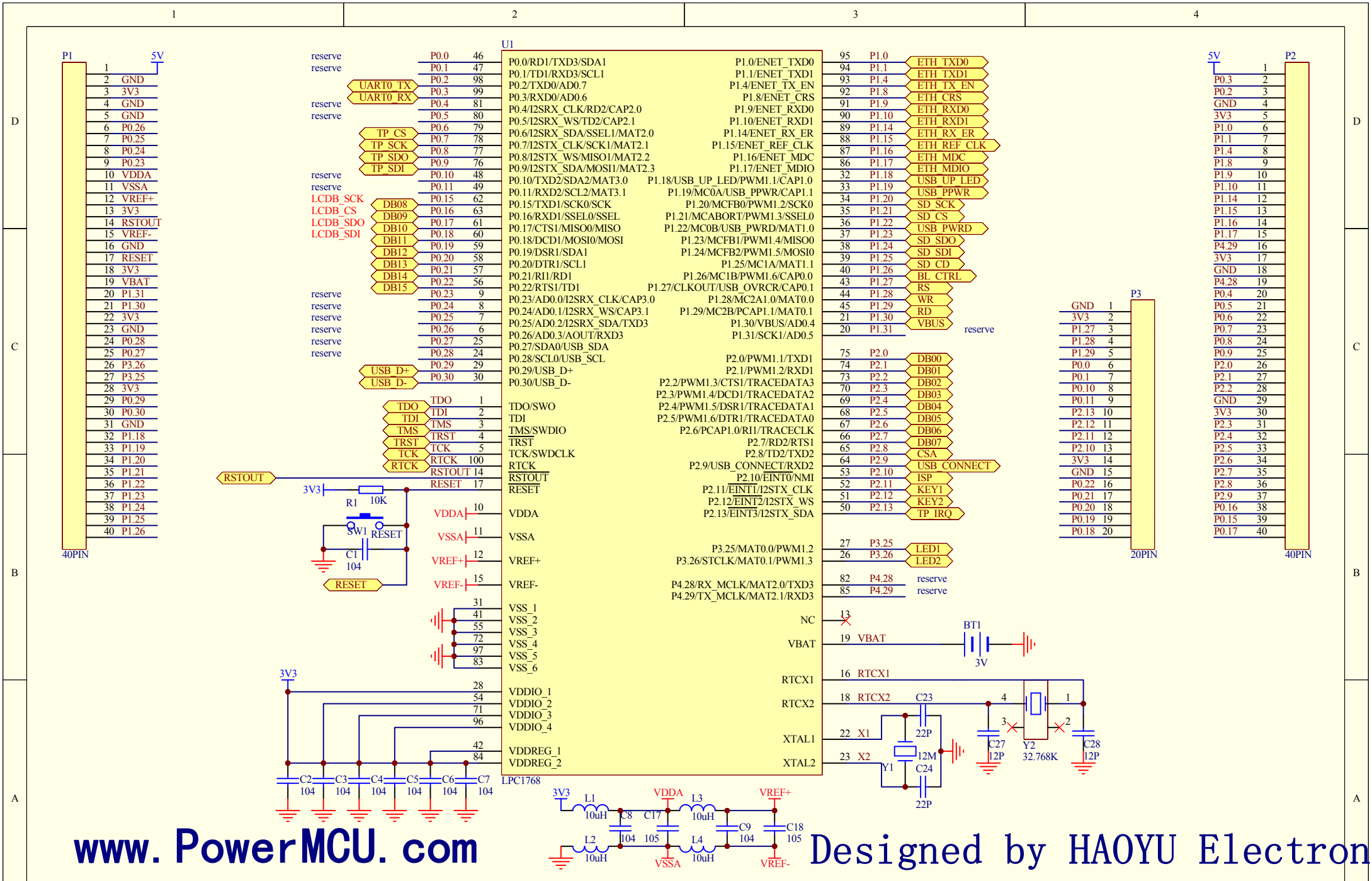
[18] https://es.aliexpress.com/es_home.htm (Última vez consultada en 21/02/2018).

[19] https://hobbyking.com/es_es/?store=es_es (Última vez consultada en 21/02/2018).

