



COMILLAS
UNIVERSIDAD PONTIFICIA

ICAI

GRADO EN INGENIERÍA EN
TECNOLOGÍAS INDUSTRIALES

TRABAJO FIN DE GRADO

DESARROLLO, CONSTRUCCIÓN Y PROGRAMACIÓN
DE UN ROBOT AUTÓNOMO PARA PARTICIPAR EN LA
COPA DE ROBÓTICA DE FRANCIA

Autor: Daniel Horcajo de la Cruz

Director: Nicolas Boullis

Madrid

Agosto de 2019

AUTORIZACIÓN PARA LA DIGITALIZACIÓN, DEPÓSITO Y DIVULGACIÓN EN RED DE PROYECTOS FIN DE GRADO, FIN DE MÁSTER, TESINAS O MEMORIAS DE BACHILLERATO

1º. Declaración de la autoría y acreditación de la misma.

El autor D. DANIEL HORCAJO DE LA CRUZ DECLARA ser el titular de los derechos de propiedad intelectual de la obra: *Desarrollo, construcción y programación de un robot autónomo para participar en la Copa de Francia de Robótica*, que ésta es una obra original, y que ostenta la condición de autor en el sentido que otorga la Ley de Propiedad Intelectual.

2º. Objeto y fines de la cesión.

Con el fin de dar la máxima difusión a la obra citada a través del Repositorio institucional de la Universidad, el autor **CEDE** a la Universidad Pontificia Comillas, de forma gratuita y no exclusiva, por el máximo plazo legal y con ámbito universal, los derechos de digitalización, de archivo, de reproducción, de distribución y de comunicación pública, incluido el derecho de puesta a disposición electrónica, tal y como se describen en la Ley de Propiedad Intelectual. El derecho de transformación se cede a los únicos efectos de lo dispuesto en la letra a) del apartado siguiente.

3º. Condiciones de la cesión y acceso

Sin perjuicio de la titularidad de la obra, que sigue correspondiendo a su autor, la cesión de derechos contemplada en esta licencia habilita para:

- a) Transformarla con el fin de adaptarla a cualquier tecnología que permita incorporarla a internet y hacerla accesible; incorporar metadatos para realizar el registro de la obra e incorporar “marcas de agua” o cualquier otro sistema de seguridad o de protección.
- b) Reproducirla en un soporte digital para su incorporación a una base de datos electrónica, incluyendo el derecho de reproducir y almacenar la obra en servidores, a los efectos de garantizar su seguridad, conservación y preservar el formato.
- c) Comunicarla, por defecto, a través de un archivo institucional abierto, accesible de modo libre y gratuito a través de internet.
- d) Cualquier otra forma de acceso (restringido, embargado, cerrado) deberá solicitarse expresamente y obedecer a causas justificadas.
- e) Asignar por defecto a estos trabajos una licencia Creative Commons.
- f) Asignar por defecto a estos trabajos un HANDLE (URL *persistente*).

4º. Derechos del autor.

El autor, en tanto que titular de una obra tiene derecho a:

- a) Que la Universidad identifique claramente su nombre como autor de la misma
- b) Comunicar y dar publicidad a la obra en la versión que ceda y en otras posteriores a través de cualquier medio.
- c) Solicitar la retirada de la obra del repositorio por causa justificada.
- d) Recibir notificación fehaciente de cualquier reclamación que puedan formular terceras personas en relación con la obra y, en particular, de reclamaciones relativas a los derechos de propiedad intelectual sobre ella.

5º. Deberes del autor.

El autor se compromete a:

- a) Garantizar que el compromiso que adquiere mediante el presente escrito no infringe ningún derecho de terceros, ya sean de propiedad industrial, intelectual o cualquier otro.
- b) Garantizar que el contenido de las obras no atenta contra los derechos al honor, a la intimidad y a la imagen de terceros.
- c) Asumir toda reclamación o responsabilidad, incluyendo las indemnizaciones por daños, que pudieran ejercitarse contra la Universidad por terceros que vieran infringidos sus derechos e intereses a causa de la cesión.

- d) Asumir la responsabilidad en el caso de que las instituciones fueran condenadas por infracción de derechos derivada de las obras objeto de la cesión.

6º. Fines y funcionamiento del Repositorio Institucional.

La obra se pondrá a disposición de los usuarios para que hagan de ella un uso justo y respetuoso con los derechos del autor, según lo permitido por la legislación aplicable, y con fines de estudio, investigación, o cualquier otro fin lícito. Con dicha finalidad, la Universidad asume los siguientes deberes y se reserva las siguientes facultades:

- La Universidad informará a los usuarios del archivo sobre los usos permitidos, y no garantiza ni asume responsabilidad alguna por otras formas en que los usuarios hagan un uso posterior de las obras no conforme con la legislación vigente. El uso posterior, más allá de la copia privada, requerirá que se cite la fuente y se reconozca la autoría, que no se obtenga beneficio comercial, y que no se realicen obras derivadas.
- La Universidad no revisará el contenido de las obras, que en todo caso permanecerá bajo la responsabilidad exclusiva del autor y no estará obligada a ejercitar acciones legales en nombre del autor en el supuesto de infracciones a derechos de propiedad intelectual derivados del depósito y archivo de las obras. El autor renuncia a cualquier reclamación frente a la Universidad por las formas no ajustadas a la legislación vigente en que los usuarios hagan uso de las obras.
- La Universidad adoptará las medidas necesarias para la preservación de la obra en un futuro.
- La Universidad se reserva la facultad de retirar la obra, previa notificación al autor, en supuestos suficientemente justificados, o en caso de reclamaciones de terceros.

Madrid, a 23 de julio de 2019.

ACEPTA

DANIEL HORCAJO DE LA CRUZ

Fdo..........

Motivos para solicitar el acceso restringido, cerrado o embargado del trabajo en el Repositorio Institucional:

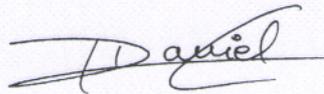
I, hereby, declare that I am the only author of the project report with title:

DEVELOPMENT, BUILDING AND PROGRAMMING OF AN
AUTONOMOUS ROBOT TO PARTICIPATE IN THE FRENCH ROBOTICS CUP
"EUROBOT"

which has been submitted to ICAI School of Engineering of Comillas Pontifical University in the academic year 2018/2019. This project is original, has not been submitted before for any other purpose and has not been copied from any other source either fully or partially. All information sources used have been rightly acknowledged.

Fdo.: DANIEL HORCAJO DE LA CRUZ

Date: 10/04/2019



I authorize the submission of this project

PROJECT SUPERVISOR



Fdo.: Nicolas BOULLIS

Date: April 14th, 2019



COMILLAS
UNIVERSIDAD PONTIFICIA

ICAI

GRADO EN INGENIERÍA EN
TECNOLOGÍAS INDUSTRIALES

TRABAJO FIN DE GRADO

DESARROLLO, CONSTRUCCIÓN Y PROGRAMACIÓN
DE UN ROBOT AUTÓNOMO PARA PARTICIPAR EN LA
COPA DE ROBÓTICA DE FRANCIA

Autor: Daniel Horcajo de la Cruz

Director: Nicolas Boullis

Madrid

Agosto de 2019

DESARROLLO, CONSTRUCCIÓN Y PROGRAMACIÓN DE UN ROBOT AUTÓNOMO PARA PARTICIPAR EN LA COPA DE ROBÓTICA DE FRANCIA

Autor: Horcajo de la Cruz, Daniel.

Director: Boullis, Nicolas.

Entidad colaboradora: Club de Robotique de CentraleSupélec.

RESUMEN DEL PROYECTO

INTRODUCCIÓN

Este proyecto se realiza en colaboración con el Club de Robótica de CentraleSupélec (CRoCS) de París, donde cada año un equipo de la universidad participa en la Copa de Robótica de Francia. En esta, los equipos participantes se enfrentan de dos en dos en un mismo terreno de juego, en el cual deberán llevar a cabo una serie de tareas para obtener el máximo número de puntos antes de que se agoten los noventa segundos de tiempo reglamentario.

El objetivo del proyecto es, pues, desarrollar la plataforma móvil autónoma regulada en posición del robot perteneciente al equipo de la universidad CentraleSupélec para participar en la edición 2019 de esta competición, “La fábrica de átomos”, donde los robots deberán clasificar, desplazar y agrupar distintos tipos de átomos, representados por discos de hockey de diferente peso y color.

ESTADO DE LA TÉCNICA

Debido a las diversas tareas a realizar en el terreno de juego, una parte crucial del robot es que pueda desplazarse de una forma rápida a la vez que precisa, ya que esto facilitará el trabajo del resto del equipo; por ello, se estudian las principales opciones y configuraciones existentes que faciliten el logro este objetivo. De entre dichas opciones, entre las cuales se encuentra la clásica tracción diferencial a dos ruedas, la elegida será la configuración Kiwi drive con ruedas omnidireccionales.



Figura 1. Rueda omnidireccional.
Fuente: Pinterest.

Explicado brevemente, una configuración Kiwi drive está formada por una base con tres ruedas orientadas en una disposición de 120° las unas respecto de las otras, como se muestra en la imagen 2. Las ruedas omnidireccionales, al estar compuestas por multitud de rodillos a lo

largo de su perímetro, permiten el movimiento no solo en la dirección habitual, sino también en la del eje de la rueda.

Así pues, haciendo uso de la configuración Kiwi drive y de la versatilidad de las ruedas omnidireccionales, el robot es capaz de desplazarse en cualquier dirección del plano sin necesidad de realizar un giro previo (característica denominada “holonómica”), tan solo controlando las velocidades individuales de las tres ruedas.

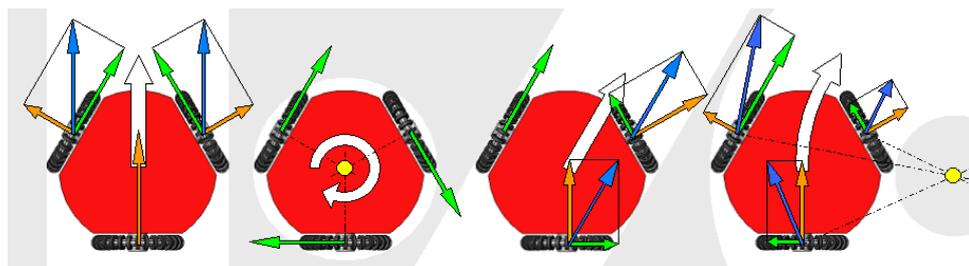


Figura 2. Representación vectorial del movimiento de un Kiwi drive con ruedas omnidireccionales.

Fuente: Wikimedia Commons.

La desventaja de este modo de desplazamiento es, sin embargo, la problemática de orientación y comandabilidad que presenta, ya que al no corresponder el giro de los motores con el movimiento global del robot, es necesaria una etapa de cálculo previo para pasar de las velocidades independientes de cada motor a la velocidad absoluta del robot.

CINEMÁTICA DEL KIWI DRIVE

Como se ha mencionado anteriormente, la comanda de los motores necesaria para hacer desplazarse al robot en una cierta dirección no es algo trivial.

Por ello, con ayuda de la figura 3, podemos calcular las matrices de cambio de base que nos permiten hacer el cambio de las velocidades individuales de las ruedas $[v_1, v_2, v_3]^T$ a la global $[V_x, V_y, \Omega]^T$ del robot:

$$\begin{bmatrix} v_1 \\ v_2 \\ v_3 \end{bmatrix} = \underbrace{\begin{bmatrix} -\sin \theta & \cos \theta & R \\ -\sin \left(\theta + \frac{2\pi}{3} \right) & \cos \left(\theta + \frac{2\pi}{3} \right) & R \\ -\sin \left(\theta + \frac{4\pi}{3} \right) & \cos \left(\theta + \frac{4\pi}{3} \right) & R \end{bmatrix}}_A \begin{bmatrix} V_x \\ V_y \\ \Omega \end{bmatrix}$$

Figura 4. Matriz A de cambio de base para obtener las velocidades lineales individuales de las ruedas dada la velocidad lineal y angular del robot.

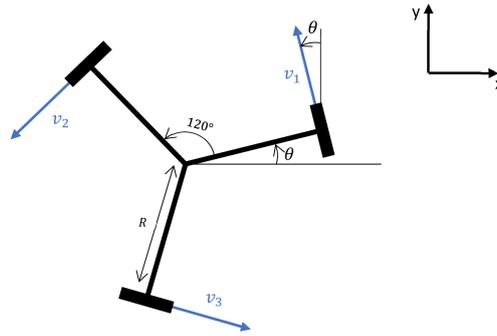


Figura 3. Representación vectorial de las velocidades independientes de cada rueda para realizar un cambio de base.

Fuente: Autor.

$$\begin{bmatrix} V_x \\ V_y \\ \Omega \end{bmatrix} = \frac{1}{3} \underbrace{\begin{bmatrix} -2 \sin \theta & -2 \sin \left(-\theta + \frac{\pi}{3}\right) & 2 \sin \left(\theta + \frac{\pi}{3}\right) \\ 2 \cos \theta & -2 \cos \left(-\theta + \frac{\pi}{3}\right) & -2 \cos \left(\theta + \frac{\pi}{3}\right) \\ \frac{1}{R} & \frac{1}{R} & \frac{1}{R} \end{bmatrix}}_{A^{-1}} \begin{bmatrix} v_1 \\ v_2 \\ v_3 \end{bmatrix}$$

Figura 5. Matriz de cambio de base A^{-1} para obtener la velocidad lineal y angular del robot dadas las velocidades lineales individuales de las ruedas.

COMPONENTES

Los motores que propulsan la base son unos Faulhaber 2642W 012 CXR de corriente continua a 12V. Incluidos en estos, encontramos unos reductores de la misma marca con una relación de reducción $\eta = 1 : 23$ que permitirán reducir la velocidad a la salida del motor a una más adecuada para el robot.

Incluido también en cada motor, encontramos un encoder óptico HEDS 5540 C02 que permitirá medir el giro de cada rueda, lo cual podrá ser más tarde utilizado para calcular el desplazamiento del robot en su conjunto.

Para ello, sabiendo que cada encoder trabaja con dos señales (A y B) en cuadratura y que tiene una resolución de 100LPR, obtenemos que el giro realizado por un motor (en grados) viene dado por la siguiente expresión:

$$\Delta\theta = \frac{360 \cdot \Delta enc}{4 \cdot 100} \quad (1)$$

lo cual, teniendo en cuenta la relación de reducción de 1:23, proporciona una resolución final de $\frac{1}{400} \cdot \frac{1}{23} = \frac{1}{9200}$ revoluciones, lo que es equivalente a 0,039°.

La velocidad de los motores a corriente continua se controlará mediante una señal PWM enviada por dos controladores de motor L298N de doble salida siguiendo el principio de un puente en H. Gracias a ellos se podrá controlar tanto la velocidad de cada rueda, como su sentido a través de un Arduino Mega 2560 Rev3. Para facilitar la conexión y desconexión de cables de esta, una PCB será diseñada por uno de los miembros del equipo.

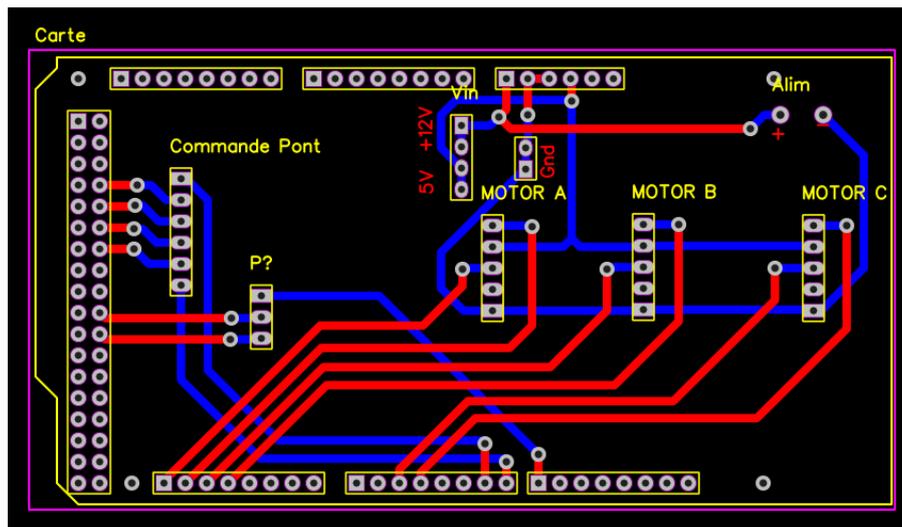


Figura 6. Diseño de la PCB para la plataforma móvil.

Fuente: Saïd Lourhaoui (equipo Eurobot 2019) usando los laboratorios CentraleSupélec.

Para evitar cortes de alimentación en la placa Arduino durante la competición, la electrónica de potencia de la base (en este caso, los controladores de motor) estará alimentada por una batería NiMH Conrad de 12V 2400mAh; las placas Arduino, entre ellas la utilizada para la plataforma, estará alimentada de forma independiente por una batería secundaria, también de NiMH, a 12V de 1600 mAh. Ambas estarán reguladas por un step-down LM2596 para garantizar que tanto los motores como las placas Arduino reciben exactamente 12V, lo cual será importante para el regulador PI.

PRINCIPIO DE FUNCIONAMIENTO Y CONTROLADOR PI

La determinación de la posición del robot se realizará por odometría, es decir, transformando la información sobre la rotación que realiza cada rueda en el desplazamiento del conjunto del robot. Para ello, una buena idea es regular en primer lugar las velocidades individuales de cada rueda.

Un motor a corriente continua puede ser modelizado de la siguiente manera:

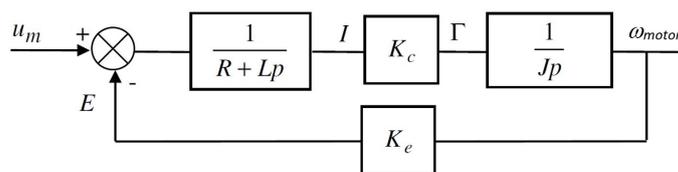


Figura 7. Modelización de un motor de corriente continua.

Fuente: Autor.

Lo cual resulta en la siguiente función de transferencia:

$$T(p) = \frac{\omega_{motor}(p)}{U_m(p)} = \frac{K_v}{(1 + \tau_e p)(1 + \tau_m p)}$$

con $K_v = 53,84 \text{ V}^{-1} \cdot \text{rad} \cdot \text{s}^{-1}$, $\tau_e = 92,5 \text{ } \mu\text{s}$ y $\tau_m = 80 \text{ ms}$.

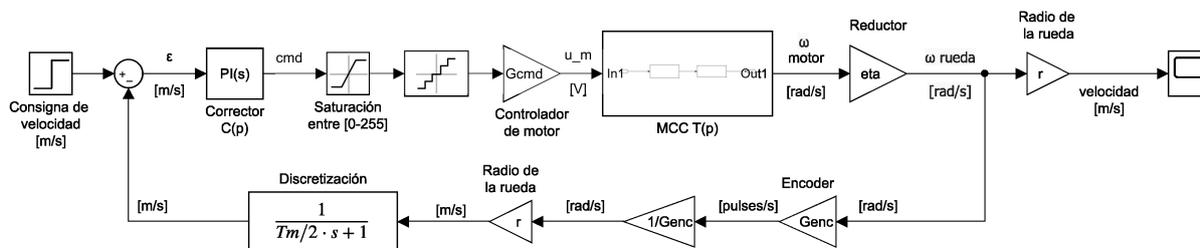


Figura 8. Diagrama de bloques de la regulación en velocidad de un solo motor.

Fuente: Autor.

Representando el conjunto de la base como se indica en el esquema 8, si seguimos un pliego de condiciones de

$$\omega_c = 70 \text{ rad/s}$$

$$\Delta\phi \geq 60^\circ$$

$$D \approx 10 \%$$

obtenemos el corrector

$$C(p) = K_P \left(1 + \frac{1}{T_i p} \right) = K_P \left(\frac{1 + T_i p}{T_i p} \right)$$

con $T_i = 25 \text{ ms}$ y $K_P = 1492$, el cual satisface las condiciones impuestas.

Una vez reguladas las velocidades individuales de cada rueda, se puede pasar a la regulación de la posición global del robot. Esta está representada por el esquema siguiente:

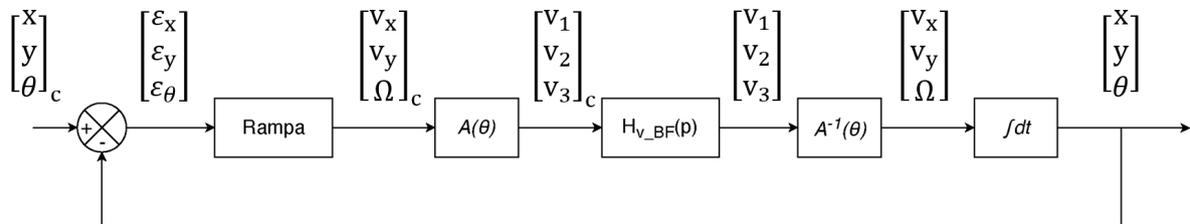


Figura 9. Diagrama de bloques de la regulación en posición del robot.

Fuente: Autor.

donde A representa la matriz de cambio de base mencionada anteriormente.

Experimentalmente, sin embargo, encontramos que el valor de $K_P = 1492$ calculado es demasiado alto y provoca grandes oscilaciones durante los desplazamientos del robot. Así pues, siguiendo un método heurístico, el valor de K_P encontrado que ofrece los mejores resultados es $K_P^{exp} = 300$, como se puede ver en las figuras 10 y 11.

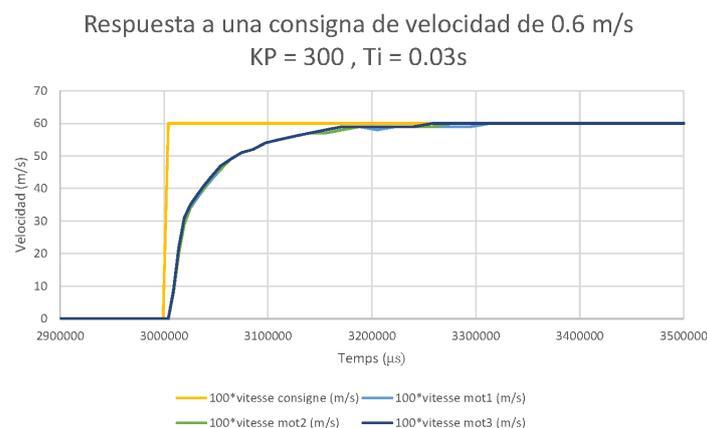


Figura 10. Respuesta temporal experimental de cada motor obtenida con un tiempo de muestreo $T_m = 5$ ms para una velocidad de consigna de 0,6 m/s.

$K_P = 300$, $T_i = 30$ ms, $T_m = 5$ ms.

Fuente: Autor.

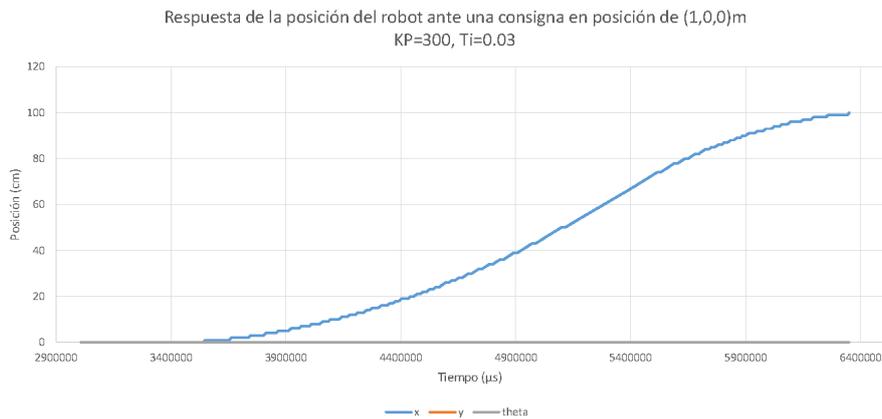


Figura 11. Respuesta en posición del robot ante una consigna de posición de (1, 0, 0) m.

$$K_P = 300, T_i = 30 \text{ ms}, T_m = 5 \text{ ms}.$$

Fuente: Autor.

ANÁLISIS DE RESULTADOS

Pese a lo calculado en el apartado anterior, en la práctica encontramos que el robot tiene una imprecisión de entre 8 % y 10 % al recorrer una distancia de un metro en línea recta.

Para solucionarlo, un factor de compensación `correction` de 0,919 es añadido al código C++, el cual disminuye este error a alrededor del 2 %. En lo que respecta a la precisión en la rotación del robot, este tiene un desvío de aproximadamente 10° tras realizar veinte vueltas sobre sí mismo, lo cual es más que suficiente para la competición.

Distancia consigna (mm)	Distancia recorrida (mm)	% Error sin corrector	% Error con corrector
200	202	10 %	1 %
500	490	9 %	2 %
800	800	8,1 %	0 %
1000	1003	9,5 %	0,3 %
1200	1200	8,8 %	0 %
1400	1344	7,9 %	4 %
1600	1552	8,4 %	3 %

Cuadro 1. Comparativa del error en la posición de la plataforma para una distancia de consigna antes y después de la adición del factor de corrección.

Fuente: Autor.

Teniendo en cuenta que las distancias recorridas por el robot en la competición no superarán los tres metros en total, los resultados finales obtenidos tras la adición del factor de corrección son más que satisfactorios y permitirán al resto del equipo Eurobot 2019 poder realizar correctamente el resto de las tareas del robot.

REFERENCIAS

- [1] **Coupe de France de Robotique**, *Sitio web de la Copa de Robótica de Francia*. Último acceso: 22/05/2019.
<https://www.coupederobotique.fr/edition-2019/le-concours/>
- [2] **Géza Szayer**, *Kinematic and Dynamic Limits of Holonomic Mobile Robots*, Budapest University of Technology and Economics, tesis doctoral. Último acceso: 15/08/2019.
<https://repozitorium.omikk.bme.hu/bitstream/handle/10890/5484/ertekezes.pdf>
- [3] **Raúl Rojas**, *Omnidirectional control*, Freie Universität Berlin, proyecto de fin de carrera. Último acceso: 15/08/2019.
<http://robocup.mi.fu-berlin.de/buch/omnidrive.pdf>
- [4] **Quentin Bennehard, Armand Bouvier, Simon Le Cléac'H, Gabriel Moneyron, Théophile Montagne, Victor Zhang, Julien Zhou**, *Rapport final de projet concours*, informe de proyecto innovación. CentraleSupélec, 2017.
- [5] **Swart, Dirk** (11 septiembre 2014). *R/C Omniwheel Robot – Electronics | Make*, blog de electrónica y robótica. Último acceso: 18/04/2019.
<https://makezine.com/projects/make-40/kiwi/>
- [6] **Christina VLAD**, *Systèmes automatiques*, asignatura del curso 2018 de la universidad de CentraleSupélec de París.
- [7] **Paul Stoffregen**, *Quadrature Encoder Library for Arduino*, repositorio GitHub. Último acceso: 02/04/2019.
<https://github.com/PaulStoffregen/Encoder>

Abstract

DEVELOPMENT, BUILDING AND PROGRAMMING OF AN AUTONOMOUS ROBOT TO PARTICIPATE IN THE FRENCH ROBOTICS CUP

Author: Horcajo de la Cruz, Daniel.

Director: Boullis, Nicolas.

Collaborating Entity: CentraleSupélec's Robotics Club (CRoCS).

SUMMARY OF THE PROJECT

INTRODUCTION

This project is conducted in collaboration with the Robotics Club at the university of CentraleSupélec (CRoCS) in Paris, where one of their teams participates in the French Robotics' Cup every year. In such competition, the teams will face one another in the same playing field – called the 'arena' – where they will have to carry out and accomplish multiple kind of tasks in order to receive the maximum amount of points, all in under ninety seconds.

The aim of this project is therefore to develop the base of this year's robot for the team at CentraleSupélec. This base must be both autonomous and controlled in position.

STATE OF THE ART

Due to the various tasks the robot will face at the playing field, a key characteristic of the robot will be its ability to move from one point of the field to another in a way that is both fast and precise. For this reason, it is important to analyse in detail the different technologies that will allow this. Amongst all such options – including the classic two wheel differential drive – the Kiwi drive configuration will be the one use in this project.



Figure 1. Omni wheel.
Source: Pinterest.

Briefly explained, a kiwi drive configuration is composed of a three-wheel system where each wheel is rotated 120° with respect to the previous one, as show in figure 2. Omni wheels have

small discs around their circumference which are perpendicular to the turning direction, which allows the wheel not only for turning around its axis as any other wheel would, but also for sliding laterally with great ease.

Hence, the combination of the kiwi drive configuration and the omni wheels' versatility, the robot will be able to move in any direction only by controlling the speeds of its wheels, without the need to turn on itself beforehand – which is why this base is called *holonomic*.

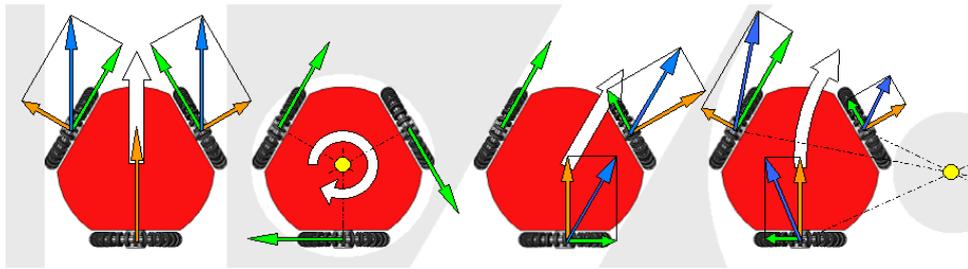


Figure 2. Representación vectorial del movimiento de un Kiwi drive con ruedas omnidireccionales.

Fuente: Wikimedia Commons.

However, the downside to this type of drive is the problems it poses regarding its commandability and orientation. Since the moving direction of each wheel will not correspond to the global movement of the robot, a change of basis matrix will be needed in order to convert the individual speeds of the three wheels into the global speed of the robot.

KINEMATICS OF THE KIWI DRIVE

As previously mentioned, controlling each of the motors in the kiwi drive is no evident task.

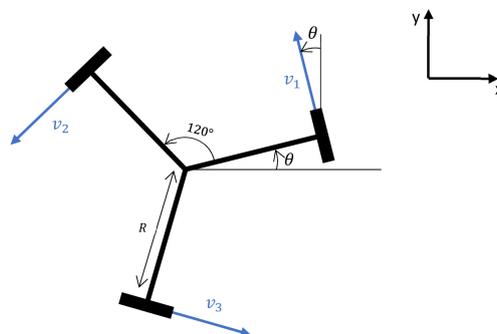


Figure 3. Vector representation of the individual wheels' velocities allowing for a change of basis..

Fuente: Author.

For this reason, with the help of figure 3, we are able to calculate the change of basis matrix that will allow for converting from the individual motor speeds $[v_1, v_2, v_3]^T$ to the global $[V_x, V_y, \Omega]^T$ of the robot:

$$\begin{bmatrix} v_1 \\ v_2 \\ v_3 \end{bmatrix} = \underbrace{\begin{bmatrix} -\sin \theta & \cos \theta & R \\ -\sin \left(\theta + \frac{2\pi}{3} \right) & \cos \left(\theta + \frac{2\pi}{3} \right) & R \\ -\sin \left(\theta + \frac{4\pi}{3} \right) & \cos \left(\theta + \frac{4\pi}{3} \right) & R \end{bmatrix}}_A \begin{bmatrix} V_x \\ V_y \\ \Omega \end{bmatrix}$$

Figure 4. Change of basis matrix A allowing for converting the motors' individual speed to the robot's global.

$$\begin{bmatrix} V_x \\ V_y \\ \Omega \end{bmatrix} = \frac{1}{3} \underbrace{\begin{bmatrix} -2 \sin \theta & -2 \sin \left(-\theta + \frac{\pi}{3} \right) & 2 \sin \left(\theta + \frac{\pi}{3} \right) \\ 2 \cos \theta & -2 \cos \left(-\theta + \frac{\pi}{3} \right) & -2 \cos \left(\theta + \frac{\pi}{3} \right) \\ \frac{1}{R} & \frac{1}{R} & \frac{1}{R} \end{bmatrix}}_{A^{-1}} \begin{bmatrix} v_1 \\ v_2 \\ v_3 \end{bmatrix}$$

Figure 5. Change of basis matrix A^{-1} allowing for converting the robot's global speed to the motors' individual.

COMPONENTS

The motors in charge of propelling the robot are the Faulhaber 2642W 012 CXR 12V DC motors. Built in them there are three 26/1 S series geardown of 1:23 which will reduce the motors' RPM to a speed more suitable for the robot.

An optical HEDS 5540 C02 encoder is also built in every motor. It will allow to measure how much each wheel has turned, which will later be used to calculate the robot's global movement.

Each encoder uses two signals – A and B – in quadrature and they each have a resolution of 100 LPR. Combining these two information, the formula expressing the rotation of a motor in degrees can be derived:

$$\Delta\theta = \frac{360 \cdot \Delta enc}{4 \cdot 100} \quad (1)$$

Taking into account the 1:23 geardown, a final resolution of $\frac{1}{400} \cdot \frac{1}{23} = \frac{1}{9200}$ revolutions, or $0,039^\circ$, is obtained.

The speed and turning direction of the three DC motors will be controlled via a PWM signal sent by two double output L298N motor drivers commanded by an Arduino Mega2560 Rev3. In order to facilitate the access to the required connections, a PCB will be designed by one of the other members of the team.

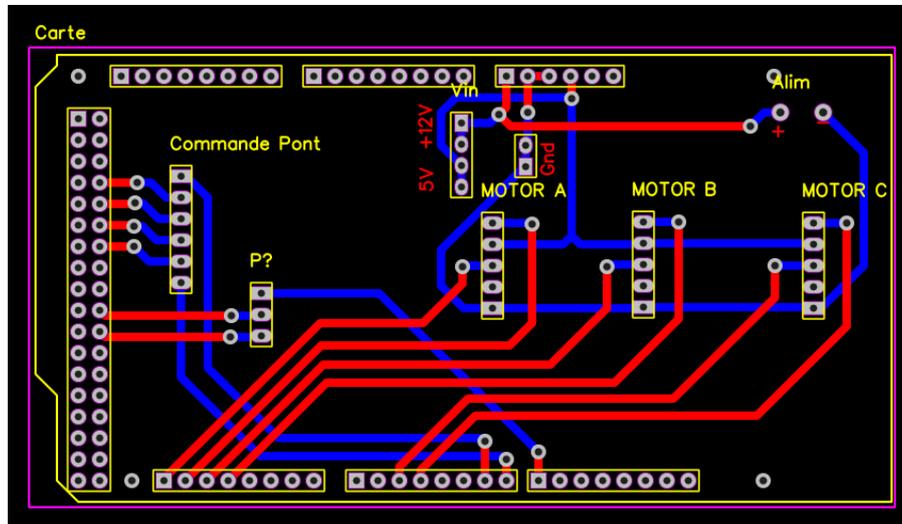


Figure 6. PCB design for the base platform.

Fuente: Saïd Lourhaoui (Eurobot 2019 team) using the laboratories at CentraleSupélec.

In order to avoid any power-related problems during the competition, the power electronics – in this case, the motor drivers – will be powered by a Conrad NiMH 12V 2400 mAh battery. The multiple Arduino boards used in the robot, on the other hand, will be powered by their own NiMH 12V 1600 mAh battery.

It is important to note that each of these batteries will use a LM2596 step-down to ensure that the input voltage for all electronics is always 12V, which will be a key component in the development of the PI control system.

OPERATING PRINCIPLE AND PI CONTROL SYSTEM

The determination of the position of the robot will be done by odometry, that is, by using the rotation of each wheel to estimate the global positioning of the robot. For this reason, it is important to control the individual motor speeds.

A DC motor may be modelled as follows:

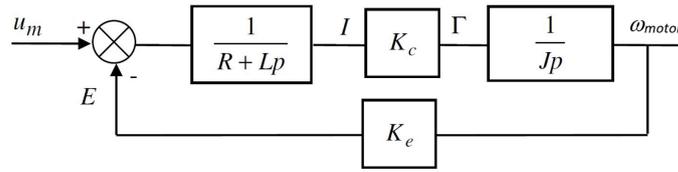


Figure 7. Block diagram model of a DC motor.
Fuente: Author.

which can then be translated into the following transfer function:

$$T(p) = \frac{\omega_{motor}(p)}{U_m(p)} = \frac{K_v}{(1 + \tau_e p)(1 + \tau_m p)}$$

where $K_v = 53,84 \text{ V}^{-1} \cdot \text{rad} \cdot \text{s}^{-1}$, $\tau_e = 92,5 \text{ } \mu\text{s}$ and $\tau_m = 80 \text{ ms}$.

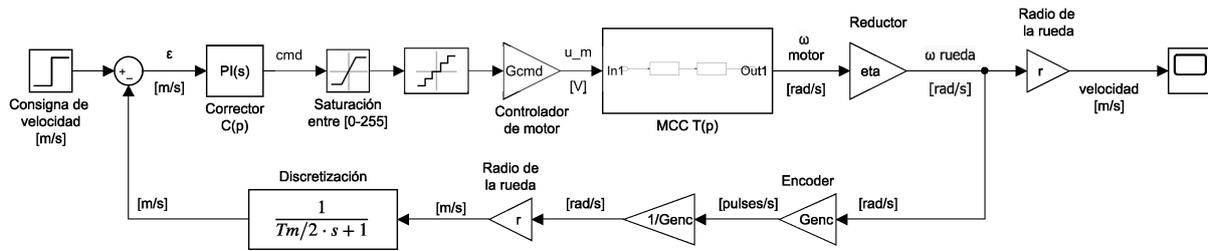


Figure 8. Black diagram of the control system for the individual speed of one motor.
Fuente: Author.

The whole of the base platform may be modelled as in figure 8. Thus, taking into account the following specifications

$$\omega_c = 70 \text{ rad/s}$$

$$\Delta\phi \geq 60^\circ$$

$$D \approx 10\%$$

the resulting controller is

$$C(p) = K_P \left(1 + \frac{1}{T_i p} \right) = K_P \left(\frac{1 + T_i p}{T_i p} \right)$$

where $T_i = 25 \text{ ms}$ and $K_P = 1492$, which satisfies the previous specifications.

Once the individual motor speed have been controlled, the next step is the control of the global position of the robot. This is represented by the following block diagram:

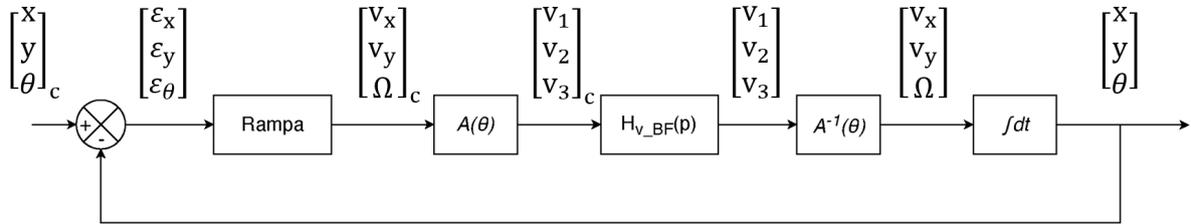


Figure 9. Block diagram of the control system for the global position of the robot.
Fuente: Author.

where A represents the change of basis matrix previously mentioned.

Experimentally, however, the calculated value of K_P seems to be too high, since the robot oscillates sharply while moving. Thus, following a heuristic approach, a value of $K_P^{exp} = 300$ is found. Tracing the response diagrams using this new value, we can conclude it offers a good performance, as shown in images 10 and 11.

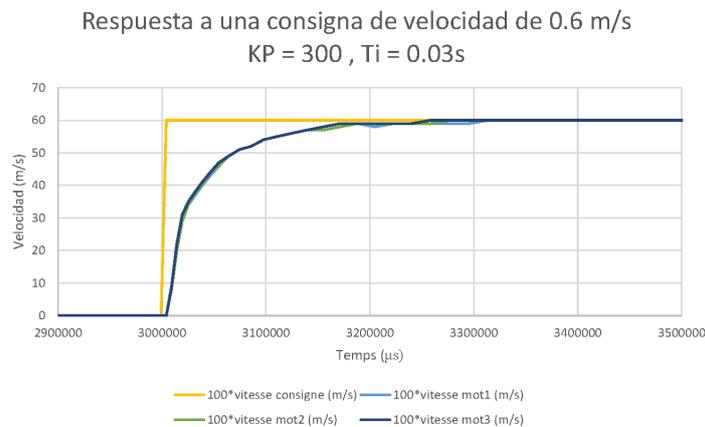


Figure 10. Experimental step response for an individual motor given an input speed of 0, 6 m/s.
 $K_P = 300, T_i = 30 \text{ ms}, T_m = 5 \text{ ms}.$
Fuente: Author.

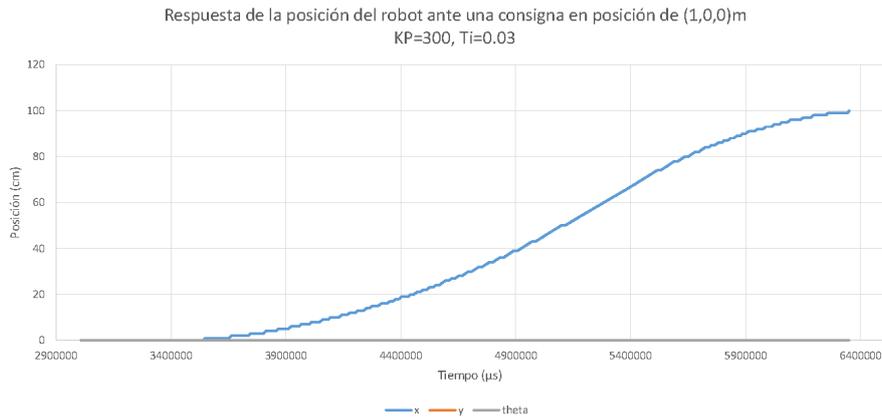


Figure 11. Response in the robot’s position for an input of (1, 0, 0) m.
 $K_P = 300, T_i = 30 \text{ ms}, T_m = 5 \text{ ms}.$
Fuente: Auhtor.

PERFORMANCE ANALYSIS

In spite of what has been calculated in previous sections, in reality we find an imprecision between 8% and 10% when making the robot move one meter in a straight line.

In order to correct this issue, a factor `correction` equal to 0,919 has been added to the C++ source code, which results in a final error of around 2%. Regarding the precision of the rotation of the robot, an error of around 10° is caused after spinning twenty times on itself, which is an acceptable value for the competition.

Input distance (mm)	Travelled distance (mm)	% Error without correction	% Error with correction
200	202	10%	1%
500	490	9%	2%
800	800	8,1%	0%
1000	1003	9,5%	0,3%
1200	1200	8,8%	0%
1400	1344	7,9%	4%
1600	1552	8,4%	3%

Table 1. Comparison of the error in the robot’s position given an input distance before and after adding the correction.
Fuente: Autor.

Taking into account that the overall travelled distance of the robot during the competition will not exceed three meters, the results obtained after adding the correction factor can be qualified as successful and will allow the rest of the team for an easy use of this platform.

REFERENCES

- [1] **Coupe de France de Robotique**, *French Robotics' Cup website*. Last access: 22/05/2019.
<https://www.coupederobotique.fr/edition-2019/le-concours/>
- [2] **Géza Szayer**, *Kinematic and Dynamic Limits of Holonomic Mobile Robots*, Budapest University of Technology and Economics, Ph.D. Thesis. Last access: 15/08/2019.
<https://repositorium.omikk.bme.hu/bitstream/handle/10890/5484/ertekezes.pdf>
- [3] **Raúl Rojas**, *Omnidirectional control*, Freie Universität Berlin, End-of-Degree Thesis. Last access: 15/08/2019.
<http://robocup.mi.fu-berlin.de/buch/omnidrive.pdf>
- [4] **Quentin Bennehard, Armand Bouvier, Simon Le Cléac'H, Gabriel Moneyron, Théophile Montagne, Victor Zhang, Julien Zhou**, *Rapport final de projet concours*, innovation project report. CentraleSupélec, 2017.
- [5] **Swart, Dirk** (11th September 2014). *R/C Omniwheel Robot – Electronics | Make*, electronics and robotics blog. Last access: 18/04/2019.
<https://makezine.com/projects/make-40/kiwi/>
- [6] **Christina VLAD**, *Systèmes automatiques*, course at CentraleSupélec University, 2018.
- [7] **Paul Stoffregen**, *Quadrature Encoder Library for Arduino*, GitHub repository. Last access: 02/04/2019.
<https://github.com/PaulStoffregen/Encoder>

DOCUMENTO I



MEMORIA



Índice

I. Memoria	13
1. Introducción	15
1.1. Descripción de la Copa de Robótica de Francia	15
1.1.1. Edición 2019: la fábrica de átomos	16
1.2. Objetivos del proyecto y colaboración con el CRoCS	18
1.3. Motivación	18
1.4. Recursos empleados	19
2. Descripción de las tecnologías: estado de la técnica	21
2.1. Modo de desplazamiento	21
2.1.1. Tracción diferencial a dos ruedas	21
2.1.2. Oruga	22
2.1.3. Kiwi drive con ruedas omnidireccionales	23
2.2. Sistema antichoque	24
2.2.1. LIDAR	24
2.2.2. Sensores infrarrojos	25
2.2.3. Sensores ultrasonidos	26
3. Diseño de la plataforma y cinemática del Kiwi drive	27
3.1. Diseño de la plataforma móvil	27
3.1.1. Fijación de motores a la plataforma	29
3.1.2. Soporte axial de los motores	29
3.2. Cinemática del Kiwi drive y matrices de cambio de base	30
4. Componentes y arquitectura electrónica	33
4.1. Ruedas	33
4.2. Conjunto motor-reductor-encoder	34
4.3. Controlador de motor	35
4.4. Arduino Mega 2560 Rev3	37
4.5. Alimentación	38
5. Principio de funcionamiento y controlador PI de la plataforma	41

5.1. Regulación de la velocidad de una sola rueda	42
5.1.1. Estudio y modelización del motor	42
5.1.2. Regulación de la velocidad y síntesis del corrector	43
5.2. Regulación de la posición del robot	48
5.2.1. Operador <i>Rampa</i>	49
5.3. Puesta en práctica y análisis de resultados	50
6. Código fuente	55
6.1. Funcionamiento general	56
7. Análisis de resultados	59
Bibliografía	62
II. Código fuente	65
1. Clases C++ utilizadas en la plataforma	67
1.1. Clase <code>Motor</code>	67
1.1.1. <code>Motor.h</code>	67
1.1.2. <code>Motor.cpp</code>	68
1.2. Clase <code>assservissement</code>	70
1.2.1. <code>assservissement.h</code>	70
1.2.2. <code>assservissement.cpp</code>	70
1.3. Clase <code>calculator</code>	71
1.3.1. <code>calculator.h</code>	71
1.3.2. <code>calculator.cpp</code>	72
1.4. Clase <code>Robot</code>	73
1.4.1. <code>Robot.h</code>	73
1.4.2. <code>Robot.cpp</code>	74
1.5. Clase <code>goto-test</code>	83
1.5.1. <code>goto-test.h</code>	83
1.5.2. <code>goto-test.cpp</code>	83
1.6. Clase <code>read-encoders-test</code>	89
1.6.1. <code>read-encoders-test.h</code>	89
1.6.2. <code>read-encoders-test.cpp</code>	90
1.7. Clase <code>test-motor</code>	90
1.7.1. <code>test-motor.h</code>	90
1.7.2. <code>test-motor.cpp</code>	91
2. Scripts MatLab para la modelización del robot	95
2.1. Script <code>assservissement_moteur</code> del regulador PI	95

2.2. Script <code>matrices_Kiwi_drive</code> de cambio de base entre la cinemática global del robot y las individuales de las ruedas	96
III. Data sheets	99
Ruedas omnidireccionales 60mm	101
Motor DC 2642W 012CXR	103
Reductor planetario serie 26/1 S	106
Encoder óptico HEDS 5540 C02	108
Controlador de motor L298N	111
Arduino Mega 2560 Rev3	125
Baterías NiMH 1.2V/celda	134
Regulador de tensión step-down LM2596	138

List of Figures

1. Ejemplos de competiciones de años anteriores	15
2. Vista global de la arena en la edición 2019	16
3. Vista detallada de la arena en la edición 2019	17
4. Antiguo logo del CRoCS	18
5. Modelización de un robot de tracción diferencial a dos ruedas	22
6. Ejemplo de un robot con una configuración de ruedas oruga	22
7. Rueda omnidireccional	23
8. Representación vectorial del movimiento de un Kiwi drive con ruedas omnidireccionales	23
9. Módulo TF Mini LIDAR unidireccional	24
10. Módulo RPLidar A1M8 de 360°	24
11. Sensor IR para detección de obstáculos	25
12. Sensor ultrasonidos HC-SR04	26
13. Test de rango angular del sensor ultrasonidos HC-SR04	26
14. Vista tridimensional de la plataforma móvil	28
15. Fotografía de la plataforma móvil	28
16. Vista inferior de la plataforma móvil	29
17. Vista tridimensional del conjunto motor-fijaciones	30
18. Vista lateral del conjunto base-motores-soporte	30
19. Representación vectorial de las velocidades independientes de cada rueda para realizar un cambio de base	31
20. Matriz A de cambio de base para obtener las velocidades lineares individuales de las ruedas dada la velocidad lineal y angular del robot.	31
21. Matriz de cambio de base para obtener la velocidad lineal y angular del robot dadas las velocidades lineares individuales de las ruedas.	32
22. Rueda omnidireccional	33
23. Características de funcionamiento del motor en régimen permanente	34
24. Estados lógicos de los lectores ópticos del encoder HEDS 5540 C02	35
25. Controlador de motor L298N	36
26. Esquema representativo de un puente en H	36
27. PCB para la plataforma móvil	37

28. Diseño de la PCB para la plataforma móvil	38
29. Batería principal Conrad 12V 2400 mAh utilizada para la electrónica de potencia	39
30. Batería secundaria 12V 1600 mAh para las placas Arduino	39
31. Regulador step-down LM2596	39
32. Modelización de un motor de corriente continua	42
33. Diagrama de bloques de la regulación en velocidad de un solo motor	43
34. Diagrama de Bode del lazo abierto $H_{v_BO}(\omega)$	44
35. Ábacos para la determinación del sobrepico D y tiempo de respuesta para una respuesta temporal en un sistema de segundo orden	45
36. Diagrama de Bode del lazo abierto corregido $H_{v_BOC}(j\omega)$	47
37. Respuesta temporal del lazo cerrado $H_{v_bf}(j\omega)$	48
38. Diagrama de bloques de la regulación en posición del robot	49
39. Perfil de la velocidad global del robot en función del tiempo calculado por el operador <i>Rampa</i>	49
40. Respuesta temporal experimental de cada motor obtenida con un tiempo de muestreo $T_m = 5$ ms para una velocidad de consigna de 0,6 m/s. $K_P = 300$, $T_i = 30$ ms, $T_m = 5$ ms	51
41. Respuesta en posición del robot ante una consigna de posición de (1, 0, 0) m. $K_P =$ 300 , $T_i = 30$ ms, $T_m = 5$ ms	51
42. Respuesta de la velocidad total del robot ante una consigna de posición de (1, 0, 0) m. $K_P = 300$, $T_i = 30$ ms, $T_m = 5$ ms	52
43. Respuesta de las velocidades individuales de los motores ante una consigna de posición de (1, 0, 0) m. $K_P = 300$, $T_i = 30$ ms, $T_m = 5$ ms	53
44. Versión final del robot del equipo de CentraleSupélec Eurobot 2019	60
45. Instantánea del robot en uno de los partidos de la Copa 2019	61

List of Tables

1. Error en la posición de la plataforma para una distancia de consigna. <i>Fuente: Autor.</i>	59
2. Error en la posición de la plataforma para una distancia de consigna tras la adición del factor de corrección. <i>Fuente: Autor.</i>	60

Índice de extractos de código

Acrónimos

<i>CAD</i>	Diseño Asistido por Ordenador (en inglés, <i>Computer-Aided Design</i>)
<i>CRoCS</i>	Club de Robotique de CentraleSupélec
<i>CS</i>	CentraleSupélec
<i>DIY</i>	Do It Yourself
<i>I2C</i>	Circuito inter-integrado (<i>Inter-Integrated Circuit</i> en inglés)
<i>IDE</i>	Entorno de desarrollo interactivo (en inglés <i>Integrated Development Environment</i>)
<i>LPR</i>	Lines Per Revolution
<i>MCC</i>	Motor de corriente continua
<i>DM</i>	Tipo de madera reconstituida
<i>IR</i>	Infrarrojos
<i>LIDAR</i>	Detección por luz y distancia (<i>Light Detection and Ranging</i> en inglés)
<i>PWM</i>	Modulación por ancho de pulsos (<i>Pulse-Width Modulation</i> en inglés)
<i>PCB</i>	Printed Circuit Board
<i>PI</i>	Controlador proporcional-integral

PART I



MEMORIA



Chapter 1

Introducción

ESTE proyecto tiene como finalidad el desarrollo de una plataforma móvil que será utilizada por el robot del Club de Robótica de la universidad CentraleSupélec de París (CRoCS) en la Copa de Francia de Robótica de 2019 [1]. Sus desarrollo está guiado pues por el reglamento de dicha competición, donde se especifican tanto las tareas que debe realizar el robot participante, como las diferentes restricciones impuestas a este.