11. Anexos

11.1 Planos electrónica: Tarjetas adquiridas

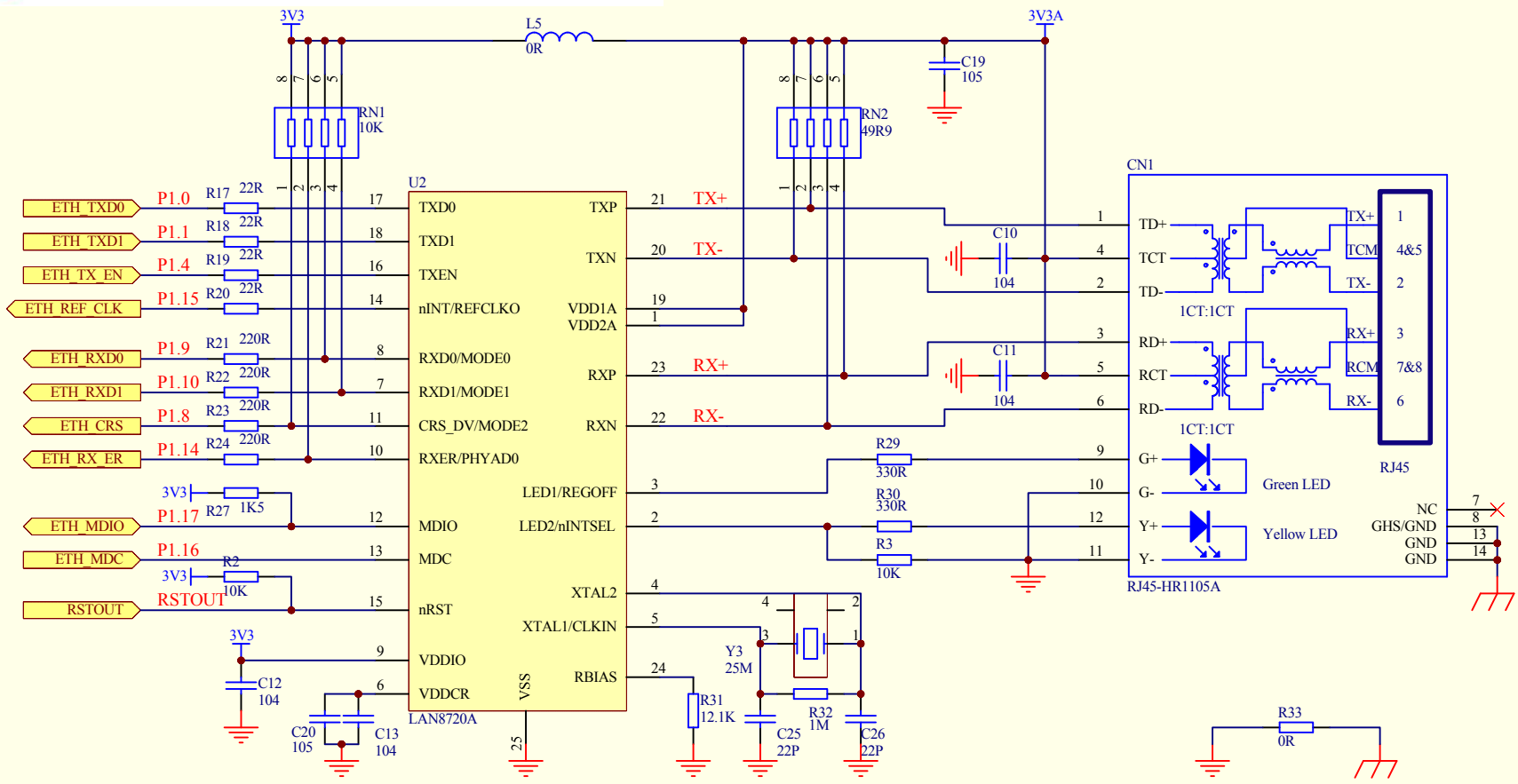
- **Mini-DK2: LPC 1768**



www.PowerMCU.com

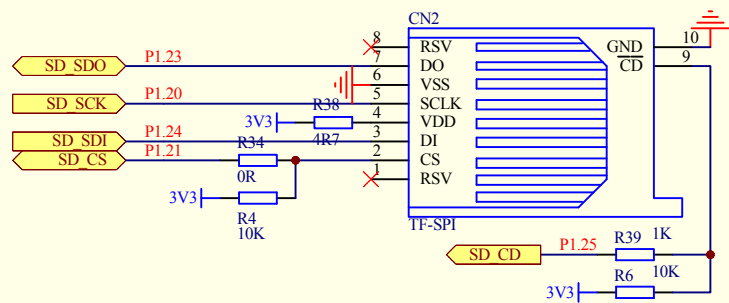
Designed by HAOYU Electroni

HAOYU ELECTRONICS

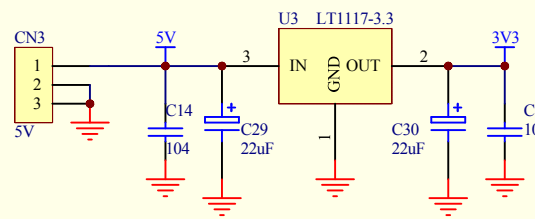


Designed by HAOYU Electronics

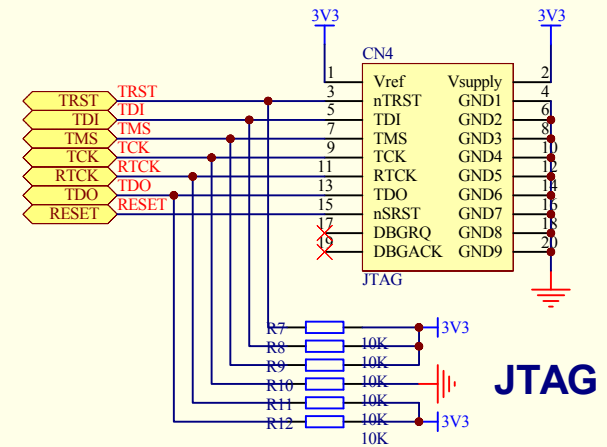