1.1. Descripción de la Copa de Robótica de Francia

La Copa de Robótica de Francia (antigua *Coupe E=M6 de robótica* creada en 1994) es una competición anual coorganizada desde 2016 por la asociación Planète Sciences y la sociedad de economía mixta Oryon [1]. A día de hoy, esta competición reúne cada año a más de 200 equipos, desde clubs de amateurs de la electrónica, hasta escuelas de ingeniería e institutos universitarios de tecnología.



Figure 1. Ejemplos de competiciones de años anteriores.
Fuente: página web de la Copa de Robótica de Francia [1].

El objetivo de esta competición es diseñar, construir y programar un robot completamente independiente capaz de llevar a cabo una serie de tareas en un área de juego, también llamada “arena”. Con cada edición de la competición, los organizadores del evento crean una arena diferente y proponen nuevos retos que los robots participantes deberán superar en menos de noventa segundos. A cada tarea le es asignado un número de puntos, por lo que cada robot deberá realizar aquellas tareas que le proporcionen un mayor número de puntos en el menor tiempo posible. En cada partido, dos equipos se enfrentan en la misma arena al mismo tiempo, por lo que deberán luchar por los recursos existentes.

1.1.1. Edición 2019: la fábrica de átomos

El tema elegido por la organización para la edición 2019 es “La fábrica de átomos”, cuya arena puede observarse en la figura 2.



Figure 2. Vista global de la arena en la edición 2019.
Fuente: página web de la Copa de Robótica de Francia [1].

Las posibles tareas a realizar por los robots participantes son las siguientes:

1. **Clasificación de átomos:** repartidos por la arena se encuentran discos de hockey de diferentes colores y tamaños que representan diferentes elementos de la tabla periódica (*redium, greenium, blueium y goldenium*) y que poseen valores diferentes para el recuento

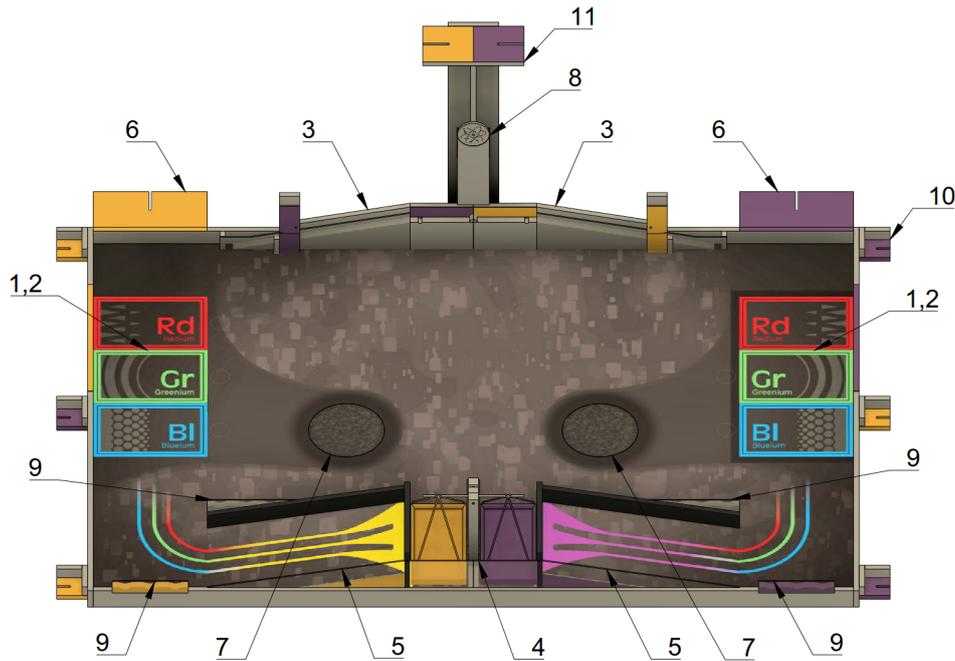


Figure 3. Vista detallada de la arena en la edición 2019.
Fuente: página web de la Copa de Robótica de Francia [1].

final de puntos¹. Cada equipo deberá recoger y clasificar los átomos en su propia tabla periódica (elementos 1, 2 de la figura 3) para obtener la mayor puntuación posible.

2. **Peso de átomos:** los robots deberán recoger los átomos y depositarlos en la balanza correspondiente (elemento 4). Cuanto más pesados sean los átomos depositados, más puntos les serán otorgados al equipo.
3. **Creación de un nuevo elemento:** en el acelerador de partículas (elemento 3), el robot deberá empujar el *blueium* que se encuentra en la parte de arriba de la rampa, que rodará por el acelerador y liberará (“creará”) el *goldenium*. Este átomo es único en su especie y el procedimiento descrito anteriormente es la única forma de obtenerlo.
4. **Predicción de la puntuación:** cada robot deberá ser capaz de predecir la puntuación final que obtendrá tras haber pasado los noventa segundos de partido. El equipo obtendrá mayor bonus de puntuación cuanto menor sea el error en la predicción.

¹El número de puntos que otorga cada elemento está dado tanto por el tipo de elemento utilizado, como por el tipo de acción realizada con ese elemento; en cualquier caso, el valor de cada átomo para una misma tarea, ordenado de menor a mayor, es: *redium*, *greenium*, *blueium*, *goldenium*.

1.2. Objetivos del proyecto y colaboración con el CRoCS

Como se ha explicado con anterioridad, la Copa de Robótica es un evento cuyo pilar base es el trabajo en equipo para lograr un objetivo común: construir el mejor robot posible. Por consiguiente, este proyecto será parte de una colaboración con el Club de Robótica (perteneciente al Club de Tecnología, el *Club Tech*) de CentraleSupélec, el cual es ya participante habitual en esta competición.



Figure 4. Antiguo logo del CRoCS.

Fuente: CRoCS.

El objetivo último de este proyecto en particular es diseñar y construir una plataforma autónoma asistida en posición sobre la cual será construido el resto del robot del CRoCS, y que será la encargada de hacer que este se desplace por todo el terreno de juego de la forma más rápida y precisa posible.

Otras tareas en el desarrollo del robot, como la gestión económica de recursos, el diseño de PCBs o el sistema de recogida y almacenamiento de los discos de hockey será llevado a cabo por el resto de los miembros del equipo del CRoCS.

1.3. Motivación

Hoy en día, es muy común que los clubs de robótica o electrónica de institutos y universidades desarrollen proyectos que les ayuden no solo a entender y profundizar conocimientos tecnológicos, sino dar solución a problemas reales que los alumnos encuentran en su día a día. La Copa de Robótica de Francia sigue también estas mismas líneas: propone versiones simplificadas de situaciones reales que los participantes deberán superar en equipo.

La parte correspondiente a este proyecto (descrita anteriormente en el apartado 1.2), al tratar únicamente la base del robot, hace que sus aplicaciones en el mundo real puedan ser aún más generalizadas, ya que las plataformas móviles autónomas son ampliamente utilizadas en diferentes industrias, sobre todo en fábricas, para el transporte eficiente de cargamentos dentro de las instalaciones.

1.4. Recursos empleados

Para lograr los objetivos mencionados anteriormente, a lo largo del proyecto se hará uso de los siguientes recursos:

1. Programas de CAD, como SolidEdge o SpaceClaim, para la modelización de la plataforma.
2. Impresoras 3D Ultimaker y Cura, para el prototipado de piezas del robot.
3. Cortadora láser y otra maquinaria de La Fabrique (laboratorio de prototipado de la universidad CentraleSupélec), para la fabricación de las diferentes partes de la base.
4. Placas Arduino, para el control y asistencia en posición de la plataforma.
5. PlatformIO en Atom, para la programación en C++.
6. Scripts MatLab, para la modelización del sistema de control de la posición.
7. Microsoft OneDrive y GitLab [2], para el almacenamiento y uso compartido de recursos como datasheets, archivos 3D y código, entre todos los miembros del equipo.
8. Foros online y canales YouTube de electrónica y robótica, donde adquirir conocimientos sobre el funcionamiento de los diferentes componentes.

Cabe destacar que todos los archivos y modelos 3D, así como los scripts MatLab y código, pueden ser accedidos a través del GitLab del equipo de este año [2].

Chapter 2

Descripción de las tecnologías: estado de la técnica

LA Copa de Robótica de Francia tiene lugar cada año, por lo que un buen punto de partida para nuevos participantes es analizar las tecnologías utilizadas por los competidores de otros años para averiguar cuáles son aquellas que se adaptan mejor a los requisitos de la última edición de la Copa.

2.1. Modo de desplazamiento

Tras un periodo de investigación al respecto de las diferentes tecnologías disponibles a nuestro alcance, se estudiaron diferentes posibilidades para elegir el modo de desplazamiento óptimo para la plataforma [3]. Esta debe moverse de forma rápida y versátil para poder realizar el máximo número de acciones en los noventa segundos de partido.

2.1.1. Tracción diferencial a dos ruedas

Este tipo de robot está constituido de dos ruedas motrices montadas a lo largo de un mismo eje y una tercera rueda que se mueve libremente. Esta última puede tratarse de una rueda como que encontramos en un carrito de la compra o incluso de una simple canica sujeta a la base a través de una conexión esférica.

El movimiento del robot, representado en la figura 5, es controlado por la diferencia en la velocidad de cada una de las dos ruedas: si ambas giran a la misma velocidad, entonces el robot se moverá en línea recta; sin embargo, si una gira más deprisa que la otra, el robot tenderá a girar en la dirección de la rueda más lenta.

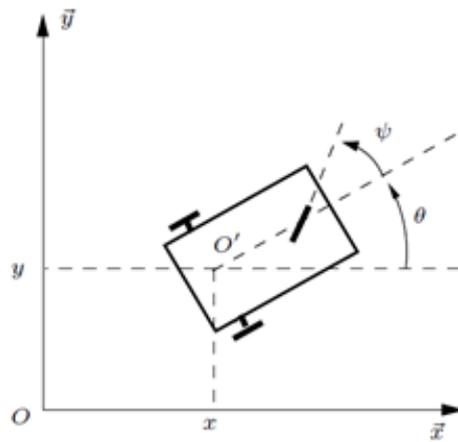


Figure 5. Modelización de un robot de tracción diferencial a dos ruedas.
Fuente: Rapport final de projet concours. CentraleSupélec, 2017. [4]

2.1.2. Oruga

Este modo de tracción está constituido por una serie de eslabones modulares unidos unos a otros por un cinturón flexible (ver figura 6) que permiten un desplazamiento estable aún en terrenos irregulares ya que permite distribuir el peso en una mayor superficie. Por esta razón, este método de desplazamiento es comúnmente utilizado en vehículos pesados, como tractores o tanques.

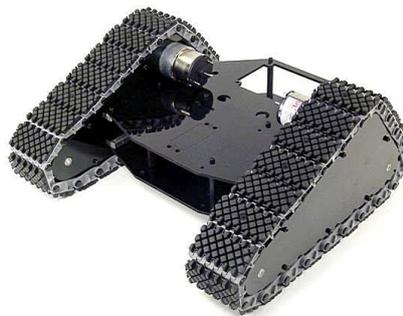


Figure 6. Ejemplo de un robot con una configuración de ruedas oruga.
Fuente: Superrobótica

Esta configuración permite alcanzar un movimiento robusto gracias a un gran par en las ruedas, lo que permitiría al robot resistir a cualquier perturbación en el terreno de juego, como puede ser un choque lateral por parte del robot adversario durante un desplazamiento. Sin embargo, este método es generalmente más lento que el anterior, además de ocupar más espacio en la parte inferior de la plataforma, la cual podría ser utilizada para guardar los discos de hockey que se encuentran repartidos por el suelo (ver figura 2 del apartado 1.1.1).

2.1.3. Kiwi drive con ruedas omnidireccionales

Un modo de desplazamiento que es utilizado de forma recurrente todos los años en la competición [5] son las ruedas omnidireccionales, como la representada en la figura 7.



Figure 7. Rueda omnidireccional.
Fuente: Pinterest.

El principio de funcionamiento de este tipo de ruedas es simple: como se puede ver en la figura, cada rueda está formada por una serie de rodillos cuyo eje se encuentra orientado de forma perpendicular al eje principal de la rueda en sí. Esta configuración consigue que se anulen los esfuerzos oblicuos en la dirección transversal a la rueda.

La gran ventaja de estas ruedas es que añaden otro grado de libertad al movimiento del robot, permitiendo su desplazamiento de forma paralela al eje de la rueda, además del movimiento clásico perpendicular al eje. Este tipo de ruedas es ampliamente utilizado, por ejemplo, en cadenas de montaje.

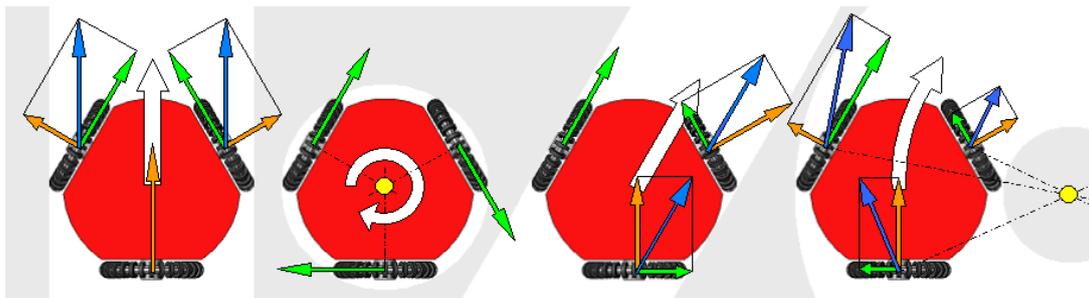


Figure 8. Representación vectorial del movimiento de un Kiwi drive con ruedas omnidireccionales.
Fuente: Wikimedia Commons.

Si tres ruedas omnidireccionales son combinadas en una disposición de 120° las unas respecto de las otras, se obtiene una configuración conocida como plataforma Kiwi drive [6]: una plataforma capaz de desplazarse en cualquier dirección sin tener que realizar una rotación previa, lo cual es conocido como plataforma holonómica [11] y se ve representado en la figura 8.

Esta configuración plantea, sin embargo, una problemática de orientación y comandabilidad de los motores, ya que al estar rotados 120° los unos de los otros, es necesario realizar una etapa de cálculo previa para poder pasar de las velocidades independientes de los motores a la velocidad absoluta del robot deseada (ver sección 3.2 sobre las matrices de cambio de base). Una vez superado esto, sin embargo, esta configuración ofrece una versatilidad de movimiento excepcional, por lo que será este el modo de desplazamiento elegido para la base del robot final.

2.2. Sistema antichoque

Un aspecto muy importante según el reglamento para que el robot pueda ser homologado y, por tanto, participar en la competición, es que el robot no debe interferir físicamente con el robot adversario [1]; es decir, no debe chocar ni tocar a este en ningún momento, o podrá ser penalizado, e incluso descalificado.

Para evitar esto, es necesario realizar un sistema antichoque para el robot². Existen diferentes tecnologías que pueden ser utilizadas para detectar al robot adversario [7].

2.2.1. LIDAR

Un LIDAR es un dispositivo que permite medir la distancia desde un emisor láser a un objeto, utilizando para ello haces de luz pulsados. El principio de funcionamiento del LIDAR es el siguiente: el dispositivo emite un haz de luz y mide en la señal que recibe (reflejada por el objeto) aspectos como la intensidad y el ángulo con el que vuelve. Estas mediciones son entonces utilizadas para calcular la distancia hasta el objeto detectado.



Figure 9. Módulo TF Mini LIDAR unidireccional.

Fuente: RobotShop.



Figure 10. Módulo RPLidar A1M8 de 360° .

Fuente: RobotShop.

Los sensores LIDAR poseen una alta precisión y frecuencia, pudiendo escanear hasta 100 veces por segundo, por lo que es difícil no detectar un objeto en movimiento. Además, estos sensores

²Aunque el sistema antichoque no será abarcado en detalle en este proyecto, es una parte esencial en el robot que está estrechamente relacionado con la plataforma móvil, por lo que es de interés que comentarlo.

tienen un rango de máximo de unos 12 metros, lo cual supera con creces las distancias máximas en las que se desplazará el robot, ya que el terreno de juego mide tres metros de largo por dos de ancho.

Si este dispositivo es montado sobre un mecanismo giratorio es posible obtener una “nube de puntos” tridimensional que da lugar a una imagen 3D de alta resolución a una distancia máxima de unos 12 metros, lo cual permitiría no solo evitar al robot adverso, sino también tener una versión digital del terreno de juego y la posibilidad de orientarse en él. Esto es posible gracias a su alta frecuencia de escaneo, ya que no es posible realizar esto mismo con otras tecnologías, como es el caso de los sensores ultrasonidos.

Además de su uso en topografía o previsiones meteorológicas, esta tecnología es altamente utilizada tanto en vehículos autónomos como en robots enviados al espacio.

2.2.2. Sensores infrarrojos

Aunque el LIDAR es técnicamente un sensor IR, en este apartado se tratarán los módulos compuestos por un diodo emisor y un receptor como los de la figura 11.

Estos sensores envían una señal cuando detectan que un objeto se encuentra a una distancia menor a la límite, la cual puede ser regulada mediante un potenciómetro.

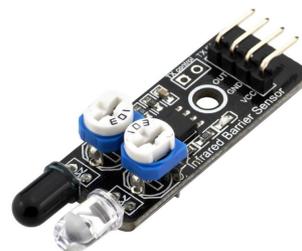


Figure 11. Sensor IR para detección de obstáculos.

Fuente: RobotShop.

Estos módulos son extremadamente asequibles y, aunque como los LIDAR, funcionan emitiendo pulsos de luz, el rango de estos no suele sobrepasar los 10 cm. Además, estos sensores no proporcionan ninguna información sobre las propiedades de la señal reflejada, por lo que son utilizados para evitar colisiones cercanas.

2.2.3. Sensores ultrasonidos

Al igual que los otros dos sensores, este módulo también envía una señal y espera a recibir la señal reflejada; en este caso, sin embargo, en vez de pulsos de luz infrarroja utiliza ondas de sonido de alta frecuencia. Tienen un rango de hasta unos 4 metros, lo que es una distancia suficiente para la Copa.



Figure 12. Sensor ultrasonidos HC-SR04.

Fuente: RobotShop.

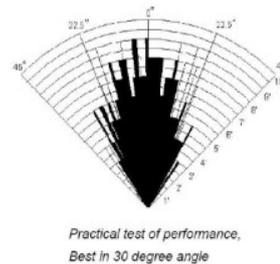


Figure 13. Test de rango angular del sensor ultrasonidos HC-SR04.

Fuente: DroneBot Workshop.

Estos sensores ultrasonidos, sin embargo, no son tan direccionales como lo son los haces de luz enviados por los sensores anteriores. Como se puede apreciar en la figura 13, estos módulos detectan objetos con un máximo de unos 45° de desvío, aunque su funcionamiento es óptimo es cuando el objeto se encuentra frente al sensor.

Por este hecho, las ondas ultrasonido no pueden ser utilizadas para crear una “nube de puntos” como en los casos anteriores. Debido a la baja velocidad de propagación del sonido en comparación con la de la luz, la cantidad de información que es posible recolectar en un periodo corto de tiempo es limitada. Asimismo, no hay que olvidar que no todos los objetos reflejan los ultrasonidos correctamente, ya que en ocasiones estos son absorbidos en mayor o menor parte, y la señal reflejada no es detectada correctamente.

A pesar de sus limitaciones, estos módulos tienen una gran ventaja al ser sencillos de usar a la vez que prácticos para la mayoría de las situaciones cotidianas, lo que hace que sean muy utilizados en el ámbito de la robótica.

Chapter 3

Diseño de la plataforma y cinemática del Kiwi drive

SIGUIENDO los requisitos de versatilidad y rapidez, de todas las configuraciones expuestas anteriormente en la sección 2.1, el Kiwi drive es la que ofrece más ventajas, ya que permitiría al robot desplazarse por el terreno sin necesidad de girar y, además, realizar en ocasiones movimientos simultáneos de rotación y traslación para invertir un menor tiempo en ciertos desplazamientos.

En este capítulo se explicará por tanto el proceso de diseño y concepción de la plataforma, así como la cinemática de la configuración Kiwi drive para su posterior implantación en el código Arduino.

Por último, esta configuración tan poco convencional supone un desafío cinemático que permitirá explorar formas de desplazamiento diferentes a las clásicas de dos o cuatro ruedas, proporcionando la posibilidad de adquirir y poner en práctica conocimientos de trigonometría, así como de las famosas bases de Euler. Esto será expuesto en la sección 3.2 de este capítulo.

3.1. Diseño de la plataforma móvil

La base es uno de los elementos principales de los cuales depende el resultado final del robot, ya que es el modo de desplazamiento del que este dispone. Dicha base deberá poder desplazarse de manera versátil y rápida por el terreno de juego, y su diseño estará condicionado por las normas del reglamento de la Copa, prestando especial atención a no superar el límite de perímetro máximo de 1200 mm.



Figure 14. Vista tridimensional de la plataforma móvil.

Fuente: Autor.

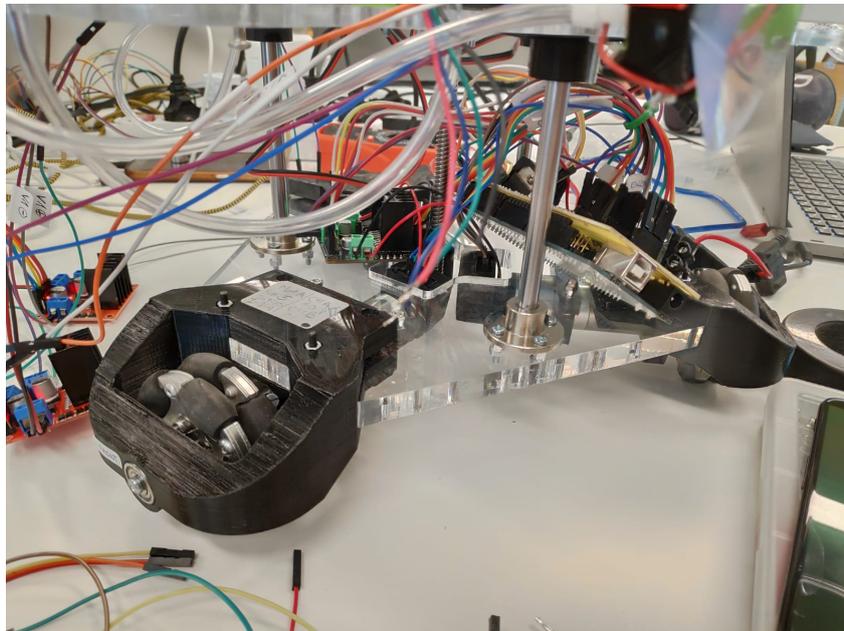


Figure 15. Fotografía de la plataforma móvil.

Fuente: Autor.

Así pues, la versión final de la plataforma posee el aspecto mostrado en las figuras 14 y 15. Como se puede observar, la base tiene forma triangular como consecuencia de la configuración Kiwi drive. La razón por la cual una base triangular es más conveniente que una redonda es que, en este caso, al poseer lados rectos el robot podrá acercarse con mayor seguridad (e incluso llegar a tocar) a las paredes del terreno para recoger los discos de hockey sin temor a que haya pequeñas rotaciones por el camino que le desvíen del disco correcto. El pequeño agujero que se encuentra en el centro de la plataforma hexagonal permitirá pasar los cables de los *encoders* y de los motores hacia la parte superior de esta, donde encontramos las placas controladoras.

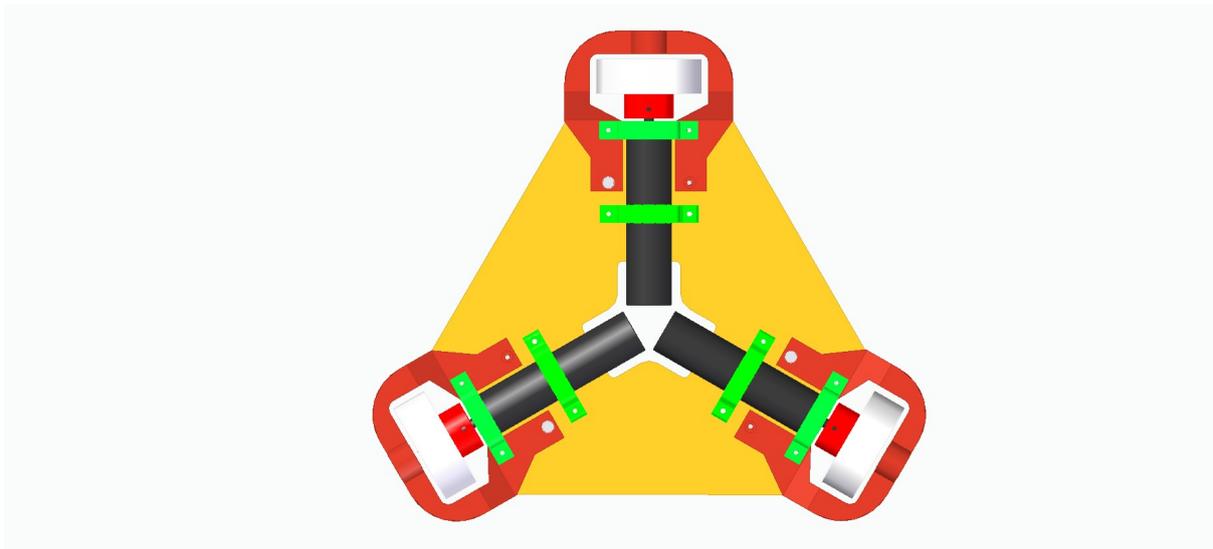


Figure 16. Vista inferior de la plataforma móvil.

Fuente: Autor.

Aunque inicialmente el cuerpo de la plataforma estaba fabricado en DM, debido a restricciones regulatorias por emisión de gases contaminantes de este material en la cortadora láser de La Fabrique, el DM ha sido sustituido por metacrilato transparente, el cual tiene la ventaja de permitir ver los motores y conexiones que se encuentran por debajo de este sin necesidad de tener que voltear el robot.

3.1.1. Fijación de motores a la plataforma

Cada uno de los tres motores del robot debe ser fijado a la plataforma de forma rígida para permitir un movimiento fiable y con las menores vibraciones posibles. Las fijaciones de los motores, mostradas en la figura 17, están impresas en 3D en PLA y permiten atornillar los motores a la base.

La sujeción exterior posee un agujero en su parte frontal que, gracias a un tornillo que llega hasta el motor, evita que este gire sobre sí mismo en vez de transmitirlo a la rueda. Por último, la razón por la cual el agujero es un arco de circunferencia, y no un simple círculo, es que la rosca destinada a este fin no se encuentra en la misma posición en todos los motores, por lo que de esta manera se resuelve fácilmente este problema.

3.1.2. Soporte axial de los motores

Con el fin de evitar una deformación del eje de los motores debido al peso del resto del robot, es conveniente distribuir esta carga de la forma más uniforme posible.

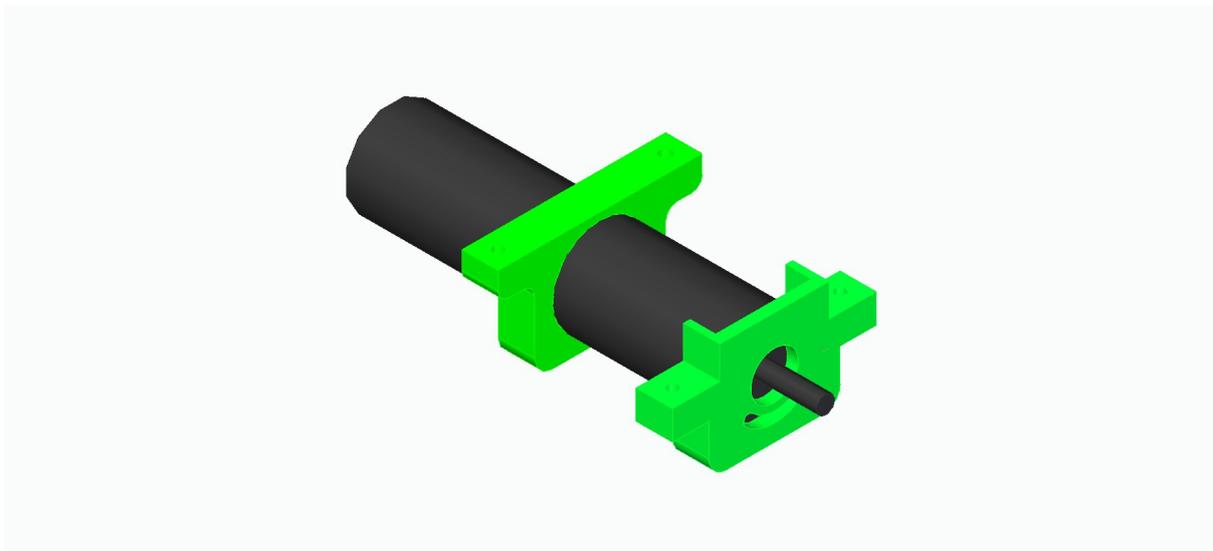


Figure 17. Vista tridimensional del conjunto motor-fijaciones.

Fuente: Autor.

Por esta razón, un soporte simple como el de la figura 18 permite pasar parte del estrés que soportaría el eje del motor al contacto de la rueda con el suelo, evitando así dañar los motores.

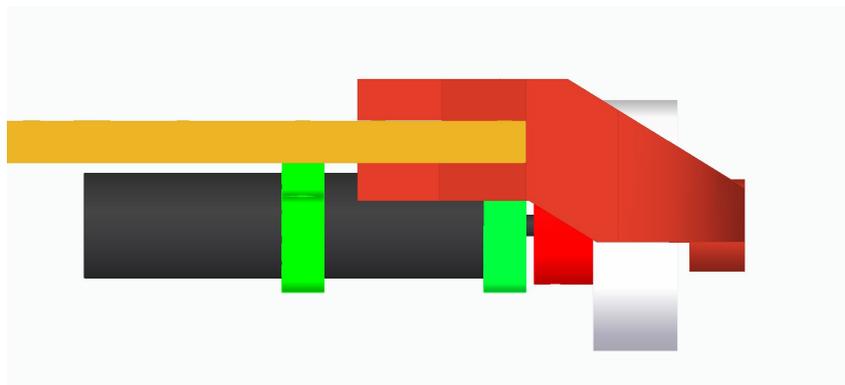


Figure 18. Vista lateral del conjunto base-motores-soporte.

Fuente: Autor.

Como las fijaciones del apartado anterior, estos soportes también están impresos en 3D gracias a las impresoras 3D de La Fabrique y del Club Tech.

3.2. Cinemática del Kiwi drive y matrices de cambio de base

La elección de un Kiwi drive de tres ruedas giradas 120° las unas respecto de las otras como modo de desplazamiento de la plataforma es muy interesante debido a los retos de cambio de bases que implica, ya que el sentido de giro de cada rueda de forma individual no se corresponde

con el movimiento global del robot; por tanto, es necesario una etapa de cálculo previa aplicando el cambio de base de Euler con rotación.

Consideremos la siguiente ilustración:

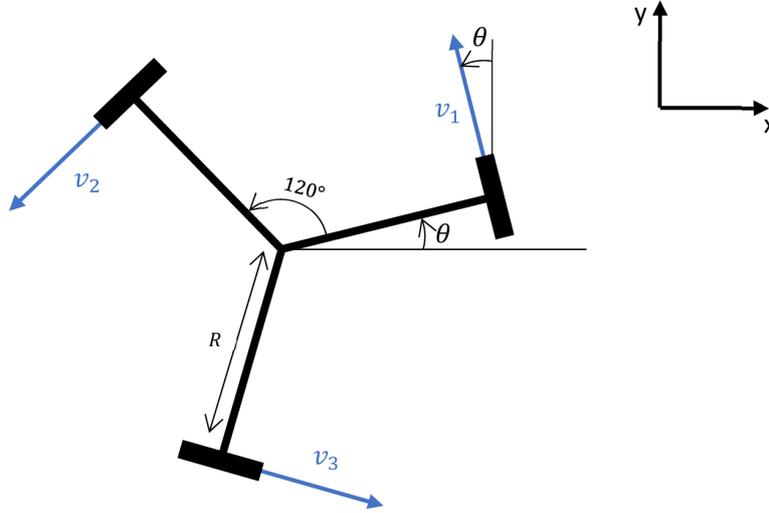


Figure 19. Representación vectorial de las velocidades independientes de cada rueda para realizar un cambio de base.

Fuente: Autor.

Si consideramos positiva la velocidad lineal las ruedas en el sentido trigonométrico y que el robot está ligeramente rotado de un ángulo θ respecto del eje de x , proyectando la velocidad lineal v_i de cada rueda sobre el eje de coordenadas (x, y) obtenemos:

$$\begin{bmatrix} v_1 \\ v_2 \\ v_3 \end{bmatrix} = \underbrace{\begin{bmatrix} -\sin \theta & \cos \theta & R \\ -\sin \left(\theta + \frac{2\pi}{3} \right) & \cos \left(\theta + \frac{2\pi}{3} \right) & R \\ -\sin \left(\theta + \frac{4\pi}{3} \right) & \cos \left(\theta + \frac{4\pi}{3} \right) & R \end{bmatrix}}_A \begin{bmatrix} V_x \\ V_y \\ \Omega \end{bmatrix} \quad (1)$$

Figure 20. Matriz A de cambio de base para obtener las velocidades lineales individuales de las ruedas dada la velocidad lineal y angular del robot.

donde R representa el radio de la plataforma del centro a cada rueda, Ω la velocidad angular total de la plataforma y (V_x, V_y) las velocidades lineales del robot según los ejes x e y , respectivamente.

Se consiguen así la matriz A de cambio de base para calcular las velocidades lineales que debe tener cada rueda según el la cinemática deseada en el robot.

En caso de necesitar realizar el cálculo inverso, es decir, obtener la cinemática del robot dadas las velocidades lineales individuales de cada rueda, simplemente hay que invertir la matriz de cambio de base anterior, con lo que se obtiene:

$$\begin{bmatrix} V_x \\ V_y \\ \Omega \end{bmatrix} = \frac{1}{3} \underbrace{\begin{bmatrix} -2 \sin \theta & -2 \sin \left(-\theta + \frac{\pi}{3}\right) & 2 \sin \left(\theta + \frac{\pi}{3}\right) \\ 2 \cos \theta & -2 \cos \left(-\theta + \frac{\pi}{3}\right) & -2 \cos \left(\theta + \frac{\pi}{3}\right) \\ \frac{1}{R} & \frac{1}{R} & \frac{1}{R} \end{bmatrix}}_{A^{-1}} \begin{bmatrix} v_1 \\ v_2 \\ v_3 \end{bmatrix} \quad (2)$$

Figure 21. Matriz de cambio de base para obtener la velocidad lineal y angular del robot dadas las velocidades lineales individuales de las ruedas.

Una vez halladas estas matrices, la dificultad principal de la configuración Kiwi drive ha sido superada, con que pasar de las velocidades individuales de las ruedas (v_1, v_2, v_3) a la global del robot (V_x, V_y, Ω) , y viceversa, es ahora cuestión de multiplicar matrices. Esta será una de las tareas del código de la plataforma.

Chapter 4

Componentes y arquitectura electrónica

UNA vez elegido el Kiwi drive como configuración final de la plataforma, es hora de realizar el montaje electrónico. Un esquema global de este se encuentra en el apartado 1 de la parte III de este documento.

Como recordatorio, las datasheets de todos los componentes mencionados en este proyecto se encuentran al final de esta memoria.

4.1. Ruedas

El Kiwi drive, como se ha explicado anteriormente, necesita unas ruedas especiales denominadas *omnidireccionales* que permiten el movimiento en cualquier dirección del plano (consultar la sección 2.1.3 para más detalle).



Figure 22. Rueda omnidireccional.
Fuente: Pinterest.

Las ruedas utilizadas en esta plataforma, de referencia RB-Nex-75 en Robotshop³, son dobles, con rodillos de goma antideslizantes, y poseen un diámetro D de 60 mm. Asimismo, cada rueda puede soportar una carga máxima de 3 kg, lo que supone que el robot pueda pesar un máximo de 9 kg.

4.2. Conjunto motor-reductor-encoder

Los motores elegidos para mover la plataforma son los 2642W 012 CXR, cedidos por la compañía Faulhaber al equipo del año anterior a modo de patrocinio.

Estos motores planetarios a corriente continua poseen las características técnicas presentadas a continuación:

Rated values for continuous operation		
29	Rated torque	M_N 25
30	Rated current (thermal limit)	I_N 1,6
31	Rated speed	n_N 4 770

Figure 23. Características de funcionamiento del motor en régimen permanente.

Fuente: Ficha técnica del motor Faulhaber 2642W 012 CXR.

El reductor que viene incluido con el motor pertenece a la serie 26/1 S y tiene una relación de reducción de 1:23. Esto permitirá reducir la velocidad de 4770 RPM en régimen permanente a la salida del motor a una más adecuada para el robot, así como aumentar el par en la misma proporción.

Para poder controlar la posición del robot es necesario conocer cuánto ha girado cada rueda en cada momento. Esta tarea será posible gracias a los encoders ópticos HEDS 5540 C02, proporcionados también por Faulhaber. Como está indicado en su datasheet, la resolución de estos encoders es de 100 LPR⁴ y utilizan dos lectores ópticos en cuadratura (A y B), como muestra la imagen 24.

Así pues, conectando los canales A y B de los encoders al Arduino, es posible calcular cuánto han girado las ruedas según la siguiente expresión:

$$\Delta\theta = \frac{360 \cdot \Delta enc}{4 \cdot 100} \quad (3)$$

³Sitio web: <https://www.robotshop.com/es/en/60mm-aluminum-omni-wheel.html>.

⁴Este valor, en el código de la plataforma utilizado durante la Copa, es de 96 LPR y no de 100; esto es debido a una confusión en la referencia de modelo del encoder por parte de uno de los miembros del equipo. Este error, sin embargo, no es significativo en el desarrollo del robot.

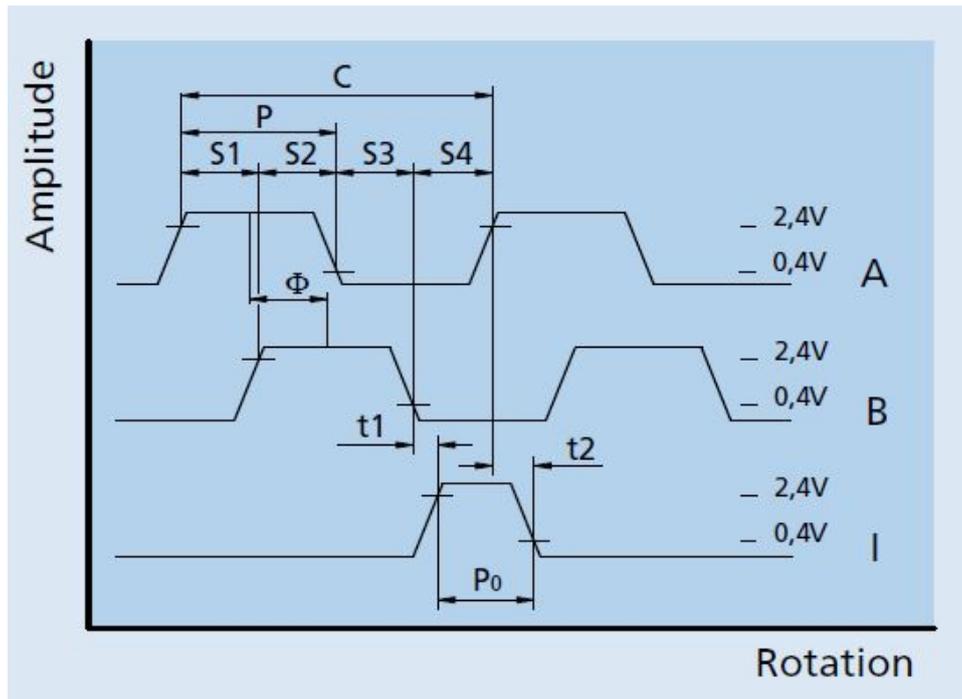


Figure 24. Estados lógicos de los lectores ópticos del encoder HEDS 5540 C02.
Fuente: Hoja de características del encoder HEDS 5540 C02.

donde $\Delta\theta$ está expresada en grados y Δenc es el incremento en las LPR muestreadas por el encoder.

En lo que respecta al factor 4 del denominador, no hay que olvidar que en los encoders, al realizar una decodificación en cuadratura, la salida proporciona dos señales, A y B, que alternan su estado entre HIGH y LOW, por lo que obtenemos 4 estados diferentes en la señal de salida (S1, S2, S3 y S4 en la figura 24). Finalmente obtenemos una resolución total de $\frac{1}{4 \cdot 100} = \frac{1}{400}$ revoluciones, es decir, 0,0157 radianes o 0,9° sobre el motor. Sin embargo, si tenemos en cuenta la relación de reducción de $\eta = \frac{1}{23}$, la resolución final con la que contamos es de $\frac{1}{4 \cdot 100 \cdot \eta^{-1}} = \frac{1}{9200}$ revoluciones, lo que equivale a $6,83 \cdot 10^{-4}$ radianes o 0,039°.

4.3. Controlador de motor

Para poder controlar tanto la velocidad como el sentido de giro de los motores mediante una señal PWM, es necesario utilizar un controlador de motor como el L298N, el cual será utilizado en esta plataforma y acepta tensiones de hasta 46V y puede proporcionar hasta 2A por cada una de sus dos salidas.

Siguiendo el esquema de puertas lógicas proporcionado en la página 1 de la datasheet del L298N, este puede ser esquematizado de la siguiente manera:



Figure 25. Controlador de motor L298N.
Fuente: Makerlab Electronics.

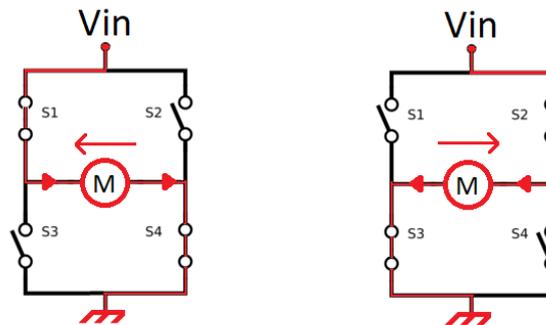


Figure 26. Esquema representativo de un puente en H.
Fuente: Autor.

Donde los distintos interruptores pueden ser expresados en función de las entradas IN1, IN2 y ENABLE del controlador:

$$\begin{aligned} S_1 &= IN1 \cdot EN & S_2 &= IN2 \cdot EN \\ S_3 &= \overline{IN1} \cdot EN & S_4 &= \overline{IN2} \cdot EN \end{aligned}$$

Se observa que si se cierran simultáneamente los interruptores S_1 y S_4 , la corriente recorre el circuito de V_{in} a tierra, lo que hace girar el motor en un sentido; viceversa, si se cierran simultáneamente los interruptores S_2 y S_3 , la corriente circula en sentido inverso, por lo que el sentido de giro del motor cambia. En este caso, los controladores de motor estarán alimentados a 12V.

Así pues, el sentido de giro del motor depende simplemente de los valores de IN1 e IN2. Para evitar que la rueda entre en rotación libre, el ENABLE del controlador debe estar siempre activo,

lo que permite tener un mayor control sobre esta. La velocidad del motor será controlada a través de una señal PWM que será enviada mediante un Arduino.

En caso de querer frenar la rueda de forma activa, es suficiente con que los valores de IN1 e IN2 sean iguales: de esta forma, el voltaje a las bornas del motor es el mismo, por lo que este estará bloqueado.

4.4. Arduino Mega 2560 Rev3

El encargado de controlar todo el movimiento de la plataforma a través de los controladores de motor es un Arduino Mega 2560 Rev3, ya que esta placa posee una gran cantidad de pines que serán necesarios para los *inputs* y *outputs* de motores y controladores. Posee también un mayor número de pines de interrupción, los cuales serán utilizados para las comunicaciones I2C con los otros Arduinos presentes en el resto del robot, siendo esta placa la placa “máster”.

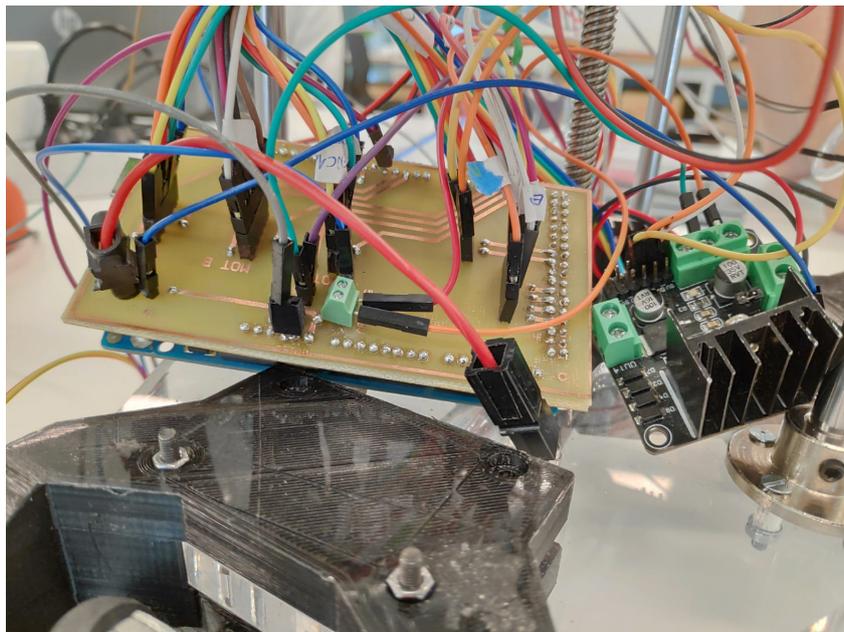


Figure 27. PCB para la plataforma móvil.

Fuente: Saïd Lourhaoui (equipo Eurobot 2019) usando los laboratorios CentraleSupélec.

La distribución electrónica y de pines se encuentra esquematizada en el Diagrama electrónico de la plataforma al final de este documento. Para garantizar una mayor facilidad de uso en la versión final de la plataforma, la PCB mostrada a en la figura 27 fue diseñada por otro de los miembros del equipo. Un diseño más detallado de esta puede encontrarse en la parte III de este documento.

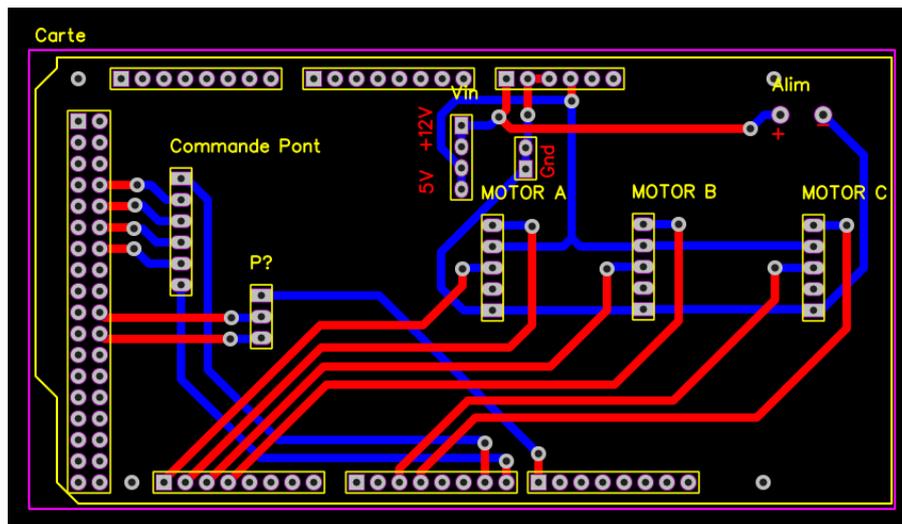


Figure 28. Diseño de la PCB para la plataforma móvil.

Fuente: Saïd Lourhaoui (equipo Eurobot 2019) usando los laboratorios CentraleSupélec.

Un detalle a tener en cuenta para el control PWM de los motores es que la frecuencia PWM del Arduino (por defecto, 490 Hz) es en este caso demasiado baja, lo que produce un ruido intenso y hace que los controladores se calienten considerablemente. Para solucionar esto, es necesario aumentar la frecuencia de los pines sobre los que se hará la comanda PWM de los motores, lo cual será explicado en detalle en el capítulo 6.

El Arduino, al igual que los controladores de motor, estará alimentado a 12V.

4.5. Alimentación

De forma general, el robot está alimentado a través de dos baterías diferentes de 12V de NiMH. La justificación del uso de este tipo de baterías en vez de unas de LiPo (más eficientes) son las altas restricciones que impone el reglamento de la Copa al respecto, debido a la peligrosidad de estas.

Así pues, la primera es una batería NiMH Conrad de 12V 2400 mAh de 10 celdas. Está será la encargada de proporcionar energía a todo aquello que necesite cierta potencia para su funcionamiento, como es el caso de los controladores de motor de la plataforma, pero también de los diferentes actuadores que se encuentran en el resto del robot, como las bombas de aire que permiten atrapar los discos de hockey.



Figure 29. Batería Conrad 12V 2400 mAh utilizada para la electrónica de potencia.

Fuente: Conrad.



Figure 30. Batería secundaria 12V 1600 mAh para las placas Arduino.

Fuente: RobotShop.

La segunda batería utilizada en el robot es una NiMH de 12V 1600 mAh⁵, y su función es alimentar las diferentes placas Arduino (entre ellas, la máster), así como otros elementos que requieren menor potencia, como los sensores ultrasonidos del sistema antichoque.



Figure 31. Regulador step-down LM2596.

Fuente: Addicore.

Un punto importante a tener en cuenta es que, como se verá más tarde en la explicación del Principio de funcionamiento y controlador PI de la plataforma, es conveniente que el voltaje de entrada al controlador de motores esté fijo a 12V; sin embargo, al cargar las baterías, estas aumentan su voltaje por encima de los 12V, lo cual puede afectar el comportamiento del PI. Por esta razón, la plataforma utiliza un regulador step-down LM2596 variable en salida, la cual está fijada a 12V. De este modo, y puesto que la batería se mantiene siempre cargada, se asegura un comportamiento óptimo del controlador PI de las ruedas.

⁵Sitio web: <https://www.robotshop.com/ca/fr/batterie-rechargeable-12v-1600-mah-ni-mh.html>

Teniendo en cuenta todo lo anterior encontramos que, por sí misma, la plataforma del robot consume a 12V unos 5A. Dado que la batería principal es de 2400 mAh, esto nos da una autonomía de una media hora de forma continuada; sin embargo, no hay que olvidar que la plataforma no es más que una de las partes que componen el robot, por lo que a estos 5A de consumo habrá que añadirles aquello consumido por el resto de sensores y actuadores utilizados por el resto del equipo.

Chapter 5

Principio de funcionamiento y controlador PI de la plataforma

CON toda la parte electrónica puesta en marcha, es hora de comenzar la parte de *software* de la plataforma.

La base, como se ha explicado con más detalle en las secciones 2.1.3 y 3.2, se desplazará gracias a las tres ruedas omnidireccionales que forman el Kiwi drive, las cuales serán comandadas a través de un Arduino Mega mediante una señal PWM enviada a los controladores de motor L298N.

En lo que respecta a la determinación de la posición, esta se realizará por *odometría*, es decir, usando la información proporcionada por los encoders sobre la rotación de las ruedas para estimar cambios en la posición del robot a lo largo del tiempo. Es importante mencionar que, ya que la odometría depende en gran medida de ecuaciones mecánicas, cualquier imperfección en la geometría de la plataforma a la hora de construirla (diferencia en el diámetro real y de fábrica de las ruedas, ángulo de montaje de motores, distancia de las ruedas al centro de la base) resultará en errores en el cálculo final de la posición.

Para asegurar un buen desplazamiento de la base, una buena idea es regular la velocidad de giro de cada una de las tres ruedas. Esto permitirá tener un control más fiable sobre la plataforma a la vez que evitar errores de comanda de los motores, como en el caso de que una misma comanda PWM haga girar un motor más rápido que otro.

5.1. Regulación de la velocidad de una sola rueda

El primer paso es pues la regulación de la velocidad individual de una de las ruedas. Esta sección tratará pues el modelizado del sistema motor-reductor-encoder junto con la rueda, así como el desarrollo de la regulación en velocidad de la misma.

El motor, como se ha explicado anteriormente, será comandado a través de una señal PWM enviada por el Arduino y cuyo valor `cmd` en el código estará comprendido entre los valores -255 y 255. En cuanto a la lectura de los encoders, esta se realizará gracias a la librería `Encoder` [10] y haciendo uso de la ecuación 3 explicada anteriormente para traducir los pulsos por segundo enviados por los encoders a velocidad angular.

5.1.1. Estudio y modelización del motor

El motor utilizado es de corriente continua e imanes permanentes, por lo que puede ser modelizado de la siguiente manera [9]:

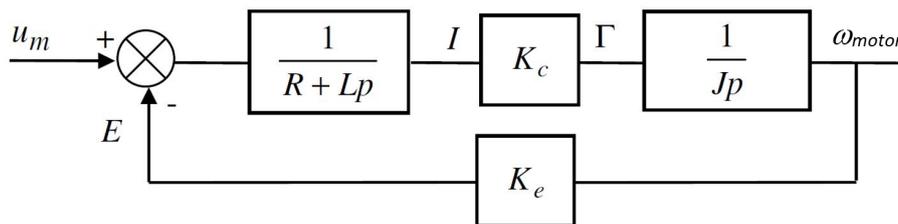


Figure 32. Modelización de un motor de corriente continua.

Fuente: Autor.

u_m	tensión de entrada	K_c	constante de par
I	corriente	K_e	constante de fcem
Γ	par motor	R	resistencia interna
ω_{motor}	velocidad de rotación del motor en rad/s	L	inductancia interna
E	fuerza contraelectromotriz (fcem)	J	inercia del rotor

Siendo conocidas las características del motor gracias a su datasheet, sabemos que

$$R = 1,46 \, \Omega ; L = 135 \, \mu\text{H} ; K_c = 18,57 \, \text{mNm/A} ; \\ K_e = 18,57 \, \text{mV} \cdot \text{rad}^{-1} \cdot \text{s} ; J = 1,5 \cdot 10^{-6} \, \text{kg} \cdot \text{m}^2$$

Así pues, sea p la variable de Laplace y según el esquema mostrado anteriormente, la función de transferencia del MCC, $T(p) = \omega_{motor}(p)/U_m(p)$, puede reescribirse bajo la forma

$$T(p) = \frac{\omega_{motor}(p)}{U_m(p)} = \frac{K_v}{(1 + \tau_e p)(1 + \tau_m p)} \quad (4)$$

donde K_v , τ_e y τ_m representan respectivamente la ganancia, la constante de tiempo eléctrica y la constante de tiempo mecánica, y cuyos valores son:

$$K_v = \frac{1}{K_e} = 53,84 \text{ V}^{-1} \cdot \text{rad} \cdot \text{s}^{-1}$$

$$\tau_e = \frac{L}{R} = 92,5 \text{ } \mu\text{s}$$

$$\tau'_m = \frac{JR}{K_c K_e} = 6,35 \text{ ms}$$

Sin embargo, tras hacer varios tests con la plataforma, se ha hallado que el valor de la constante de tiempo mecánica es en realidad mayor de lo esperado e igual a $\tau_m = 80 \text{ ms}$ a causa del peso del resto del robot, por lo que el valor de la inercia del motor utilizada en los cálculos siguientes será tomado en consecuencia, obteniendo así:

$$\tau_m = \tau_{m_exp} = 80 \text{ ms}$$

$$J = J_{exp} = \frac{\tau_m K_c K_e}{R} = 1,89 \cdot 10^{-5} \text{ kg} \cdot \text{m}^2$$

5.1.2. Regulación de la velocidad y síntesis del corrector

Una vez calculada la función de transferencia $T(p)$ del motor MCC, podemos representar el diagrama de bloques de la regulación de la velocidad del motor de la siguiente manera:

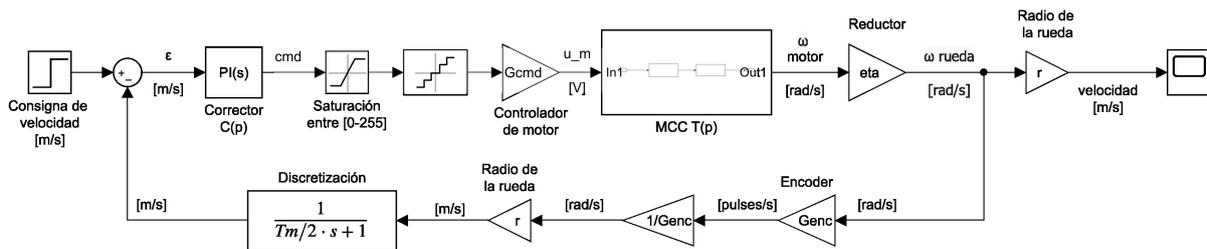


Figure 33. Diagrama de bloques de la regulación en velocidad de un solo motor.

Fuente: Autor.

donde

- $C(p)$ es el corrector a desarrollar.
- $G_{cmd} = \frac{12}{255}$ representa la conversión de la variable de comanda cmd en el voltaje que recibe el motor a través de los L298N, tomando como valor máximo 12V.

- $G_{enc} = \frac{4 \cdot 100}{2\pi}$ es el factor de conversión de rad/s a LPR/s (o pulsos/s) de los encoders.
- $H_d(p) = \frac{1}{1 + \frac{T_m}{2}p}$ es la función de discretización del modelo, con $T_m = 5$ ms el tiempo de muestreo elegido.

Así pues, si consideramos un corrector $C(p) = 1$, el lazo abierto del diagrama 33 está representado por

$$H_{v_BO}(p) = \frac{G_{cmd} \cdot T(p) \cdot \eta \cdot r \cdot H_d(p)}{G} \quad (5)$$

$$= \frac{G}{(1 + \tau_e p)(1 + \tau_m p)(1 + \frac{T_m}{2} p)}$$

con $G = G_{cmd} \cdot \eta \cdot r \cdot K_v = 0,0033$ la ganancia total del lazo.

Si trazamos su diagrama de Bode (figura 34), observamos que este posee los polos $\omega_m = \frac{1}{\tau_m} = 12,5$ rad/s, $\omega_M = \frac{1}{T_m} = 400$ rad/s y $\omega_e = \frac{1}{\tau_e} = 1,0815 \cdot 10^4$ rad/s. Observamos también que nunca llega a cortar la línea de los 0 dB y que el sistema es de clase 0 (es decir, no incluye un integrador puro), por lo que el sistema en lazo cerrado presentará un error estático no nulo frente a consignas en escalón.

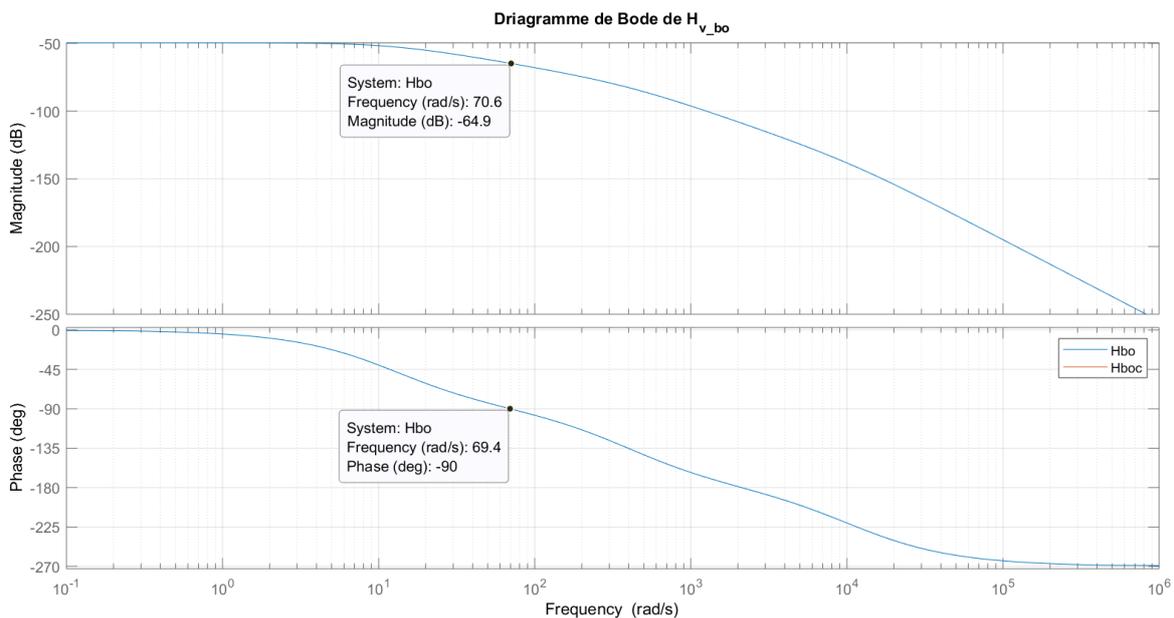


Figure 34. Diagrama de Bode del lazo abierto $H_{v_BO}(\omega)$.
Fuente: Autor.

El siguiente paso es calcular un corrector $C(p)$ que permita obtener una pulsación de corte $\omega_c = 70$ rad/s (pulsación a la cual la fase del lazo abierto pasa por los -90°). Asimismo, haciendo

Annexe 5 : Abaques définissant le dépassement et le temps du premier maximum en régime indiciel pour un second ordre

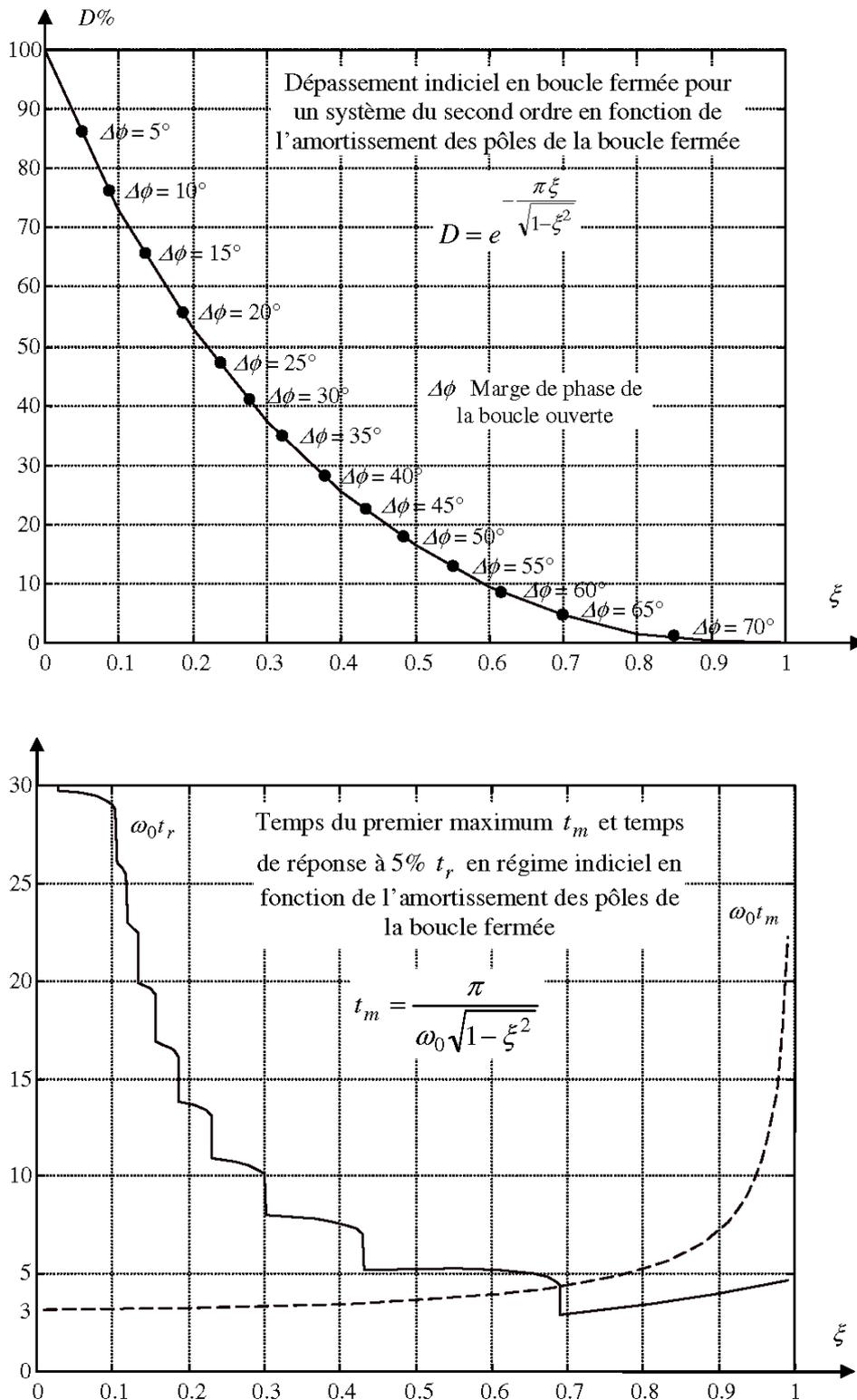


Figure 35. Ábacos para la determinación del sobrepico D y tiempo de respuesta para una respuesta temporal en un sistema de segundo orden.

Fuente: Systèmes automatiques, Christina VLAD [9]

uso del ábaco mostrado en la figura 35, se busca conseguir un sobrepico D de alrededor de 10%, lo que implica un margen de fase $\Delta\phi$ de alrededor de 60° .

Nos encontramos pues con el pliego de condiciones siguiente:

$$\omega_c = 70 \text{ rad/s}$$

$$\Delta\phi \geq 60^\circ$$

$$D \approx 10\%$$

Para lograr este objetivo, el corrector $C(p)$ elegido es del tipo PI

$$C(p) = K_P \left(1 + \frac{1}{T_i p} \right) = K_P \left(\frac{1 + T_i p}{T_i p} \right) \quad (6)$$

ya que, teniendo en cuenta lo explicado anteriormente, la acción integral del PI permite anular el error estático frente a una consigna en escalón; y la acción proporcional, ajustar la pulsación de corte para obtener los 70 rad/s calculados y, consecuentemente, un margen de fase $\Delta\phi$ de unos 60° .

Como se puede observar en la figura 34, a $\omega_c = 70 \text{ rad/s}$, la fase del lazo abierto $H_{v_BO}(j\omega_c)$ es de $\phi = -90^\circ$. Además, sabemos que un margen de fase de $\Delta\phi = 60^\circ$ del lazo abierto corregido H_{v_BOC} equivale a una fase de -120° . Con estos datos, calculamos el valor de T_i correspondiente para obtener un margen de fase de 60° a ω_c :

$$\arg(H_{v_BOC}(j\omega_c)) = -120^\circ;$$

$$\arg(C(j\omega_c)) + \arg(H_{v_BO}(j\omega_c)) = -120^\circ;$$

$$-90^\circ + \arctg(T_i\omega_c) - 90^\circ = -120^\circ;$$

$$T_i\omega_c = \tan 60 = \sqrt{3}$$

con lo que

$$T_i = \frac{\sqrt{3}}{\omega_c} = 25 \text{ ms}^6$$

⁶Por temas de comodidad, este valor ha sido redondeado a $T_i = 30 \text{ ms}$ en el código fuente de la plataforma.

La ganancia K_P del corrector puede ahora ser determinada fijando la pulsación de corte a $\omega_c = 70$ rad/s y conociendo la ganancia del valor del lazo abierto a ω_c :

$$\begin{aligned} |H_{v_BOC}(j\omega_c)|_{dB} &= 0; \\ 20 \log |H_{v_BO}(j\omega_c)C(j\omega_c)| &= 0 \text{ dB}; \\ 20 \log |H_{v_BO}(j\omega_c)| + 20 \log C(j\omega_c) &= 0 \text{ dB}; \\ 20 \log \left(K_P \frac{1 + T_i j\omega_c}{T_i j\omega_c} \right) - 64,7 \text{ dB} &= 0 \text{ dB}; \end{aligned}$$

por lo que finalmente obtenemos

$$K_P = 1492$$

Con lo que el corrector deseado es el siguiente:

$$C(p) = K_P \left(\frac{1 + T_i p}{T_i p} \right) = 1492 \left(\frac{1 + 0,025p}{0,025p} \right) \quad (7)$$

que, según comprobamos en la figura 36, tiene en efecto una pulsación de corte $\omega_c \approx 70$ rad/s para 0 dB y un margen de fase $\phi = 60^\circ$.

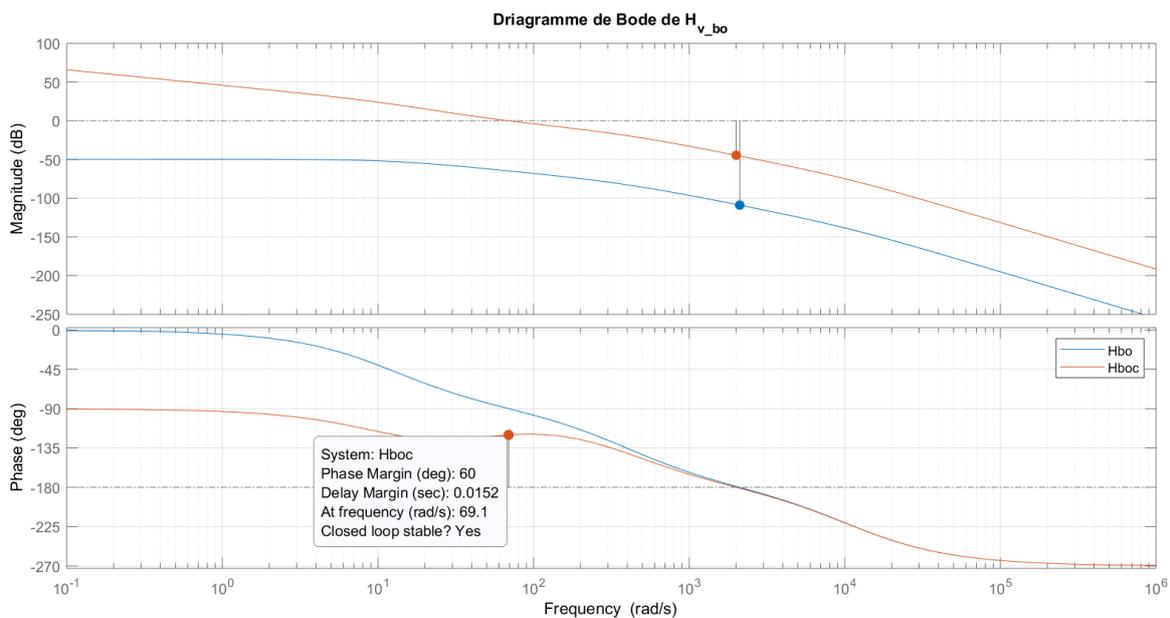


Figure 36. Diagrama de Bode del lazo abierto corregido $H_{v_BOC}(j\omega)$.

Fuente: Autor.

Con ayuda del ábaco mostrado en la figura 35 y sabiendo que $\omega_c \cdot t_{pico} \approx 3$, calculamos un tiempo de pico $t_{pico} \approx \frac{3}{\omega_c} = 43$ ms y un tiempo de respuesta $t_{resp} \approx \frac{5.5}{\omega_c} = 80$ ms, lo cual concuerda con lo reflejado en la figura 37. Observamos también que, aunque el sobrepico es de $D = 16\%$ (algo mayor que el 10% esperado), el error estático obtenido es nulo.

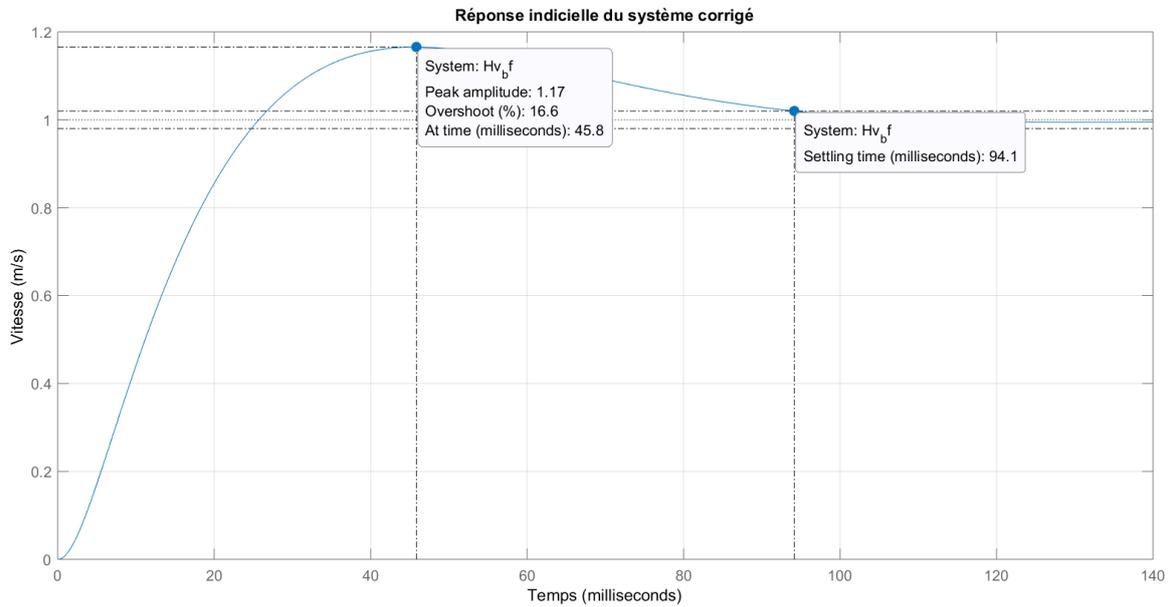


Figure 37. Respuesta temporal del lazo cerrado $H_{v_bf}(j\omega)$.

Fuente: Autor.

5.2. Regulación de la posición del robot

Una vez regulada la velocidad individual de cada rueda, es hora de regular la posición global del robot.

Recordemos que la posición y rotación del robot está representada por el vector $[x, y, \theta]^T$; su velocidad global, por $[v_x, v_y, \Omega]^T$; y las velocidades individuales de los motores, por $[v_1, v_2, v_3]^T$. Asimismo, gracias a la matriz de cambio de base A explicada anteriormente en la sección 3.2, el paso de la cinemática global del robot a la de cada una de las ruedas, y viceversa, es trivial.

Lo único que falta pues es el operador encargado de convertir el error respecto a la posición deseada $[\varepsilon_x, \varepsilon_y, \varepsilon_\theta]^T$ a la consigna en velocidad total $[v_x, v_y, \Omega]^T_c$. Dicho operador será denominado *Rampa*.

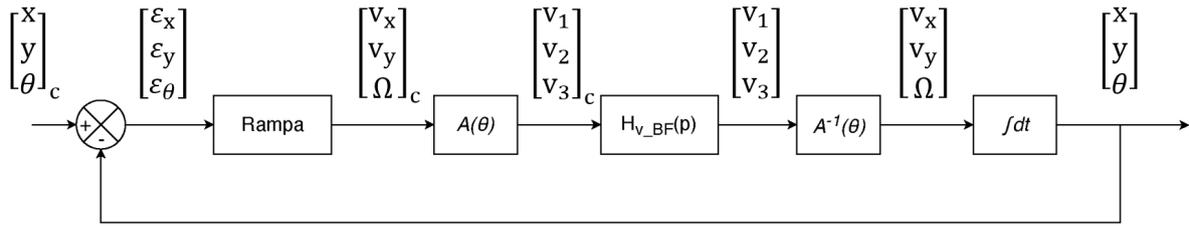


Figure 38. Diagrama de bloques de la regulación en posición del robot.

Fuente: Autor.

5.2.1. Operador *Rampa*

El objetivo de este operador es convertir un error en la posición global del robot en una comanda de velocidad de este.

Para ello, debemos tener en cuenta las siguientes condiciones:

1. La aceleración del robot debe estar limitada a $\pm a_{max}$ para evitar un posible deslizamiento de las ruedas y, por lo tanto, un error en el cálculo de la posición del robot.
2. La velocidad máxima de la plataforma debe ser de $\pm v_{max}$, ya que esta se encuentra físicamente limitada por la velocidad que pueden proporcionar los propios motores.

Teniendo en cuenta estas condiciones, obtenemos un perfil de velocidad trapezoidal como el mostrado en la figura 39.

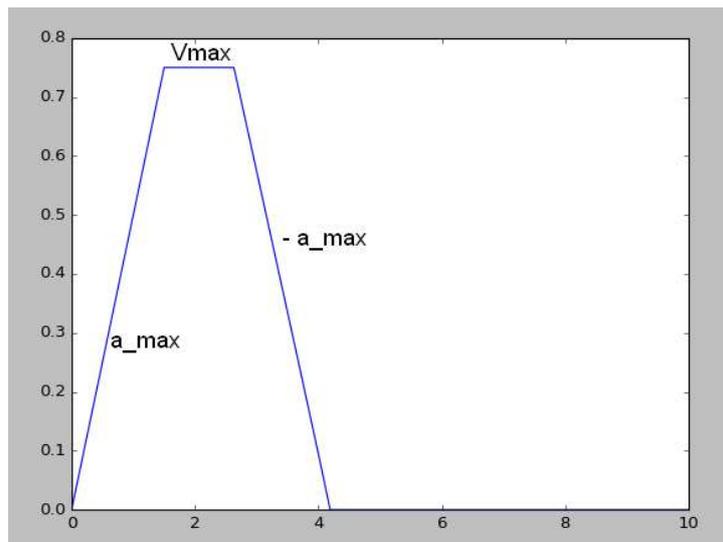


Figure 39. Perfil de la velocidad global del robot en función del tiempo calculado por el operador *Rampa*.

Fuente: Autor.

Así pues, la velocidad del robot será siempre la mínima entre:

- La recta aceleración, de pendiente a_{max} .
- La velocidad máxima v_{max} .
- La recta de desaceleración, de pendiente $-a_{max}$, tras la cual el robot deberá tener una velocidad nula al llegar a su destino.

5.3. Puesta en práctica y análisis de resultados

Tras haber hecho un estudio teórico del movimiento del robot y haber programado el código fuente de la plataforma (ver capítulo 6), es hora de poner en práctica el modelo.

Mientras que los valores calculados anteriormente para el corrector PI dan buenos resultados para tests de regulación de las velocidades individuales de las ruedas en vacío, al hacer pruebas de posición usando la base completa (es decir, usando los tres motores) el robot oscila de manera muy brusca, llegando casi a vibrar.

Ya que este comportamiento suele ser síntoma de una ganancia proporcional K_P demasiado alta, lo lógico es disminuir esta variable y observar su comportamiento. Siguiendo pues un procedimiento heurístico de forma a obtener una ganancia K_P que haga al robot ser lo suficientemente rápido sin producir oscilaciones, el valor óptimo encontrado es

$$K_P^{exp} = 300$$

Esta diferencia de ganancias puede ser debido a haber realizado todas las pruebas de velocidades individuales en vacío, ya que la geometría de la base no permite hacer rodar la plataforma usando un solo motor, así como a no haber tenido en cuenta la influencia del resto del cuerpo del robot sobre ella.

Se obtiene así pues la respuesta temporal del sistema que muestra la figura 40 para una velocidad de consigna de 0,6 m/s. Como se puede comprobar, se obtiene una respuesta temporal que, aunque tiene un tiempo de respuesta de algo más de 100 ms, no posee error estático y, además, en ningún momento sobrepasa la velocidad de consigna de 0,6 m/s.

Respuesta a una consigna de velocidad de 0.6 m/s
 $K_P = 300$, $T_i = 0.03s$

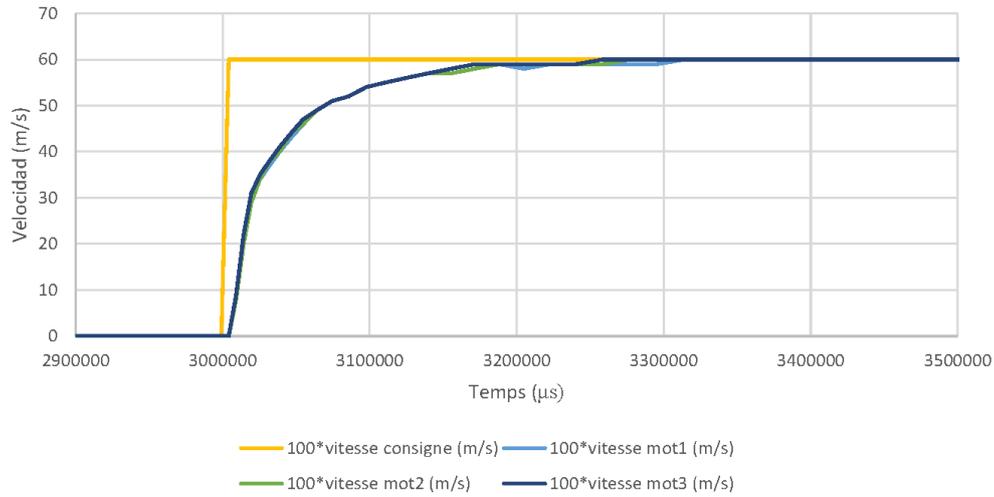


Figure 40. Respuesta temporal experimental de cada motor obtenida con un tiempo de muestreo $T_m = 5$ ms para una velocidad de consigna de 0,6 m/s.
 $K_P = 300$, $T_i = 30$ ms , $T_m = 5$ ms.
Fuente: Autor.

Una vez obtenidos unos resultados de regulación de velocidades individuales satisfactorios, la última prueba es la regulación de la posición del robot. Para ello, se hace avanzar al robot en línea recta una distancia de un metro en la dirección del eje x y se observa su comportamiento.

Respuesta de la posición del robot ante una consigna en posición de (1,0,0)m
 $K_P=300$, $T_i=0.03$

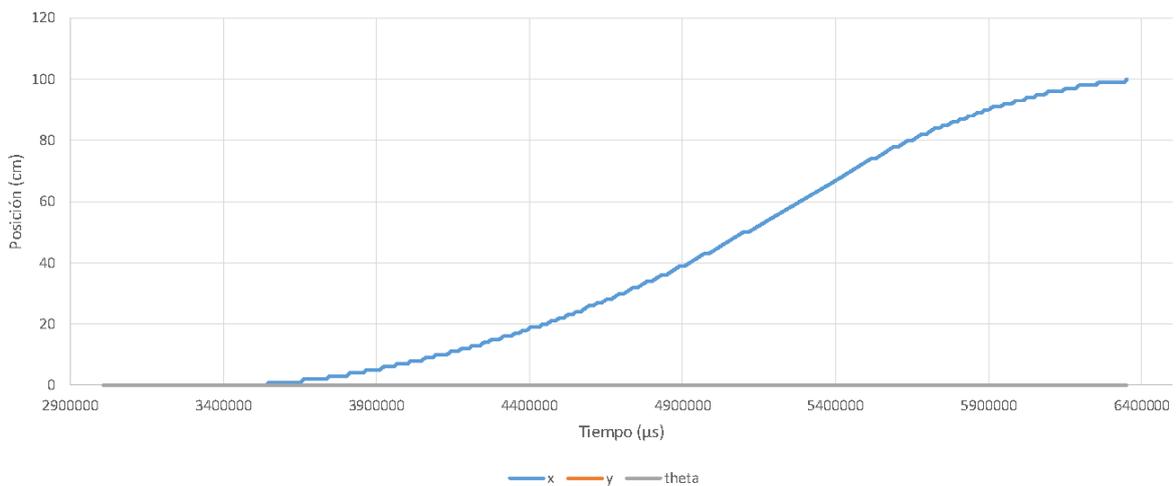


Figure 41. Respuesta en posición del robot ante una consigna de posición de (1,0,0) m.
 $K_P = 300$, $T_i = 30$ ms , $T_m = 5$ ms.
Fuente: Autor.

Como se puede apreciar en la figura 41, el robot realiza una trayectoria a lo largo del eje x de manera suave y no violenta, acelerando al inicio y desacelerando al final del recorrido, hasta llegar a destino.



Figure 42. Respuesta de la velocidad total del robot ante una consigna de posición de (1, 0, 0) m.

$$K_P = 300, T_i = 30 \text{ ms}, T_m = 5 \text{ ms.}$$

Fuente: Autor.

La figura 42 muestra que la velocidad horizontal v_x del robot (en azul) es igual a la velocidad total v de este (en amarillo), y que ambas tienen el aspecto trapezoidal esperado, similar al de la figura 39, gracias al operador *Rampa*.

Las velocidades individuales de cada motor se encuentran representadas en la figura 43.

Visto esto, gracias al nuevo valor de K_P obtenemos el corrector $C(p)$ definitivo:

$$C(p) = K_P^{exp} \left(\frac{1 + T_i p}{T_i p} \right) = 300 \left(\frac{1 + 0,025p}{0,025p} \right) \quad (8)$$

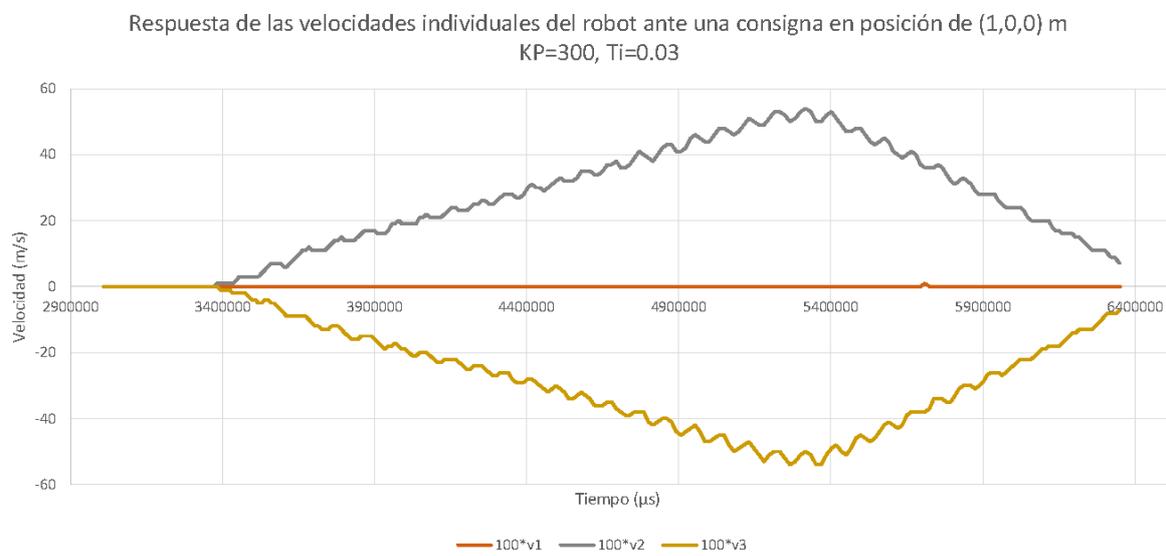


Figure 43. Respuesta de las velocidades individuales de los motores ante una consigna de posición de (1, 0, 0) m.

$$K_P = 300, T_i = 30 \text{ ms}, T_m = 5 \text{ ms}.$$

Fuente: Autor.

Chapter 6

Código fuente

ESTE capítulo abordará el código del robot responsable de controlar la plataforma en el terreno de juego y proporcionará una perspectiva general al respecto de la estructura que sigue⁷.

Al utilizar como placa controladora un Arduino Mega 2560, la elección natural de lenguaje de programación es C++, ya que el objetivo principal de este lenguaje es la programación orientada a objetos. En este caso, dichos objetos corresponderán a los tres motores que, finalmente, compondrán el conjunto `Robot`. Asimismo, el IDE utilizado es PlatformIO (ecosistema de Atom) debido a su facilidad de uso y buena interfaz en comparación con el IDE nativo de Arduino.

No hay que olvidar que, de forma general, cada clase está compuesta por dos tipos de archivos:

- Archivo `Clase.h` o encabezado, donde se encuentran las declaraciones de las diferentes funciones y variables de la clase.
- Archivo `Clase.cpp` o archivo de implementación, donde, como su propio nombre indica, se implementan y desarrollan las funciones de la clase.

Asimismo, como fue mencionado anteriormente en la sección 4.4, es necesario aumentar la frecuencia de los pines sobre los que se hará la comanda PWM de los motores en el Arduino (que en este caso serán el 6, 7 y 8) siguiendo las instrucciones del foro Arduino [8], lo cual se resume en añadir la siguiente línea de código al `void setup`:

```
TCCR4B = TCCR4B & B11111000 | B00000001
```

⁷Si se desea obtener más detalle sobre el código fuente de la plataforma, este, junto con el resto del código del robot, puede encontrarse al final de este documento o en el GitLab del equipo Eurobot 2019 [2].

Nótese que la elección de los pines 6, 7 y 8 se ha realizado con el objetivo de utilizar el `timer 4` del Arduino y no el `timer 0`, ya que esto alteraría el funcionamiento de funciones como `delay()` y `millis()`.

6.1. Funcionamiento general

El control de la posición del robot está comandado por la clase `Robot`, en la cual están definidos los diferentes atributos de este, como son su diámetro, su posición y velocidad actuales y de consigna, o las restricciones de movimiento (aceleración y velocidad máximas). Asimismo, en esta clase se encuentran las funciones que le permitirán desplazarse hacia una coordenada específica $[x, y, \theta]_c^T$ impuesta en el programa principal, `goto_test.cpp`.

Para ello, dentro de la clase `Robot` encontramos la función `go_to()`, que recibe las coordenadas de consigna $[x, y, \theta]_c^T$ y las procesa para hacer avanzar al robot hacia allí siempre y cuando no haya obstáculos en su camino, los cuales serán detectados mediante los sensores ultrasonidos que posee el robot a lo largo del perímetro del piso superior.

Una vez el robot haya obtenido permiso para avanzar, se actualizan la velocidad y posición actuales de este mediante la función `actualize_speed_position()` según el giro de las ruedas medido por la librería `Encoder` [10]. Acto seguido, la función `calcul_consigne_robot()` calcula las componentes de velocidad de consigna globales del robot $[v_x, v_y, \Omega]_c^T$ teniendo para ello en cuenta la aceleración y velocidad máxima definidas (ver sección 5.2.1 sobre el Operador *Rampa*).

Finalmente, una vez obtenidas dichas velocidades globales de consigna, se pueden calcular las velocidades individuales de los motores mediante la función `calculate_command_lmotor()` que se encuentra en la clase `Motor` y la clase `Calculator`, que cumple la función de la matriz de cambio de base A calculada en la sección 3.2. Es además en esta función donde se encuentra la regulación PI de las velocidades individuales de las ruedas.

Así pues, el comportamiento del código puede ser resumido en las siguiente etapas:

1. Definición de la posición deseada del robot a través de la función `go_to()`.
2. Medida del desplazamiento de las ruedas gracias a los encoders.
3. Cálculo de la velocidad global de consigna del robot $[v_x, v_y, \Omega]_c^T$ haciendo uso del operador *Rampa*.

4. Cálculo de las velocidades individuales de consigna $[v_1, v_2, v_3]_c^T$ de cada motor gracias a la clase `Calculator` y aplicando la regulación PI.
5. Comanda de la velocidad de los motores a través de la función `calculate_command_1motor()`.

Chapter 7

Análisis de resultados

COMO se ha visto en los capítulos anteriores, los motores están correctamente regulados en velocidad, y el robot, en posición. Sin embargo, a pesar de lo mostrado en la figura 41 respecto a la distancia recorrida por el robot, en la realidad este no es tan exacto como cree ser.

Tras hacer varios tests de distancia, se han obtenido los siguientes resultados:

Distancia consigna (mm)	Distancia recorrida (mm)	Diferencia (mm)	% Error
200	180	20	10%
500	455	45	9%
800	735	65	8,1%
1000	905	95	9,5%
1200	1095	105	8,8%
1400	1290	110	7,9%
1600	1465	135	8,4%

Table 1. Error en la posición de la plataforma para una distancia de consigna.

Fuente: Autor.

Estos muestran que, aunque el propio robot haya medido que ha recorrido la distancia de consigna, en la realidad comete un error de alrededor del 9%, como muestra el cuadro anterior. Este hecho es debido principalmente a las imprecisiones acumuladas en la medida de los encoders causado por el deslizamiento de las ruedas al desplazarse de forma no “habitual”; es decir, lateralmente.

Para solucionar este problema, y teniendo en cuenta que el error medido está comprendido entre 8% y 10%, un factor de corrección `correction` de 0,919 ha sido calculado de forma experimental y añadido en el código a la distancia medida por el robot con el objetivo de compensar las imprecisiones mencionadas anteriormente, obteniendo los siguientes resultados:

Distancia consigna (mm)	Distancia recorrida (mm)	Diferencia (mm)	% Error
200	202	2	1%
500	490	10	2%
800	800	0	0%
1000	1003	3	0,3%
1200	1200	0	0%
1400	1344	56	4%
1600	1552	48	3%

Table 2. Error en la posición de la plataforma para una distancia de consigna tras la adición del factor de corrección.

Fuente: Autor.

Como se puede observar, el error en distancia ha sido disminuido de forma considerable; en lo que respecta a la precisión en la rotación sobre sí mismo, el robot comete un error de alrededor de 10° tras haber girado 20 veces sobre su eje, por lo que, en lo que respecta a la rotación, los resultados son correctos.

Teniendo en cuenta que las distancias recorridas por el robot en la competición no superarán los tres metros en total, los resultados finales obtenidos tras la adición del factor de corrección son más que satisfactorios y permitirán al resto del equipo Eurobot 2019 poder realizar correctamente el resto de las tareas del robot.

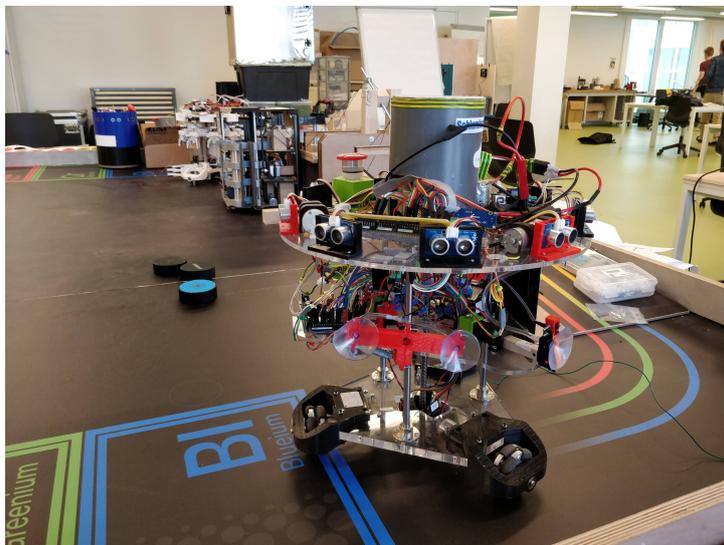


Figure 44. Versión final del robot del equipo de CentraleSupélec Eurobot 2019.

Fuente: Autor.

Todos los partidos de la competición, incluidos aquellos en los que participa el equipo CroCS, pueden encontrarse en el canal YouTube de Planète Sciences [5].

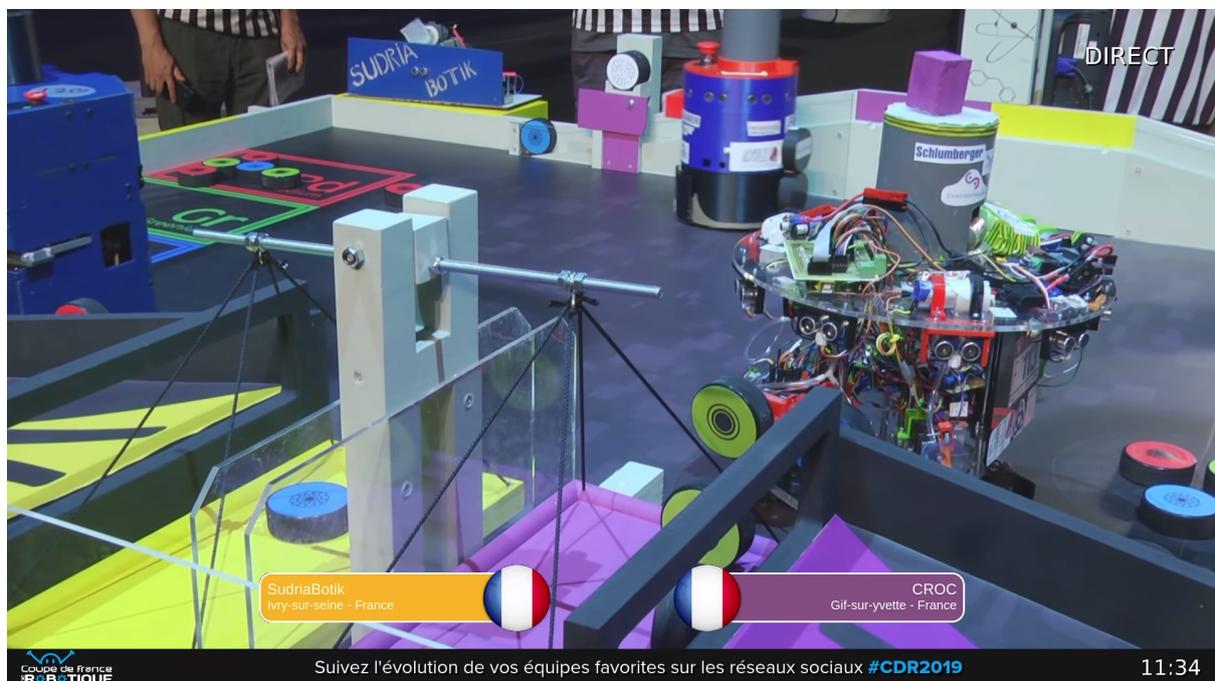


Figure 45. Instantánea del robot en uno de los partidos de la Copa 2019.

Fuente: Canal YouTube de Planète Sciences.