www.PowerMCU.com



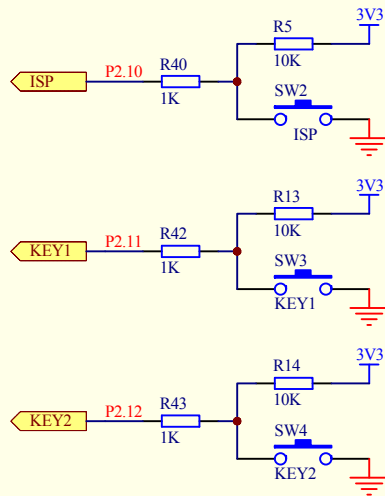
SD_CARD



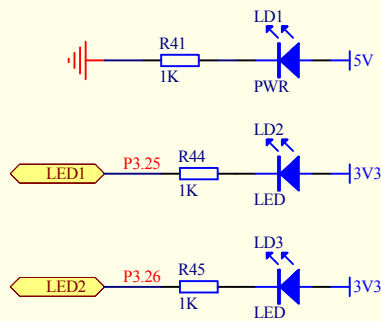
Power



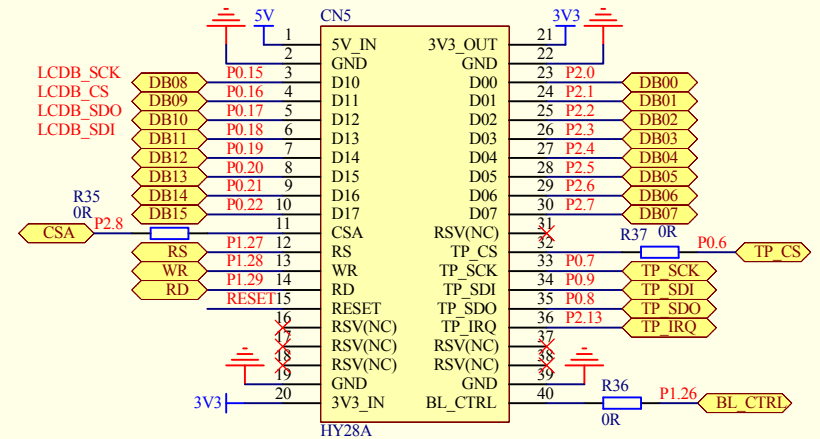
JTAG



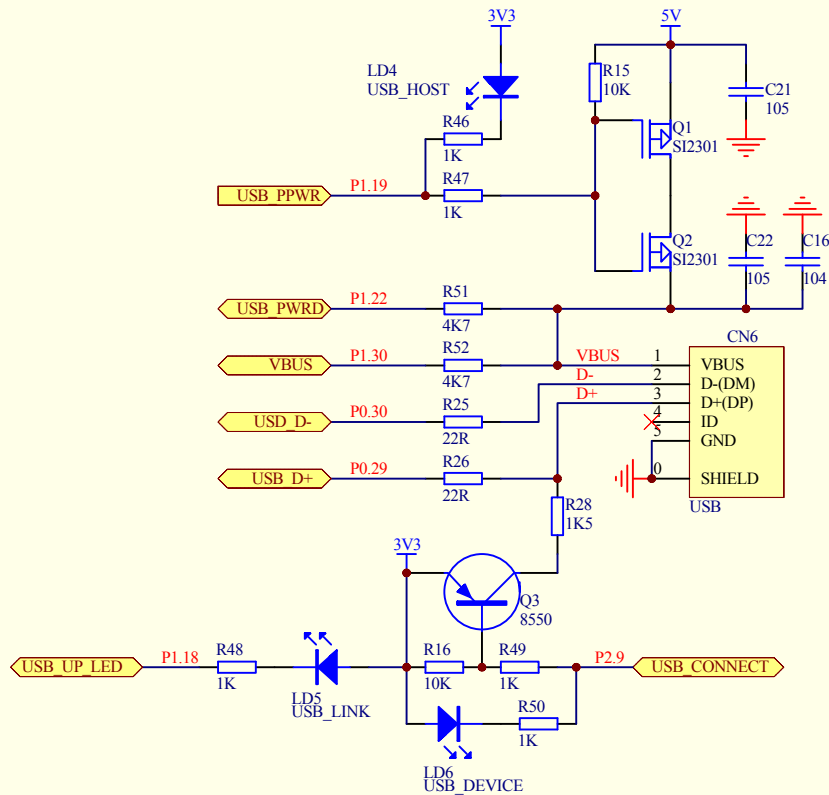
USER_KEY



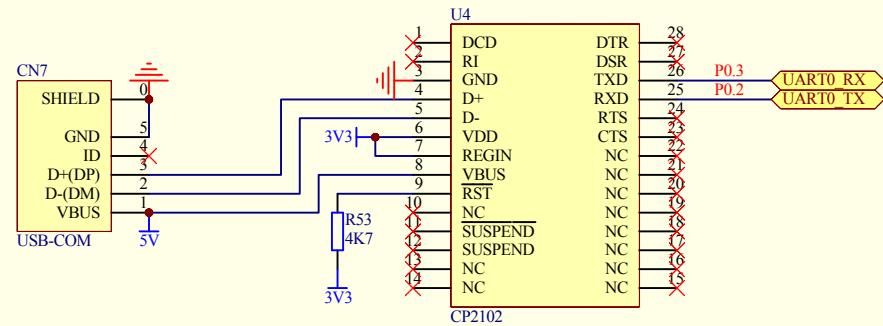
LEDs



LCD



USB Host/Device

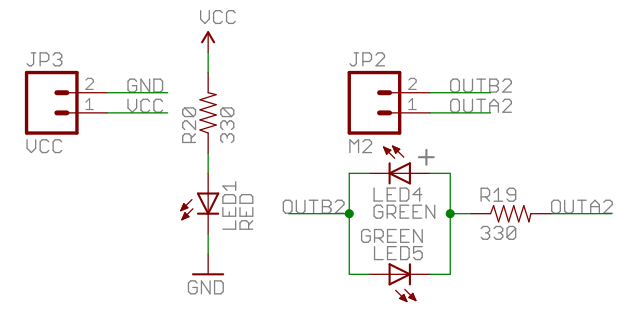