Bibliography

- [1] **Coupe de France de Robotique**, *Sitio web de la Copa de Robótica de Francia*. Último acceso: 22/05/2019.
<https://www.coupederobotique.fr/edition-2019/le-concours/>
- [2] **Lison Abecassis, Daniel Horcajo de la Cruz, Juan Mateo Vaello, Ayman El Moskatin, Saïd Lourhaoui, Stéphane Chevallier, Daniel Calderas**, *Eurobot 2019*, (2019), repositorio GitLab del equipo CRoCS 2019. Último acceso: 02/06/2019.
<https://gitlab.viarezo.fr/2017horcajod/eurobot-2019>
- [3] **DroneBotWorkshop**, *Building a Real Robot – Part 4 – Selecting Wheels*, blog de electrónica y robótica. Last access: 18/06/2019.
<https://dronebotworkshop.com/real-robot-004/>
- [4] **Quentin Bennehard, Armand Bouvier, Simon Le Cléac'H, Gabriel Moneyron, Théophile Montagne, Victor Zhang, Julien Zhou**, *Rapport final de projet concours*, informe de proyecto innovación. CentraleSupélec, 2017.
- [5] **Planète Sciences**, canal YouTube oficial de la asociación co-organizadora Planète Sciences. Último acceso: 30/06/2019.
<https://www.youtube.com/user/PlaneteSciences/>
- [6] **Swart, Dirk** (11 septiembre 2014). *R/C Omniwheel Robot – Electronics | Make*, blog de electrónica y robótica. Último acceso: 18/04/2019.
<https://makezine.com/projects/make-40/kiwi/>
- [7] **DroneBotWorkshop**, *DF Robot LIDAR Sensors – Getting Started with LIDAR*, blog de electrónica y robótica. Último acceso: 13/08/2019.
https://dronebotworkshop.com/getting-started-with-lidar/#LIDAR_vs_RADAR
- [8] **valerio_sperati** en el foro Arduino, *Changing PWM frequency in Arduino Mega 2560*. Último acceso: 20/02/2019.
<https://forum.arduino.cc/index.php?topic=72092.0>

- [9] **Christina VLAD**, *Systèmes automatiques*, asignatura del curso 2018 de la universidad CentraleSupélec de París.
- [10] **Paul Stoffregen**, *Quadrature Encoder Library for Arduino*, repositorio GitHub. Último acceso: 02/04/2019.
<https://github.com/PaulStoffregen/Encoder>
- [11] **Géza Szayer**, *Kinematic and Dynamic Limits of Holonomic Mobile Robots*, Budapest University of Technology and Economics, tesis doctoral. Último acceso: 15/08/2019.
<https://repozitorium.omikk.bme.hu/bitstream/handle/10890/5484/ertekezes.pdf>
- [12] **Raúl Rojas**, *Omnidirectional control*, Freie Universität Berlin, proyecto de fin de carrera. Último acceso: 15/08/2019.
<http://robocup.mi.fu-berlin.de/buch/omnidrive.pdf>

PART II

CÓDIGO FUENTE



Chapter 1

Clases C++ utilizadas en la plataforma

1.1. Clase Motor

1.1.1. Motor.h

```
#ifndef MOTOR_H
#define MOTOR_H
#include <Encoder.h>

class Motor
{
public:
    Motor(int pin_IN1, int pin_IN2, int pin_enable, int pin_encoder_main, int
        pin_encoder_secondary);
    void command_motor(int command);
    void initialize_motor();
    void initPins();
    void read_wheel_speed(int dt);
    void calculate_command_lmotor(int dt);
    void test_motor(double vitesse_consigne, long dt);
    void reset_motor();

    float integrale;
    float erreur;
    int commande;
    float v;
    float v_consigne;

    int pin_IN1;
    int pin_IN2;
    int pin_enable;
    Encoder encoder;
    int encoder_last_value;
    boolean goes_forward; // true if the motor was going forward, else if it was going
        backwards (arbitrary decided).

protected:
```

```

private:

};

#endif // MOTOR_H

```

1.1.2. Motor.cpp

```

#include "Motor.h"
#include <Encoder.h>
float const D = 0.06; // Wheel's diameter
float const KP = 300;
float const Ti = 0.03;
float const BORNE_INTEGRALE = 255*Ti/KP;
//float const KI = 1000; //not used with new schema

Motor::Motor(int pin_IN1, int pin_IN2, int pin_enable, int pin_encoder_main, int
pin_encoder_secondary):
encoder(pin_encoder_main, pin_encoder_secondary)
{
    this->pin_IN1 = pin_IN1;
    this->pin_IN2 = pin_IN2;
    this->pin_enable = pin_enable;
    this->integrale = 0;
    this->erreur = 0;
    this->commande = 0;
    this->v = 0; // in m/s
    this->v_consigne = 0;
    this->goes_forward = false;
}

void Motor::initPins() { // initialisation des pins liees au moteur
pinMode(this->pin_IN1, OUTPUT);
pinMode(this->pin_IN2, OUTPUT);
pinMode(this->pin_enable, OUTPUT);
digitalWrite(this->pin_enable, LOW);
}

void Motor::command_motor(int command){ // command between -255 and 255
if (command > 0) {
    if ( !this->goes_forward ){ // passage au "point mort" pour changer de sens
        digitalWrite(this->pin_enable, LOW);
        digitalWrite(this->pin_IN1, LOW);
    }
    digitalWrite(this->pin_IN2, LOW);
    digitalWrite(this->pin_enable, HIGH);
    analogWrite(this->pin_IN1, command);
    this->goes_forward = true;
}
else if (command < 0) {
    if (this->goes_forward){
        digitalWrite(this->pin_enable, LOW);

```

```

        digitalWrite(this->pin_IN1, HIGH);
    }
    digitalWrite(this->pin_IN2, HIGH);
    digitalWrite(this->pin_enable, HIGH);
    analogWrite(this->pin_IN1, 255+command);
    this->goes_forward = false;
}
else {
    if(goes_forward){
        digitalWrite(pin_enable, HIGH);
        digitalWrite(pin_IN2, LOW);
        digitalWrite(pin_IN1, LOW);
    }
    else {
        digitalWrite(pin_enable, HIGH);
        digitalWrite(pin_IN2, HIGH);
        digitalWrite(pin_IN1, HIGH);
    }
}
}
}

/*
  Reads the encoder and actualises the motor speed.
*/
void Motor::read_wheel_speed(int dt)
{
    int encl_value = this->encoder.read();
    float wheel_distance = D*PI*(encl_value - this->encoder_last_value)/384.0/23.0; //en m
    this->v = 1000000.0*wheel_distance/dt;
    this->encoder_last_value = encl_value;
}

//Calculates command of one motor, after previous asservissement
void Motor::calculate_command_1motor(int dt){ // calcul de la commande
    this->erreur = this->v_consigne - this->v; // erreur entre la consigne et la realite
    this->integrale = this->integrale + this->erreur*dt/1000000.0; // integrale de
        cette erreur approximee avec les rectangles
    this->integrale = max(-BORNE_INTEGRALE,min(BORNE_INTEGRALE,this->integrale)); // on borne
        l'integrale pour ne pas qu'elle explose
    int cmd = KP *(this->erreur + (1/Ti)*this->integrale); // calcul de la commande
    this->commande = max(-255,min(255,cmd)); // borne de la commande
}

void Motor::test_motor(double vitesse_consigne, long dt){
    this->v_consigne = vitesse_consigne;

    if (dt > 5000) {
        this->read_wheel_speed(dt);
        // Serial.print("Motor : ");
        // Serial.println(this->v);
        calculate_command_1motor(dt);
        command_motor(this->commande);
    }
}
}

```

```
void Motor::reset_motor(){
    this->integrale = 0.;
    this->commande = 0;
    this->v = 0;
    this->v_consigne = 0;
}
```

1.2. Clase asservissement

1.2.1. asservissement.h

```
#ifndef ASSERVISSEMENT_H_INCLUDED
#define ASSERVISSEMENT_H_INCLUDED
#include "Robot.h"

// Parametres a determiner et modifier

//void consigne_moteur(float vx_consigne, float vy_consigne, float omega_consigne, Robot&
    robot);
float distance_critique (float v, float a_max);
float vitesse_critique(float erreur, float a_max);
float omega_critique(float erreur_theta, float alpha_max);
//float min(float a, float b);
//float max(float a, float b);

#endif // ASSERVISSEMENT_H_INCLUDED
```

1.2.2. asservissement.cpp

```
#include "asservissement.h"

float distance_critique (float v, float a_max){
    return v*v/(2*a_max);
}

float vitesse_critique(float erreur, float a_max){
    return sqrt(2*erreur*a_max);
}

float omega_critique(float erreur_theta, float alpha_max){
    return sqrt(2*erreur_theta*alpha_max);
}
/*
void consigne_moteur(float vx_consigne, float vy_consigne, float omega_consigne, Robot robot){
    float multiplier = 1;
    float V1_consigne = calculV1(vx_consigne, vy_consigne,omega_consigne,robot.theta,R);
    float V2_consigne = calculV2(vx_consigne, vy_consigne,omega_consigne,robot.theta,R);
    float V3_consigne = calculV3(vx_consigne, vy_consigne,omega_consigne,robot.theta,R);

    if (abs(robot.x - robot.x_consigne)<distance_critique(robot.vx, A_MAX_ROBOT))
```

```

    {
        multiplicier = abs(robot.x - robot.x_consigne)/distance_critique(robot.vx, A_MAX_ROBOT);
    }
    if (abs(robot.y - robot.y_consigne)<distance_critique(robot.vy, A_MAX_ROBOT))
    {
        multiplicier = min(abs(robot.y - robot.y_consigne)/distance_critique(robot.vy,
            A_MAX_ROBOT), multiplicier);
    }
    if (abs(robot.omega - robot.omega_consigne)<distance_critique(robot.omega, A_MAX_ROBOT))
    {
        multiplicier = min(abs(robot.omega - robot.omega_consigne)/distance_critique(robot.omega,
            A_MAX_ROBOT), multiplicier);
    }
    if (robot.v+robot.omega*D> V_MAX_MOTEUR)
    {
        multiplicier = min(V_MAX_MOTEUR/robot.v+robot.omega*D, multiplicier);
    }

    V1_consigne = V1_consigne * multiplicier;
    V2_consigne = V2_consigne * multiplicier;
    V3_consigne = V3_consigne * multiplicier;

    robot.motor1.v_consigne = V1_consigne;
    robot.motor2.v_consigne = V2_consigne;
    robot.motor3.v_consigne = V3_consigne;
}
*/

//
// float min(float a, float b){
//     float m = a;
//     if (a>b){m = b;}
//     return m;
// }
//
// float max(float a, float b){
//     float m = a;
//     if (a<b){m = b;}
//     return m;
// }

```

1.3. Clase calculator

1.3.1. calculator.h

```

#ifndef CALCULATOR_H
#define CALCULATOR_H

#include "Arduino.h"

// const float PI = 3.14159265;

```

```

float calculVx( float v1, float v2, float v3, float theta, float R);

float calculVy( float v1, float v2, float v3, float theta, float R);

float calculOmega( float v1, float v2, float v3, float theta, float R);

float calculV1(float vx, float vy, float omega, float theta, float R);

float calculV2(float vx, float vy, float omega, float theta, float R);

float calculV3(float vx, float vy, float omega, float theta, float R);

//bool arrive(float x1, float y1, float theta1, float x2, float y2, float theta2);

float distance (float x, float y);

#endif

```

1.3.2. calculator.cpp

```

#include "calculator.h"

//A-1 ligne 1
float calculVx( float v1, float v2, float v3, float theta, float R){
    float v_moy = (v1 + v2 + v3) /3;
    return (2.0/3)*((v1)*sin(theta) + (v2)*sin(-theta+(PI/3)) - (v3)*sin(theta + (PI/3)));
}

//A-1 ligne 2
float calculVy( float v1, float v2, float v3, float theta, float R){
    float v_moy = (v1 + v2 + v3) /3;
    return (2.0/3)*(-(v1)*cos(theta) + (v2)*cos(-theta + (PI/3)) + (v3)*cos(theta + (PI/3)));
}

//A-1 ligne 3
// the variable 'theta' is not actually used
float calculOmega( float v1, float v2, float v3, float theta, float R){
    return -(v1 + v2 + v3)/(3*R);
}

//A ligne 1
float calculV1(float vx, float vy, float omega, float theta, float R){
    return vx*sin(theta) - vy*cos(theta) - omega*(R);
}

```

```

//A ligne 2
float calculV2(float vx, float vy, float omega, float theta, float R){
    return vx*sin(theta + (2*PI/3)) - vy*cos(theta + (2*PI/3)) - omega*(R);
}

//A ligne3
float calculV3(float vx, float vy, float omega, float theta, float R){
    return vx*sin(theta + (4*PI/3)) - vy*cos(theta + (4*PI/3)) - omega*(R);
}

float distance (float x, float y)
{
    return sqrt(x*x + y*y);
}

```

1.4. Clase Robot

1.4.1. Robot.h

```

#ifndef ROBOT_H
#define ROBOT_H
#include "Motor.h"
#include "calculator.h"
#include "asservissement.h"
#include "check_distances.h"

const float R = 0.1453; //en metres. Rayon du robot (avant : 0.147) => 142
const float D = 0.06; //en metres. Diametre de la roue du robot
const float A_MAX_ROBOT = 1.6; // en m/s2
const float A_MAX_ROBOT_DEC = 0.5; // en m/s2
const float ALPHA_MAX = 5; // en rad/s2
const float ALPHA_MAX_DEC = 1;
const float OMEGA_MAX_ROBOT = 1.6; // en rad/s
const float V_MAX_MOTEUR = 0.4; // en m/s, max = 0.75
const float TOLERANCE_XY = 0.005;
const float TOLERANCE_THETA = 0.01;
const float V_ARRET = 0.1;
const float OMEGA_ARRET = 0.1;
const float V_MAX_ROBOT = 0.6; //TODO change this value to something that takes into account
    the position sensor range
const float distance_detection = 19; // in cm
const float distance_detection_balance = 4;
const float correction = 0.919;

class Robot
{
public:
    Robot(Motor& motor1, Motor& motor2, Motor& motor3);
    void start_ultrasounds();
    void actualize_speed_position(int dt);
    //float get_consigne (float v, float dx, float a_max, int dt);
    void calcul_consigne_robot(int dt);
    //void calcul_consigne_moteurs();
    void calcul_commande_all_motors(int dt);
}

```

```

void activate_command();
bool arrived();
void display();
void update_position(float real_x, float real_y, float real_theta);
//void display_courbes();
void go_to(float x, float y, float theta);
void go_to_close_to_balance(float x, float y, float theta);
void stop_motors();
void deactivate_ultrasounds(bool value);
float x;
float y;
float theta;
float x_consigne;
float y_consigne;
float theta_consigne;
float x_saved;
float y_saved;
float theta_saved;

float vx;
float vy;
float omega;
float vx_consigne;
float vy_consigne;
float omega_consigne;

float v;

bool waiting;
Check_distances Checker;

Motor& motor1;
Motor& motor2;
Motor& motor3;

bool ultrasound_desactivated;

protected:

private:

};

#endif // ROBOT_H

```

1.4.2. Robot.cpp

```

#include "Robot.h"
#include "Motor.h"
#include "check_distances.h"
#include "Time.h"

Robot::Robot(Motor& motor1, Motor& motor2, Motor& motor3):
motor1(motor1), motor2(motor2), motor3(motor3)
{
    this->x = 0;    // in m

```

```

this->y = 0;    // in m
this->theta = 0; // in rad
this->x_consigne = 0;
this->y_consigne = 0;
this->theta_consigne = 0;

this->vx = 0; // in m/s
this->vy = 0; // in m/s
this->omega = 0; // in rad/s
this->vx_consigne = 0;
this->vy_consigne = 0;
this->omega_consigne = 0;
this->v = 0;

this->Checker = Check_distances();
}

void Robot::start_ultrasounds() {
    this->Checker.setup_wire();
}

void Robot::actualize_speed_position(int dt)
{
    this->motor1.read_wheel_speed(dt);
    this->motor2.read_wheel_speed(dt);
    this->motor3.read_wheel_speed(dt);

    this->omega = calculOmega(motor1.v, motor2.v, motor3.v, this->theta, R);
    this->theta += dt * this->omega / 1000000;

    this->vx = calculVx(motor1.v, motor2.v, motor3.v, this->theta, R);
    this->vy = calculVy(motor1.v, motor2.v, motor3.v, this->theta, R);
    this->x += dt * this->vx / 1000000 * correction;
    this->y += dt * this->vy / 1000000 * correction;

    this->v = sqrt(this->vx * this->vx + this->vy * this->vy);
}

void Robot::calcul_consigne_robot(int dt)
{
    float dx = this->x_consigne - this->x;
    float dy = this->y_consigne - this->y;
    float dtheta = this->theta_consigne - this->theta;

    float vx_acc = abs(this->vx) + A_MAX_ROBOT*dt/1000000; // this->vx could be
        this->vx_consigne
    float vy_acc = abs(this->vy) + A_MAX_ROBOT*dt/1000000; // this->vy could be
        this->vy_consigne
    float omega_acc = abs(this->omega) + ALPHA_MAX*dt/1000000;
    // float omega_preconsigne = 1.6; //TODO how to calculate omega right
        // 1.6 est une bonne valeur max
    float vx_dec = vitesse_critique(abs(dx), A_MAX_ROBOT_DEC);
    float vy_dec = vitesse_critique(abs(dy), A_MAX_ROBOT_DEC);
    float omega_dec = omega_critique(abs(dtheta), ALPHA_MAX_DEC);

```

```

// If the robot is going forward (from x=1 to x=4, for example)
if(dx>0){
    this->vx_consigne = min(vx_dec,min(vx_acc,V_MAX_ROBOT));
    // if(abs(this->vx_consigne-vx_dec)<0.0001){
    //     Serial.print("VX_DEC \n");
    // }
    // else if(abs(this->vx_consigne-vx_acc)<0.0001){
    //     Serial.print("VX_ACC \n");
    // }
    // else if(abs(this->vx_consigne-V_MAX_ROBOT)<0.0001){
    //     Serial.print("VX_MAX \n");
    // }
    // else{
    //     Serial.print("NOOOOOONE OF THEM");
    // }
}

// If the robot is going backwards (from x=4 to x=1, for example)
else{
    this->vx_consigne = -min(vx_dec,min(vx_acc,V_MAX_ROBOT));
    // if(abs(this->vx_consigne+vx_dec)<0.0001){
    //     Serial.print("VX_DEC \n");
    // }
    // else if(abs(this->vx_consigne+vx_acc)<0.0001){
    //     Serial.print("VX_ACC \n");
    // }
    // else if(abs(this->vx_consigne+V_MAX_ROBOT)<0.0001){
    //     Serial.print("VX_MAX \n");
    // }
    // else{
    //     Serial.print("NOOOOOONE OF THEM");
    // }
}

// If the robot is going forward (from y=1 to y=4, for example)
if(dy>0){
    this->vy_consigne = min(vy_dec,min(vy_acc,V_MAX_ROBOT));
    // if(abs(this->vy_consigne-vy_dec)<0.0001){
    //     Serial.print("VY_DEC \n");
    // }
    // else if(abs(this->vy_consigne-vy_acc)<0.0001){
    //     Serial.print("VY_ACC \n");
    // }
    // else if(abs(this->vy_consigne-V_MAX_ROBOT)<0.0001){
    //     Serial.print("VY_MAX \n");
    // }
    // else{
    //     Serial.print("NOOOOOONE OF THEM");
    // }
}

// If the robot is going backwards (from y=4 to y=1, for example)
else{
    this->vy_consigne = -min(vy_dec,min(vy_acc,V_MAX_ROBOT));
    // if(abs(this->vy_consigne+vy_dec)<0.0001){

```

```

// Serial.print("VY_DEC \n");
// }
// else if(abs(this->vy_consigne+vy_acc)<0.0001){
// Serial.print("VY_ACC \n");
// }
// else if(abs(this->vy_consigne+V_MAX_ROBOT)<0.0001){
// Serial.print("VY_MAX \n");
// }
// else{
// Serial.print("NOOOOOONE OF THEM");
// }
}

if(dtheta>0){
    this->omega_consigne = min(omega_dec, min(omega_acc, OMEGA_MAX_ROBOT));
}

else {
    this->omega_consigne = - min(omega_dec, min(omega_acc, OMEGA_MAX_ROBOT));
}

// // If going from theta=0o to theta=30o for example
// if(dtheta>TOLERANCE_THETA){
// this->omega_consigne = omega_preconsigne;
// }
//
// // If going from theta=30o to theta=0o for example
// else if(dtheta < - TOLERANCE_THETA){
// this->omega_consigne = - omega_preconsigne;
// }
//
// else {
// this->omega_consigne = 0;
// }

float V1_consigne = calculV1(this->vx_consigne, this->vy_consigne,
    this->omega_consigne,this->theta,R);
float V2_consigne = calculV2(this->vx_consigne,
    this->vy_consigne,this->omega_consigne,this->theta,R);
float V3_consigne = calculV3(this->vx_consigne,
    this->vy_consigne,this->omega_consigne,this->theta,R);

this->motor1.v_consigne = V1_consigne;
this->motor2.v_consigne = V2_consigne;
this->motor3.v_consigne = V3_consigne;

// Serial.print("v1: ");
// Serial.print(motor1.v_consigne);
// Serial.print("\t");
// Serial.print("v2: ");
// Serial.print(motor2.v_consigne);
// Serial.print("\t");
// Serial.print("v3: ");
// Serial.println(motor3.v_consigne);
}

void Robot::calcul_commande_all_motors(int dt)

```

```

{
  motor1.calculate_command_1motor(dt);
  motor2.calculate_command_1motor(dt);
  motor3.calculate_command_1motor(dt);
}

void Robot::activate_command()
{
  motor1.command_motor(motor1.commande);
  motor2.command_motor(motor2.commande);
  motor3.command_motor(motor3.commande);
}

bool Robot::arrived() {
  bool hasArrived = ((abs(this->x - this->x_saved) <= TOLERANCE_XY) &&
    (abs(this->y - this->y_saved) <= TOLERANCE_XY) &&
    (abs(this->theta - this->theta_saved) <= TOLERANCE_THETA) &&
    (abs(this->omega) <= OMEGA_ARRET) && (abs(this->vx) <= V_ARRET) && (abs(this->vy) <=
      V_ARRET));

  return hasArrived;
}

void Robot::display() {
  Serial.print(micros());
  Serial.print(";");

  Serial.print(this->x);
  Serial.print(";");
  Serial.print(this->y);
  Serial.print(";");
  Serial.print(this->theta);
  Serial.print(";");

  Serial.print(this->v);
  Serial.print(";");
  Serial.print(this->vx);
  Serial.print(";");
  Serial.print(this->vy);
  Serial.print(";");
  Serial.print(this->omega);
  Serial.print(";");

  Serial.print(motor1.commande);
  Serial.print(";");
  Serial.print(motor2.commande);
  Serial.print(";");
  Serial.print(motor3.commande);
  Serial.print(";");

  Serial.print(motor1.integrale);
  Serial.print(";");
  Serial.print(motor2.integrale);
  Serial.print(";");
  Serial.print(motor3.integrale);
  Serial.print(";");
}

```

```

Serial.print (motor1.erreur);
Serial.print ("");
Serial.print (motor2.erreur);
Serial.print ("");
Serial.print (motor3.erreur);
Serial.print ("");

Serial.print (motor1.v_consigne);
Serial.print ("");
Serial.print (motor2.v_consigne);
Serial.print ("");
Serial.print (motor3.v_consigne);
Serial.print ("");

Serial.print (motor1.v);
Serial.print ("");
Serial.print (motor2.v);
Serial.print ("");
Serial.print (motor3.v);
Serial.print ("");

Serial.println(""); //saut de ligne
}

void Robot::go_to (float x, float y, float theta)
{
    this->x_consigne = x;
    this->y_consigne = -y;
    this->theta_consigne = theta*2*PI/360;
    this->x_saved = x;
    this->y_saved = -y;
    this->theta_saved = theta*2*PI/360;
    this->waiting = false;
    long t1 = micros();
    long tprim1 = micros();
    long dt1 = tprim1 - t1;
    long t2 = micros();
    long tprim2 = micros();
    long dt2 = tprim2 - t2;
    this->motor1.reset_motor();
    this->motor2.reset_motor();
    this->motor3.reset_motor();

    while (!this->arrived()){
        tprim1 = micros();
        dt1 = tprim1 - t1;

        // For the ultrasound sensors:
        if (dt1 > 50000){
            float alpha = atan2(y - y_saved, x - x_saved);
            this->Checker.calcul_distances();
            int num_capteur_to_look = this->Checker.num_capteur(alpha, theta);
            // Serial.println(num_capteur_to_look);
            // float c1 = this->Checker.get_a_distance(num_capteur_to_look);
            // float c1 = this->Checker.get_distances(num_capteur_to_look);

```

```

float c1 = this->Checker.get_all_distances();
float readed_distance = c1; //We keep the value of the lowest distance
c1 = 100; //Increase the value of c1 for next interaction
// Serial.println(readed_distance);
if (readed_distance < distance_detection) {
    if (!waiting) {
        waiting = true;
        // //rampeVx.set_a_max(acceleration_freinage);
        // //rampeVy.set_a_max(acceleration_freinage);
        // x_saved = x_consigne;
        // y_saved = y_consigne;
        // x_consigne = x;
        // y_consigne = y;
        // // delay(1000); // wait for a second
        Serial.println("OBJECT DETECTED");
        this->stop_motors();
    }
    if (waiting){
        // x_consigne = x;
        // y_consigne = y;
        this->stop_motors();
    }
}
else if (readed_distance < distance_detection + 3 and waiting){
    // x_consigne = x;
    // y_consigne = y;
    this->stop_motors();
}
else if (waiting) {
    Serial.println("No longer detecting object");
    waiting = false;
    x_consigne = x_saved;
    y_consigne = y_saved;
    // delay(1000); // wait for a second
}
t1 = micros();
}

// For the motors:
tprim2 = micros();
dt2 = tprim2 - t2;
if ( dt2 > 5000 && !(Time::timesUp()) ){
    //display();
    t2=tprim2;
    if(!waiting){
        this->actualize_speed_position(dt2);
        this->calcul_consigne_robot(dt2);
        this->calcul_commande_all_motors(dt2);
        this->activate_command();
    }
    // if(waiting){
    //     x_consigne = x;
    //     y_consigne = y;
    // }
    // this->display();
}
}

```

```

}

Serial.println("Arrived");

this->motor1.commande = 0;
this->motor2.commande = 0;
this->motor3.commande = 0;
this->vx_consigne = 0.;
this->vy_consigne = 0.;
this->omega_consigne = 0.;
this->vx = 0.;
this->vy = 0.;
this->omega = 0.0;
this->activate_command();
// delay(2000);
}

void Robot::go_to_close_to_balance (float x, float y, float theta)
{
    this->x_consigne = x;
    this->y_consigne = -y;
    this->theta_consigne = theta*2*PI/360;
    this->x_saved = x;
    this->y_saved = -y;
    this->theta_saved = theta*2*PI/360;
    this->waiting = false;
    long t1 = micros();
    long tprim1 = micros();
    long dt1 = tprim1 - t1;
    long t2 = micros();
    long tprim2 = micros();
    long dt2 = tprim2 - t2;
    this->motor1.reset_motor();
    this->motor2.reset_motor();
    this->motor3.reset_motor();

    while (!this->arrived()){
        tprim1 = micros();
        dt1 = tprim1 - t1;
        if (dt1 > 50000){
            // Serial.println("ENTERED");
            float alpha = atan2(y - y_saved, x - x_saved);
            this->Checker.calcul_distances();
            int num_capteur_to_look = this->Checker.num_capteur(alpha, theta);
            // Serial.println(num_capteur_to_look);
            // float c1 = this->Checker.get_a_distance (num_capteur_to_look);
            // float c1 = this->Checker.get_distances (num_capteur_to_look);
            float c1 = this->Checker.get_all_distances ();
            float readed_distance = c1; //We keep the walue of the lowest distance
            c1 = 100; //Increase the value of c1 for next interaction
            // Serial.println(readed_distance);
            if (ultrasound_desactivated){

            }
            else if (readed_distance < distance_detection_balance) {
                if (!waiting) {

```

```

        waiting = true;
        // //rampeVx.set_a_max(acceleration_freinage);
        // //rampeVy.set_a_max(acceleration_freinage);
        // x_saved = x_consigne;
        // y_saved = y_consigne;
        // x_consigne = x;
        // y_consigne = y;
        // // delay(1000); // wait for a second
        Serial.println("OBJECT DETECTED");
        this->stop_motors();
    }
    if (waiting){
        // x_consigne = x;
        // y_consigne = y;
        this->stop_motors();
    }
}
else if(readed_distance < distance_detection_balance and waiting){
    // x_consigne = x;
    // y_consigne = y;
    this->stop_motors();
}
else if (waiting) {
    Serial.println("Back in businessssssss");
    waiting = false;
    x_consigne = x_saved;
    y_consigne = y_saved;
    // delay(1000); // wait for a second
}
t1 = micros();
}

tprim2 = micros();
dt2 = tprim2 - t2;
if ( dt2 > 5000 && !(Time::timesUp())){
    //display();
    t2=tprim2;
    if(!waiting){
        this->actualize_speed_position(dt2);
        this->calcul_consigne_robot(dt2);
        this->calcul_commande_all_motors(dt2);
        this->activate_command();
    }
    // if(waiting){
    //     x_consigne = x;
    //     y_consigne = y;
    // }
    // this->display();
}
}

Serial.println("Arrived");

this->motor1.commande = 0;
this->motor2.commande = 0;
this->motor3.commande = 0;

```

```

this->vx_consigne = 0.;
this->vy_consigne = 0.;
this->omega_consigne = 0.;
this->vx = 0.;
this->vy = 0.;
this->omega = 0.0;
this->activate_command();
// delay(2000);
}

void Robot::desactivate_ultrasounds(bool value){
    ultrasound_desactivated = value;
}

void Robot::stop_motors(){
    this->motor1.commande = 0;
    this->motor2.commande = 0;
    this->motor3.commande = 0;
    this->vx_consigne = 0.;
    this->vy_consigne = 0.;
    this->omega_consigne = 0.;
    this->vx = 0.;
    this->vy = 0.;
    this->omega = 0.0;
    this->activate_command();
}

void Robot::update_position(float real_x, float real_y, float real_theta){
    this->x = real_x;    // in m
    this->y = - real_y;  // in m
    this->theta = 2*PI/360 * real_theta;    // in o
}

```

1.5. Clase goto-test

1.5.1. goto-test.h

```

#ifndef goto_test_included
#define goto_test_included
#include "Robot.h"
#include "Connect_actuators.h"

void init_goto_test();
bool action_goto_test(int side_of_the_arena);

#endif

```

1.5.2. goto-test.cpp

```

#include "Robot.h"
#include "test_motor.h"
#include "goto_test.h"
#include "Connect_actuators.h"

```

```

Motor motor20(6,46,48,19,17); // Motor A
Motor motor10(7,42,44,18,16); // Motor B
Motor motor30(8,36,34,2,4); // Motor C

Robot myRobot(motor10, motor20, motor30);
Connect_actuators myConnect = Connect_actuators();

void init_goto_test(){
    pinMode(LED_BUILTIN, OUTPUT);
    myConnect.setup_wire();
}

bool action_goto_test(int side_of_the_arena_purple){
    int correctYaxis = 1;
    // 0 = right side looking at the balance
    if (!side_of_the_arena_purple) {
        Serial.println(side_of_the_arena_purple);
        correctYaxis = 1;
        Serial.println("Playing in the right side of the arena (yellow)");

        myConnect.do_next_action();
        myRobot.update_position(0.21867, correctYaxis*0.6747, 0);
        // myRobot.go_to(0.11867, 3, 0);
        // delay(3000);
        Serial.println("Starting move.");
        // Petit distributeur :
        Serial.println("Starting petit distributeur: ");
        myRobot.desactivate_ultrasounds(true);
        // myRobot.go_to_close_to_balance(0.21867, correctYaxis*1.210, correctYaxis*0);
        myRobot.go_to_close_to_balance(0.21867, correctYaxis*1.790, correctYaxis*0);
        myRobot.go_to_close_to_balance(0, correctYaxis*1.790, correctYaxis*0);
        myRobot.go_to_close_to_balance(0, correctYaxis*1.910, correctYaxis*0);
        myRobot.go_to_close_to_balance(0.02, correctYaxis*1.910, correctYaxis*0);
        myRobot.update_position(0.07637, correctYaxis*1.850, 1);
        Serial.println("Finished petit distributeur ");
        // Palets bleu et vert
        Serial.println("Starting palets bleu et vert: ");
        myConnect.do_next_action();
        // delay(2000);
        Serial.println("Finished palets bleu et vert ");
        // Grand distributeur :
        Serial.println("Starting grand distributeur spin: ");
        // myRobot.go_to(0.09637, correctYaxis*1.34, correctYaxis*30);
        // delay(1000);
        myRobot.go_to_close_to_balance(0.2, correctYaxis*1.34, 1);
        myRobot.go_to_close_to_balance(0.2, correctYaxis*1.34, correctYaxis*30);
        Serial.println("Finished grand distributeur spin");
        // Recuperer le palet vert :
        Serial.println("Starting palet vert grand distributeur: ");
        // delay(500);
        myRobot.go_to_close_to_balance(0.588, correctYaxis*1.34, correctYaxis*30);
        myConnect.do_next_action();
        myRobot.go_to_close_to_balance(0.588, correctYaxis*1.3, correctYaxis*30);
        myRobot.desactivate_ultrasounds(false);
        // delay(2000);
    }
}

```

```

Serial.println("Finished palet vert");

myRobot.go_to(0.790, correctYaxis*1.3, correctYaxis*(30+120));
// delay(500);
myRobot.go_to(0.790, correctYaxis*1.355, correctYaxis*(30+120));
myConnect.do_next_action();
// delay(2000);
Serial.println("Finished palet bleu");
myRobot.go_to(0.796, correctYaxis*1.3, correctYaxis*(30+120));
// myRobot.go_to(1.295, 1.342, 30+120);
// delay(500);
Serial.println("Starting balance 1");

myRobot.go_to_close_to_balance(1.305, correctYaxis*1.31, correctYaxis*(30+120));
myRobot.go_to_close_to_balance(1.305, correctYaxis*1.417, correctYaxis*(30+120));
myConnect.do_next_action();
// delay(2000);
myRobot.go_to_close_to_balance(1.205, correctYaxis*1.317, correctYaxis*(30+120));
myRobot.go_to_close_to_balance(1.205, correctYaxis*1.317, correctYaxis*40);
myRobot.go_to_close_to_balance(1.300, correctYaxis*1.420, correctYaxis*40);
myConnect.do_next_action();
// delay(2000);
myRobot.go_to_close_to_balance(1.255, correctYaxis*1.367, correctYaxis*40);
myRobot.desactivate_ultrasounds(true);
myRobot.go_to_close_to_balance(1.255, correctYaxis*1.367, correctYaxis*0);
myRobot.go_to_close_to_balance(1.290, correctYaxis*1.437, correctYaxis*0);
myConnect.do_next_action();
myRobot.desactivate_ultrasounds(false);
// delay(2000);
Serial.println("Finished balance 1");
myRobot.go_to_close_to_balance(1.205, correctYaxis*1.347, correctYaxis*0);
myRobot.go_to_close_to_balance(0.876, correctYaxis*1.347, correctYaxis*0);
myRobot.go_to(0.876, correctYaxis*1.394, correctYaxis*0);
Serial.println("Taking two lasts palets");
myConnect.do_next_action();
myRobot.go_to_close_to_balance(1.205, correctYaxis*1.347, correctYaxis*0);
myRobot.desactivate_ultrasounds(true);
myRobot.go_to_close_to_balance(1.305, correctYaxis*1.397, correctYaxis*0);
myRobot.go_to_close_to_balance(1.305, correctYaxis*1.430, correctYaxis*0);
myRobot.go_to_close_to_balance(1.305, correctYaxis*1.477, correctYaxis*0);
myConnect.do_next_action();
myRobot.desactivate_ultrasounds(false);

/* ### THIS WAS THE FIRST VERSION WITH TWO INDIVIDUAL VENTOSES ### */
// Serial.println("Backup from balance");
// myRobot.go_to(1.295, correctYaxis*1.342, correctYaxis*30);
// Serial.println("We go to the red closer to balance");
// myRobot.go_to(0.906, correctYaxis*1.342, correctYaxis*30);
// myRobot.go_to(0.906, correctYaxis*1.362, correctYaxis*30);
//
// Serial.println("Starting picking first red palet : ");
// myConnect.do_next_action();
// // delay(2000);
// Serial.println("Finished picking up red palet");

```

```

// myRobot.go_to(0.9055, correctYaxis*1.3, correctYaxis*30);
// myRobot.go_to(0.9055, correctYaxis*1.342, correctYaxis*(30+120));
//
// myRobot.go_to(1.002, correctYaxis*1.342, correctYaxis*(30+120));
//
// Serial.println("Starting picking second green palet : ");
// myConnect.do_next_action();
// // delay(2000);
// Serial.println("Finished picking up second green palet");
//
// Serial.println("Go back to the balance");
// myRobot.go_to_close_to_balance(1.310, correctYaxis*1.342, correctYaxis*(30+120));
//
// myRobot.go_to_close_to_balance(1.310, correctYaxis*1.400, correctYaxis*(30+120));
//
// Serial.println("Deposer second green palet in the balance");
// myConnect.do_next_action();
// Serial.println("Deposer first red palet in the balance");
// myRobot.go_to_close_to_balance(1.310, correctYaxis*1.400, correctYaxis*30);
// myConnect.do_next_action();

Serial.println("Backup from balance");
myRobot.go_to(1.310, correctYaxis*1.342, correctYaxis*30);

myRobot.go_to(0.700, correctYaxis*1.342, correctYaxis*30);
Serial.println("Starting picking first red palet : ");
myConnect.do_next_action();
// delay(2000);
Serial.println("Finished picking up red palet");

myRobot.go_to(0.700, correctYaxis*1.342, correctYaxis*(30+120));

myRobot.go_to(0.495, correctYaxis*1.342, correctYaxis*(30+120));
Serial.println("Starting picking second red palet : ");
myConnect.do_next_action();
// delay(2000);
Serial.println("Finished picking up second red palet ");

myRobot.go_to(0.24, correctYaxis*1.342, correctYaxis*(30+120));
myRobot.go_to(0.24, correctYaxis*0.4667, correctYaxis*(30+120));
myConnect.do_next_action();
Serial.println("Deposed second red palet ");
// delay(2000);

// myRobot.go_to(0.3, 0.5, 30+120+120);
myConnect.do_next_action();
Serial.println("Deposed first red palet ");

Serial.println("\nCOMPETITION FINISHED\n");

// myRobot.go_to(0.11867, correctYaxis*0.4667, correctYaxis*0);
// delay(500);

// myRobot.go_to(0, 0, 0);
delay(5000);
}

```

```

// 1 = left side looking at the balance
else{
  Serial.println(side_of_the_arena_purple);
  correctYaxis = -1;
  Serial.println("Playing in the left side of the arena (purple)");

  Serial.println(correctYaxis);

  myConnect.do_next_action();
  myRobot.update_position(0.21867, correctYaxis*0.6747, 0);
  // myRobot.go_to(0.11867, 3, 0);
  // delay(3000);
  Serial.println("Starting move.");
  // Petit distributeur :
  Serial.println("Starting petit distributeur: ");
  myRobot.desactivate_ultrasounds(true);
  // myRobot.go_to_close_to_balance(0.21867, correctYaxis*1.210, correctYaxis*0);
  myRobot.go_to_close_to_balance(0.21867, correctYaxis*1.770, correctYaxis*0);
  myRobot.go_to_close_to_balance(0, correctYaxis*1.770, correctYaxis*0);
  myRobot.go_to_close_to_balance(0, correctYaxis*1.910, correctYaxis*0);
  myRobot.go_to_close_to_balance(0.02, correctYaxis*1.910, correctYaxis*0);
  myRobot.update_position(0.08637, correctYaxis*1.840, 1);
  Serial.println("Finished petit distributeur ");
  // Palets bleu et vert
  Serial.println("Starting palets bleu et vert: ");
  myConnect.do_next_action();
  // delay(2000);
  Serial.println("Finished palets bleu et vert ");
  // Grand distributeur :
  Serial.println("Starting grand distributeur spin: ");
  // myRobot.go_to(0.09637, correctYaxis*1.34, correctYaxis*30);
  // delay(1000);
  myRobot.go_to_close_to_balance(0.2, correctYaxis*1.34, 1);
  myRobot.go_to_close_to_balance(0.2, correctYaxis*1.34, correctYaxis*30);
  Serial.println("Finished grand distributeur spin");
  // Recuperer le palet vert :
  Serial.println("Starting palet vert grand distributeur: ");
  // delay(500);
  myRobot.go_to_close_to_balance(0.588, correctYaxis*1.34, correctYaxis*30);
  myConnect.do_next_action();
  myRobot.go_to_close_to_balance(0.588, correctYaxis*1.3, correctYaxis*30);
  myRobot.desactivate_ultrasounds(false);
  // delay(2000);
  Serial.println("Finished palet vert");

  myRobot.go_to(0.781, correctYaxis*1.3, correctYaxis*(30+120));
  // delay(500);
  myRobot.go_to(0.781, correctYaxis*1.355, correctYaxis*(30+120));
  myConnect.do_next_action();
  // delay(2000);
  Serial.println("Finished palet bleu");
  myRobot.go_to(0.781, correctYaxis*1.3, correctYaxis*(30+120));
  // myRobot.go_to(1.295, 1.342, 30+120);
  // delay(500);
  Serial.println("Starting balance 1");

```

```

myRobot.go_to_close_to_balance(1.305, correctYaxis*1.34, correctYaxis*(30+120));
myRobot.go_to_close_to_balance(1.305, correctYaxis*1.442, correctYaxis*(30+120));
myConnect.do_next_action();
// delay(2000);
myRobot.go_to_close_to_balance(1.205, correctYaxis*1.342, correctYaxis*(30+120));
myRobot.go_to_close_to_balance(1.205, correctYaxis*1.342, correctYaxis*40);
myRobot.go_to_close_to_balance(1.300, correctYaxis*1.442, correctYaxis*40);
myConnect.do_next_action();
// delay(2000);
myRobot.go_to_close_to_balance(1.255, correctYaxis*1.352, correctYaxis*40);
myRobot.desactivate_ultrasounds(true);
myRobot.go_to_close_to_balance(1.255, correctYaxis*1.352, correctYaxis*0);
myRobot.go_to_close_to_balance(1.290, correctYaxis*1.452, correctYaxis*0);
myConnect.do_next_action();
myRobot.desactivate_ultrasounds(false);
// delay(2000);
Serial.println("Finished balance 1");
myRobot.go_to_close_to_balance(1.205, correctYaxis*1.36, correctYaxis*(-4));
myRobot.go_to_close_to_balance(0.866, correctYaxis*1.36, correctYaxis*(-4));
myRobot.go_to(0.866, correctYaxis*1.394, correctYaxis*(-2));
Serial.println("Taking two lasts palets");
myConnect.do_next_action();
myRobot.desactivate_ultrasounds(true);
myRobot.go_to_close_to_balance(1.205, correctYaxis*1.347, correctYaxis*(-4));
myRobot.go_to_close_to_balance(1.305, correctYaxis*1.397, correctYaxis*(-4));
myRobot.go_to_close_to_balance(1.305, correctYaxis*1.430, correctYaxis*(-4));
myRobot.go_to_close_to_balance(1.315, correctYaxis*1.490, correctYaxis*(-4));
myConnect.do_next_action();

/* ### THIS WAS THE FIRST VERSION WITH TWO INDIVIDUAL VENTOSES ### */
// Serial.println("Backup from balance");
// myRobot.go_to(1.295, correctYaxis*1.342, correctYaxis*30);
// Serial.println("We go to the red closer to balance");
// myRobot.go_to(0.906, correctYaxis*1.342, correctYaxis*30);
// myRobot.go_to(0.906, correctYaxis*1.362, correctYaxis*30);
//
// Serial.println("Starting picking first red palet : ");
// myConnect.do_next_action();
// // delay(2000);
// Serial.println("Finished picking up red palet");
// myRobot.go_to(0.9055, correctYaxis*1.3, correctYaxis*30);
// myRobot.go_to(0.9055, correctYaxis*1.342, correctYaxis*(30+120));
//
// myRobot.go_to(1.002, correctYaxis*1.342, correctYaxis*(30+120));
//
// Serial.println("Starting picking second green palet : ");
// myConnect.do_next_action();
// // delay(2000);
// Serial.println("Finished picking up second green palet");
//
// Serial.println("Go back to the balance");
// myRobot.go_to_close_to_balance(1.310, correctYaxis*1.342, correctYaxis*(30+120));
//
// myRobot.go_to_close_to_balance(1.310, correctYaxis*1.400, correctYaxis*(30+120));

```

```

//
// Serial.println("Deposer second green palet in the balance");
// myConnect.do_next_action();
// Serial.println("Deposer first red palet in the balance");
// myRobot.go_to_close_to_balance(1.310, correctYaxis*1.400, correctYaxis*30);
// myConnect.do_next_action();

Serial.println("Backup from balance");
myRobot.go_to(1.310, correctYaxis*1.342, correctYaxis*30);

myRobot.go_to(0.700, correctYaxis*1.342, correctYaxis*30);
Serial.println("Starting picking first red palet : ");
myConnect.do_next_action();
// delay(2000);
Serial.println("Finished picking up red palet");

myRobot.go_to(0.700, correctYaxis*1.342, correctYaxis*(30+120));

myRobot.go_to(0.495, correctYaxis*1.342, correctYaxis*(30+120));
Serial.println("Starting picking second red palet : ");
myConnect.do_next_action();
// delay(2000);
Serial.println("Finished picking up second red palet ");

myRobot.go_to(0.24, correctYaxis*1.342, correctYaxis*(30+120));
myRobot.go_to(0.24, correctYaxis*0.4667, correctYaxis*(30+120));
myConnect.do_next_action();
Serial.println("Deposed second red palet ");
// delay(2000);

// myRobot.go_to(0.3, 0.5, 30+120+120);
myConnect.do_next_action();
Serial.println("Deposed first red palet ");

Serial.println("\nCOMPETITION FINISHED\n");

// myRobot.go_to(0.11867, correctYaxis*0.4667, correctYaxis*0);
// delay(500);

// myRobot.go_to(0, 0, 0);
delay(5000);
}

return true;
}

```

1.6. Clase read-encoders-test

1.6.1. read-encoders-test.h

```
#ifndef test_encoders_included
```

```

#define test_encoders_included
#include <Encoder.h>

void init_read_encoders_test();
void action_read_encoders_test();

#endif

```

1.6.2. read-encoders-test.cpp

```

#include <Encoder.h>
#include "Motor.h"

// Motor motorTest(34,36,8,3,2); // Motor C

long t2;
long dt2;
long tprim2;

void init_read_encoders_test(){
    // motorTest.initPins();
    // motorTest.v_consigne = 0.0;

    // regle le pwm de la mega
    int myEraser = 7; // this is 111 in binary and is used as an eraser
    TCCR4B &= ~myEraser; // this operation (AND plus NOT), set the three bits in TCCR4B to 0
    int myPrescaler = 1; // this could be a number in [1 , 6]. In this case, 3
    // corresponds in binary to 011.
    TCCR4B |= myPrescaler; //this change pwm for pins 6,7,8

    t2 = micros();
    Serial.begin(115200);
}

void action_read_encoders_test(){
    tprim2 = micros();
    dt2 = tprim2 - t2;

    if(dt2 > 5){
        // Serial.println(motorTest.encoder.read());
    }
}

```

1.7. Clase test-motor

1.7.1. test-motor.h

```

#ifndef test_motor_included
#define test_motor_included
#include <Encoder.h>

void init_test_motor();
void action_test_motor();

```

```
#endif
```

1.7.2. test-motor.cpp

```
// #include <Encoder.h>
// #include "Motor.h"
//
// Motor motor2(6,46,48,19,17); // Motor A
// Motor motor1(7,42,44,18,16); // Motor B
// Motor motor3(8,36,34,2,4); // Motor C
//
// long t;
//
// /*
//  Vitesse consigne pour le test d'un seul moteur:
//  */
// float v_consigne_test = 0.2; // en m/s
// long tprim;
// long dt;
//
// // Command to use for the second part of the code
// int cmd = 70;
//
// void init_test_motor(){
//
//  motor1.initPins();
//  motor1.v_consigne = 0.0;
//
//  motor2.initPins();
//  motor2.v_consigne = 0.0;
//
//  motor3.initPins();
//  motor3.v_consigne = 0.0;
//
//  t = micros();
//
//
//
// }
//
// void action_test_motor(){
//
//  tprim = micros();
//  dt = tprim - t;
//
//  /*
//  Use this part of the code in order to command any of the three motors via an controlled
//  speed.
//  */
//
//
//  // if(dt > 5000){
//  //  t = tprim;
//  //  if(tprim > 3000000){
//  //    motor3.test_motor(v_consigne_test, dt);
//  //    motor2.test_motor(0, dt);
//  //    motor1.test_motor(0, dt);
```

```

// // } else {
// //     motor3.test_motor(0, dt);
// //     motor2.test_motor(0, dt);
// //     motor1.test_motor(0, dt);
// // }
// //
// // // Serial.print(tprim);
// // // // Serial.print(";");
// // // Serial.print(" Command moteur: ");
// // // // Serial.print(motor1.commande);
// // // // Serial.print(";");
// // // // Serial.print(motor2.commande);
// // // // Serial.print(";");
// // // Serial.print(motor3.commande);
// // // Serial.print(";");
// // //
// // // Serial.print(" Encoder: ");
// // // Serial.print(motor3.encoder.read());
// // // // Serial.print("\t");
// // // // Serial.print("Consigne moteur: ");
// // //
// // // // Serial.print(motor3.v_consigne);
// // // // Serial.print(";");
// // //
// // // Serial.print(" Vitesse du moteur: ");
// // // // Serial.print(";");
// // // // Serial.print(motor1.v);
// // // // Serial.print(";");
// // // // Serial.print(motor2.v);
// // // // Serial.print(";");
// // // Serial.print(motor3.v);
// // // Serial.print(";");
// // //
// // // Serial.println();
// // // }
// //
// //
// //
// //
// // /*
// // Use this part of the code in order to command any of the three motors via fixed command.
// // */
// //
// //
// // // if(tprim > 3000000 && tprim < 10000000 ){
// //     motor3.command_motor(cmd);
// //     motor2.command_motor(cmd);
// //     motor1.command_motor(cmd);
// // // } else if (tprim > 15000000 && tprim < 20000000) {
// //     // motor3.command_motor(cmd);
// //     // motor2.command_motor(cmd);
// //     // motor1.command_motor(cmd);
// // // }
// // // else {
// // //     motor3.command_motor(0);
// // //     motor2.command_motor(0);

```

```

// // motor1.command_motor(0);
// // }
//
// Serial.print(tprim);
// Serial.print(";");
// // Serial.print("\t");
// // Serial.print("Command moteur: ");
// // Serial.print(motor1.commande);
// // Serial.print(";");
// // Serial.print(motor2.commande);
// // Serial.print(";");
// // Serial.print(motor3.commande);
// // Serial.print(";");
//
// Serial.print("\t");
// Serial.print("Encoders: ");
// Serial.print(motor1.encoder.read());
// Serial.print(";");
// Serial.print(motor2.encoder.read());
// Serial.print(";");
// Serial.print(motor3.encoder.read());
// // Serial.print("\t");
// // Serial.print("Consigne moteur: ");
//
// // Serial.print(motor3.v_consigne);
// // Serial.print(";");
//
// // Serial.print("Vitesse du moteur: ");
// // Serial.print(";");
// // Serial.print(motor1.v);
// // Serial.print(";");
// // Serial.print(motor2.v);
// // Serial.print(";");
// // Serial.print(motor3.v);
// // Serial.print(";");
//
// Serial.println("");
//
//
//
// }

```


Chapter 2

Scripts MatLab para la modelización del robot

2.1. Script asservissement_moteur del regulador PI

```
%% Commande du moteur a courant continu Faulhaber 2642W-012CXR
clear all
close all
clc

% parametres du systeme
R = 1.46; % Resistance de l'induit
L = 0.135e-3; % Inductance
Kc = 18.57e-3; % Constante de couple
Ke = Kc; % Constante de fcem
taum = 0.08; % en sec, Cf excel reponse en vitesse a un echelon en cmd
Je = Ke*Kc*taum/R;

eta = 1/23; % Rapport de reduction
r = 0.06/2; % Rayon de la roue (en m)

%% Regulation de vitesse d'un seul moteur

Kv = 1/Ke;
taue = L/R;
taum = Je*R/Ke/Kc;

% reponse frequentielle de la boucle ouverte de vitesse
p = tf('p');
% Fonction de transfer du Mcc :
Mcc = Kv/( (taue*p+1)*(taum*p+1) );
% Commande moteur :
Gcmd = 12/255;
% Lecture encodeurs :
Genc = 4*100/2/pi;

%% Modele de discretisation
Tm = 5e-3; % temps d'echantillonnage en se-
```

```

Hd = 1/(1+Tm/2*p) % retard a cause de la discretisation du signal

Hv_bo = Gcmd*Mcc*eta*r*Hd
figure(1); bode(Hv_bo); hold on; grid on; title('Driagramme de Bode de H_{v\_bo}');
% figure(3); hold on; clf; nyquist(Hv_bo); grid on; title('Driagramme de Nyquist');

% [num,den] = tfdata(Hv_bo, 'v');
% bode_asymptotic(num, den) % diagramme de Bode asymptotique

%% Correction PI

PI = 1;

Hv_boc = PI*Hv_bo;

Hv_bf = feedback(Hv_boc,1);
figure(2); step(Hv_bf); grid on; title('Reponse indicielle du systeme'); xlabel('Temps');
ylabel('Vitesse (m/s)');

% Cahier des charges
wc = 70; % passage de la phase de la BO a -90 degrees
trep = 3/wc;

% Parametres du correcteur PI
Ti = 25e-3;
Kp = 1492;
PI = Kp*(1+1/Ti/p)
Hv_boc = Hv_bo*PI

figure(1); bode(Hv_boc); hold on; grid on; legend('Hbo', 'Hboc');

Hv_bf = feedback(Hv_boc,1);
figure(2); step(Hv_bf); grid on; title('Reponse indicielle du systeme corrige');
xlabel('Temps'); ylabel('Vitesse (m/s)');

```

2.2. Script matrices_Kiwi_drive de cambio de base entre la cinemática global del robot y las individuales de las ruedas

```

-%% Conversor del Kiwi Drive

R = 0.03; % radius of the robot, in m
theta = 0; % offset angle, in rad

v_xyw = [2 2 0]';

% from v_xyw to v_123
A = [ -sin(theta), cos(theta), R;
      -sin(theta+2*pi/3), cos(theta+2*pi/3), R;
      -sin(theta+4*pi/3), cos(theta+4*pi/3), R]

```

```
v_123 = A*v_xyw

%from v_123 to v_xyw
Ainv = 1/3*[ -2*sin(theta), -2*sin(-theta + pi/3), 2*sin(theta+pi/3);
            2*cos(theta), -2*cos(-theta+pi/3), -2*cos(theta+pi/3);
            1/R, 1/R, 1/R]
```


PART III

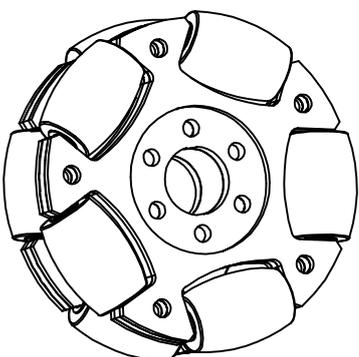
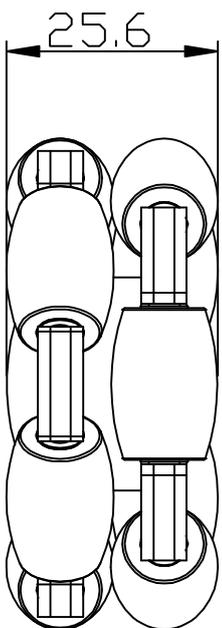
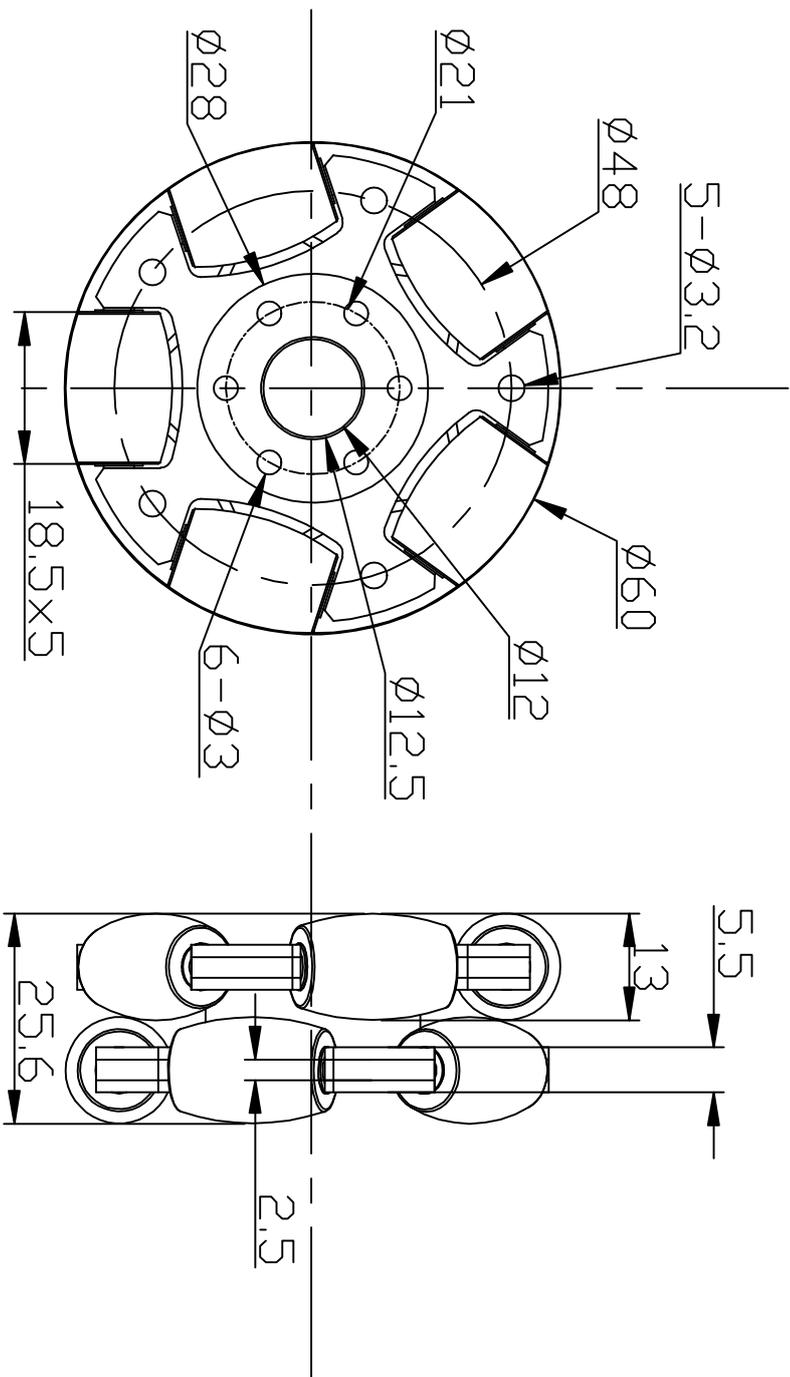


DATA SHEETS



Ruedas omnidireccionales 60mm





TITLE

60mm Aluminum Double OmniWheel

DIMENSIONS ARE IN MM

WWW.NEXUSROBOT.COM

Motor DC 2642W 012CXR



DC-Micromotors

Graphite Commutation

26 mNm
21 W

Series 2642 ... CXR

Values at 22°C and nominal voltage	2642 W	012 CXR	015 CXR	018 CXR	024 CXR	036 CXR	048 CXR		
1 Nominal voltage	U_N	12	15	18	24	36	48	V	
2 Terminal resistance	R	1,46	2,17	3,29	5,84	13,78	24,06	Ω	
3 Efficiency, max.	η_{max}	76	81	80	78	80	79	%	
4 No-load speed	n_0	5 800	5 600	5 800	5 900	5 800	5 900	min^{-1}	
5 No-load current, typ. (with shaft \varnothing 4 mm)	I_0	0,092	0,07	0,06	0,045	0,03	0,022	A	
6 Stall torque	M_H	144,6	165,3	153,2	150,5	148	149	mNm	
7 Friction torque	M_R	1,7	1,7	1,7	1,7	1,7	1,7	mNm	
8 Speed constant	k_n	514	395	337	252	167	125	min^{-1}/V	
9 Back-EMF constant	k_E	1,945	2,53	2,965	3,962	6,001	7,994	$\text{mV}/\text{min}^{-1}$	
10 Torque constant	k_M	18,57	24,16	28,31	37,83	57,31	76,34	mNm/A	
11 Current constant	k_I	0,054	0,041	0,035	0,026	0,017	0,013	A/mNm	
12 Slope of n-M curve	$\Delta n/\Delta M$	40,4	35,5	39,2	39	40,1	39,4	$\text{min}^{-1}/\text{mNm}$	
13 Rotor inductance	L	135	232	313	560	1 283	2 280	μH	
14 Mechanical time constant	τ_m	5,1	4,5	4,9	4,9	5	5	ms	
15 Rotor inertia	J	12	12	12	12	12	12	gcm^2	
16 Angular acceleration	α_{max}	121	138	128	125	123	124	$\cdot 10^3 \text{rad}/\text{s}^2$	
17 Thermal resistance	R_{th1} / R_{th2}	4,7 / 15,2						K/W	
18 Thermal time constant	τ_{w1} / τ_{w2}	20 / 720						s	
19 Operating temperature range:									
- motor		-30 ... +100						$^{\circ}\text{C}$	
- winding, max. permissible		+125						$^{\circ}\text{C}$	
20 Shaft bearings		sintered bearings			ball bearings, preloaded (optional version)				
21 Shaft load max.:		(standard)			(optional version)				
- with shaft diameter		4			4				mm
- radial at 3 000 min^{-1} (3 mm from bearing)		10			20				N
- axial at 3 000 min^{-1}		2			2				N
- axial at standstill		50			20				N
22 Shaft play:									
- radial	\leq	0,03			0,015				mm
- axial	\leq	0,15			0				mm
23 Housing material		steel, zinc galvanized and passivated							
24 Mass		114						g	
25 Direction of rotation		clockwise, viewed from the front face							
26 Speed up to	n_{max}	7 000						min^{-1}	
27 Number of pole pairs		1							
28 Magnet material		NdFeB							
Rated values for continuous operation									
29 Rated torque	M_N	25	26	26	26	26	26	mNm	
30 Rated current (thermal limit)	I_N	1,6	1,32	1,08	0,82	0,54	0,41	A	
31 Rated speed	n_N	4 770	4 660	4 750	4 770	4 710	4 750	min^{-1}	

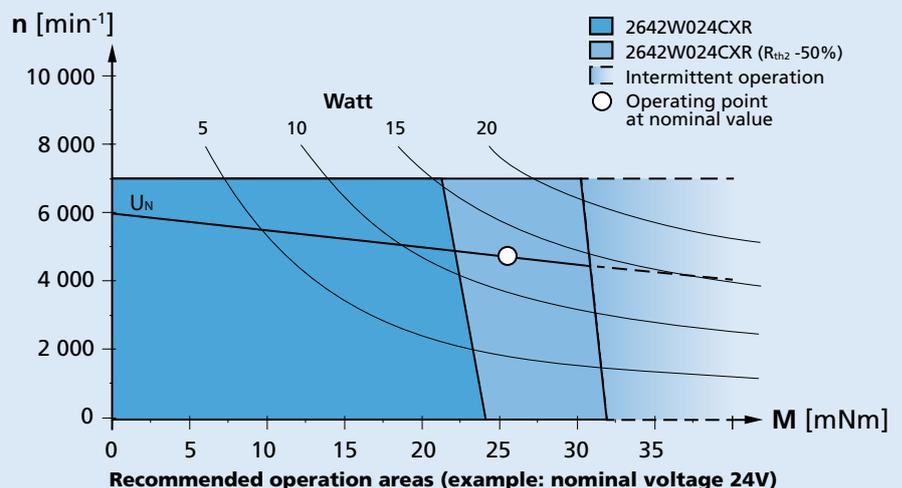
Note: Rated values are calculated with nominal voltage and at a 22°C ambient temperature. The R_{th2} value has been reduced by 25%.

Note:

The diagram indicates the recommended speed in relation to the available torque at the output shaft for a given ambient temperature of 22°C.

The diagram shows the motor in a completely insulated as well as thermally coupled condition (R_{th2} 50% reduced).

The nominal voltage (U_N) curve shows the operating point at nominal voltage in the insulated and thermally coupled condition. Any points of operation above the curve at nominal voltage will require a higher operating voltage. Any points below the nominal voltage curve will require less voltage.



Reductor planetario serie 26/1 S



Réducteurs planétaires

3,5 Nm

Combinaisons avec
Micromoteurs C.C.
Moteurs C.C. sans balais
Moteurs pas à pas

Série 26/1 S

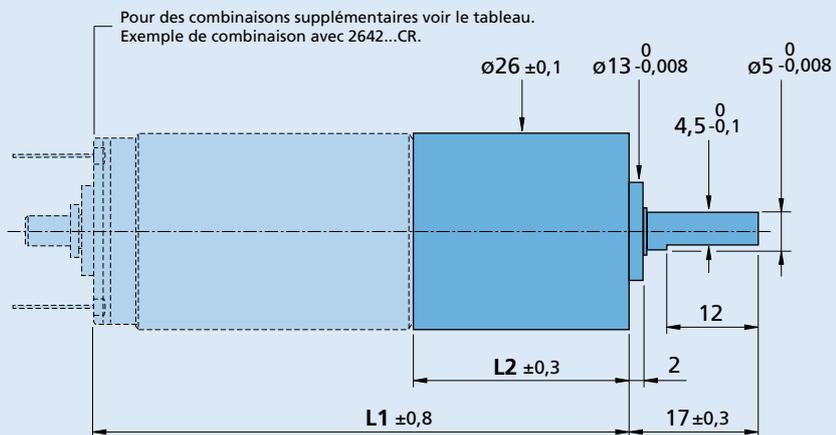
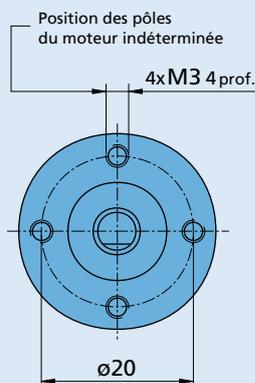
	26/1 S
Matériau du boîtier	acier
Matériau des engrenages	acier
Vitesse max. recommandée à l'entrée:	
– pour service permanent	4 000 min ⁻¹
Jeu angulaire typique, sans charge	≤ 1 °
Palier de l'arbre de sortie	roulements à billes précontraints
Charge de l'arbre max.:	
– radiale (à 10 mm de la face)	≤ 150 N
– axiale	≤ 100 N
Pression sur l'arbre max.	≤ 150 N
Jeu de l'arbre:	
– radial (à 10 mm de la face)	≤ 0,03 mm
– axial	≤ 0,1 mm
Température d'utilisation	- 30 ... + 100 °C

Spécifications

	2	3	4	5
Nombre des étages				
Couple permanent	Nm 3,5	3,5	3,5	3,5
Couple intermittent	Nm 4,5	4,5	4,5	4,5
Poids sans moteur, env.	g 116	139	162	185
Rendement, max.	% 80	70	60	55
Sens de rotation, entrée vers sortie	=	=	=	=
Rapport de réduction ¹⁾ (arrondi)	9,7:1 14:1 23:1	43:1 66:1 86:1	134:1 159:1 246:1	415:1 592:1 989:1 1 526:1
L2 [mm] = Longueur du réducteur	36,4	44,4	52,4	60,5
L1 [mm] = Long. avec moteur	78,4	86,4	94,4	102,5
2342S...CR	78,4	86,4	94,4	102,5
2642W...CR	78,4	86,4	94,4	102,5
2642W...CXR	78,4	86,4	94,4	102,5
2657W...CR	93,4	101,4	109,4	117,5
2657W...CXR	93,4	101,4	109,4	117,5
2668W...CR	104,4	112,4	120,4	128,5
2444S...B	80,4	88,4	96,4	104,5
AM2224-R3...-30	67,3	75,3	83,3	91,4

¹⁾ Les rapports de réduction indiqués sont des valeurs arrondies, les valeurs exactes sont disponibles sur demande ou sur www.faulhaber.com.

Note: Les réducteurs type S ont des engrenages en acier et lubrification renforcée qui permettent d'augmenter la durée de vie.



26/1 S

Encoder óptico HEDS 5540 C02



Encoders

optical Encoder, digital outputs,
3 channels, 100 - 500 lines per revolution

For combination with
DC-Micromotors
Brushless DC-Motors

Series HEDS 5540

		HEDS 5540 C	HEDS 5540 A	
Lines per revolution	N	100	500	
Frequency range, up to ¹⁾²⁾	f	100	100	kHz
Signal output, square wave		2+1 Index		Channels
Supply voltage	U_{DD}	4,5 ... 5,5		V
Current consumption, typical ³⁾	I_{DD}	57		mA
Pulse width	P	180 ± 35		°e
Phase shift, channel A to B	Φ	90 ± 15		°e
Logic state width	S	90 ± 35		°e
Cycle	C	360 ± 5,5		°e
Signal rise/fall time, max. ($C_{LOAD} = pF$)	tr/tf	0,25 / 0,25		µs
Inertia of code disc	J	0,6		gcm ²
Operating temperature range		-40 ... +100		°C

¹⁾ Velocity (min⁻¹) = f (Hz) x 60/ N

²⁾ HEDS 5540 requires pull-up resistors of 2,7 kΩ between pins 2, 3, 5 and 4 (U_{DD})

³⁾ $U_{DD} = 5$ V: with unloaded outputs

For combination with Motor

Dimensional drawing A	<L1 [mm]		
2230 ... S	52,8	3863 ... CR	86,1
2233 ... S	55,6	3890 ... CR	112,1
2342 ... CR	63,8	2036 ... B - K312	56,8
2642 ... CXR	64,8	2057 ... B - K312	75,8
2642 ... CR	64,8	2057 ... BHS - K312	75,8
2642 ... CR	64,8	2444 ... B - K312	64,9
2657 ... CXR	79,8	3056 ... B - K312	76,1
2657 ... CR	79,8	3274 ... BP4	94,0
2668 ... CR	90,8	3564 ... B - K312	84,1
3242 ... CR	65,3	4490 ... B - K312	116,3
3257 ... CR	80,3	4490 ... BS - K312	116,3
3272 ... CR	95,3		

Characteristics

These incremental shaft encoders in combination with the DC-Motors are designed for the indication and control of both shaft velocity and direction of rotation as well as for positioning.

Motors with ball bearings are recommended for continuous operation at low and high speeds and for elevated radial shaft load.

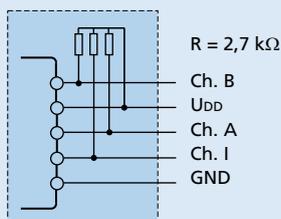
A LED source and lens system transmits collimated light through a low inertia metal disc to give two channels with 90° phase shift.

Details for the Motors and suitable reduction gearheads are on separate catalogue pages.

The single 5 volt supply and the two or three channel digital output signals are interfaced with a 5-pin connector.

Circuit diagram / Output signals

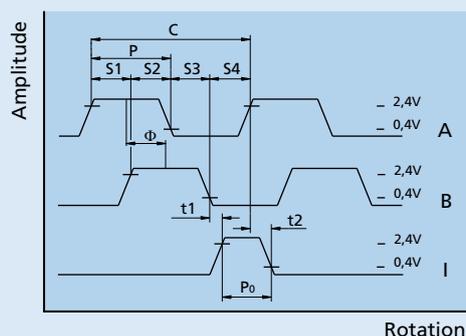
Output circuit



Note:
HEDS 5540 requires
pull-up resistors

Output signals

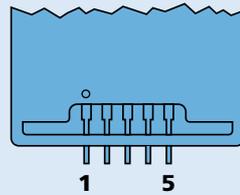
with clockwise rotation as seen
from the shaft end



Connector information / Variants

No.	Function
1	GND
2	Channel I
3	Channel A
4	U _{DD}
5	Channel B

Connection Encoder



Recommended connector

AMP 103686-4/640442-5,
Molex 2695/2759
FCI 65039-032/4825x-000

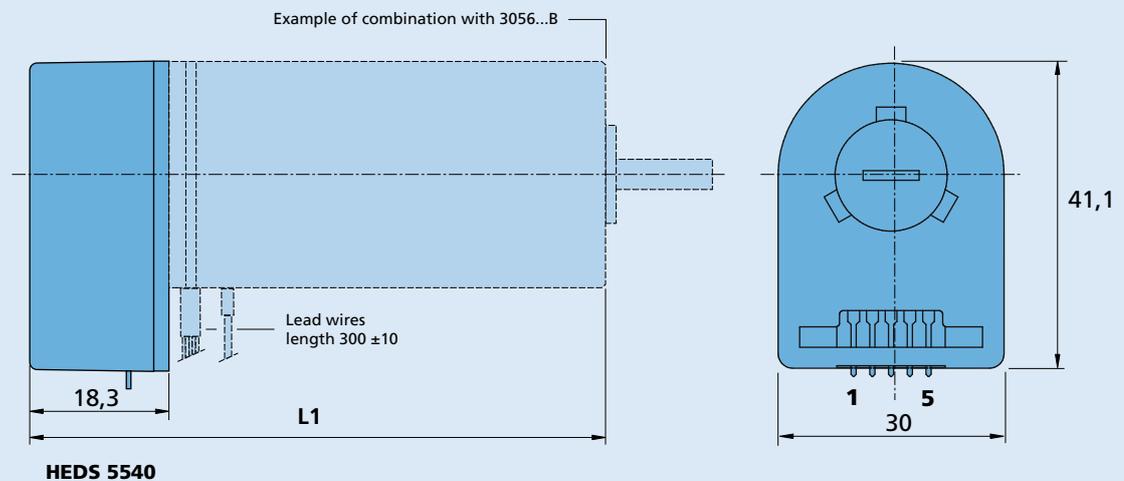
Option

- HEDS 5540 Interlocking connector, extension cables 300 mm length (Part No.: K799)

Full product description

- Example:
2444S024B K312 HEDS5540C
3863H048CR HEDS5540A

Dimensional drawing A



Controlador de motor L298N

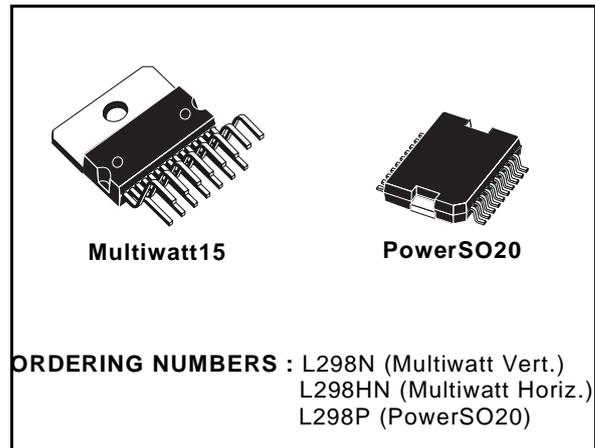


DUAL FULL-BRIDGE DRIVER

- OPERATING SUPPLY VOLTAGE UP TO 46 V
- TOTAL DC CURRENT UP TO 4 A
- LOW SATURATION VOLTAGE
- OVERTEMPERATURE PROTECTION
- LOGICAL "0" INPUT VOLTAGE UP TO 1.5 V (HIGH NOISE IMMUNITY)

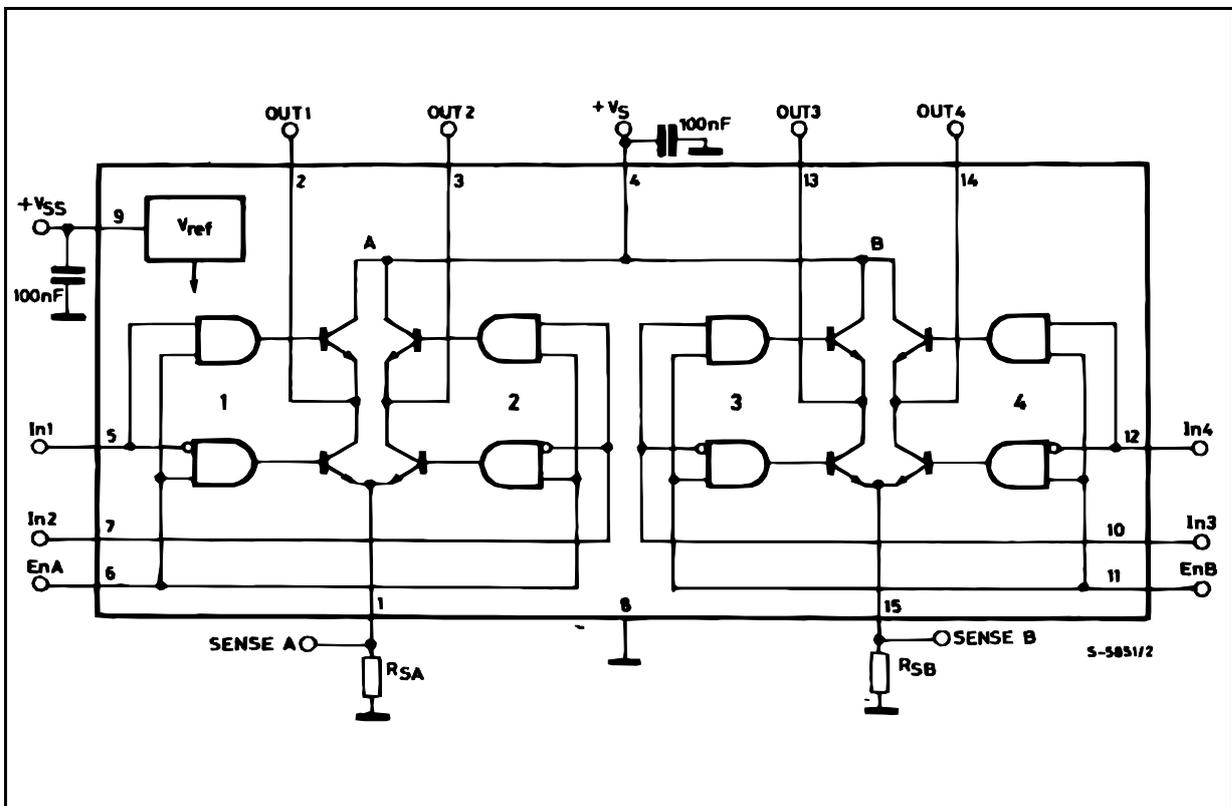
DESCRIPTION

The L298 is an integrated monolithic circuit in a 15-lead Multiwatt and PowerSO20 packages. It is a high voltage, high current dual full-bridge driver designed to accept standard TTL logic levels and drive inductive loads such as relays, solenoids, DC and stepping motors. Two enable inputs are provided to enable or disable the device independently of the input signals. The emitters of the lower transistors of each bridge are connected together and the corresponding external terminal can be used for the con-



nection of an external sensing resistor. An additional supply input is provided so that the logic works at a lower voltage.

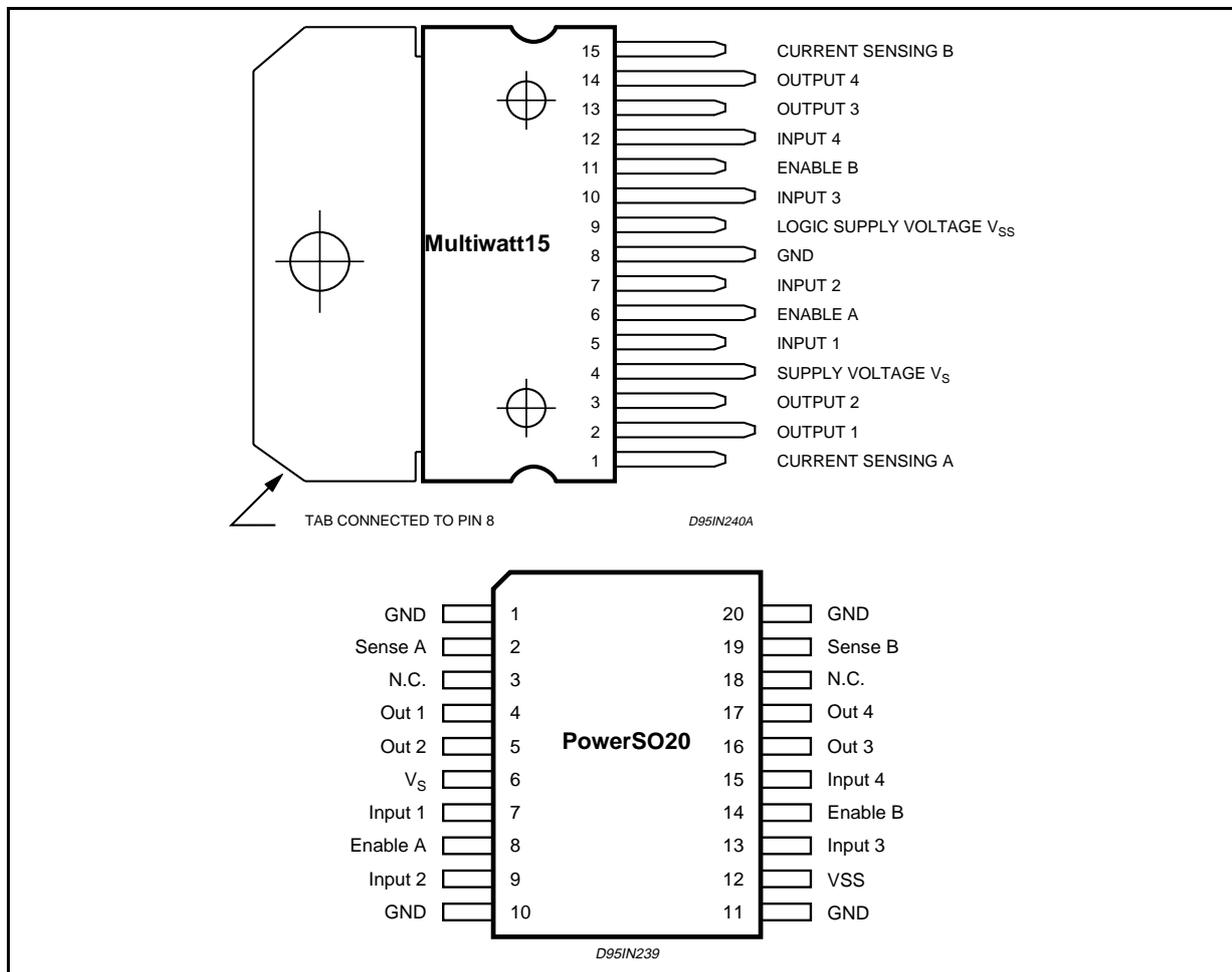
BLOCK DIAGRAM



ABSOLUTE MAXIMUM RATINGS

Symbol	Parameter	Value	Unit
V_S	Power Supply	50	V
V_{SS}	Logic Supply Voltage	7	V
V_i, V_{en}	Input and Enable Voltage	-0.3 to 7	V
I_O	Peak Output Current (each Channel)		
	- Non Repetitive (t = 100µs)	3	A
	- Repetitive (80% on -20% off; t _{on} = 10ms)	2.5	A
	-DC Operation	2	A
V_{sens}	Sensing Voltage	-1 to 2.3	V
P_{tot}	Total Power Dissipation (T _{case} = 75°C)	25	W
T_{op}	Junction Operating Temperature	-25 to 130	°C
T_{stg}, T_j	Storage and Junction Temperature	-40 to 150	°C

PIN CONNECTIONS (top view)



THERMAL DATA

Symbol	Parameter	PowerSO20	Multiwatt15	Unit
$R_{th\ j-case}$	Thermal Resistance Junction-case	Max. -	3	°C/W
$R_{th\ j-amb}$	Thermal Resistance Junction-ambient	Max. 13 (*)	35	°C/W

(*) Mounted on aluminum substrate



PIN FUNCTIONS (refer to the block diagram)

MW.15	PowerSO	Name	Function
1;15	2;19	Sense A; Sense B	Between this pin and ground is connected the sense resistor to control the current of the load.
2;3	4;5	Out 1; Out 2	Outputs of the Bridge A; the current that flows through the load connected between these two pins is monitored at pin 1.
4	6	V _s	Supply Voltage for the Power Output Stages. A non-inductive 100nF capacitor must be connected between this pin and ground.
5;7	7;9	Input 1; Input 2	TTL Compatible Inputs of the Bridge A.
6;11	8;14	Enable A; Enable B	TTL Compatible Enable Input: the L state disables the bridge A (enable A) and/or the bridge B (enable B).
8	1,10,11,20	GND	Ground.
9	12	V _{SS}	Supply Voltage for the Logic Blocks. A100nF capacitor must be connected between this pin and ground.
10; 12	13;15	Input 3; Input 4	TTL Compatible Inputs of the Bridge B.
13; 14	16;17	Out 3; Out 4	Outputs of the Bridge B. The current that flows through the load connected between these two pins is monitored at pin 15.
–	3;18	N.C.	Not Connected

ELECTRICAL CHARACTERISTICS (V_S = 42V; V_{SS} = 5V, T_j = 25°C; unless otherwise specified)

Symbol	Parameter	Test Conditions	Min.	Typ.	Max.	Unit
V _S	Supply Voltage (pin 4)	Operative Condition	V _{IH} +2.5		46	V
V _{SS}	Logic Supply Voltage (pin 9)		4.5	5	7	V
I _S	Quiescent Supply Current (pin 4)	V _{en} = H; I _L = 0 V _i = L V _i = H		13 50	22 70	mA mA
		V _{en} = L V _i = X			4	mA
I _{SS}	Quiescent Current from V _{SS} (pin 9)	V _{en} = H; I _L = 0 V _i = L V _i = H		24 7	36 12	mA mA
		V _{en} = L V _i = X			6	mA
V _{iL}	Input Low Voltage (pins 5, 7, 10, 12)		–0.3		1.5	V
V _{iH}	Input High Voltage (pins 5, 7, 10, 12)		2.3		V _{SS}	V
I _{iL}	Low Voltage Input Current (pins 5, 7, 10, 12)	V _i = L			–10	μA
I _{iH}	High Voltage Input Current (pins 5, 7, 10, 12)	V _i = H ≤ V _{SS} –0.6V		30	100	μA
V _{en} = L	Enable Low Voltage (pins 6, 11)		–0.3		1.5	V
V _{en} = H	Enable High Voltage (pins 6, 11)		2.3		V _{SS}	V
I _{en} = L	Low Voltage Enable Current (pins 6, 11)	V _{en} = L			–10	μA
I _{en} = H	High Voltage Enable Current (pins 6, 11)	V _{en} = H ≤ V _{SS} –0.6V		30	100	μA
V _{CEsat} (H)	Source Saturation Voltage	I _L = 1A I _L = 2A	0.95	1.35 2	1.7 2.7	V V
V _{CEsat} (L)	Sink Saturation Voltage	I _L = 1A (5) I _L = 2A (5)	0.85	1.2 1.7	1.6 2.3	V V
V _{CEsat}	Total Drop	I _L = 1A (5) I _L = 2A (5)	1.80		3.2 4.9	V V
V _{sens}	Sensing Voltage (pins 1, 15)		–1 (1)		2	V

ELECTRICAL CHARACTERISTICS (continued)

Symbol	Parameter	Test Conditions	Min.	Typ.	Max.	Unit
T ₁ (V _i)	Source Current Turn-off Delay	0.5 V _i to 0.9 I _L (2); (4)		1.5		μs
T ₂ (V _i)	Source Current Fall Time	0.9 I _L to 0.1 I _L (2); (4)		0.2		μs
T ₃ (V _i)	Source Current Turn-on Delay	0.5 V _i to 0.1 I _L (2); (4)		2		μs
T ₄ (V _i)	Source Current Rise Time	0.1 I _L to 0.9 I _L (2); (4)		0.7		μs
T ₅ (V _i)	Sink Current Turn-off Delay	0.5 V _i to 0.9 I _L (3); (4)		0.7		μs
T ₆ (V _i)	Sink Current Fall Time	0.9 I _L to 0.1 I _L (3); (4)		0.25		μs
T ₇ (V _i)	Sink Current Turn-on Delay	0.5 V _i to 0.9 I _L (3); (4)		1.6		μs
T ₈ (V _i)	Sink Current Rise Time	0.1 I _L to 0.9 I _L (3); (4)		0.2		μs
f _c (V _i)	Commutation Frequency	I _L = 2A		25	40	KHz
T ₁ (V _{en})	Source Current Turn-off Delay	0.5 V _{en} to 0.9 I _L (2); (4)		3		μs
T ₂ (V _{en})	Source Current Fall Time	0.9 I _L to 0.1 I _L (2); (4)		1		μs
T ₃ (V _{en})	Source Current Turn-on Delay	0.5 V _{en} to 0.1 I _L (2); (4)		0.3		μs
T ₄ (V _{en})	Source Current Rise Time	0.1 I _L to 0.9 I _L (2); (4)		0.4		μs
T ₅ (V _{en})	Sink Current Turn-off Delay	0.5 V _{en} to 0.9 I _L (3); (4)		2.2		μs
T ₆ (V _{en})	Sink Current Fall Time	0.9 I _L to 0.1 I _L (3); (4)		0.35		μs
T ₇ (V _{en})	Sink Current Turn-on Delay	0.5 V _{en} to 0.9 I _L (3); (4)		0.25		μs
T ₈ (V _{en})	Sink Current Rise Time	0.1 I _L to 0.9 I _L (3); (4)		0.1		μs

- 1) Sensing voltage can be -1 V for t ≤ 50 μsec; in steady state V_{sens} min ≥ -0.5 V.
- 2) See fig. 2.
- 3) See fig. 4.
- 4) The load must be a pure resistor.

Figure 1 : Typical Saturation Voltage vs. Output Current.

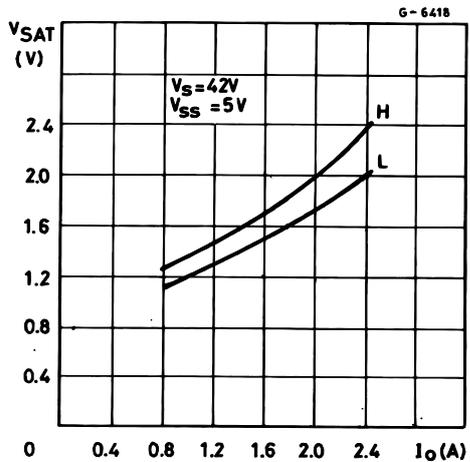
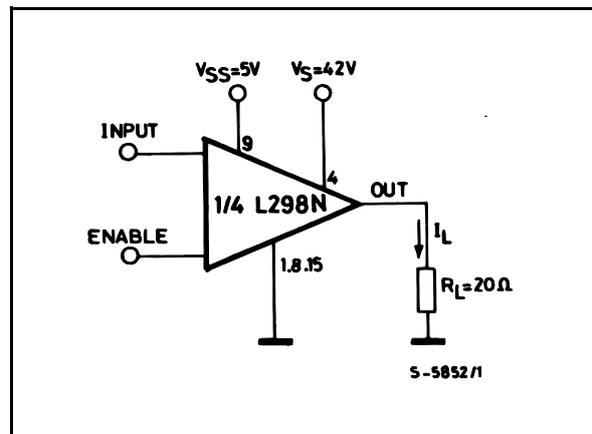


Figure 2 : Switching Times Test Circuits.



Note : For INPUT Switching, set EN = H
 For ENABLE Switching, set IN = H

Figure 3 : Source Current Delay Times vs. Input or Enable Switching.

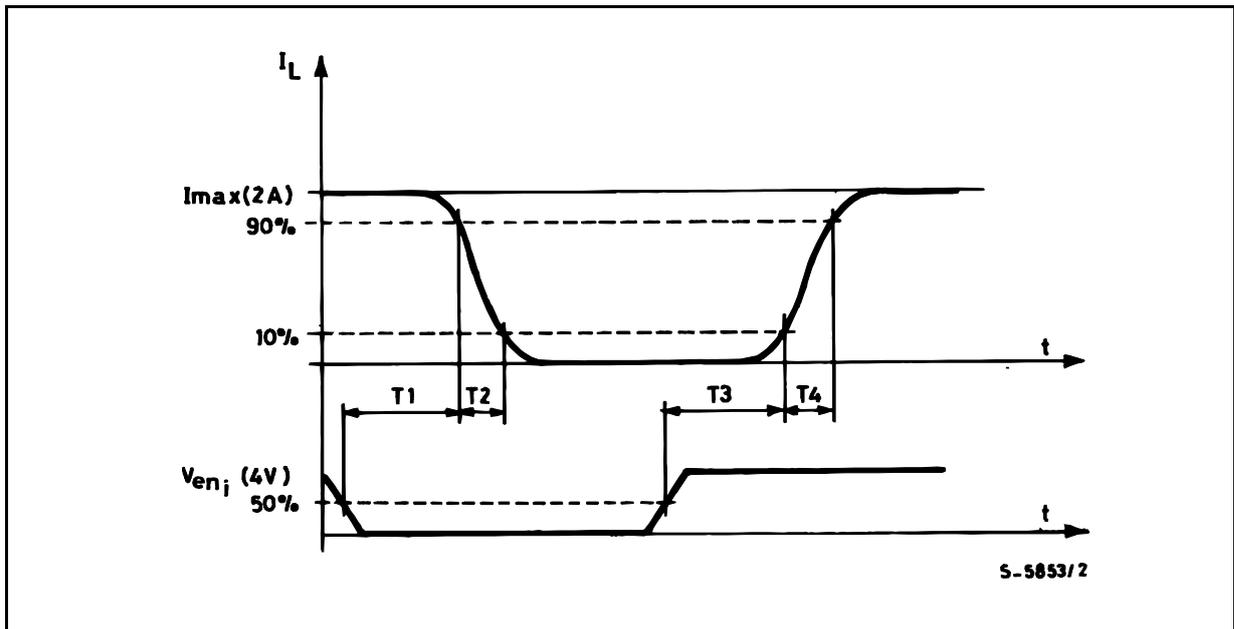
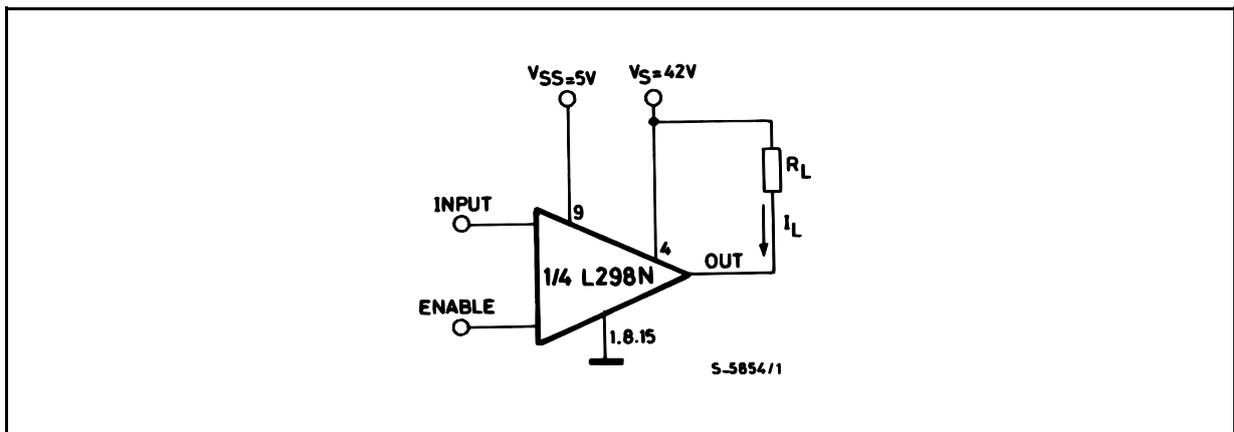


Figure 4 : Switching Times Test Circuits.



Note : For INPUT Switching, set EN = H
 For ENABLE Switching, set IN = L

Figure 5 : Sink Current Delay Times vs. Input 0 V Enable Switching.

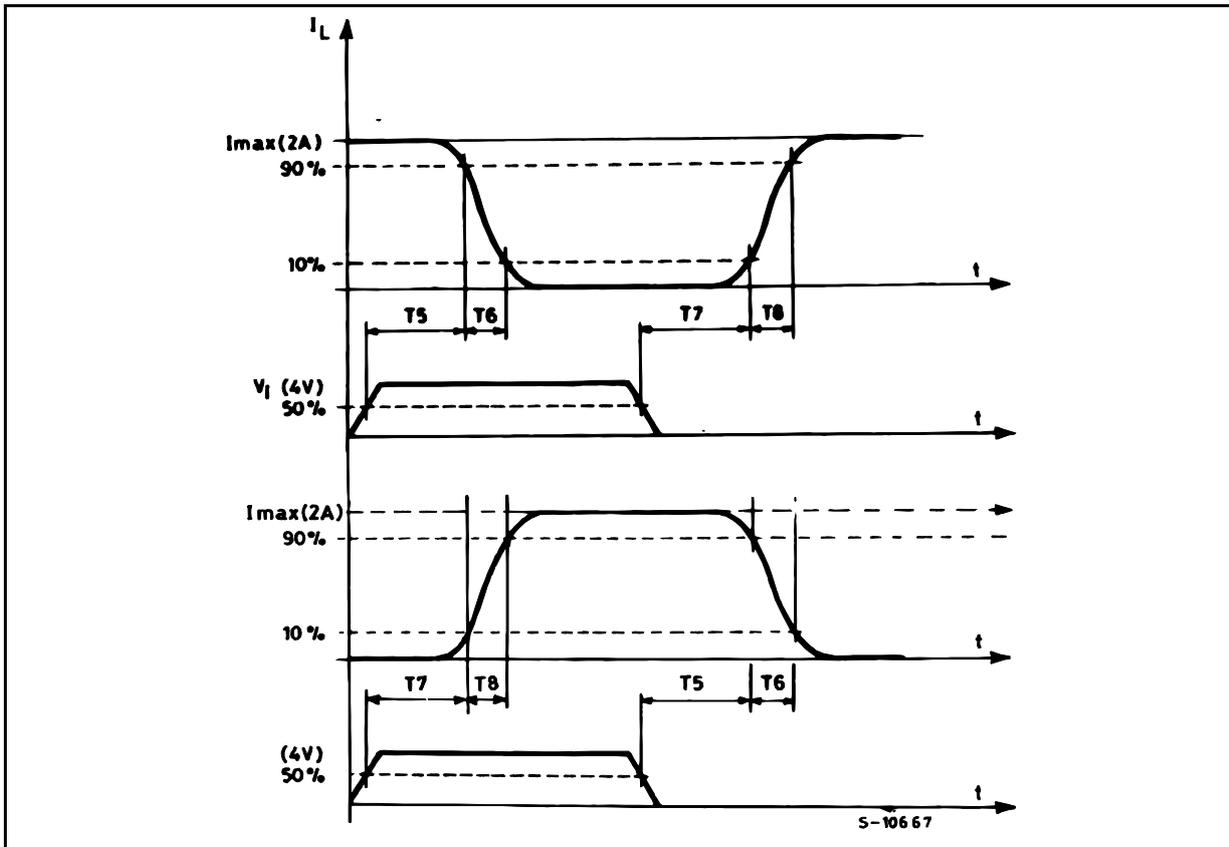


Figure 6 : Bidirectional DC Motor Control.

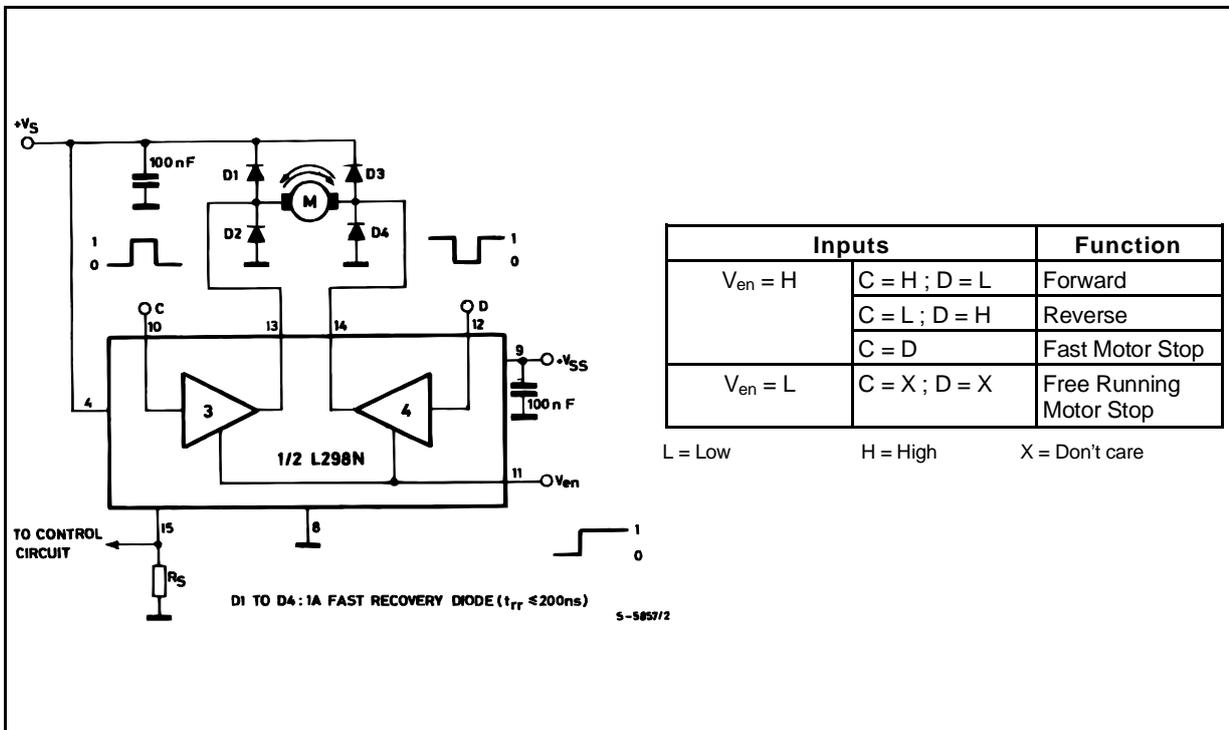
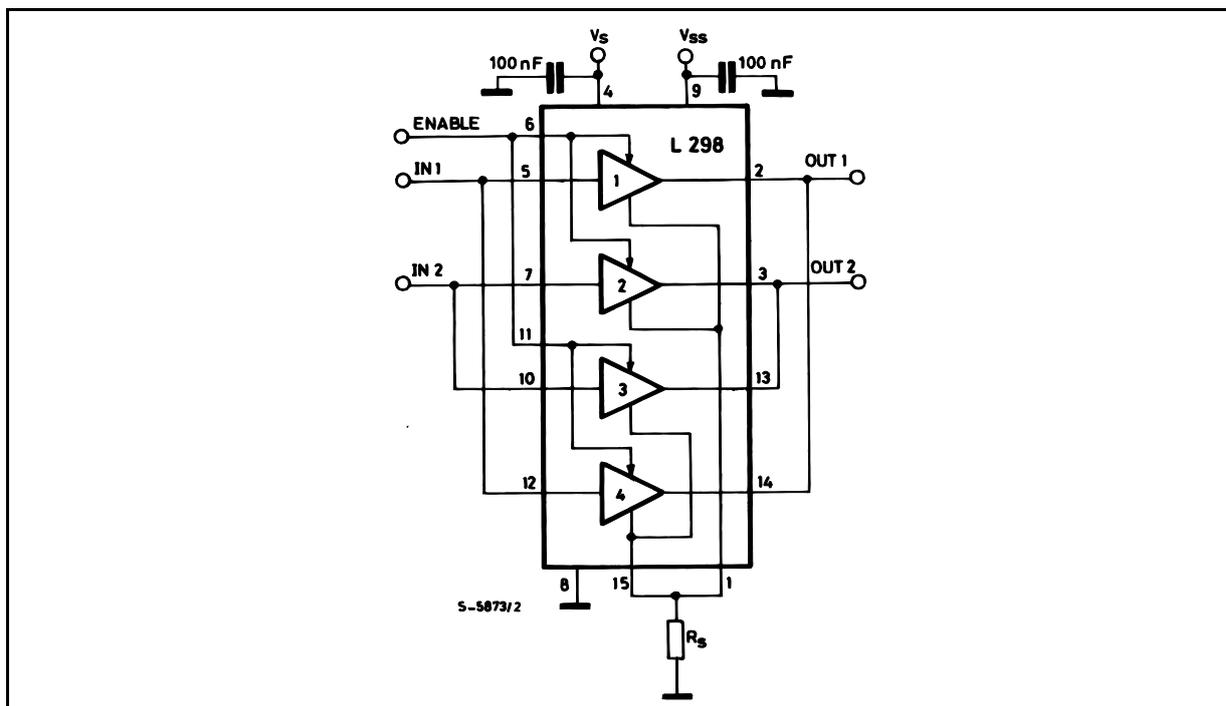


Figure 7 : For higher currents, outputs can be paralleled. Take care to parallel channel 1 with channel 4 and channel 2 with channel 3.



APPLICATION INFORMATION (Refer to the block diagram)

1.1. POWER OUTPUT STAGE

The L298 integrates two power output stages (A ; B). The power output stage is a bridge configuration and its outputs can drive an inductive load in common or differential mode, depending on the state of the inputs. The current that flows through the load comes out from the bridge at the sense output : an external resistor (R_{SA} ; R_{SB} .) allows to detect the intensity of this current.

1.2. INPUT STAGE

Each bridge is driven by means of four gates the input of which are In_1 ; In_2 ; EnA and In_3 ; In_4 ; EnB . The In inputs set the bridge state when The En input is high ; a low state of the En input inhibits the bridge. All the inputs are TTL compatible.