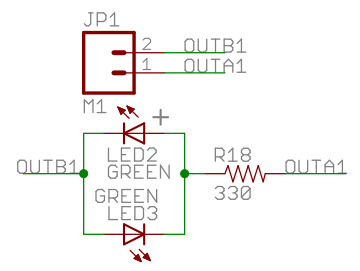
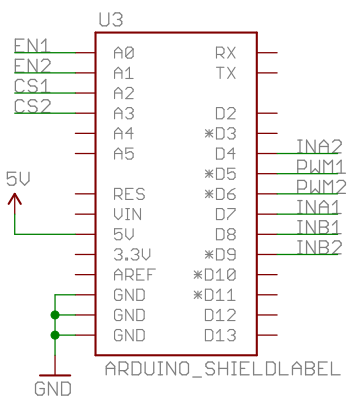
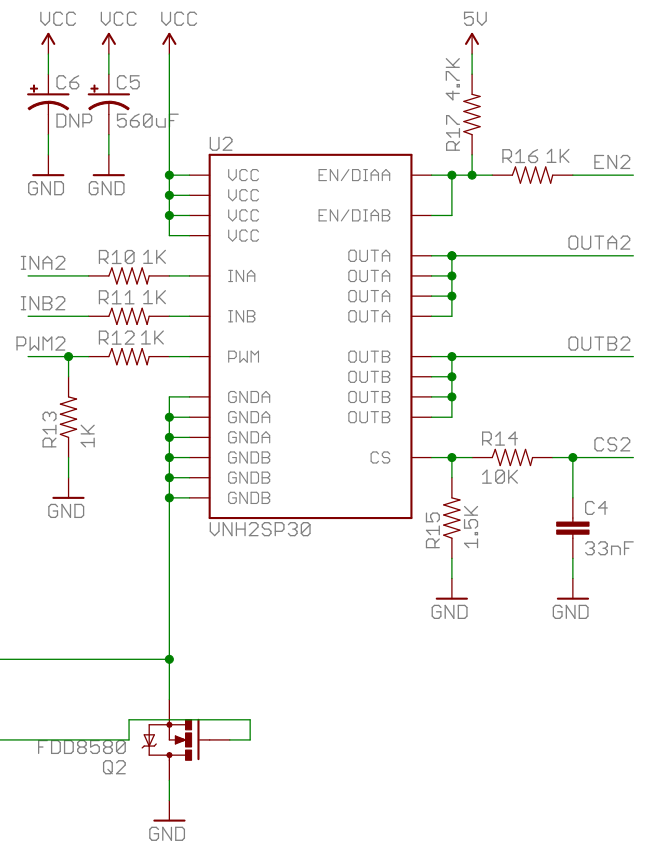
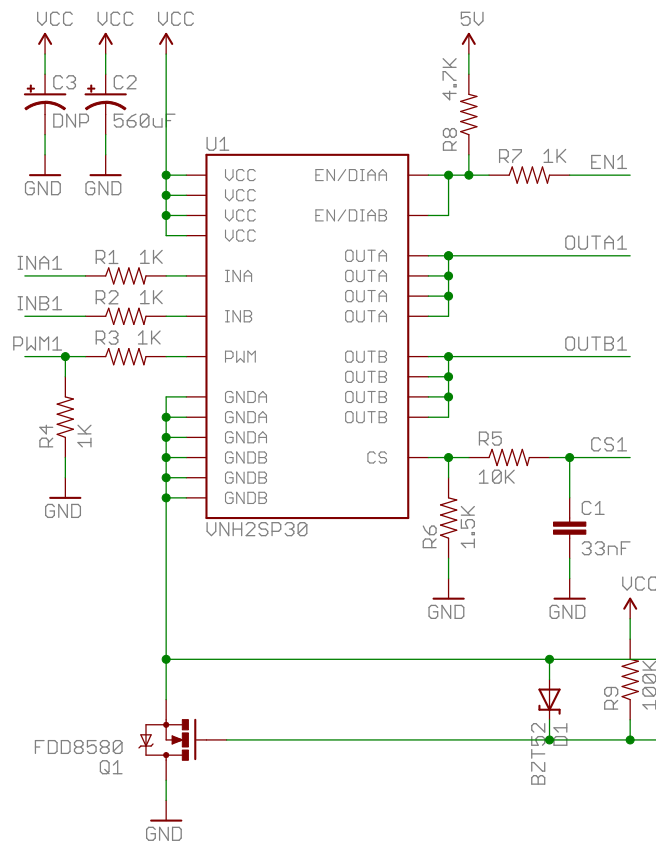


USB to UART

Designed by HAOYU Electronics

www.PowerMCU.com

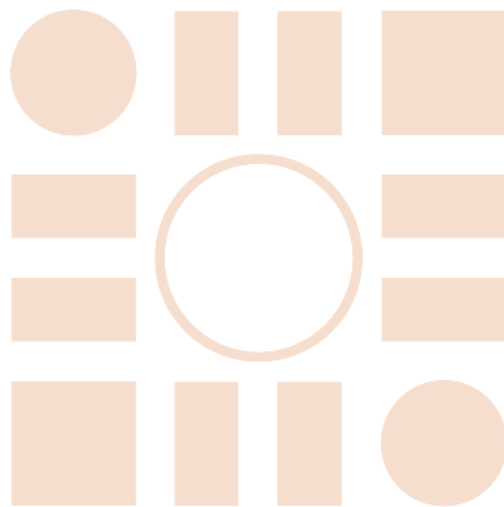
- **Monster Moto Shield – Dual VNH2SP30**



⊗ ⊗	
TITLE: MonsterMoto-Shield-v12	
SFE	
Document Number:	REV:
Date: 11/3/2010 12:02:08 PM	Sheet: 1/1

Released under the Creative Commons Attribution Share-Alike 3.0 License
<http://creativecommons.org/licenses/by-sa/3.0>
 Design by: Nate Bernstein
 Jim Lindblom

Universidad de Alcalá
Escuela Politécnica Superior



ESCUELA POLITECNICA
SUPERIOR



Universidad
de Alcalá