2. SUGGESTIONS

A non inductive capacitor, usually of 100 nF, must be foreseen between both V_s and V_{ss} , to ground, as near as possible to GND pin. When the large capacitor of the power supply is too far from the IC, a second smaller one must be foreseen near the L298.

The sense resistor, not of a wire wound type, must be grounded near the negative pole of V_s that must be near the GND pin of the I.C.

Each input must be connected to the source of the driving signals by means of a very short path.

Turn-On and Turn-Off : Before to Turn-ON the Supply Voltage and before to Turn it OFF, the Enable input must be driven to the Low state.

3. APPLICATIONS

Fig 6 shows a bidirectional DC motor control Schematic Diagram for which only one bridge is needed. The external bridge of diodes $D1$ to $D4$ is made by four fast recovery elements ($trr \leq 200$ nsec) that must be chosen of a VF as low as possible at the worst case of the load current.

The sense output voltage can be used to control the current amplitude by chopping the inputs, or to provide overcurrent protection by switching low the enable input.

The brake function (Fast motor stop) requires that the Absolute Maximum Rating of 2 Amps must never be overcome.

When the repetitive peak current needed from the load is higher than 2 Amps, a paralleled configuration can be chosen (See Fig.7).

An external bridge of diodes are required when inductive loads are driven and when the inputs of the IC are chopped ; Schottky diodes would be preferred.

This solution can drive until 3 Amps In DC operation and until 3.5 Amps of a repetitive peak current.

On Fig 8 it is shown the driving of a two phase bipolar stepper motor ; the needed signals to drive the inputs of the L298 are generated, in this example, from the IC L297.

Fig 9 shows an example of P.C.B. designed for the application of Fig 8.

Figure 8 : Two Phase Bipolar Stepper Motor Circuit.

This circuit drives bipolar stepper motors with winding currents up to 2 A. The diodes are fast 2 A types.

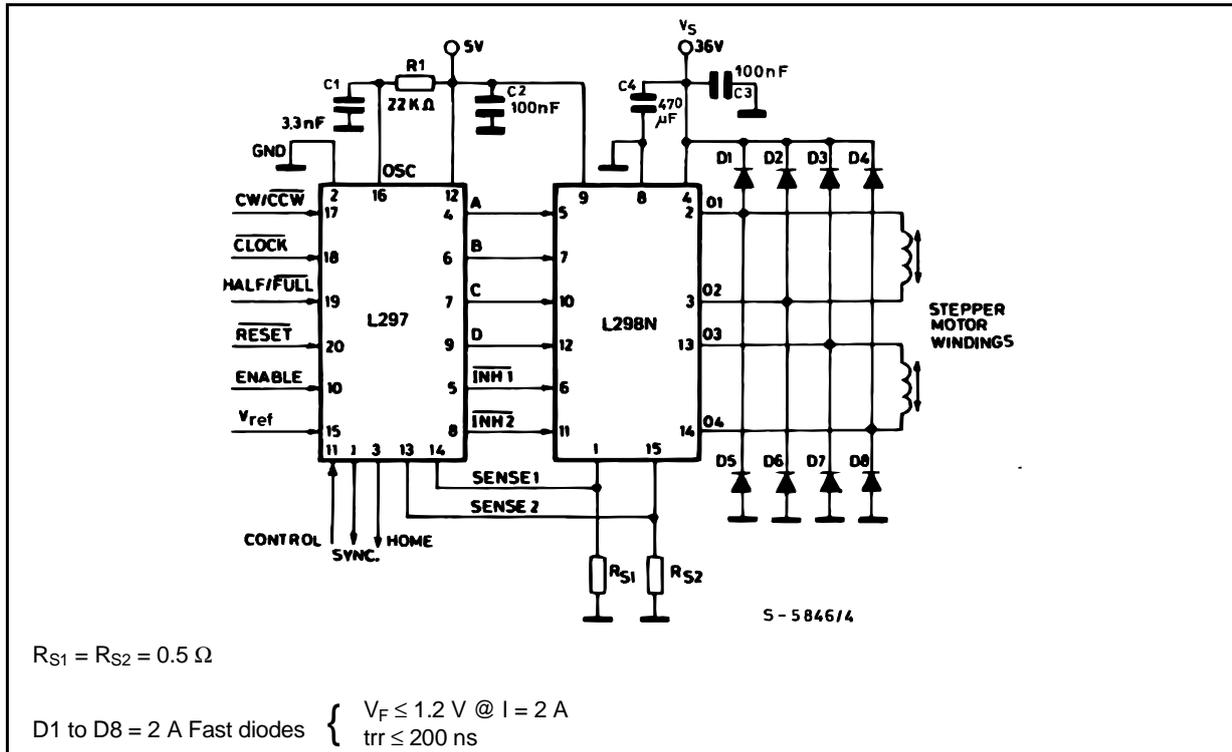


Fig 10 shows a second two phase bipolar stepper motor control circuit where the current is controlled by the I.C. L6506.

Figure 9 : Suggested Printed Circuit Board Layout for the Circuit of fig. 8 (1:1 scale).

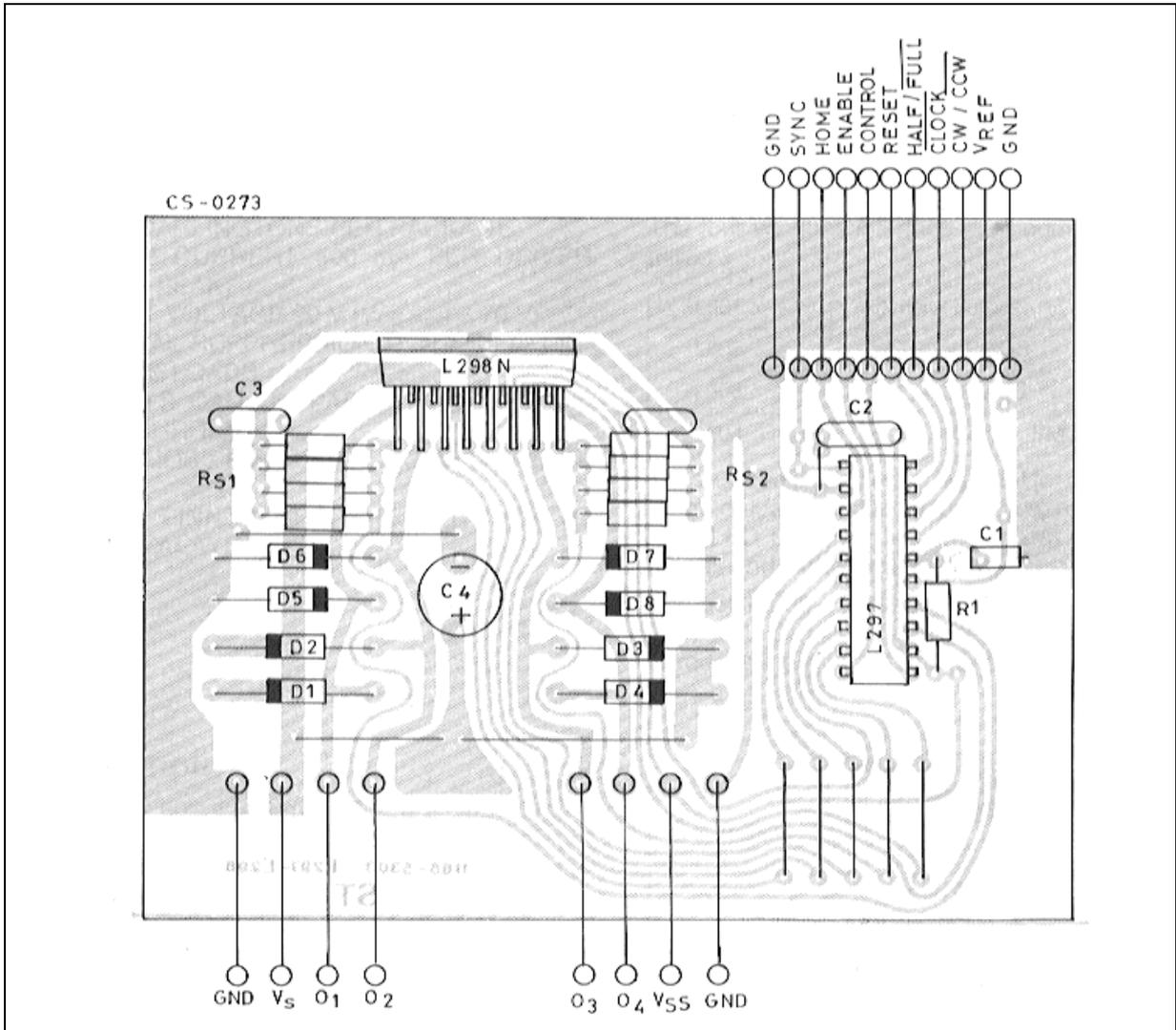
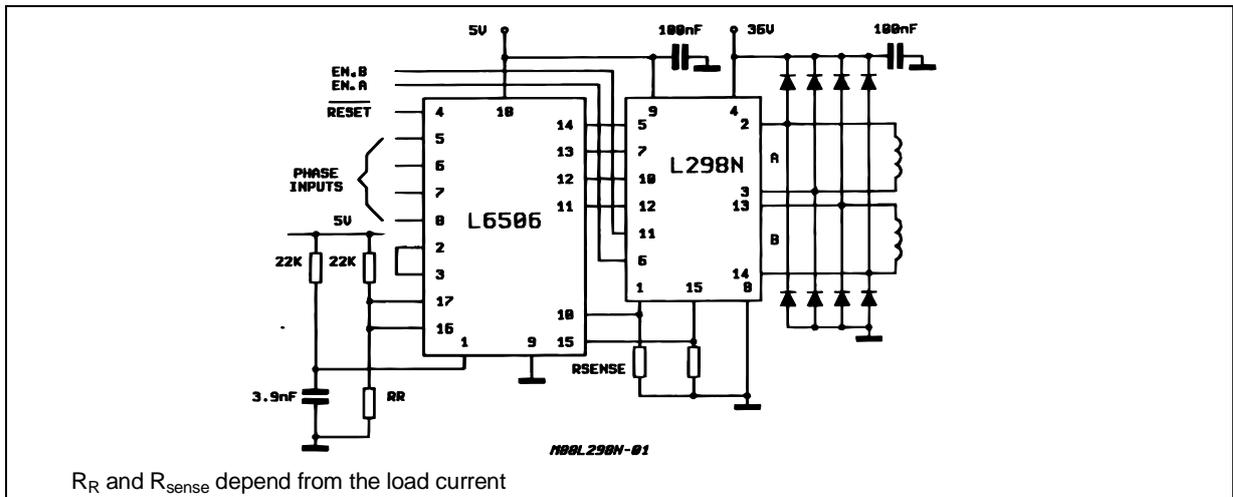


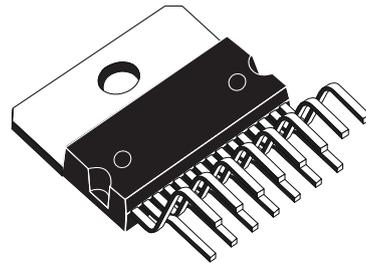
Figure 10 : Two Phase Bipolar Stepper Motor Control Circuit by Using the Current Controller L6506.



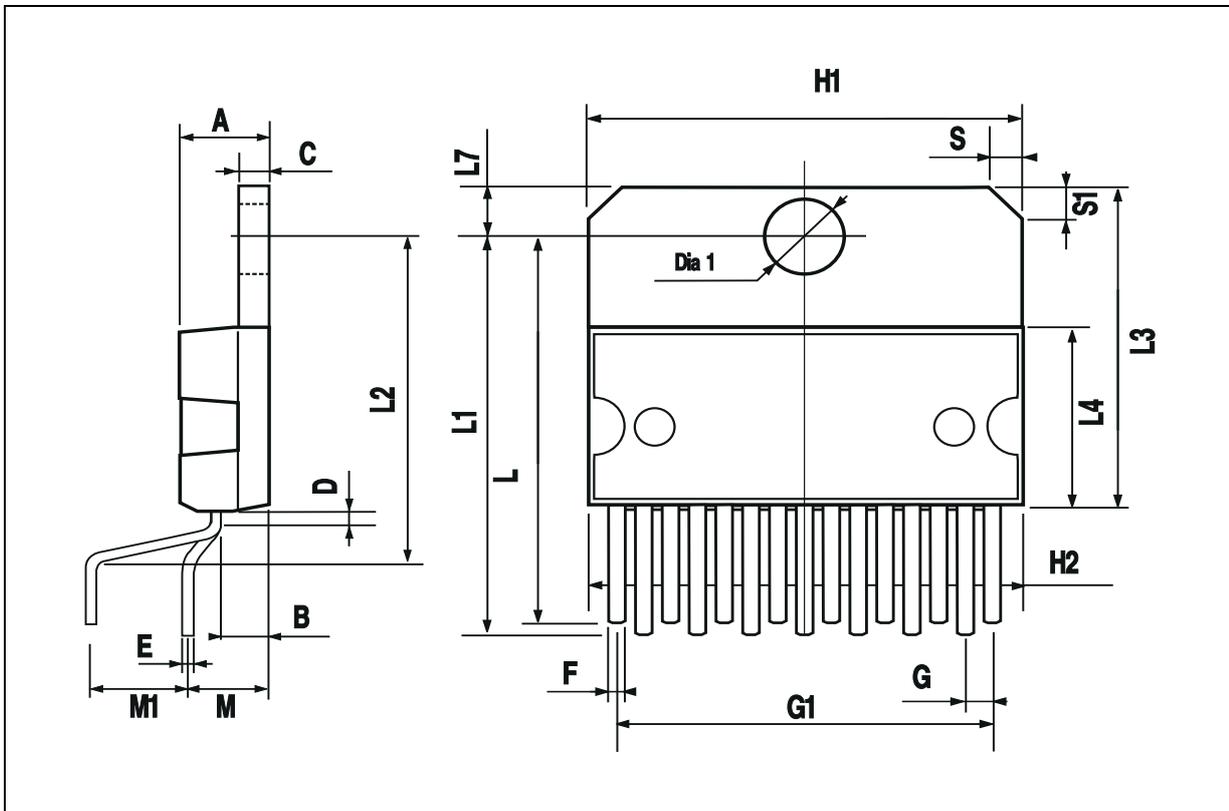
L298

DIM.	mm			inch		
	MIN.	TYP.	MAX.	MIN.	TYP.	MAX.
A			5			0.197
B			2.65			0.104
C			1.6			0.063
D		1			0.039	
E	0.49		0.55	0.019		0.022
F	0.66		0.75	0.026		0.030
G	1.02	1.27	1.52	0.040	0.050	0.060
G1	17.53	17.78	18.03	0.690	0.700	0.710
H1	19.6			0.772		
H2			20.2			0.795
L	21.9	22.2	22.5	0.862	0.874	0.886
L1	21.7	22.1	22.5	0.854	0.870	0.886
L2	17.65		18.1	0.695		0.713
L3	17.25	17.5	17.75	0.679	0.689	0.699
L4	10.3	10.7	10.9	0.406	0.421	0.429
L7	2.65		2.9	0.104		0.114
M	4.25	4.55	4.85	0.167	0.179	0.191
M1	4.63	5.08	5.53	0.182	0.200	0.218
S	1.9		2.6	0.075		0.102
S1	1.9		2.6	0.075		0.102
Dia1	3.65		3.85	0.144		0.152

OUTLINE AND MECHANICAL DATA

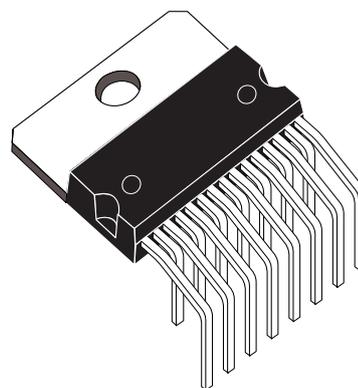


Multiwatt15 V

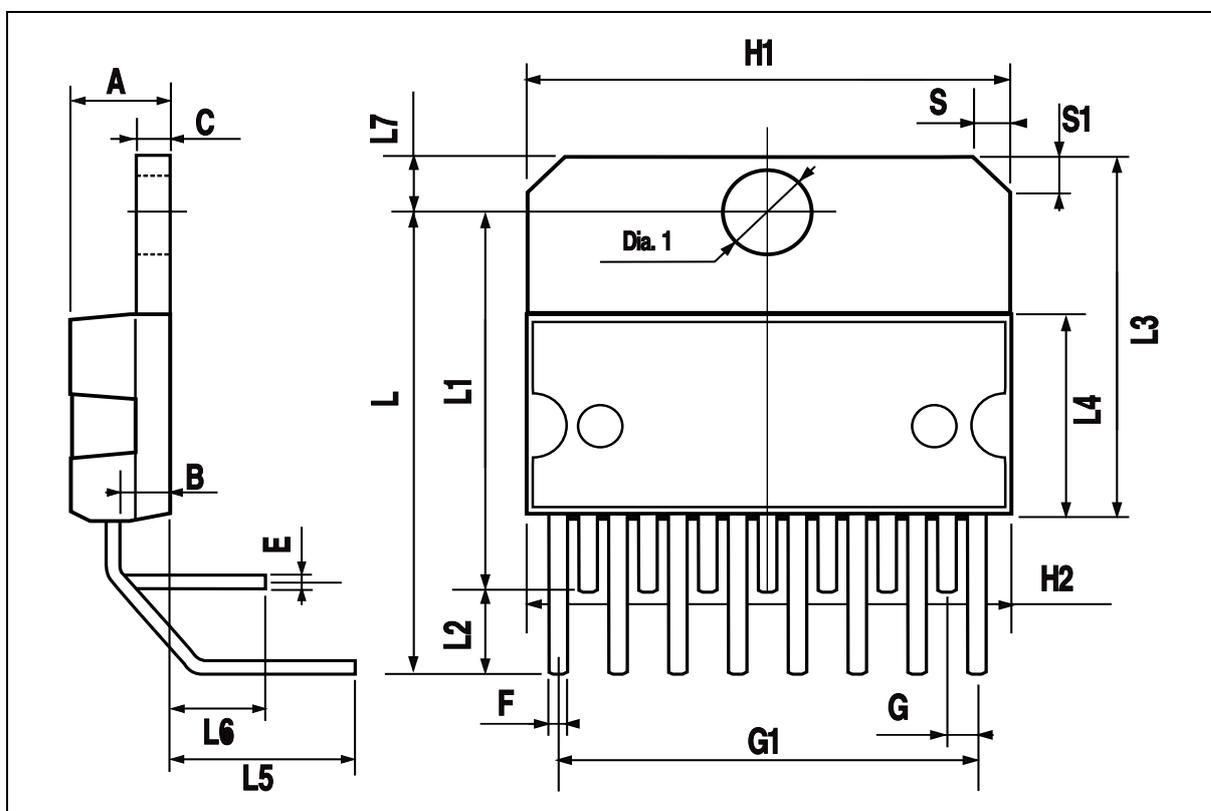


DIM.	mm			inch		
	MIN.	TYP.	MAX.	MIN.	TYP.	MAX.
A			5			0.197
B			2.65			0.104
C			1.6			0.063
E	0.49		0.55	0.019		0.022
F	0.66		0.75	0.026		0.030
G	1.14	1.27	1.4	0.045	0.050	0.055
G1	17.57	17.78	17.91	0.692	0.700	0.705
H1	19.6			0.772		
H2			20.2			0.795
L		20.57			0.810	
L1		18.03			0.710	
L2		2.54			0.100	
L3	17.25	17.5	17.75	0.679	0.689	0.699
L4	10.3	10.7	10.9	0.406	0.421	0.429
L5		5.28			0.208	
L6		2.38			0.094	
L7	2.65		2.9	0.104		0.114
S	1.9		2.6	0.075		0.102
S1	1.9		2.6	0.075		0.102
Dia1	3.65		3.85	0.144		0.152

OUTLINE AND MECHANICAL DATA



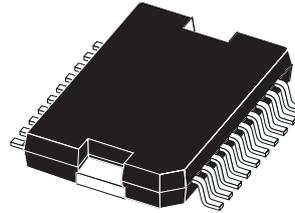
Multiwatt15 H



DIM.	mm			inch		
	MIN.	TYP.	MAX.	MIN.	TYP.	MAX.
A			3.6			0.142
a1	0.1		0.3	0.004		0.012
a2			3.3			0.130
a3	0		0.1	0.000		0.004
b	0.4		0.53	0.016		0.021
c	0.23		0.32	0.009		0.013
D (1)	15.8		16	0.622		0.630
D1	9.4		9.8	0.370		0.386
E	13.9		14.5	0.547		0.570
e		1.27			0.050	
e3		11.43			0.450	
E1 (1)	10.9		11.1	0.429		0.437
E2			2.9			0.114
E3	5.8		6.2	0.228		0.244
G	0		0.1	0.000		0.004
H	15.5		15.9	0.610		0.626
h			1.1			0.043
L	0.8		1.1	0.031		0.043
N	10° (max.)					
S	8° (max.)					
T		10			0.394	

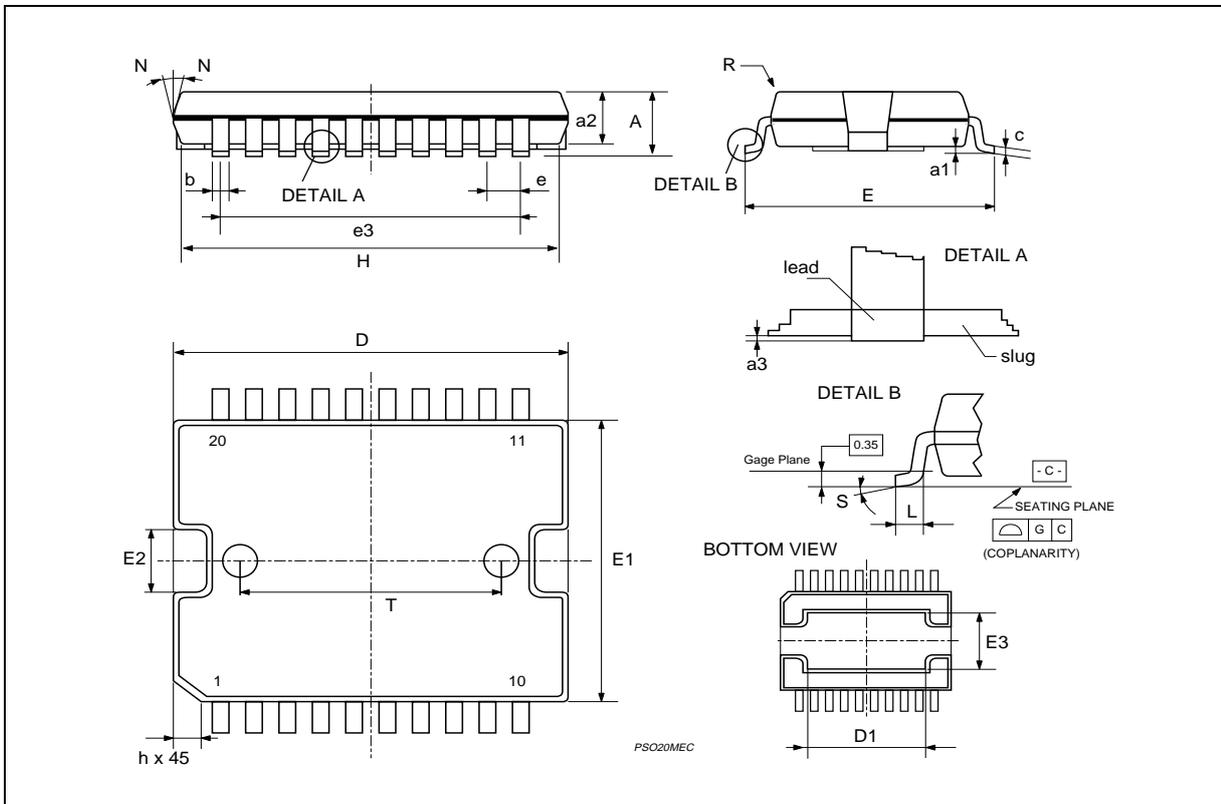
(1) "D and F" do not include mold flash or protrusions.
 - Mold flash or protrusions shall not exceed 0.15 mm (0.006").
 - Critical dimensions: "E", "G" and "a3"

OUTLINE AND MECHANICAL DATA



JEDEC MO-166

PowerSO20



Information furnished is believed to be accurate and reliable. However, STMicroelectronics assumes no responsibility for the consequences of use of such information nor for any infringement of patents or other rights of third parties which may result from its use. No license is granted by implication or otherwise under any patent or patent rights of STMicroelectronics. Specification mentioned in this publication are subject to change without notice. This publication supersedes and replaces all information previously supplied. STMicroelectronics products are not authorized for use as critical components in life support devices or systems without express written approval of STMicroelectronics.

The ST logo is a registered trademark of STMicroelectronics
© 2000 STMicroelectronics – Printed in Italy – All Rights Reserved
STMicroelectronics GROUP OF COMPANIES

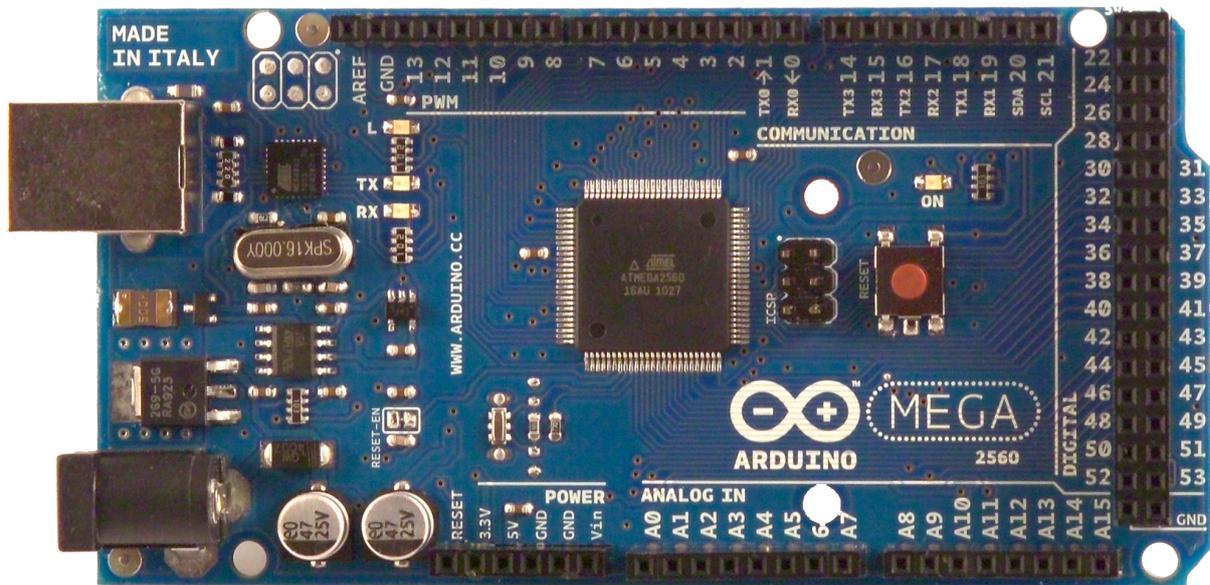
Australia - Brazil - China - Finland - France - Germany - Hong Kong - India - Italy - Japan - Malaysia - Malta - Morocco -
Singapore - Spain - Sweden - Switzerland - United Kingdom - U.S.A.

<http://www.st.com>

Arduino Mega 2560 Rev3



Arduino MEGA 2560



Product Overview

The Arduino Mega 2560 is a microcontroller board based on the ATmega2560 ([datasheet](#)). It has 54 digital input/output pins (of which 14 can be used as PWM outputs), 16 analog inputs, 4 UARTs (hardware serial ports), a 16 MHz crystal oscillator, a USB connection, a power jack, an ICSP header, and a reset button. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with a AC-to-DC adapter or battery to get started. The Mega is compatible with most shields designed for the Arduino Duemilanove or Diecimila.

Index

Technical Specifications

Page 2

How to use Arduino
Programming Enviroment, Basic Tutorials

Page 6

Terms & Conditions

Page 7

Enviromental Policies
half sqm of green via Impatto Zero®

Page 7



radiospares

RADIONICS



Technical Specification

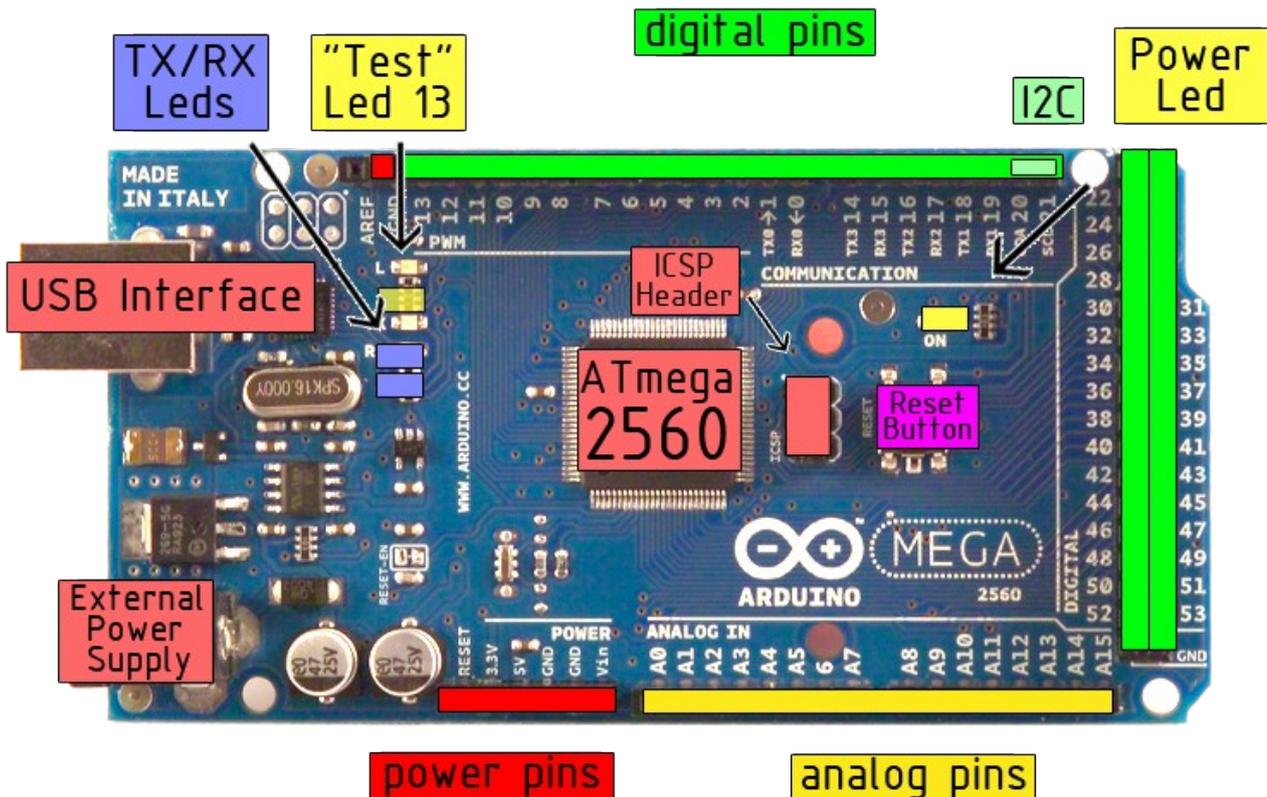


EAGLE files: [arduino-mega2560-reference-design.zip](#) Schematic: [arduino-mega2560-schematic.pdf](#)

Summary

Microcontroller	ATmega2560
Operating Voltage	5V
Input Voltage (recommended)	7-12V
Input Voltage (limits)	6-20V
Digital I/O Pins	54 (of which 14 provide PWM output)
Analog Input Pins	16
DC Current per I/O Pin	40 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	256 KB of which 8 KB used by bootloader
SRAM	8 KB
EEPROM	4 KB
Clock Speed	16 MHz

the board



radiospares

RADIONICS



Power

The Arduino Mega2560 can be powered via the USB connection or with an external power supply. The power source is selected automatically. External (non-USB) power can come either from an AC-to-DC adapter (wall-wart) or battery. The adapter can be connected by plugging a 2.1mm center-positive plug into the board's power jack. Leads from a battery can be inserted in the Gnd and Vin pin headers of the POWER connector.

The board can operate on an external supply of 6 to 20 volts. If supplied with less than 7V, however, the 5V pin may supply less than five volts and the board may be unstable. If using more than 12V, the voltage regulator may overheat and damage the board. The recommended range is 7 to 12 volts.

The Mega2560 differs from all preceding boards in that it does not use the FTDI USB-to-serial driver chip. Instead, it features the Atmega8U2 programmed as a USB-to-serial converter.

The power pins are as follows:

- **VIN.** The input voltage to the Arduino board when it's using an external power source (as opposed to 5 volts from the USB connection or other regulated power source). You can supply voltage through this pin, or, if supplying voltage via the power jack, access it through this pin.
- **5V.** The regulated power supply used to power the microcontroller and other components on the board. This can come either from VIN via an on-board regulator, or be supplied by USB or another regulated 5V supply.
- **3V3.** A 3.3 volt supply generated by the on-board regulator. Maximum current draw is 50 mA.
- **GND.** Ground pins.

Memory

The ATmega2560 has 256 KB of flash memory for storing code (of which 8 KB is used for the bootloader), 8 KB of SRAM and 4 KB of EEPROM (which can be read and written with the [EEPROM library](#)).

Input and Output

Each of the 54 digital pins on the Mega can be used as an input or output, using [pinMode\(\)](#), [digitalWrite\(\)](#), and [digitalRead\(\)](#) functions. They operate at 5 volts. Each pin can provide or receive a maximum of 40 mA and has an internal pull-up resistor (disconnected by default) of 20-50 kOhms. In addition, some pins have specialized functions:

- **Serial: 0 (RX) and 1 (TX); Serial 1: 19 (RX) and 18 (TX); Serial 2: 17 (RX) and 16 (TX); Serial 3: 15 (RX) and 14 (TX).** Used to receive (RX) and transmit (TX) TTL serial data. Pins 0 and 1 are also connected to the corresponding pins of the ATmega8U2 USB-to-TTL Serial chip .
- **External Interrupts: 2 (interrupt 0), 3 (interrupt 1), 18 (interrupt 5), 19 (interrupt 4), 20 (interrupt 3), and 21 (interrupt 2).** These pins can be configured to trigger an interrupt on a low value, a rising or falling edge, or a change in value. See the [attachInterrupt\(\)](#) function for details.
- **PWM: 0 to 13.** Provide 8-bit PWM output with the [analogWrite\(\)](#) function.
- **SPI: 50 (MISO), 51 (MOSI), 52 (SCK), 53 (SS).** These pins support SPI communication, which, although provided by the underlying hardware, is not currently included in the Arduino language. The SPI pins are also broken out on the ICSP header, which is physically compatible with the Duemilanove and Diecimila.
- **LED: 13.** There is a built-in LED connected to digital pin 13. When the pin is HIGH value, the LED is on, when the pin is LOW, it's off.
- **I²C: 20 (SDA) and 21 (SCL).** Support I²C (TWI) communication using the [Wire library](#) (documentation on the Wiring website). Note that these pins are not in the same location as the I²C pins on the Duemilanove.

The Mega2560 has 16 analog inputs, each of which provide 10 bits of resolution (i.e. 1024 different values). By default they measure from ground to 5 volts, though is it possible to change the upper end of their range using the AREF pin and [analogReference\(\)](#) function.

There are a couple of other pins on the board:

- **AREF.** Reference voltage for the analog inputs. Used with [analogReference\(\)](#).
- **Reset.** Bring this line LOW to reset the microcontroller. Typically used to add a reset button to shields which block the one on the board.



radiospares

RADIONICS



Communication

The Arduino Mega2560 has a number of facilities for communicating with a computer, another Arduino, or other microcontrollers. The ATmega2560 provides four hardware UARTs for TTL (5V) serial communication. An ATmega8U2 on the board channels one of these over USB and provides a virtual com port to software on the computer (Windows machines will need a .inf file, but OSX and Linux machines will recognize the board as a COM port automatically). The Arduino software includes a serial monitor which allows simple textual data to be sent to and from the board. The RX and TX LEDs on the board will flash when data is being transmitted via the ATmega8U2 chip and USB connection to the computer (but not for serial communication on pins 0 and 1).

A [SoftwareSerial library](#) allows for serial communication on any of the Mega's digital pins.

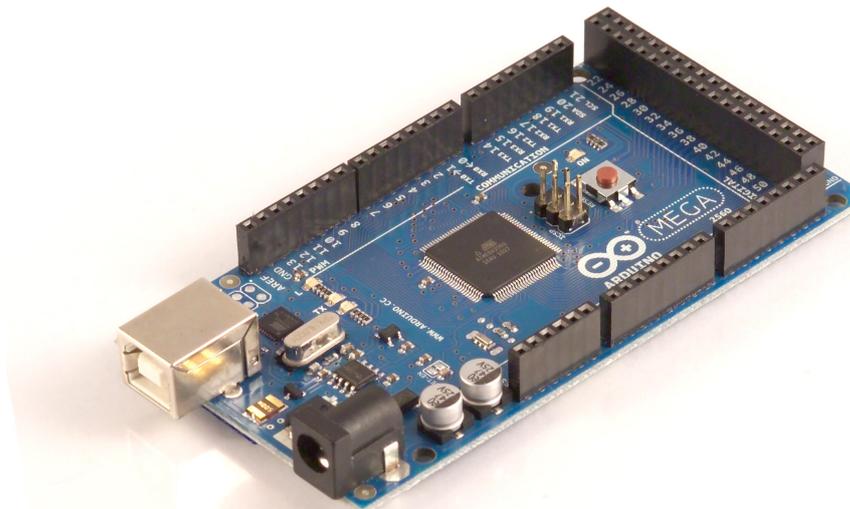
The ATmega2560 also supports I2C (TWI) and SPI communication. The Arduino software includes a Wire library to simplify use of the I2C bus; see the [documentation on the Wiring website](#) for details. To use the SPI communication, please see the ATmega2560 datasheet.

Programming

The Arduino Mega2560 can be programmed with the Arduino software ([download](#)). For details, see the [reference](#) and [tutorials](#).

The ATmega2560 on the Arduino Mega comes preburned with a [bootloader](#) that allows you to upload new code to it without the use of an external hardware programmer. It communicates using the original STK500 protocol ([reference](#), [C header files](#)).

You can also bypass the bootloader and program the microcontroller through the ICSP (In-Circuit Serial Programming) header; see [these instructions](#) for details.



radiospares

RADIONICS



Automatic (Software) Reset

Rather than requiring a physical press of the reset button before an upload, the Arduino Mega2560 is designed in a way that allows it to be reset by software running on a connected computer. One of the hardware flow control lines (DTR) of the ATmega8U2 is connected to the reset line of the ATmega2560 via a 100 nanofarad capacitor. When this line is asserted (taken low), the reset line drops long enough to reset the chip. The Arduino software uses this capability to allow you to upload code by simply pressing the upload button in the Arduino environment. This means that the bootloader can have a shorter timeout, as the lowering of DTR can be well-coordinated with the start of the upload.

This setup has other implications. When the Mega2560 is connected to either a computer running Mac OS X or Linux, it resets each time a connection is made to it from software (via USB). For the following half-second or so, the bootloader is running on the Mega2560. While it is programmed to ignore malformed data (i.e. anything besides an upload of new code), it will intercept the first few bytes of data sent to the board after a connection is opened. If a sketch running on the board receives one-time configuration or other data when it first starts, make sure that the software with which it communicates waits a second after opening the connection and before sending this data.

The Mega contains a trace that can be cut to disable the auto-reset. The pads on either side of the trace can be soldered together to re-enable it. It's labeled "RESET-EN". You may also be able to disable the auto-reset by connecting a 110 ohm resistor from 5V to the reset line; see [this forum thread](#) for details.

USB Overcurrent Protection

The Arduino Mega has a resettable polyfuse that protects your computer's USB ports from shorts and overcurrent. Although most computers provide their own internal protection, the fuse provides an extra layer of protection. If more than 500 mA is applied to the USB port, the fuse will automatically break the connection until the short or overload is removed.

Physical Characteristics and Shield Compatibility

The maximum length and width of the Mega PCB are 4 and 2.1 inches respectively, with the USB connector and power jack extending beyond the former dimension. Three screw holes allow the board to be attached to a surface or case. Note that the distance between digital pins 7 and 8 is 160 mil (0.16"), not an even multiple of the 100 mil spacing of the other pins.

The Mega is designed to be compatible with most shields designed for the Diecimila or Duemilanove. Digital pins 0 to 13 (and the adjacent AREF and GND pins), analog inputs 0 to 5, the power header, and ICSP header are all in equivalent locations. Further the main UART (serial port) is located on the same pins (0 and 1), as are external interrupts 0 and 1 (pins 2 and 3 respectively). SPI is available through the ICSP header on both the Mega and Duemilanove / Diecimila. **Please note that I²C is not located on the same pins on the Mega (20 and 21) as the Duemilanove / Diecimila (analog inputs 4 and 5).**



Radiospares

RADIONICS



How to use Arduino



Arduino can sense the environment by receiving input from a variety of sensors and can affect its surroundings by controlling lights, motors, and other actuators. The microcontroller on the board is programmed using the [Arduino programming language](#) (based on [Wiring](#)) and the Arduino development environment (based on [Processing](#)). Arduino projects can be stand-alone or they can communicate with software on running on a computer (e.g. Flash, Processing, MaxMSP).

Arduino is a cross-platform program. You'll have to follow different instructions for your personal OS. Check on the [Arduino site](#) for the latest instructions. <http://arduino.cc/en/Guide/HomePage>

Linux Install

Windows Install

Mac Install

Once you have downloaded/unzipped the arduino IDE, you can Plug the Arduino to your PC via USB cable.

Blink led

Now you're actually ready to "burn" your first program on the arduino board. To select "blink led", the physical translation of the well known programming "hello world", select

**File>Sketchbook>
Arduino-0017>Examples>
Digital>Blink**

Once you have your sketch you'll see something very close to the screenshot on the right.

In **Tools>Board** select MEGA

Now you have to go to **Tools>SerialPort** and select the right serial port, the one arduino is attached to.

```
Blink | Arduino 0017
File Edit Sketch Tools Help
Blink $
int ledPin = 13; // LED connected to digital pin 13

// The setup() method runs once, when the sketch starts

void setup() {
  // initialize the digital pin as an output:
  pinMode(ledPin, OUTPUT);
}

// the loop() method runs over and over again,
// as long as the Arduino has power

void loop()
{
  digitalWrite(ledPin, HIGH); // set the LED on
  delay(1000); // wait for a second
  digitalWrite(ledPin, LOW); // set the LED off
  delay(1000); // wait for a second
}
```



Done compiling.

Press Compile button
(to check for errors)



Upload



TX RX Flashing



Blinking Led!



radiospares

RADIONICS



Terms & Conditions



1. Warranties

1.1 The producer warrants that its products will conform to the Specifications. This warranty lasts for one (1) years from the date of the sale. The producer shall not be liable for any defects that are caused by neglect, misuse or mistreatment by the Customer, including improper installation or testing, or for any products that have been altered or modified in any way by a Customer. Moreover, The producer shall not be liable for any defects that result from Customer's design, specifications or instructions for such products. Testing and other quality control techniques are used to the extent the producer deems necessary.

1.2 If any products fail to conform to the warranty set forth above, the producer's sole liability shall be to replace such products. The producer's liability shall be limited to products that are determined by the producer not to conform to such warranty. If the producer elects to replace such products, the producer shall have a reasonable time to replacements. Replaced products shall be warranted for a new full warranty period.

1.3 EXCEPT AS SET FORTH ABOVE, PRODUCTS ARE PROVIDED "AS IS" AND "WITH ALL FAULTS." THE PRODUCER DISCLAIMS ALL OTHER WARRANTIES, EXPRESS OR IMPLIED, REGARDING PRODUCTS, INCLUDING BUT NOT LIMITED TO, ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE

1.4 Customer agrees that prior to using any systems that include the producer products, Customer will test such systems and the functionality of the products as used in such systems. The producer may provide technical, applications or design advice, quality characterization, reliability data or other services. Customer acknowledges and agrees that providing these services shall not expand or otherwise alter the producer's warranties, as set forth above, and no additional obligations or liabilities shall arise from the producer providing such services.

1.5 The Arduino™ products are not authorized for use in safety-critical applications where a failure of the product would reasonably be expected to cause severe personal injury or death. Safety-Critical Applications include, without limitation, life support devices and systems, equipment or systems for the operation of nuclear facilities and weapons systems. Arduino™ products are neither designed nor intended for use in military or aerospace applications or environments and for automotive applications or environment. Customer acknowledges and agrees that any such use of Arduino™ products which is solely at the Customer's risk, and that Customer is solely responsible for compliance with all legal and regulatory requirements in connection with such use.

1.6 Customer acknowledges and agrees that it is solely responsible for compliance with all legal, regulatory and safety-related requirements concerning its products and any use of Arduino™ products in Customer's applications, notwithstanding any applications-related information or support that may be provided by the producer.

2. Indemnification

The Customer acknowledges and agrees to defend, indemnify and hold harmless the producer from and against any and all third-party losses, damages, liabilities and expenses it incurs to the extent directly caused by: (i) an actual breach by a Customer of the representation and warranties made under this terms and conditions or (ii) the gross negligence or willful misconduct by the Customer.

3. Consequential Damages Waiver

In no event the producer shall be liable to the Customer or any third parties for any special, collateral, indirect, punitive, incidental, consequential or exemplary damages in connection with or arising out of the products provided hereunder, regardless of whether the producer has been advised of the possibility of such damages. This section will survive the termination of the warranty period.

4. Changes to specifications

The producer may make changes to specifications and product descriptions at any time, without notice. The Customer must not rely on the absence or characteristics of any features or instructions marked "reserved" or "undefined." The producer reserves these for future definition and shall have no responsibility whatsoever for conflicts or incompatibilities arising from future changes to them. The product information on the Web Site or Materials is subject to change without notice. Do not finalize a design with this information.



Environmental Policies



The producer of Arduino™ has joined the Impatto Zero® policy of LifeGate.it. For each Arduino board produced is created / looked after half squared Km of Costa Rica's forest's.



radiospares

RADIONICS

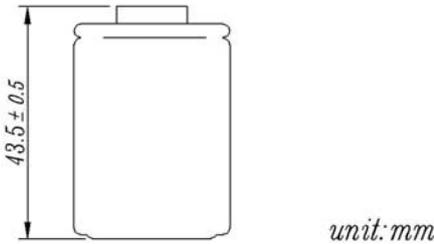
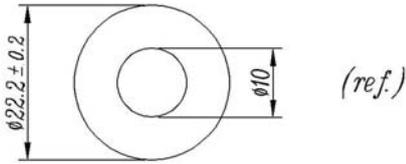


Baterías NiMH 1.2V/celda



Sealed Rechargeable Cylindrical cell Nickel Metal Hydride

SC-2400 mAh



Date Sheet

Type-----EnrichpowerSC2400
System-----Sealed rechargeable Ni-MHCylindrical cell
Nominal Voltage-----1.2V
Application-----Discharge current 240mA to 24A (Continuous)

Capacity (discharged at 0.2C to 1.0v at 20°C)

Nominal 2400mAh
 Minimum 2300mAh
 Typical 2400mAh

Internal resistance (at 1 kHz)

Fully charged, 20°C (approx) --- 5 ~ 7mΩ

Weight approx-----47g

Dimensions(including shrink sleeve)

-Diameter -----22.2(±0.2mm)

-Height -----43.5(±0.5mm)

Cd content(%cell w):-----0%

Typical Discharge time&Avg. Voltage (20°C)

30A--280S--1.100V (TURBOMATCHER , discharged at 30A to 0.9v at 20°C)

Max.discharge current(continuous)-----20C

Charging conditions(20°C)

standard charge-----240mA 14h
 quick charge-----2400mA 1.1h
 fast charge -----2A to 6A

Charge termination control recommended control parameter

-ΔV 5mV

DT/dt 0.8°C/min(0.5-1C)

0.8°C-1°C/min(>1C)

TCO 40-45°C

Timer 105%nominal input

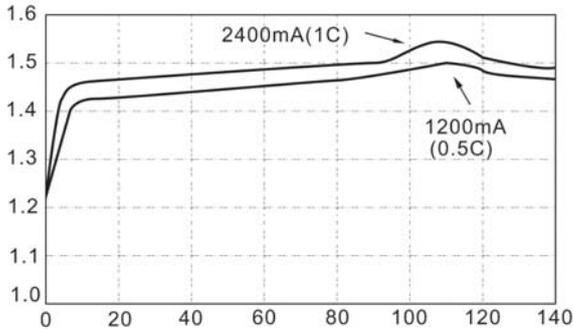
Continuous 100mA maximum current for 1 year

Overcharge No conspicuous deformation and/or leakage

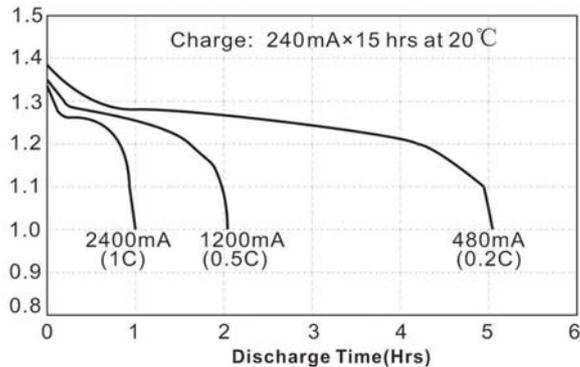
Max.charging voltage 1.5V at 240mA charging

Operation temperatures(recommended)

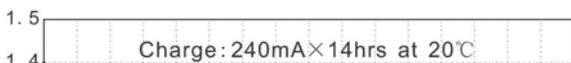
Fast Charge (Charge Control required) Voltage(V)



Low Rate Discharge Voltage(V)



High Rate Discharge Voltage (V)



Storage----- -20°C up to +35°C

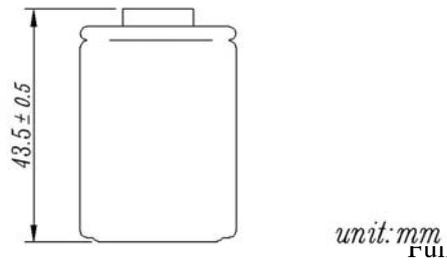
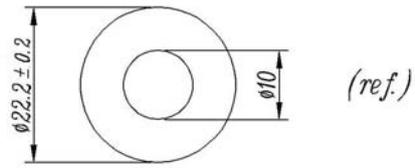
Discharge----- -20°C up to +50°C

standard charge----- 0°C up to +40°C

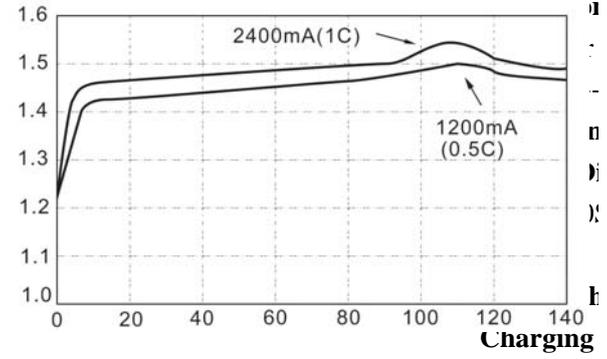
quickcharge,fastcharge--10°C upto+40°C

permanent charge-----+10°C up to +45°C

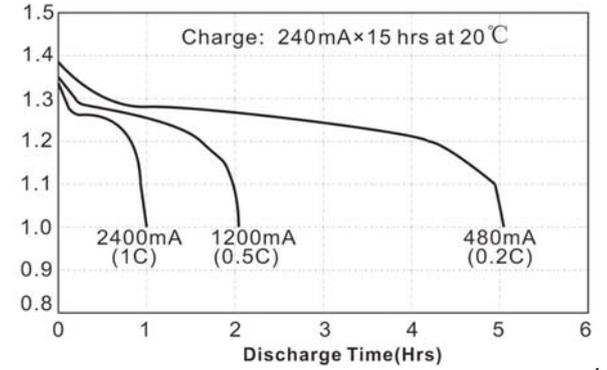
Life expectancy(20°C)----->500 cycles



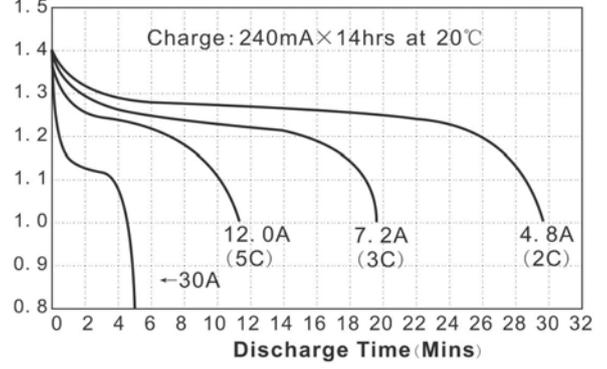
Fast Charge (Charge Control required)
Voltage(V)



Low Rate Discharge
Voltage(V)



High Rate Discharge
Voltage (V)



Regulador de tensión step-down

LM2596





LM2596 SIMPLE SWITCHER® Power Converter 150-kHz 3-A Step-Down Voltage Regulator

1 Features

- 3.3-V, 5-V, 12-V, and Adjustable Output Versions
- Adjustable Version Output Voltage Range: 1.2-V to 37-V \pm 4% Maximum Over Line and Load Conditions
- Available in TO-220 and TO-263 Packages
- 3-A Output Load Current
- Input Voltage Range Up to 40 V
- Requires Only 4 External Components
- Excellent Line and Load Regulation Specifications
- 150-kHz Fixed-Frequency Internal Oscillator
- TTL Shutdown Capability
- Low Power Standby Mode, I_Q , Typically 80 μ A
- High Efficiency
- Uses Readily Available Standard Inductors
- Thermal Shutdown and Current-Limit Protection
- Create a Custom Design Using the LM2596 with the [WEBENCH Power Designer](#)

2 Applications

- Simple High-Efficiency Step-Down (Buck) Regulator
- On-Card Switching Regulators
- Positive to Negative Converter

3 Description

The LM2596 series of regulators are monolithic integrated circuits that provide all the active functions for a step-down (buck) switching regulator, capable of driving a 3-A load with excellent line and load regulation. These devices are available in fixed output voltages of 3.3 V, 5 V, 12 V, and an adjustable output version.

Requiring a minimum number of external components, these regulators are simple to use and include internal frequency compensation, and a fixed-frequency oscillator.

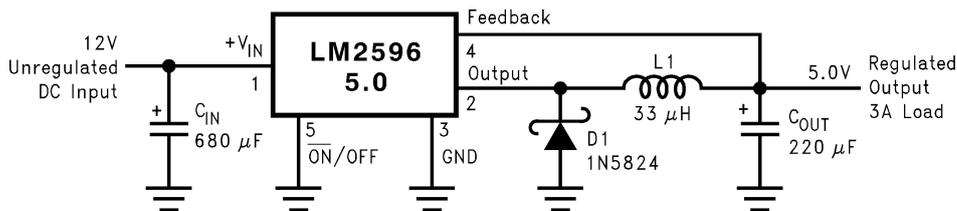
The LM2596 series operates at a switching frequency of 150 kHz, thus allowing smaller sized filter components than what would be required with lower frequency switching regulators. Available in a standard 7-pin TO-220 package with several different lead bend options, and a 7-pin TO-263 surface mount package.

Device Information⁽¹⁾

PART NUMBER	PACKAGE	BODY SIZE (NOM)
LM2596	TO-220 (7)	14.986 mm x 10.16 mm
	TO-263 (7)	10.10 mm x 8.89 mm

(1) For all available packages, see the orderable addendum at the end of the data sheet.

Typical Application



Copyright © 2016, Texas Instruments Incorporated

(Fixed Output Voltage Versions)

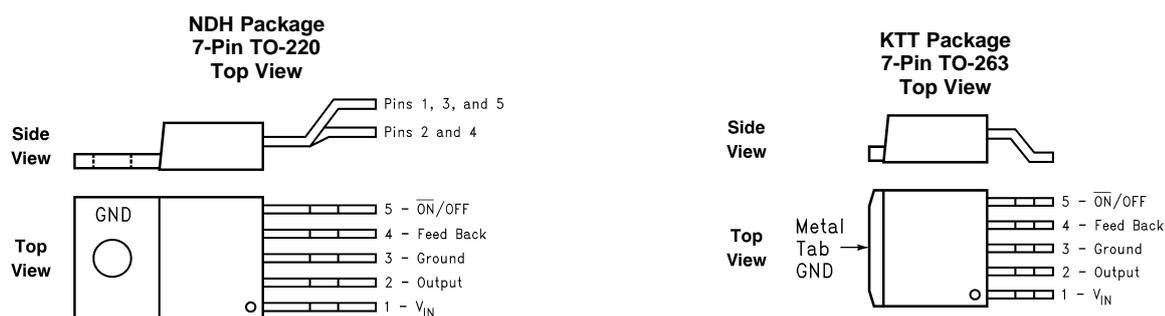


5 Description (continued)

A standard series of inductors are available from several different manufacturers optimized for use with the LM2596 series. This feature greatly simplifies the design of switch-mode power supplies.

Other features include a $\pm 4\%$ tolerance on output voltage under specified input voltage and output load conditions, and $\pm 15\%$ on the oscillator frequency. External shutdown is included, featuring typically $80 \mu\text{A}$ standby current. Self-protection features include a two stage frequency reducing current limit for the output switch and an overtemperature shutdown for complete protection under fault conditions.

6 Pin Configuration and Functions



Pin Functions

PIN		I/O	DESCRIPTION
NO.	NAME		
1	V_{IN}	I	This is the positive input supply for the IC switching regulator. A suitable input bypass capacitor must be present at this pin to minimize voltage transients and to supply the switching currents required by the regulator.
2	Output	O	Internal switch. The voltage at this pin switches between approximately $(+V_{\text{IN}} - V_{\text{SAT}})$ and approximately -0.5 V , with a duty cycle of $V_{\text{OUT}} / V_{\text{IN}}$. To minimize coupling to sensitive circuitry, the PCB copper area connected to this pin must be kept to a minimum.
3	Ground	—	Circuit ground.
4	Feedback	I	Senses the regulated output voltage to complete the feedback loop.
5	$\overline{\text{ON/OFF}}$	I	Allows the switching regulator circuit to be shut down using logic signals thus dropping the total input supply current to approximately $80 \mu\text{A}$. Pulling this pin below a threshold voltage of approximately 1.3 V turns the regulator on, and pulling this pin above 1.3 V (up to a maximum of 25 V) shuts the regulator down. If this shutdown feature is not required, the $\overline{\text{ON/OFF}}$ pin can be wired to the ground pin or it can be left open. In either case, the regulator will be in the ON condition.

7 Specifications

7.1 Absolute Maximum Ratings

 over operating free-air temperature range (unless otherwise noted)⁽¹⁾⁽²⁾

		MIN	MAX	UNIT
Maximum supply voltage (V_{IN})			45	V
\overline{SD}/SS pin input voltage ⁽³⁾			6	V
Delay pin voltage ⁽³⁾			1.5	V
Flag pin voltage		–0.3	45	V
Feedback pin voltage		–0.3	25	V
Output voltage to ground, steady-state			–1	V
Power dissipation		Internally limited		
Lead temperature	KTW package	Vapor phase (60 s)		°C
		Infrared (10 s)		
	NDZ package, soldering (10 s)		260	
Maximum junction temperature			150	°C
Storage temperature, T_{stg}		–65	150	°C

- Stresses beyond those listed under *Absolute Maximum Ratings* may cause permanent damage to the device. These are stress ratings only, which do not imply functional operation of the device at these or any other conditions beyond those indicated under *Recommended Operating Conditions*. Exposure to absolute-maximum-rated conditions for extended periods may affect device reliability.
- If Military/Aerospace specified devices are required, please contact the Texas Instruments Sales Office/ Distributors for availability and specifications.
- Voltage internally clamped. If clamp voltage is exceeded, limit current to a maximum of 1 mA.

7.2 ESD Ratings

			VALUE	UNIT
V_{ESD}	Electrostatic discharge	Human-body model (HBM), per ANSI/ESDA/JEDEC JS-001 ⁽¹⁾	±2000	V

- JEDEC document JEP155 states that 500-V HBM allows safe manufacturing with a standard ESD control process.

7.3 Operating Conditions

	MIN	MAX	UNIT
Supply voltage	4.5	40	V
Temperature	–40	125	°C

7.4 Thermal Information

THERMAL METRIC ⁽¹⁾		LM2596		UNIT
		KTW (TO-263)	NDZ (TO-220)	
		7 PINS	7 PINS	
$R_{\theta JA}$	Junction-to-ambient thermal resistance ⁽²⁾⁽³⁾	See ⁽⁴⁾	50	°C/W
		See ⁽⁵⁾	—	
		See ⁽⁶⁾	—	
		See ⁽⁷⁾	—	
$R_{\theta JC(top)}$	Junction-to-case (top) thermal resistance	2	2	°C/W

- For more information about traditional and new thermal metrics, see the *Semiconductor and IC Package Thermal Metrics* application report, [SPRA953](#).
- The package thermal impedance is calculated in accordance to JESD 51-7.
- Thermal Resistances were simulated on a 4-layer, JEDEC board.
- Junction to ambient thermal resistance (no external heat sink) for the package mounted TO-220 package mounted vertically, with the leads soldered to a printed circuit board with (1 oz.) copper area of approximately 1 in².
- Junction to ambient thermal resistance with the TO-263 package tab soldered to a single sided printed circuit board with 0.5 in² of 1-oz copper area.
- Junction to ambient thermal resistance with the TO-263 package tab soldered to a single sided printed circuit board with 2.5 in² of 1-oz copper area.
- Junction to ambient thermal resistance with the TO-263 package tab soldered to a double sided printed circuit board with 3 in² of 1-oz copper area on the LM2596S side of the board, and approximately 16 in² of copper on the other side of the PCB.

7.5 Electrical Characteristics – 3.3-V Version

Specifications are for $T_J = 25^\circ\text{C}$ (unless otherwise noted)

PARAMETER		TEST CONDITIONS		MIN ⁽¹⁾	TYP ⁽²⁾	MAX ⁽¹⁾	UNIT
SYSTEM PARAMETERS⁽³⁾ (see Figure 35 for test circuit)							
V_{OUT}	Output voltage	$4.75\text{ V} \leq V_{IN} \leq 40\text{ V}$, $0.2\text{ A} \leq I_{LOAD} \leq 3\text{ A}$	$T_J = 25^\circ\text{C}$	3.168	3.3	3.432	V
			$-40^\circ\text{C} \leq T_J \leq 125^\circ\text{C}$	3.135		3.465	
η	Efficiency	$V_{IN} = 12\text{ V}$, $I_{LOAD} = 3\text{ A}$		73%			

- (1) All room temperature limits are 100% production tested. All limits at temperature extremes are specified via correlation using standard Statistical Quality Control (SQC) methods. All limits are used to calculate Average Outgoing Quality Level (AOQL).
- (2) Typical numbers are at 25°C and represent the most likely norm.
- (3) External components such as the catch diode, inductor, input and output capacitors can affect switching regulator system performance. When the LM2596 is used as shown in Figure 35, system performance is shown in the test conditions column.

7.6 Electrical Characteristics – 5-V Version

Specifications are for $T_J = 25^\circ\text{C}$ (unless otherwise noted)

PARAMETER		TEST CONDITIONS		MIN ⁽¹⁾	TYP ⁽²⁾	MAX ⁽¹⁾	UNIT
SYSTEM PARAMETERS⁽³⁾ (see Figure 35 for test circuit)							
V_{OUT}	Output voltage	$7\text{ V} \leq V_{IN} \leq 40\text{ V}$, $0.2\text{ A} \leq I_{LOAD} \leq 3\text{ A}$	$T_J = 25^\circ\text{C}$	4.8	5	5.2	V
			$-40^\circ\text{C} \leq T_J \leq 125^\circ\text{C}$	4.75		5.25	
η	Efficiency	$V_{IN} = 12\text{ V}$, $I_{LOAD} = 3\text{ A}$		80%			

- (1) All room temperature limits are 100% production tested. All limits at temperature extremes are specified via correlation using standard Statistical Quality Control (SQC) methods. All limits are used to calculate Average Outgoing Quality Level (AOQL).
- (2) Typical numbers are at 25°C and represent the most likely norm.
- (3) External components such as the catch diode, inductor, input and output capacitors can affect switching regulator system performance. When the LM2596 is used as shown in Figure 35, system performance is shown in the test conditions column.

7.7 Electrical Characteristics – 12-V Version

Specifications are for $T_J = 25^\circ\text{C}$ (unless otherwise noted)

PARAMETER		TEST CONDITIONS		MIN ⁽¹⁾	TYP ⁽²⁾	MAX ⁽¹⁾	UNIT
SYSTEM PARAMETERS⁽³⁾ (see Figure 35 for test circuit)							
V_{OUT}	Output voltage	$15\text{ V} \leq V_{IN} \leq 40\text{ V}$, $0.2\text{ A} \leq I_{LOAD} \leq 3\text{ A}$	$T_J = 25^\circ\text{C}$	11.52	12	12.48	V
			$-40^\circ\text{C} \leq T_J \leq 125^\circ\text{C}$	11.4		12.6	
η	Efficiency	$V_{IN} = 25\text{ V}$, $I_{LOAD} = 3\text{ A}$		90%			

- (1) All room temperature limits are 100% production tested. All limits at temperature extremes are specified via correlation using standard Statistical Quality Control (SQC) methods. All limits are used to calculate Average Outgoing Quality Level (AOQL).
- (2) Typical numbers are at 25°C and represent the most likely norm.
- (3) External components such as the catch diode, inductor, input and output capacitors can affect switching regulator system performance. When the LM2596 is used as shown in Figure 35, system performance is shown in the test conditions column.

7.8 Electrical Characteristics – Adjustable Voltage Version

Specifications are for $T_J = 25^\circ\text{C}$ (unless otherwise noted)

PARAMETER		TEST CONDITIONS		MIN ⁽¹⁾	TYP ⁽²⁾	MAX ⁽¹⁾	UNIT
SYSTEM PARAMETERS⁽³⁾ (see Figure 35 for test circuit)							
V_{FB}	Feedback voltage	$4.5\text{ V} \leq V_{IN} \leq 40\text{ V}$, $0.2\text{ A} \leq I_{LOAD} \leq 3\text{ A}$		1.23			V
			V_{OUT} programmed for 3 V (see Figure 35 for test circuit)	$T_J = 25^\circ\text{C}$	1.193	1.267	
				$-40^\circ\text{C} \leq T_J \leq 125^\circ\text{C}$	1.18	1.28	
η	Efficiency	$V_{IN} = 12\text{ V}$, $V_{OUT} = 3\text{ V}$, $I_{LOAD} = 3\text{ A}$		73%			

- (1) All room temperature limits are 100% production tested. All limits at temperature extremes are specified via correlation using standard Statistical Quality Control (SQC) methods. All limits are used to calculate Average Outgoing Quality Level (AOQL).
- (2) Typical numbers are at 25°C and represent the most likely norm.
- (3) External components such as the catch diode, inductor, input and output capacitors can affect switching regulator system performance. When the LM2596 is used as shown in Figure 35, system performance is shown in the test conditions column.

7.9 Electrical Characteristics – All Output Voltage Versions

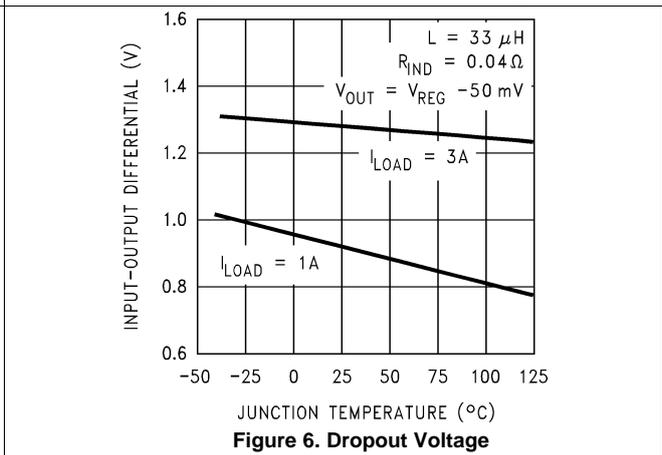
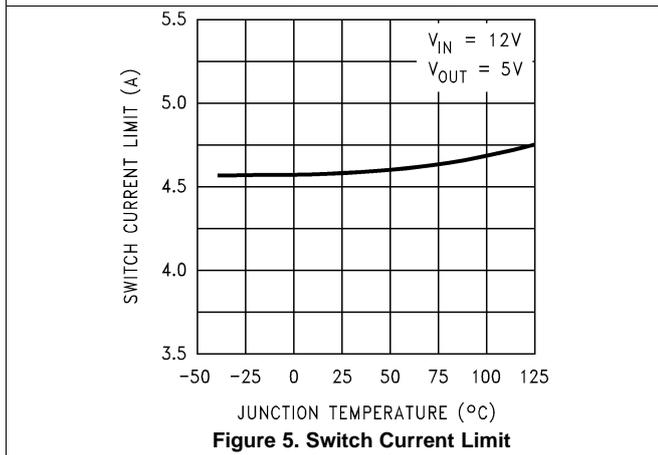
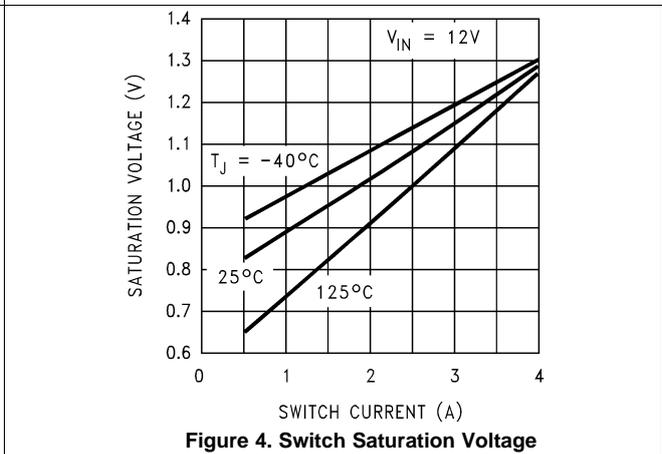
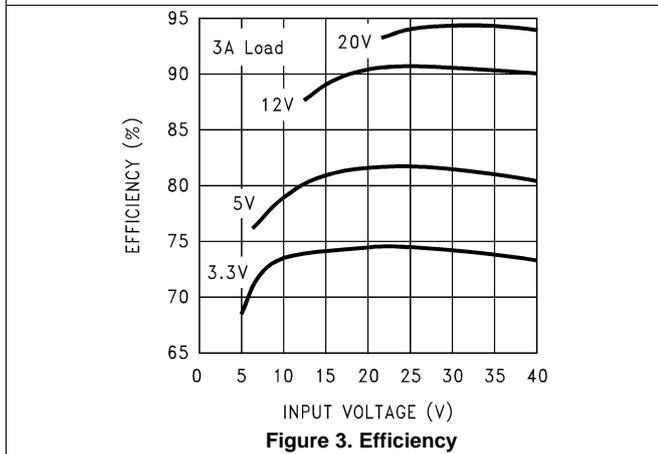
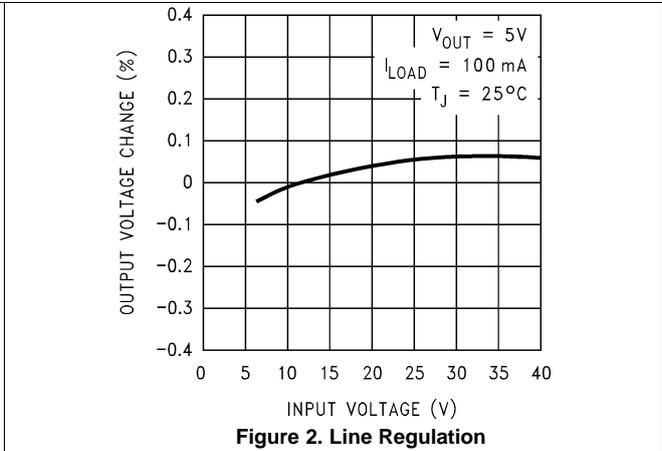
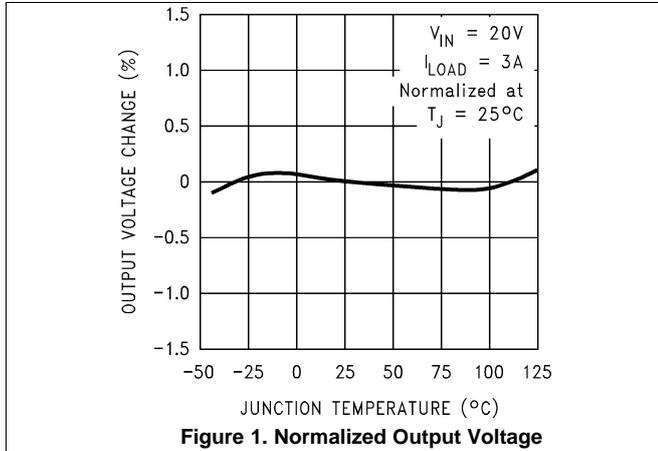
Specifications are for $T_J = 25^\circ\text{C}$, $I_{\text{LOAD}} = 500\text{ mA}$, $V_{\text{IN}} = 12\text{ V}$ for the 3.3-V, 5-V, and adjustable version, and $V_{\text{IN}} = 24\text{ V}$ for the 12-V version (unless otherwise noted).

PARAMETER		TEST CONDITIONS		MIN ⁽¹⁾	TYP ⁽²⁾	MAX ⁽¹⁾	UNIT
DEVICE PARAMETERS							
I_b	Feedback bias current	Adjustable version only, $V_{\text{FB}} = 1.3\text{ V}$	$T_J = 25^\circ\text{C}$	10	50		nA
			$-40^\circ\text{C} \leq T_J \leq 125^\circ\text{C}$			100	
f_O	Oscillator frequency ⁽³⁾		$T_J = 25^\circ\text{C}$	127	150	173	kHz
			$-40^\circ\text{C} \leq T_J \leq 125^\circ\text{C}$	110		173	
V_{SAT}	Saturation voltage ^{(4) (5)}	$I_{\text{OUT}} = 3\text{ A}$	$T_J = 25^\circ\text{C}$	1.16	1.4		V
			$-40^\circ\text{C} \leq T_J \leq 125^\circ\text{C}$			1.5	
DC	Max duty cycle (ON) ⁽⁵⁾			100%			
	Min duty cycle (OFF) ⁽⁶⁾			0%			
I_{CL}	Current limit ^{(4) (5)}	Peak current	$T_J = 25^\circ\text{C}$	3.6	4.5	6.9	A
			$-40^\circ\text{C} \leq T_J \leq 125^\circ\text{C}$	3.4		7.5	
I_L	Output leakage current ^{(4) (6)}	Output = 0 V, $V_{\text{IN}} = 40\text{ V}$				50	μA
		Output = -1 V			2	30	mA
I_Q	Operating quiescent current ⁽⁶⁾	See ⁽⁶⁾		5	10		mA
I_{STBY}	Current standby quiescent	$\overline{\text{ON}}/\text{OFF}$ pin = 5 V (OFF) ⁽⁷⁾	$T_J = 25^\circ\text{C}$	80	200		μA
			$-40^\circ\text{C} \leq T_J \leq 125^\circ\text{C}$			250	μA
SHUTDOWN/SOFT-START CONTROL (see Figure 35 for test circuit)							
V_{IH}	$\overline{\text{ON}}/\text{OFF}$ pin logic input threshold voltage	Low (regulator ON)	$T_J = 25^\circ\text{C}$	1.3			V
			$-40^\circ\text{C} \leq T_J \leq 125^\circ\text{C}$			0.6	
V_{IL}		High (regulator OFF)	$T_J = 25^\circ\text{C}$	1.3			V
			$-40^\circ\text{C} \leq T_J \leq 125^\circ\text{C}$	2			
I_H	$\overline{\text{ON}}/\text{OFF}$ pin input current	$V_{\text{LOGIC}} = 2.5\text{ V}$ (regulator OFF)		5	15		μA
I_L		$V_{\text{LOGIC}} = 0.5\text{ V}$ (regulator ON)		0.02	5		μA

- (1) All room temperature limits are 100% production tested. All limits at temperature extremes are specified via correlation using standard Statistical Quality Control (SQC) methods. All limits are used to calculate Average Outgoing Quality Level (AOQL).
- (2) Typical numbers are at 25°C and represent the most likely norm.
- (3) The switching frequency is reduced when the second stage current limit is activated. The amount of reduction is determined by the severity of current overload.
- (4) No diode, inductor, or capacitor connected to output pin.
- (5) Feedback pin removed from output and connected to 0 V to force the output transistor switch ON.
- (6) Feedback pin removed from output and connected to 12 V for the 3.3-V, 5-V, and the adjustable versions, and 15 V for the 12-V version, to force the output transistor switch OFF.
- (7) $V_{\text{IN}} = 40\text{ V}$.

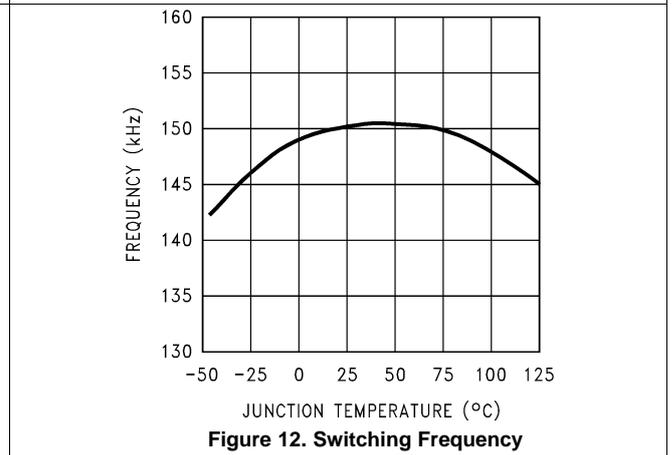
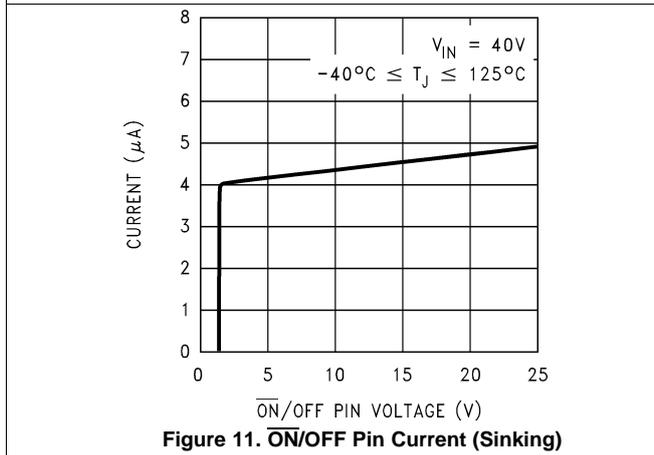
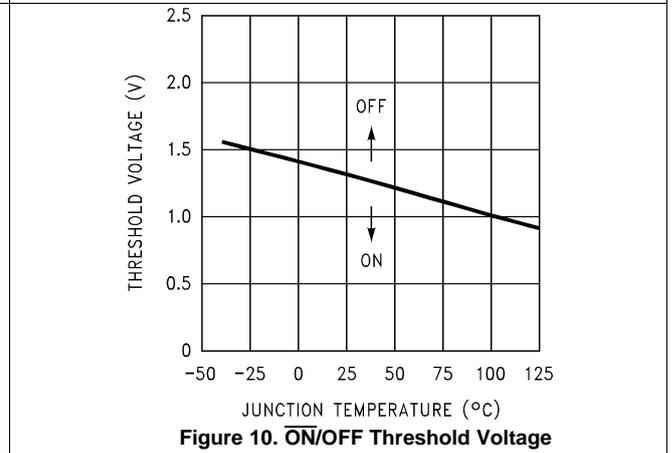
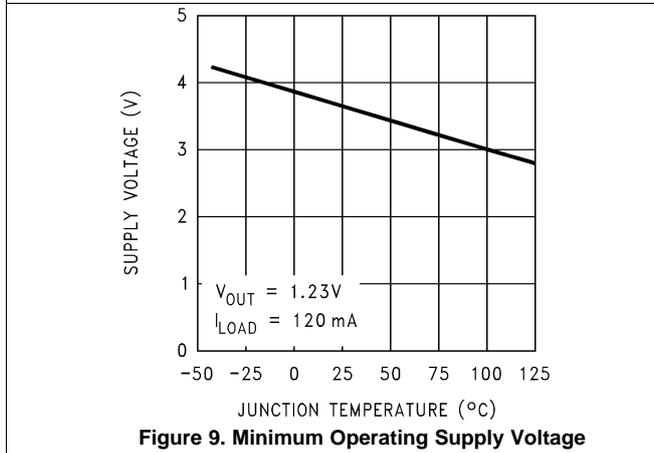
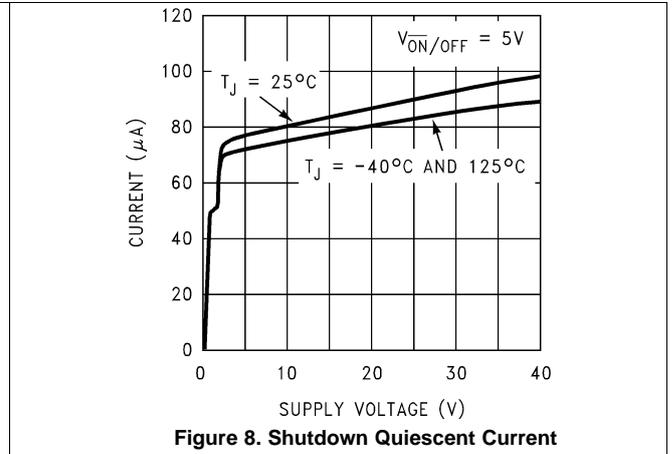
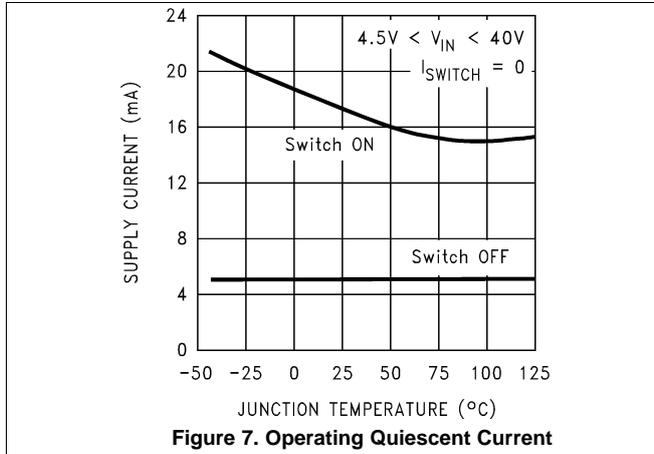
7.10 Typical Characteristics

See Figure 35 for test circuit



Typical Characteristics (continued)

See Figure 35 for test circuit



Typical Characteristics (continued)

See [Figure 35](#) for test circuit

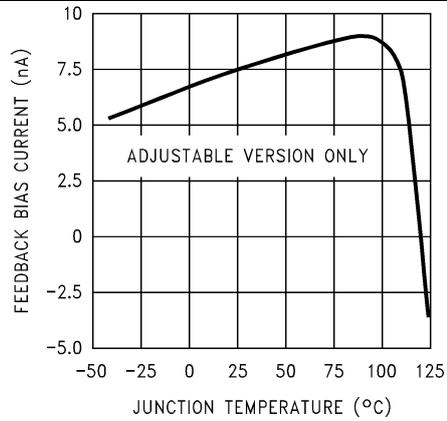


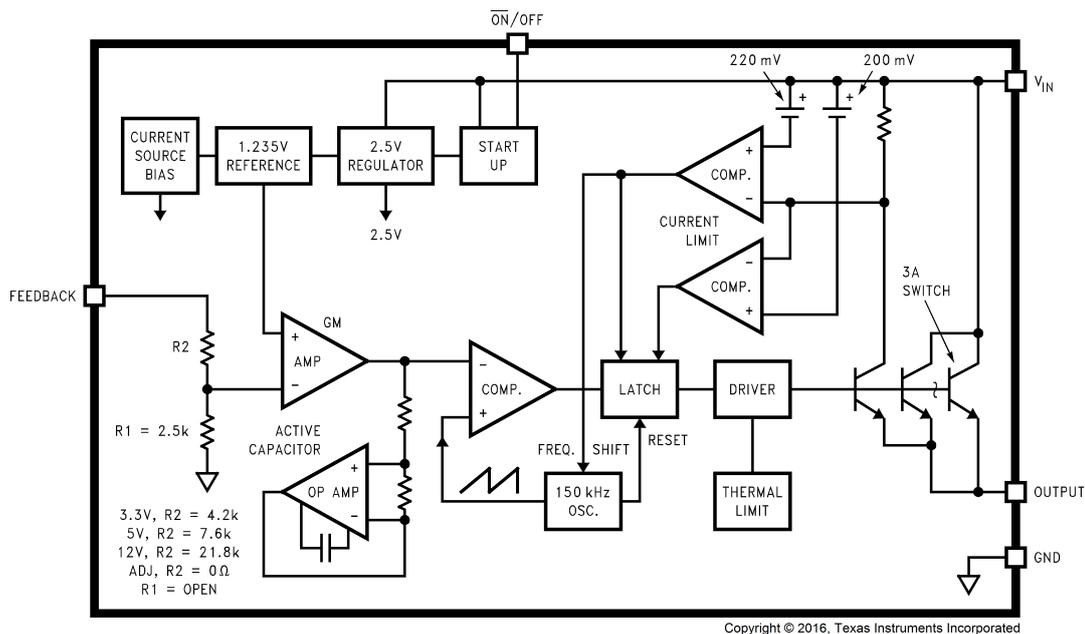
Figure 13. Feedback Pin Bias Current

8 Detailed Description

8.1 Overview

The LM2596 SIMPLE SWITCHER[®] regulator is an easy-to-use, nonsynchronous, step-down DC-DC converter with a wide input voltage range up to 40 V. The regulator is capable of delivering up to 3-A DC load current with excellent line and load regulation. These devices are available in fixed output voltages of 3.3-V, 5-V, 12-V and an adjustable output version. The family requires few external components, and the pin arrangement was designed for simple, optimum PCB layout.

8.2 Functional Block Diagram



8.3 Feature Description

8.3.1 Delayed Start-Up

The circuit in [Figure 14](#) uses the $\overline{\text{ON/OFF}}$ pin to provide a time delay between the time the input voltage is applied and the time the output voltage comes up (only the circuitry pertaining to the delayed start-up is shown). As the input voltage rises, the charging of capacitor C1 pulls the $\overline{\text{ON/OFF}}$ pin high, keeping the regulator OFF. Once the input voltage reaches its final value and the capacitor stops charging, resistor R₂ pulls the $\overline{\text{ON/OFF}}$ pin low, thus allowing the circuit to start switching. Resistor R₁ is included to limit the maximum voltage applied to the $\overline{\text{ON/OFF}}$ pin (maximum of 25 V), reduces power supply noise sensitivity, and also limits the capacitor C1 discharge current. When high input ripple voltage exists, avoid long delay time, because this ripple can be coupled into the $\overline{\text{ON/OFF}}$ pin and cause problems.

This delayed start-up feature is useful in situations where the input power source is limited in the amount of current it can deliver. It allows the input voltage to rise to a higher voltage before the regulator starts operating. Buck regulators require less input current at higher input voltages.

Feature Description (continued)

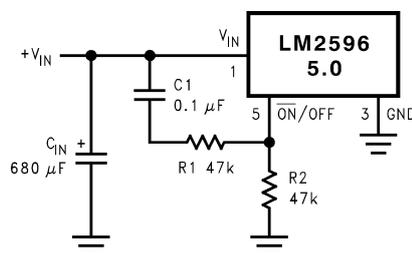


Figure 14. Delayed Start-Up

8.3.2 Undervoltage Lockout

Some applications require the regulator to remain off until the input voltage reaches a predetermined voltage. Figure 15 shows an undervoltage lockout feature applied to a buck regulator, while Figure 16 and Figure 17 apply the same feature to an inverting circuit. The circuit in Figure 16 features a constant threshold voltage for turnon and turnoff (Zener voltage plus approximately one volt). If hysteresis is required, the circuit in Figure 17 has a turnon voltage which is different than the turnoff voltage. The amount of hysteresis is approximately equal to the value of the output voltage. If Zener voltages greater than 25 V are used, an additional 47-kΩ resistor is required from the $\overline{\text{ON/OFF}}$ pin to the ground pin to stay within the 25 V maximum limit of the $\overline{\text{ON/OFF}}$ pin.

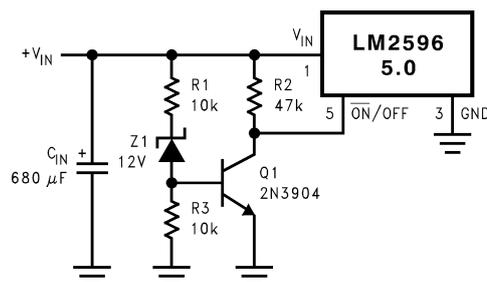
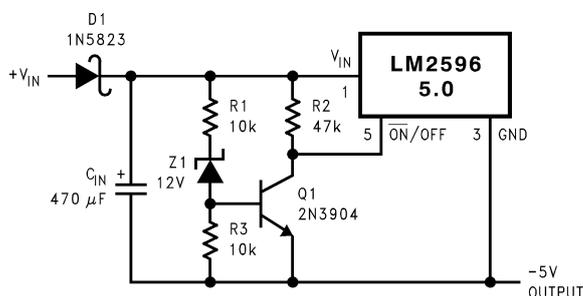


Figure 15. Undervoltage Lockout for Buck Regulator

8.3.3 Inverting Regulator

The circuit in Figure 18 converts a positive input voltage to a negative output voltage with a common ground. The circuit operates by bootstrapping the ground pin of the regulator to the negative output voltage, then grounding the feedback pin, the regulator senses the inverted output voltage and regulates it.



This circuit has an $\overline{\text{ON/OFF}}$ threshold of approximately 13 V.

Figure 16. Undervoltage Lockout for Inverting Regulator

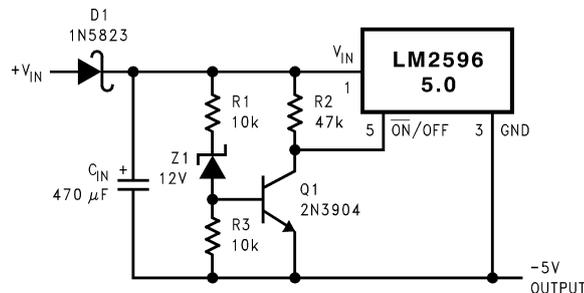
Feature Description (continued)

This example uses the LM2596-5.0 to generate a -5V output, but other output voltages are possible by selecting other output voltage versions, including the adjustable version. Because this regulator topology can produce an output voltage that is either greater than or less than the input voltage, the maximum output current greatly depends on both the input and output voltage. Figure 19 provides a guide as to the amount of output load current possible for the different input and output voltage conditions.

The maximum voltage appearing across the regulator is the absolute sum of the input and output voltage, and this must be limited to a maximum of 40V . For example, when converting $+20\text{V}$ to -12V , the regulator would see 32V between the input pin and ground pin. The LM2596 has a maximum input voltage spec of 40V .

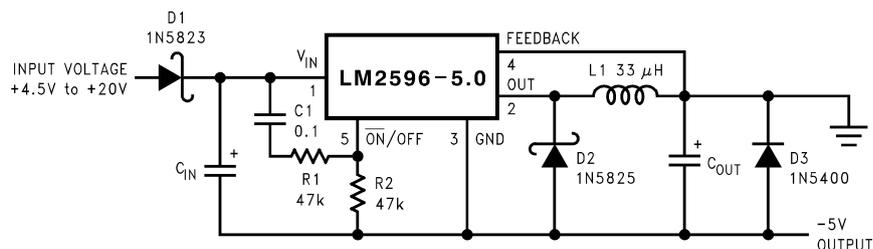
Additional diodes are required in this regulator configuration. Diode D1 is used to isolate input voltage ripple or noise from coupling through the C_{IN} capacitor to the output, under light or no load conditions. Also, this diode isolation changes the topology to closely resemble a buck configuration, thus providing good closed-loop stability. TI recommends using a Schottky diode for low input voltages, (because of its lower voltage drop) but for higher input voltages, a fast recovery diode could be used.

Without diode D3, when the input voltage is first applied, the charging current of C_{IN} can pull the output positive by several volts for a short period of time. Adding D3 prevents the output from going positive by more than a diode voltage.



This circuit has hysteresis
 Regulator starts switching at $V_{IN} = 13\text{V}$
 Regulator stops switching at $V_{IN} = 8\text{V}$

Figure 17. Undervoltage Lockout With Hysteresis for Inverting Regulator



C_{IN} — $68\text{-}\mu\text{F}$, 25-V Tant. Sprague 595D
 $470\text{-}\mu\text{F}$, 50-V Elec. Panasonic HFQ
 C_{OUT} — $47\text{-}\mu\text{F}$, 20-V Tant. Sprague 595D
 $220\text{-}\mu\text{F}$, 25-V Elec. Panasonic HFQ

Figure 18. Inverting -5V Regulator With Delayed Start-Up

Feature Description (continued)

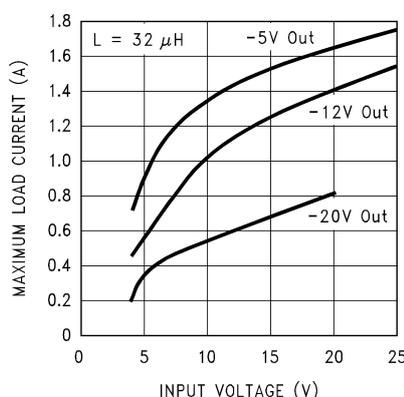


Figure 19. Inverting Regulator Typical Load Current

Because of differences in the operation of the inverting regulator, the standard design procedure is not used to select the inductor value. In the majority of designs, a 33- μ H, 3.5-A inductor is the best choice. Capacitor selection can also be narrowed down to just a few values. Using the values shown in Figure 18 will provide good results in the majority of inverting designs.

This type of inverting regulator can require relatively large amounts of input current when starting up, even with light loads. Input currents as high as the LM2596 current limit (approximately 4.5 A) are required for at least 2 ms or more, until the output reaches its nominal output voltage. The actual time depends on the output voltage and the size of the output capacitor. Input power sources that are current limited or sources that can not deliver these currents without getting loaded down, may not work correctly. Because of the relatively high start-up currents required by the inverting topology, the delayed start-up feature (C1, R1, and R2) shown in Figure 18 is recommended. By delaying the regulator start-up, the input capacitor is allowed to charge up to a higher voltage before the switcher begins operating. A portion of the high input current required for start-up is now supplied by the input capacitor (C_{IN}). For severe start-up conditions, the input capacitor can be made much larger than normal.

8.3.4 Inverting Regulator Shutdown Methods

Using the $\overline{\text{ON/OFF}}$ pin in a standard buck configuration is simple. To turn the regulator ON, pull the $\overline{\text{ON/OFF}}$ pin below 1.3 V (at 25°C, referenced to ground). To turn the regulator OFF, pull the $\overline{\text{ON/OFF}}$ pin above 1.3 V. With the inverting configuration, some level shifting is required, because the ground pin of the regulator is no longer at ground, but is now setting at the negative output voltage level. Two different shutdown methods for inverting regulators are shown in Figure 20 and Figure 21.

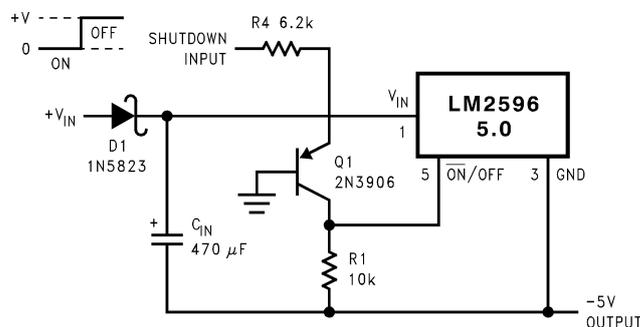


Figure 20. Inverting Regulator Ground Referenced Shutdown

Feature Description (continued)

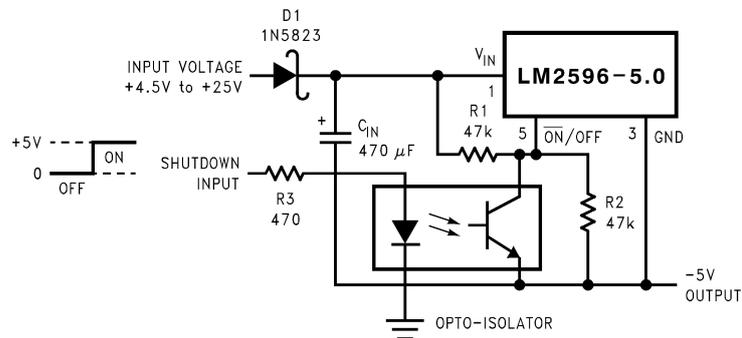


Figure 21. Inverting Regulator Ground Referenced Shutdown Using Opto Device

8.4 Device Functional Modes

8.4.1 Discontinuous Mode Operation

The selection guide chooses inductor values suitable for continuous mode operation, but for low current applications or high input voltages, a discontinuous mode design may be a better choice. A discontinuous mode design would use an inductor that would be physically smaller, and would require only one half to one third the inductance value required for a continuous mode design. The peak switch and inductor currents will be higher in a discontinuous design, but at these low load currents (1 A and below), the maximum switch current will still be less than the switch current limit.

Discontinuous operation can have voltage waveforms that are considerably different than a continuous design. The output pin (switch) waveform can have some damped sinusoidal ringing present (see [Figure 36](#)). This ringing is normal for discontinuous operation, and is not caused by feedback loop instabilities. In discontinuous operation, there is a period of time where neither the switch nor the diode are conducting, and the inductor current has dropped to zero. During this time, a small amount of energy can circulate between the inductor and the switch/diode parasitic capacitance causing this characteristic ringing. Normally this ringing is not a problem, unless the amplitude becomes great enough to exceed the input voltage, and even then, there is very little energy present to cause damage.

Different inductor types or core materials produce different amounts of this characteristic ringing. Ferrite core inductors have very little core loss and therefore produce the most ringing. The higher core loss of powdered iron inductors produce less ringing. If desired, a series RC could be placed in parallel with the inductor to dampen the ringing.

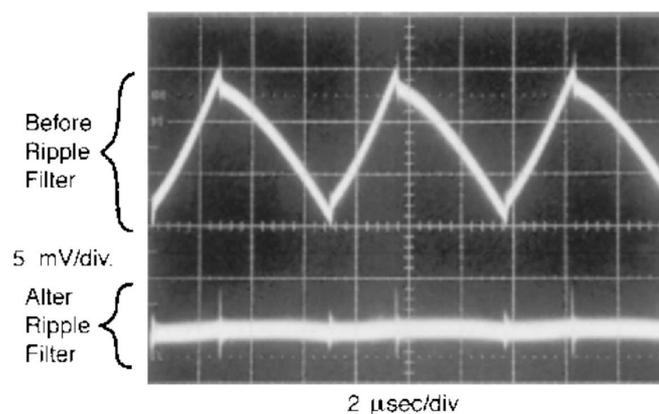


Figure 22. Post Ripple Filter Waveform

9 Application and Implementation

NOTE

Information in the following applications sections is not part of the TI component specification, and TI does not warrant its accuracy or completeness. TI's customers are responsible for determining suitability of components for their purposes. Customers should validate and test their design implementation to confirm system functionality.

9.1 Application Information

9.1.1 Input Capacitor (C_{IN})

A low ESR aluminum or tantalum bypass capacitor is required between the input pin and ground pin. It must be placed near the regulator using short leads. This capacitor prevents large voltage transients from occurring at the input, and provides the instantaneous current required each time the switch turns ON.

The important parameters for the Input capacitor are the voltage rating and the RMS current rating. Because of the relatively high RMS currents flowing in a buck regulator's input capacitor, this capacitor must be chosen for its RMS current rating rather than its capacitance or voltage ratings, although the capacitance value and voltage rating are directly related to the RMS current rating.

The RMS current rating of a capacitor could be viewed as a capacitor's power rating. The RMS current flowing through the capacitors internal ESR produces power which causes the internal temperature of the capacitor to rise. The RMS current rating of a capacitor is determined by the amount of current required to raise the internal temperature approximately 10°C above an ambient temperature of 105°C. The ability of the capacitor to dissipate this heat to the surrounding air will determine the amount of current the capacitor can safely sustain. For a given capacitor value, a higher voltage electrolytic capacitor will be physically larger than a lower voltage capacitor, and thus be able to dissipate more heat to the surrounding air, and therefore will have a higher RMS current rating.

The consequences of operating an electrolytic capacitor above the RMS current rating is a shortened operating life. The higher temperature speeds up the evaporation of the capacitor's electrolyte, resulting in eventual failure.

Selecting an input capacitor requires consulting the manufacturers data sheet for maximum allowable RMS ripple current. For a maximum ambient temperature of 40°C, a general guideline would be to select a capacitor with a ripple current rating of approximately 50% of the DC load current. For ambient temperatures up to 70°C, a current rating of 75% of the DC load current would be a good choice for a conservative design. The capacitor voltage rating must be at least 1.25 times greater than the maximum input voltage, and often a much higher voltage capacitor is required to satisfy the RMS current requirements.

[Figure 23](#) shows the relationship between an electrolytic capacitor value, its voltage rating, and the RMS current it is rated for. These curves were obtained from the Nichicon *PL* series of low-ESR, high-reliability electrolytic capacitors designed for switching regulator applications. Other capacitor manufacturers offer similar types of capacitors, but always check the capacitor data sheet.

Standard electrolytic capacitors typically have much higher ESR numbers, lower RMS current ratings and typically have a shorter operating lifetime.

Because of their small size and excellent performance, surface-mount solid tantalum capacitors are often used for input bypassing, but several precautions must be observed. A small percentage of solid tantalum capacitors can short if the inrush current rating is exceeded. This can happen at turnon when the input voltage is suddenly applied, and of course, higher input voltages produce higher inrush currents. Several capacitor manufacturers do a 100% surge current testing on their products to minimize this potential problem. If high turnon currents are expected, it may be necessary to limit this current by adding either some resistance or inductance before the tantalum capacitor, or select a higher voltage capacitor. As with aluminum electrolytic capacitors, the RMS ripple current rating must be sized to the load current.

Application Information (continued)

9.1.2 Feedforward Capacitor (C_{FF})

NOTE

For adjustable output voltage version only.

A feedforward capacitor, shown across R2 in [Table 6](#), is used when the output voltage is greater than 10 V or when C_{OUT} has a very low ESR. This capacitor adds lead compensation to the feedback loop and increases the phase margin for better loop stability. For C_{FF} selection, see the [Detailed Design Procedure](#) section.

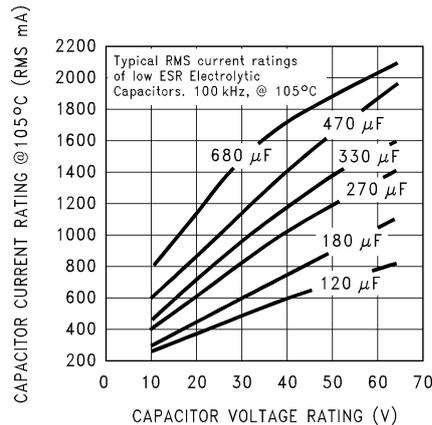


Figure 23. RMS Current Ratings for Low ESR Electrolytic Capacitors (Typical)

9.1.3 Output Capacitor (C_{OUT})

An output capacitor is required to filter the output and provide regulator loop stability. Low impedance or low-ESR electrolytic or solid tantalum capacitors designed for switching regulator applications must be used. When selecting an output capacitor, the important capacitor parameters are the 100-kHz ESR, the RMS ripple current rating, voltage rating, and capacitance value. For the output capacitor, the ESR value is the most important parameter.

The output capacitor requires an ESR value that has an upper and lower limit. For low output ripple voltage, a low ESR value is required. This value is determined by the maximum allowable output ripple voltage, typically 1% to 2% of the output voltage. But if the selected capacitor's ESR is extremely low, there is a possibility of an unstable feedback loop, resulting in an oscillation at the output. Using the capacitors listed in the tables, or similar types, will provide design solutions under all conditions.

If very low output ripple voltage (less than 15 mV) is required, see [Output Voltage Ripple and Transients](#) for a post ripple filter.

An aluminum electrolytic capacitor's ESR value is related to the capacitance value and its voltage rating. In most cases, higher voltage electrolytic capacitors have lower ESR values (see [Figure 24](#)). Often, capacitors with much higher voltage ratings may be required to provide the low ESR values required for low output ripple voltage.

The output capacitor for many different switcher designs often can be satisfied with only three or four different capacitor values and several different voltage ratings. See [Table 3](#) and [Table 4](#) for typical capacitor values, voltage ratings, and manufacturers capacitor types.

Electrolytic capacitors are not recommended for temperatures below $-25^{\circ}C$. The ESR rises dramatically at cold temperatures and is typically 3 times as large at $-25^{\circ}C$ and as much as 10 times as large at $-40^{\circ}C$. See [Figure 25](#).

Solid tantalum capacitors have a much better ESR specifications for cold temperatures and are recommended for temperatures below $-25^{\circ}C$.

Application Information (continued)

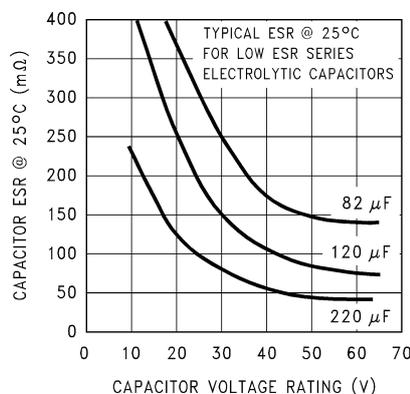


Figure 24. Capacitor ESR vs Capacitor Voltage Rating (Typical Low-ESR Electrolytic Capacitor)

9.1.4 Catch Diode

Buck regulators require a diode to provide a return path for the inductor current when the switch turns off. This must be a fast diode and must be placed close to the LM2596 using short leads and short printed-circuit traces.

Because of their very fast switching speed and low forward voltage drop, Schottky diodes provide the best performance, especially in low output voltage applications (5 V and lower). Ultra-fast recovery, or high-efficiency rectifiers are also a good choice, but some types with an abrupt turnoff characteristic may cause instability or EMI problems. Ultra-fast recovery diodes typically have reverse recovery times of 50 ns or less. Rectifiers such as the 1N5400 series are much too slow and should not be used.

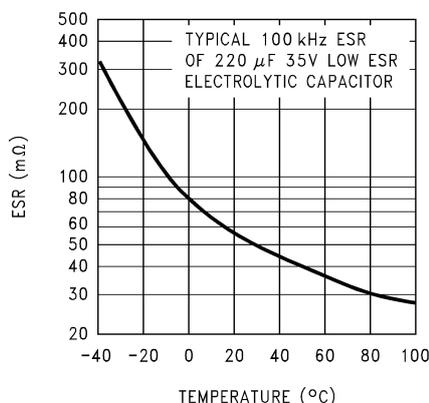


Figure 25. Capacitor ESR Change vs Temperature

9.1.5 Inductor Selection

All switching regulators have two basic modes of operation; continuous and discontinuous. The difference between the two types relates to the inductor current, whether it is flowing continuously, or if it drops to zero for a period of time in the normal switching cycle. Each mode has distinctively different operating characteristics, which can affect the regulators performance and requirements. Most switcher designs will operate in the discontinuous mode when the load current is low.

The LM2596 (or any of the SIMPLE SWITCHER™ family) can be used for both continuous or discontinuous modes of operation.

In many cases the preferred mode of operation is the continuous mode, which offers greater output power, lower peak switch, lower inductor and diode currents, and can have lower output ripple voltage. However, the continuous mode does require larger inductor values to keep the inductor current flowing continuously, especially at low output load currents or high input voltages.

Application Information (continued)

To simplify the inductor selection process, an inductor selection guide (nomograph) was designed (see [Figure 27](#) through [Figure 30](#)). This guide assumes that the regulator is operating in the continuous mode, and selects an inductor that will allow a peak-to-peak inductor ripple current to be a certain percentage of the maximum design load current. This peak-to-peak inductor ripple current percentage is not fixed, but is allowed to change as different design load currents are selected (see [Figure 26](#).)

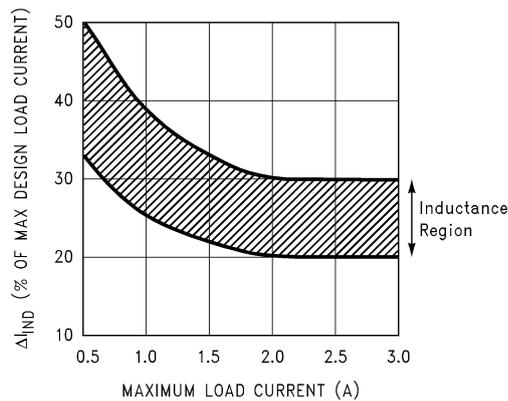


Figure 26. (ΔI_{IND}) Peak-to-Peak Inductor Ripple Current (as a Percentage of the Load Current) vs Load Current

By allowing the percentage of inductor ripple current to increase for low load currents, the inductor value and size can be kept relatively low.

When operating in the continuous mode, the inductor current waveform ranges from a triangular to a sawtooth type of waveform (depending on the input voltage), with the average value of this current waveform equal to the DC output load current.

Inductors are available in different styles such as pot core, toroid, E-core, bobbin core, and so forth, as well as different core materials, such as ferrites and powdered iron. The least expensive, the bobbin, rod or stick core, consists of wire wound on a ferrite bobbin. This type of construction makes for an inexpensive inductor, but because the magnetic flux is not completely contained within the core, it generates more Electro-Magnetic Interference (EMI). This magnetic flux can induce voltages into nearby printed-circuit traces, thus causing problems with both the switching regulator operation and nearby sensitive circuitry, and can give incorrect scope readings because of induced voltages in the scope probe (see [Open-Core Inductors](#)).

When multiple switching regulators are located on the same PCB, open-core magnetics can cause interference between two or more of the regulator circuits, especially at high currents. A toroid or E-core inductor (closed magnetic structure) should be used in these situations.

The inductors listed in the selection chart include ferrite E-core construction for Schottky, ferrite bobbin core for Renco and Coilcraft, and powdered iron toroid for Pulse Engineering.

Exceeding an inductor's maximum current rating may cause the inductor to overheat because of the copper wire losses, or the core may saturate. If the inductor begins to saturate, the inductance decreases rapidly and the inductor begins to look mainly resistive (the DC resistance of the winding). This can cause the switch current to rise very rapidly and force the switch into a cycle-by-cycle current limit, thus reducing the DC output load current. This can also result in overheating of the inductor or the LM2596. Different inductor types have different saturation characteristics, so consider this when selecting an inductor.

The inductor manufacturer's data sheets include current and energy limits to avoid inductor saturation.

For continuous mode operation, see the inductor selection graphs in [Figure 27](#) through [Figure 30](#).

Application Information (continued)

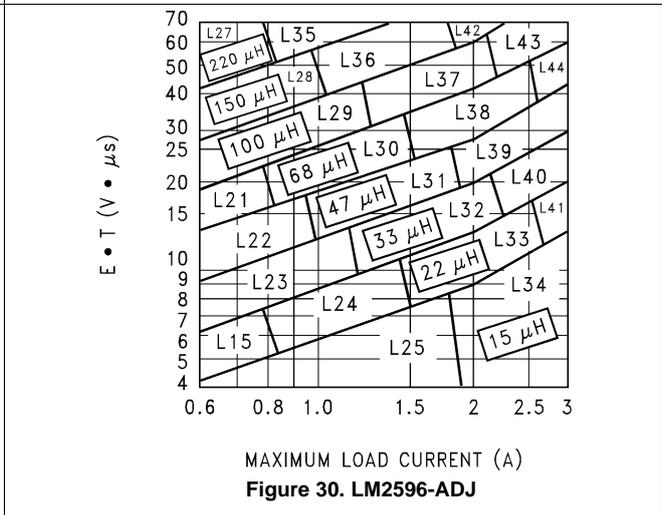
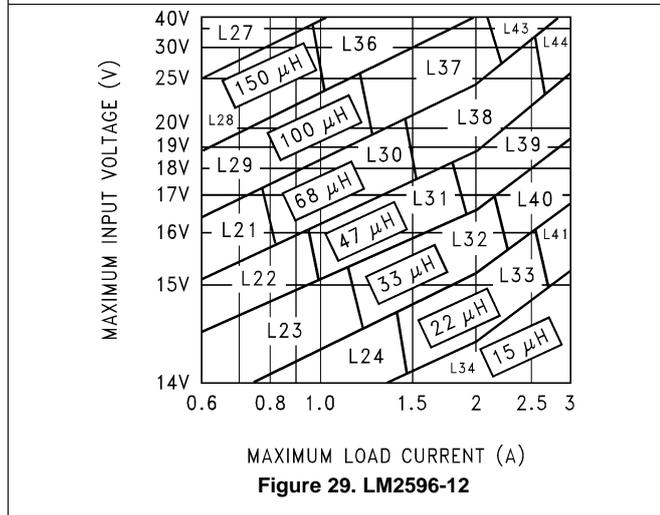
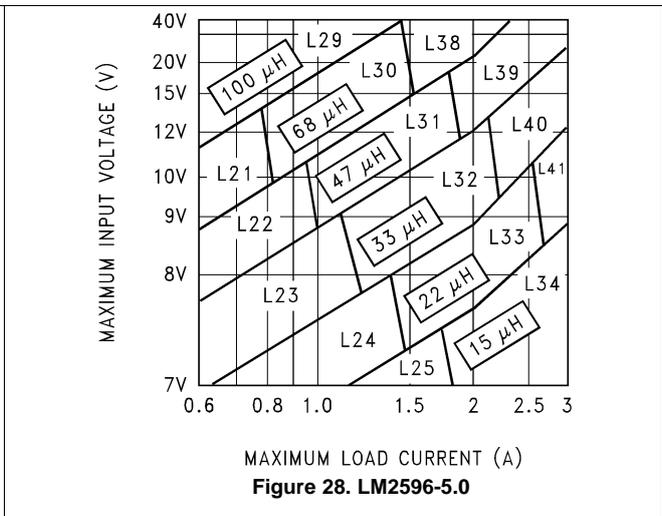
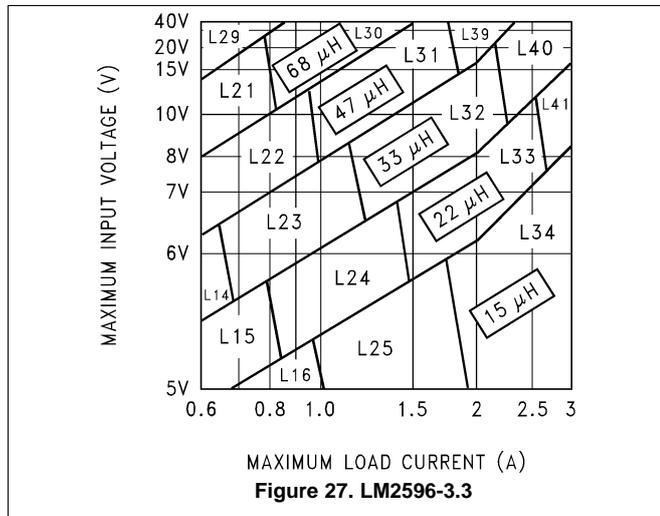


Table 1. Inductor Manufacturers Part Numbers

	INDUCTANCE (μH)	CURRENT (A)	SCHOTTKY		RENCO		PULSE ENGINEERING		COILCRAFT
			THROUGH-HOLE	SURFACE-MOUNT	THROUGH-HOLE	SURFACE-MOUNT	THROUGH-HOLE	SURFACE-MOUNT	SURFACE-MOUNT
L15	22	0.99	67148350	67148460	RL-1284-22-43	RL1500-22	PE-53815	PE-53815-S	DO3308-223
L21	68	0.99	67144070	67144450	RL-5471-5	RL1500-68	PE-53821	PE-53821-S	DO3316-683
L22	47	1.17	67144080	67144460	RL-5471-6	—	PE-53822	PE-53822-S	DO3316-473
L23	33	1.40	67144090	67144470	RL-5471-7	—	PE-53823	PE-53823-S	DO3316-333
L24	22	1.70	67148370	67148480	RL-1283-22-43	—	PE-53824	PE-53825-S	DO3316-223
L25	15	2.10	67148380	67148490	RL-1283-15-43	—	PE-53825	PE-53824-S	DO3316-153
L26	330	0.80	67144100	67144480	RL-5471-1	—	PE-53826	PE-53826-S	DO5022P-334
L27	220	1.00	67144110	67144490	RL-5471-2	—	PE-53827	PE-53827-S	DO5022P-224
L28	150	1.20	67144120	67144500	RL-5471-3	—	PE-53828	PE-53828-S	DO5022P-154
L29	100	1.47	67144130	67144510	RL-5471-4	—	PE-53829	PE-53829-S	DO5022P-104
L30	68	1.78	67144140	67144520	RL-5471-5	—	PE-53830	PE-53830-S	DO5022P-683
L31	47	2.20	67144150	67144530	RL-5471-6	—	PE-53831	PE-53831-S	DO5022P-473

Application Information (continued)
Table 1. Inductor Manufacturers Part Numbers (continued)

	INDUCTANCE (μH)	CURRENT (A)	SCHOTTKY		RENCO		PULSE ENGINEERING		COILCRAFT
			THROUGH- HOLE	SURFACE- MOUNT	THROUGH- HOLE	SURFACE- MOUNT	THROUGH- HOLE	SURFACE- MOUNT	SURFACE- MOUNT
L32	33	2.50	67144160	67144540	RL-5471-7	—	PE-53932	PE-53932-S	DO5022P-333
L33	22	3.10	67148390	67148500	RL-1283-22-43	—	PE-53933	PE-53933-S	DO5022P-223
L34	15	3.40	67148400	67148790	RL-1283-15-43	—	PE-53934	PE-53934-S	DO5022P-153
L35	220	1.70	67144170	—	RL-5473-1	—	PE-53935	PE-53935-S	—
L36	150	2.10	67144180	—	RL-5473-4	—	PE-54036	PE-54036-S	—
L37	100	2.50	67144190	—	RL-5472-1	—	PE-54037	PE-54037-S	—
L38	68	3.10	67144200	—	RL-5472-2	—	PE-54038	PE-54038-S	—
L39	47	3.50	67144210	—	RL-5472-3	—	PE-54039	PE-54039-S	—
L40	33	3.50	67144220	67148290	RL-5472-4	—	PE-54040	PE-54040-S	—
L41	22	3.50	67144230	67148300	RL-5472-5	—	PE-54041	PE-54041-S	—
L42	150	2.70	67148410	—	RL-5473-4	—	PE-54042	PE-54042-S	—
L43	100	3.40	67144240	—	RL-5473-2	—	PE-54043		—
L44	68	3.40	67144250	—	RL-5473-3	—	PE-54044		—

9.1.6 Output Voltage Ripple and Transients

The output voltage of a switching power supply operating in the continuous mode will contain a sawtooth ripple voltage at the switcher frequency, and may also contain short voltage spikes at the peaks of the sawtooth waveform.

The output ripple voltage is a function of the inductor sawtooth ripple current and the ESR of the output capacitor. A typical output ripple voltage can range from approximately 0.5% to 3% of the output voltage. To obtain low ripple voltage, the ESR of the output capacitor must be low; however, exercise caution when using extremely low ESR capacitors because they can affect the loop stability, resulting in oscillation problems. TI recommends a post ripple filter if very low output ripple voltage is required (less than 20 mV) (see [Figure 32](#)). The inductance required is typically between 1 μH and 5 μH , with low DC resistance, to maintain good load regulation. A low ESR output filter capacitor is also required to assure good dynamic load response and ripple reduction. The ESR of this capacitor may be as low as desired, because it is out of the regulator feedback loop. [Figure 22](#) shows a typical output ripple voltage, with and without a post ripple filter.

When observing output ripple with a scope, it is essential that a short, low inductance scope probe ground connection be used. Most scope probe manufacturers provide a special probe terminator which is soldered onto the regulator board, preferably at the output capacitor. This provides a very short scope ground, thus eliminating the problems associated with the 3-inch ground lead normally provided with the probe, and provides a much cleaner and more accurate picture of the ripple voltage waveform.

The voltage spikes are caused by the fast switching action of the output switch and the diode, the parasitic inductance of the output filter capacitor, and its associated wiring. To minimize these voltage spikes, the output capacitor should be designed for switching regulator applications, and the lead lengths must be kept very short. Wiring inductance, stray capacitance, as well as the scope probe used to evaluate these transients, all contribute to the amplitude of these spikes.

When a switching regulator is operating in the continuous mode, the inductor current waveform ranges from a triangular to a sawtooth type of waveform (depending on the input voltage). For a given input and output voltage, the peak-to-peak amplitude of this inductor current waveform remains constant. As the load current increases or decreases, the entire sawtooth current waveform also rises and falls. The average value (or the center) of this current waveform is equal to the DC load current.

If the load current drops to a low enough level, the bottom of the sawtooth current waveform reaches zero, and the switcher smoothly changes from a continuous to a discontinuous mode of operation. Most switcher designs (regardless of how large the inductor value is) is forced to run discontinuous if the output is lightly loaded. This is a perfectly acceptable mode of operation.

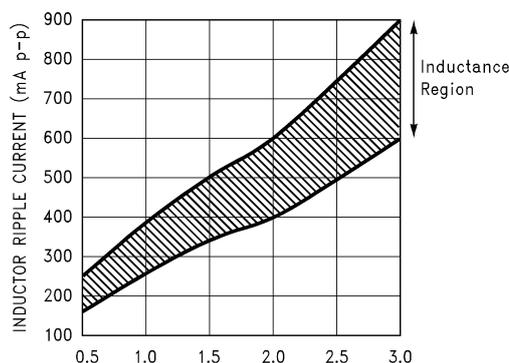


Figure 31. Peak-to-Peak Inductor Ripple Current vs Load Current

In a switching regulator design, knowing the value of the peak-to-peak inductor ripple current (ΔI_{IND}) can be useful for determining a number of other circuit parameters. Parameters such as peak inductor or peak switch current, minimum load current before the circuit becomes discontinuous, output ripple voltage, and output capacitor ESR can all be calculated from the peak-to-peak ΔI_{IND} . When the inductor nomographs in Figure 27 through Figure 30 are used to select an inductor value, the peak-to-peak inductor ripple current can immediately be determined. Figure 31 shows the range of (ΔI_{IND}) that can be expected for different load currents. Figure 31 also shows how the peak-to-peak inductor ripple current (ΔI_{IND}) changes as you go from the lower border to the upper border (for a given load current) within an inductance region. The upper border represents a higher input voltage, while the lower border represents a lower input voltage.

These curves are only correct for continuous mode operation, and only if the inductor selection guides are used to select the inductor value.

Consider the following example:

$V_{OUT} = 5\text{ V}$, maximum load current of 2.5 A

$V_{IN} = 12\text{ V}$, nominal, varying between 10 V and 16 V.

The selection guide in Figure 28 shows that the vertical line for a 2.5-A load current and the horizontal line for the 12-V input voltage intersect approximately midway between the upper and lower borders of the 33- μH inductance region. A 33- μH inductor allows a peak-to-peak inductor current (ΔI_{IND}), which is a percentage of the maximum load current, to flow. In Figure 31, follow the 2.5-A line approximately midway into the inductance region, and read the peak-to-peak inductor ripple current (ΔI_{IND}) on the left hand axis (approximately 620 mA p-p).

As the input voltage increases to 16 V, approaching the upper border of the inductance region, the inductor ripple current increases. Figure 31 shows that for a load current of 2.5 A, the peak-to-peak inductor ripple current (ΔI_{IND}) is 620 mA with 12 V_{IN} , and can range from 740 mA at the upper border (16 V_{IN}) to 500 mA at the lower border (10 V_{IN}).

Once the ΔI_{IND} value is known, use these equations to calculate additional information about the switching regulator circuit.

1. Peak Inductor or peak switch current = $\left(I_{LOAD} + \frac{\Delta I_{IND}}{2} \right) = \left(2.5\text{A} + \frac{0.62}{2} \right) = 2.81\text{A}$
2. Minimum load current before the circuit becomes discontinuous = $\frac{\Delta I_{IND}}{2} = \frac{0.62}{2} = 0.31\text{A}$
3. Output Ripple Voltage = $(\Delta I_{IND}) \times (\text{ESR of } C_{OUT}) = 0.62\text{ A} \times 0.1\ \Omega = 62\text{ mVp-p}$
- 4.

$$\begin{aligned} \text{ESR of } C_{OUT} &= \frac{\text{Output Ripple Voltage } (\Delta V_{OUT})}{\Delta I_{IND}} \\ &= \frac{0.062\text{V}}{0.62\text{A}} = 0.1\ \Omega \end{aligned}$$

9.1.7 Open-Core Inductors

Another possible source of increased output ripple voltage or unstable operation is from an open-core inductor. Ferrite bobbin or stick inductors have magnetic lines of flux flowing through the air from one end of the bobbin to the other end. These magnetic lines of flux will induce a voltage into any wire or PCB copper trace that comes within the inductor's magnetic field. The strength of the magnetic field, the orientation and location of the PCB copper trace to the magnetic field, and the distance between the copper trace and the inductor determine the amount of voltage generated in the copper trace. Another way of looking at this inductive coupling is to consider the PCB copper trace as one turn of a transformer (secondary) with the inductor winding as the primary. Many millivolts can be generated in a copper trace located near an open-core inductor, which can cause stability problems or high output ripple voltage problems.

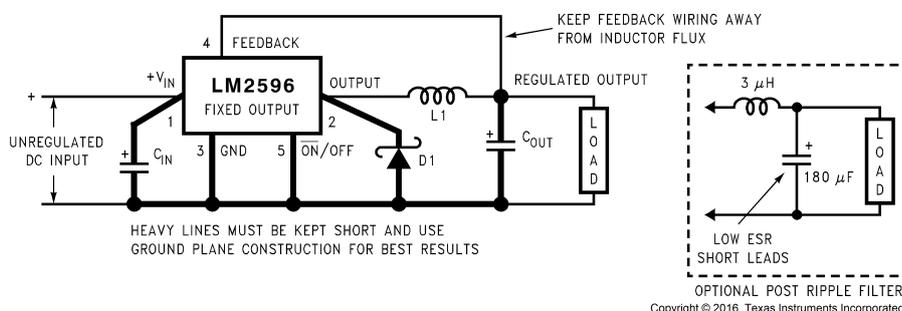
If unstable operation is seen, and an open-core inductor is used, it is possible that the location of the inductor with respect to other PC traces may be the problem. To determine if this is the problem, temporarily raise the inductor away from the board by several inches and then check circuit operation. If the circuit now operates correctly, then the magnetic flux from the open core inductor is causing the problem. Substituting a closed core inductor such as a torroid or E-core will correct the problem, or re-arranging the PC layout may be necessary. Magnetic flux cutting the IC device ground trace, feedback trace, or the positive or negative traces of the output capacitor should be minimized.

Sometimes, placing a trace directly beneath a bobbin inductor will provide good results, provided it is exactly in the center of the inductor (because the induced voltages cancel themselves out). However, problems could arise if the trace is off center one direction or the other. If flux problems are present, even the direction of the inductor winding can make a difference in some circuits.

This discussion on open core inductors is not to frighten users, but to alert users on what kind of problems to watch out for. Open-core bobbin or *stick* inductors are an inexpensive, simple way of making a compact, efficient inductor, and they are used by the millions in many different applications.

9.2 Typical Applications

9.2.1 LM2596 Fixed Output Series Buck Regulator



C_{IN} — 470- μ F, 50-V, Aluminum Electrolytic Nichicon *PL Series*
 C_{OUT} — 220- μ F, 25-V Aluminum Electrolytic, Nichicon *PL Series*
 D1 — 5-A, 40-V Schottky Rectifier, 1N5825
 L1 — 68 μ H, L38

Figure 32. Fixed Output Voltage Version

9.2.1.1 Design Requirements

Table 2 lists the design parameters for this example.

Table 2. Design Parameters

PARAMETER	EXAMPLE VALUE
Regulated Output Voltage (3.3 V, 5 V or 12 V), V_{OUT}	5 V
Maximum DC Input Voltage, $V_{IN(max)}$	12 V
Maximum Load Current, $I_{LOAD(max)}$	3 A

9.2.1.2 Detailed Design Procedure

9.2.1.2.1 Inductor Selection (L1)

1. Select the correct inductor value selection guide from [Figure 27](#), [Figure 28](#), or [Figure 29](#) (output voltages of 3.3V, 5V, or 12V respectively). Use the inductor selection guide for the 5-V version shown in [Figure 28](#).
2. From the inductor value selection guide, identify the inductance region intersected by the maximum input voltage line and the maximum load current line. Each region is identified by an inductance value and an inductor code (LXX). From the inductor value selection guide shown in [Figure 28](#), the inductance region intersected by the 12-V horizontal line and the 3-A vertical line is 33 μ H, and the inductor code is L40.
3. Select an appropriate inductor from the four manufacturer's part numbers listed in [Table 1](#). The inductance value required is 33 μ H. See row L40 of [Table 1](#) and choose an inductor part number from any of the manufacturers shown. In most instances, both through-hole and surface-mount inductors are available.

9.2.1.2.2 Output Capacitor Selection (C_{OUT})

1. In the majority of applications, low ESR (Equivalent Series Resistance) electrolytic capacitors between 82 μ F and 820 μ F and low ESR solid tantalum capacitors between 10 μ F and 470 μ F provide the best results. This capacitor must be placed close to the IC using short capacitor leads and short copper traces. Do not use capacitors larger than 820 μ F.

NOTE

For additional information, see section on output capacitors in [Table 3](#).

2. To simplify the capacitor selection procedure, see [Table 3](#) for quick design component selection. This table contains different input voltages, output voltages, and load currents, and lists various inductors and output capacitors that will provide the best design solutions.

From [Table 3](#), locate the 5-V output voltage section. In the load current column, choose the load current line that is closest to the current required for the application; for this example, use the 3-A line. In the maximum input voltage column, select the line that covers the input voltage required for the application; in this example, use the 15-V line. The rest of the line shows recommended inductors and capacitors that will provide the best overall performance.

Table 3. LM2596 Fixed Voltage Quick Design Component Selection Table

CONDITIONS			INDUCTOR		OUTPUT CAPACITOR			
OUTPUT VOLTAGE (V)	LOAD CURRENT (A)	MAX INPUT VOLTAGE (V)	INDUCTANCE (μ H)	INDUCTOR (#)	THROUGH-HOLE ELECTROLYTIC		SURFACE-MOUNT TANTALUM	
					PANASONIC HFQ SERIES (μ F/V)	NICHICON PL SERIES (μ F/V)	AVX TPS SERIES (μ F/V)	SPRAGUE 595D SERIES (μ F/V)
3.3	3	5	22	L41	470/25	560/16	330/6.3	390/6.3
		7	22	L41	560/35	560/35	330/6.3	390/6.3
		10	22	L41	680/35	680/35	330/6.3	390/6.3
	2	40	33	L40	560/35	470/35	330/6.3	390/6.3
		6	22	L33	470/25	470/35	330/6.3	390/6.3
		10	33	L32	330/35	330/35	330/6.3	390/6.3
5	3	40	47	L39	330/35	270/50	220/10	330/10
		8	22	L41	470/25	560/16	220/10	330/10
		10	22	L41	560/25	560/25	220/10	330/10
		15	33	L40	330/35	330/35	220/10	330/10
	2	40	47	L39	330/35	270/35	220/10	330/10
		9	22	L33	470/25	560/16	220/10	330/10
		20	68	L38	180/35	180/35	100/10	270/10
		40	68	L38	180/35	180/35	100/10	270/10

Table 3. LM2596 Fixed Voltage Quick Design Component Selection Table (continued)

CONDITIONS			INDUCTOR		OUTPUT CAPACITOR			
OUTPUT VOLTAGE (V)	LOAD CURRENT (A)	MAX INPUT VOLTAGE (V)	INDUCTANCE (μH)	INDUCTOR (#)	THROUGH-HOLE ELECTROLYTIC		SURFACE-MOUNT TANTALUM	
					PANASONIC HFQ SERIES (μF/V)	NICHICON PL SERIES (μF/V)	AVX TPS SERIES (μF/V)	SPRAGUE 595D SERIES (μF/V)
12	3	15	22	L41	470/25	470/25	100/16	180/16
		18	33	L40	330/25	330/25	100/16	180/16
		30	68	L44	180/25	180/25	100/16	120/20
		40	68	L44	180/35	180/35	100/16	120/20
	2	15	33	L32	330/25	330/25	100/16	180/16
		20	68	L38	180/25	180/25	100/16	120/20
		40	150	L42	82/25	82/25	68/20	68/25

The capacitor list contains both through-hole electrolytic and surface-mount tantalum capacitors from four different capacitor manufacturers. TI recommends that both the manufacturers and the manufacturer's series that are listed in [Table 3](#).

In this example aluminum electrolytic capacitors from several different manufacturers are available with the range of ESR numbers required.

- 330-μF, 35-V Panasonic HFQ Series
- 330-μF, 35-V Nichicon PL Series

3. The capacitor voltage rating for electrolytic capacitors should be at least 1.5 times greater than the output voltage, and often require much higher voltage ratings to satisfy the low ESR requirements for low output ripple voltage.

For a 5-V output, a capacitor voltage rating of at least 7.5 V is required. But even a low ESR, switching grade, 220-μF, 10-V aluminum electrolytic capacitor would exhibit approximately 225 mΩ of ESR (see [Figure 24](#) for the ESR vs voltage rating). This amount of ESR would result in relatively high output ripple voltage. To reduce the ripple to 1% or less of the output voltage, a capacitor with a higher value or with a higher voltage rating (lower ESR) must be selected. A 16-V or 25-V capacitor will reduce the ripple voltage by approximately half.

9.2.1.2.3 Catch Diode Selection (D1)

1. The catch diode current rating must be at least 1.3 times greater than the maximum load current. Also, if the power supply design must withstand a continuous output short, the diode must have a current rating equal to the maximum current limit of the LM2596. The most stressful condition for this diode is an overload or shorted output condition. See [Table 4](#). In this example, a 5-A, 20-V, 1N5823 Schottky diode will provide the best performance, and will not be overstressed even for a shorted output.

Table 4. Diode Selection Table

VR	3-A DIODES				4-A TO 6-A DIODES			
	SURFACE-MOUNT		THROUGH-HOLE		SURFACE-MOUNT		THROUGH-HOLE	
	SCHOTTKY	ULTRA FAST RECOVERY	SCHOTTKY	ULTRA FAST RECOVERY	SCHOTTKY	ULTRA FAST RECOVERY	SCHOTTKY	ULTRA FAST RECOVERY
20 V		All of these diodes are rated to at least 50V.	1N5820	All of these diodes are rated to at least 50V.		All of these diodes are rated to at least 50V.	SR502	All of these diodes are rated to at least 50V.
	SK32		SR302				1N5823	
			MBR320				SB520	
30 V	30WQ03		1N5821				SR503	
	SK33		MBR330		50WQ03		1N5824	
			31DQ03				SB530	
40 V			1N5822				SR504	
	SK34		SR304		50WQ04		1N5825	
	MBRS340		MBR340				SB540	
	30WQ04	MURS320	31DQ04	MUR320		MURS620	SB540	MUR620
50 V	SK35	30WF10	SR305			50WF10		HER601
	or		MBR350		50WQ05		SB550	
More	30WQ05		31DQ05				50SQ080	

2. The reverse voltage rating of the diode must be at least 1.25 times the maximum input voltage.
3. This diode must be fast (short reverse recovery time) and must be placed close to the LM2596 using short leads and short-printed circuit traces. Because of their fast switching speed and low forward voltage drop, Schottky diodes provide the best performance and efficiency, and must be the first choice, especially in low output voltage applications. Ultra-fast recovery, or high-efficiency rectifiers also provide good results. Ultra-fast recovery diodes typically have reverse recovery times of 50 ns or less. Rectifiers such as the 1N5400 series must not be used because they are too slow.

9.2.1.2.4 Input Capacitor (C_{IN})

A low ESR aluminum or tantalum bypass capacitor is required between the input pin and ground pin to prevent large voltage transients from appearing at the input. This capacitor must be placed close to the IC using short leads. In addition, the RMS current rating of the input capacitor should be selected to be at least $\frac{1}{2}$ the DC load current. The capacitor manufacturers data sheet must be checked to assure that this current rating is not exceeded. [Figure 23](#) shows typical RMS current ratings for several different aluminum electrolytic capacitor values.

For an aluminum electrolytic, the capacitor voltage rating must be approximately 1.5 times the maximum input voltage. Exercise caution if solid tantalum capacitors are used (see [Input Capacitor \(\$C_{IN}\$ \)](#)). The tantalum capacitor voltage rating should be 2 times the maximum input voltage and TI recommends that they be surge current tested by the manufacturer.

Use caution when using ceramic capacitors for input bypassing, because it may cause severe ringing at the V_{IN} pin.

The important parameters for the Input capacitor are the input voltage rating and the RMS current rating. With a nominal input voltage of 12 V, an aluminum electrolytic capacitor with a voltage rating greater than 18 V ($1.5 \times V_{IN}$) is necessary. The next higher capacitor voltage rating is 25 V.

The RMS current rating requirement for the input capacitor in a buck regulator is approximately $\frac{1}{2}$ the DC load current. In this example, with a 3-A load, a capacitor with a RMS current rating of at least 1.5 A is required. [Figure 23](#) can be used to select an appropriate input capacitor. From the curves, locate the 35-V line and note which capacitor values have RMS current ratings greater than 1.5 A. A 680- μ F, 35-V capacitor could be used.

For a through-hole design, a 680- μ F, 35-V electrolytic capacitor (Panasonic HFQ series or Nichicon PL series or equivalent) would be adequate. Other types or other manufacturers' capacitors can be used provided the RMS ripple current ratings are adequate.

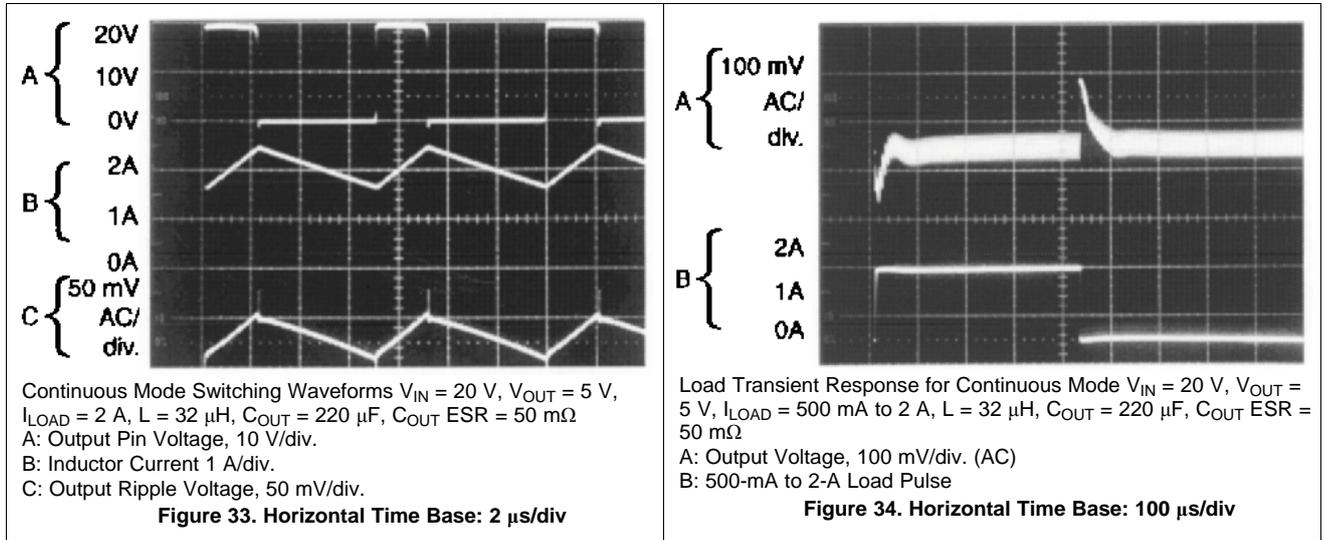
For surface-mount designs, solid tantalum capacitors can be used, but exercise caution with regard to the capacitor surge current rating (see [Input Capacitor \(\$C_{IN}\$ \)](#) in this data sheet). The TPS series available from AVX, and the 593D series from Sprague are both surge current tested.

LM2596

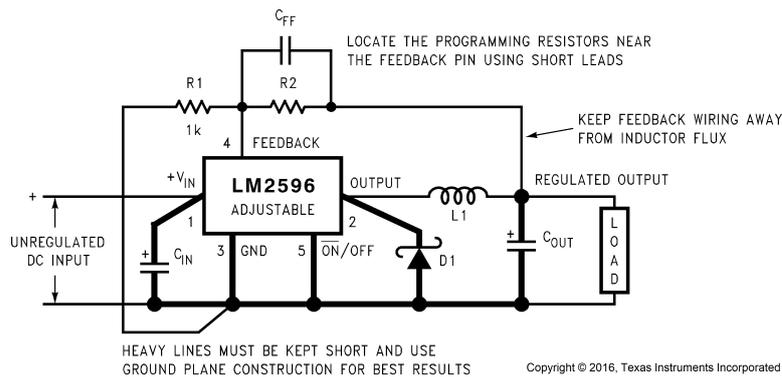
SNVS124D – NOVEMBER 1999 – REVISED MAY 2016

www.ti.com

9.2.1.3 Application Curves



9.2.2 LM2596 Adjustable Output Series Buck Regulator



$$V_{OUT} = V_{REF} \left(1 + \frac{R_2}{R_1} \right)$$

where $V_{REF} = 1.23$ V

$$R_2 = R_1 \left(\frac{V_{OUT}}{V_{REF}} - 1 \right)$$

Select R_1 to be approximately 1 k Ω , use a 1% resistor for best stability.

C_{IN} — 470- μ F, 50-V, Aluminum Electrolytic Nichicon *PL Series*

C_{OUT} — 220- μ F, 35-V Aluminum Electrolytic, Nichicon *PL Series*

D1 — 5-A, 40-V Schottky Rectifier, 1N5825

L1 — 68 μ H, L38

R1 — 1 k Ω , 1%

C_{FF} — See [Feedforward Capacitor \(\$C_{FF}\$ \)](#)

Figure 35. Adjustable Output Voltage Version

9.2.2.1 Design Requirements

Table 5 lists the design parameters for this example.

Table 5. Design Parameters

PARAMETER	EXAMPLE VALUE
Regulated output voltage (3.3V, 5V or 12V), V_{OUT}	20 V
Maximum DC input voltage, $V_{IN(max)}$	28 V
Maximum load current, $I_{LOAD(max)}$	3 A
Switching frequency, F	Fixed at a nominal 150 kHz

9.2.2.2 Detailed Design Procedure

9.2.2.2.1 Custom Design with WEBENCH Tools

Click [here](#) to create a custom design using the LM2596 device with the WEBENCH® Power Designer.

1. Start by entering your V_{IN} , V_{OUT} and I_{OUT} requirements.
2. Optimize your design for key parameters like efficiency, footprint and cost using the optimizer dial and compare this design with other possible solutions from Texas Instruments.
3. WEBENCH Power Designer provides you with a customized schematic along with a list of materials with real time pricing and component availability.
4. In most cases, you will also be able to:
 - Run electrical simulations to see important waveforms and circuit performance,
 - Run thermal simulations to understand the thermal performance of your board,
 - Export your customized schematic and layout into popular CAD formats,
 - Print PDF reports for the design, and share your design with colleagues.
5. Get more information about WEBENCH tools at www.ti.com/webench.

9.2.2.2.2 Programming Output Voltage

Select R_1 and R_2 , as shown in Table 6

Use Equation 1 to select the appropriate resistor values.

$$V_{OUT} = V_{REF} \left(1 + \frac{R_2}{R_1} \right) \quad \text{where } V_{REF} = 1.23V \quad (1)$$

Select a value for R_1 between 240 Ω and 1.5 k Ω . The lower resistor values minimize noise pickup in the sensitive feedback pin. (For the lowest temperature coefficient and the best stability with time, use 1% metal film resistors.). Calculate R_2 with Equation 2.

$$R_2 = R_1 \left(\frac{V_{OUT}}{V_{REF}} - 1 \right) \quad (2)$$

Select R_1 to be 1 k Ω , 1%. Solve for R_2 in Equation 3.

$$R_2 = R_1 \left(\frac{V_{OUT}}{V_{REF}} - 1 \right) = 1k \left(\frac{20V}{1.23V} - 1 \right) \quad (3)$$

$R_2 = 1k (16.26 - 1) = 15.26k$, closest 1% value is 15.4 k Ω .

$R_2 = 15.4 \text{ k}\Omega$.

9.2.2.2.3 Inductor Selection (L1)

1. Calculate the inductor Volt • microsecond constant $E \times T$ ($V \times \mu s$), with Equation 4:

$$E \cdot T = (V_{IN} - V_{OUT} - V_{SAT}) \cdot \frac{V_{OUT} + V_D}{V_{IN} - V_{SAT} + V_D} \cdot \frac{1000}{150 \text{ kHz}} \quad (V \cdot \mu s)$$

where

- V_{SAT} = internal switch saturation voltage = 1.16 V
 - V_D = diode forward voltage drop = 0.5 V
- (4)

Calculate the inductor Volt • microsecond constant

(E × T),

$$E \cdot T = (28 - 20 - 1.16) \cdot \frac{20 + 0.5}{28 - 1.16 + 0.5} \cdot \frac{1000}{150} (V \cdot \mu s)$$

$$E \cdot T = (6.84) \cdot \frac{20.5}{27.34} \cdot 6.67 (V \cdot \mu s) = 34.2 (V \cdot \mu s)$$

(5)

- Use the E × T value from the previous formula and match it with the E × T number on the vertical axis of the Inductor Value Selection Guide shown in [Figure 30](#).

$$E \times T = 34.2 (V \times \mu s)$$

- On the horizontal axis, select the maximum load current.

$$I_{LOAD(max)} = 3 A$$

- Identify the inductance region intersected by the E × T value and the maximum load current value. Each region is identified by an inductance value and an inductor code (LXX). From the inductor value selection guide shown in [Figure 30](#), the inductance region intersected by the 34 (V • μs) horizontal line and the 3-A vertical line is 47 μH, and the inductor code is L39.
- Select an appropriate inductor from the manufacturers' part numbers listed in [Table 1](#). From the table in [Table 1](#), locate line L39, and select an inductor part number from the list of manufacturers part numbers.

9.2.2.2.4 Output Capacitor Selection (C_{OUT})

- In the majority of applications, low ESR electrolytic or solid tantalum capacitors between 82 μF and 820 μF provide the best results. This capacitor must be placed close to the IC using short capacitor leads and short copper traces. Do not use capacitors larger than 820 μF.

NOTE

For additional information, see section on output capacitors in [Output Capacitor \(C_{OUT}\)](#) section.

- To simplify the capacitor selection procedure, see [Table 6](#) for a quick design guide. This table contains different output voltages, and lists various output capacitors that will provide the best design solutions.

From [Table 6](#), locate the output voltage column. From that column, locate the output voltage closest to the output voltage in your application. In this example, select the 24-V line. Under the [Output Capacitor \(C_{OUT}\)](#) section, select a capacitor from the list of through-hole electrolytic or surface-mount tantalum types from four different capacitor manufacturers. TI recommends that both the manufacturers and the manufacturers' series that are listed in [Table 6](#) be used.

In this example, through hole aluminum electrolytic capacitors from several different manufacturers are available.

- 220-μF, 35-V Panasonic HFQ Series
- 150-μF, 35-V Nichicon PL Series

- The capacitor voltage rating must be at least 1.5 times greater than the output voltage, and often much higher voltage ratings are required to satisfy the low ESR requirements required for low output ripple voltage.

For a 20-V output, a capacitor rating of at least 30 V is required. In this example, either a 35-V or 50-V capacitor would work. A 35-V rating was chosen, although a 50-V rating could also be used if a lower output ripple voltage is required.

Other manufacturers or other types of capacitors may also be used, provided the capacitor specifications (especially the 100-kHz ESR) closely match the types listed in [Table 6](#). Refer to the capacitor manufacturers data sheet for this information.

9.2.2.2.5 Feedforward Capacitor (C_{FF})

See [Table 6](#).

For output voltages greater than approximately 10 V, an additional capacitor is required. The compensation capacitor is typically between 100 pF and 33 nF, and is wired in parallel with the output voltage setting resistor, R₂. It provides additional stability for high output voltages, low input or output voltages, or very low ESR output capacitors, such as solid tantalum capacitors. Calculate the value for C_{FF} with [Equation 6](#):

$$C_{FF} = \frac{1}{31 \times 10^3 \times R_2} \quad (6)$$

This capacitor type can be ceramic, plastic, silver mica, and so forth. Because of the unstable characteristics of ceramic capacitors made with Z5U material, they are not recommended.

Table 6 contains feedforward capacitor values for various output voltages. In this example, a 560-pF capacitor is required.

Table 6. Output Capacitor and Feedforward Capacitor Selection Table

OUTPUT VOLTAGE (V)	THROUGH-HOLE OUTPUT CAPACITOR			SURFACE-MOUNT OUTPUT CAPACITOR		
	PANASONIC HFQ SERIES (μF/V)	NICHICON PL SERIES (μF/V)	FEEDFORWARD CAPACITOR	AVX TPS SERIES (μF/V)	SPRAGUE 595D SERIES (μF/V)	FEEDFORWARD CAPACITOR
2	820/35	820/35	33 nF	330/6.3	470/4	33 nF
4	560/35	470/35	10 nF	330/6.3	390/6.3	10 nF
6	470/25	470/25	3.3 nF	220/10	330/10	3.3 nF
9	330/25	330/25	1.5 nF	100/16	180/16	1.5 nF
12	330/25	330/25	1 nF	100/16	180/16	1 nF
15	220/35	220/35	680 pF	68/20	120/20	680 pF
24	220/35	150/35	560 pF	33/25	33/25	220 pF
28	100/50	100/50	390 pF	10/35	15/50	220 pF

9.2.2.2.6 Catch Diode Selection (D1)

1. The catch diode current rating must be at least 1.3 times greater than the maximum load current. Also, if the power supply design must withstand a continuous output short, the diode must have a current rating equal to the maximum current limit of the LM2596. The most stressful condition for this diode is an overload or shorted output condition. See Table 4. Schottky diodes provide the best performance, and in this example a 5-A, 40-V, 1N5825 Schottky diode would be a good choice. The 5-A diode rating is more than adequate and will not be overstressed even for a shorted output.
2. The reverse voltage rating of the diode must be at least 1.25 times the maximum input voltage.
3. This diode must be fast (short reverse recovery time) and must be placed close to the LM2596 using short leads and short-printed circuit traces. Because of their fast switching speed and low forward voltage drop, Schottky diodes provide the best performance and efficiency, and must be the first choice, especially in low output voltage applications. Ultra-fast recovery or high-efficiency rectifiers are also good choices, but some types with an abrupt turnoff characteristic may cause instability or EMI problems. Ultra-fast recovery diodes typically have reverse recovery times of 50 ns or less. Rectifiers such as the 1N4001 series must not be used because they are too slow.

9.2.2.2.7 Input Capacitor (C_{IN})

A low ESR aluminum or tantalum bypass capacitor is required between the input pin and ground to prevent large voltage transients from appearing at the input. In addition, the RMS current rating of the input capacitor should be selected to be at least ½ the DC load current. The capacitor manufacturers data sheet must be checked to assure that this current rating is not exceeded. Figure 23 shows typical RMS current ratings for several different aluminum electrolytic capacitor values.

This capacitor must be placed close to the IC using short leads and the voltage rating must be approximately 1.5 times the maximum input voltage.

If solid tantalum input capacitors are used, TI recommends that they be surge current tested by the manufacturer.

Use caution when using a high dielectric constant ceramic capacitor for input bypassing, because it may cause severe ringing at the V_{IN} pin.

The important parameters for the input capacitor are the input voltage rating and the RMS current rating. With a nominal input voltage of 28 V, an aluminum electrolytic capacitor with a voltage rating greater than 42 V ($1.5 \times V_{IN}$) is required. Because the next higher capacitor voltage rating is 50 V, a 50-V capacitor must be used. The capacitor voltage rating of ($1.5 \times V_{IN}$) is a conservative guideline, and can be modified somewhat if desired.

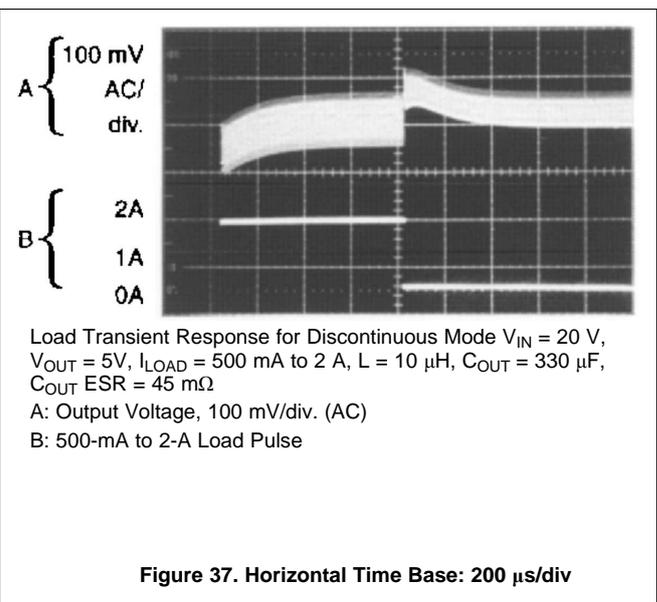
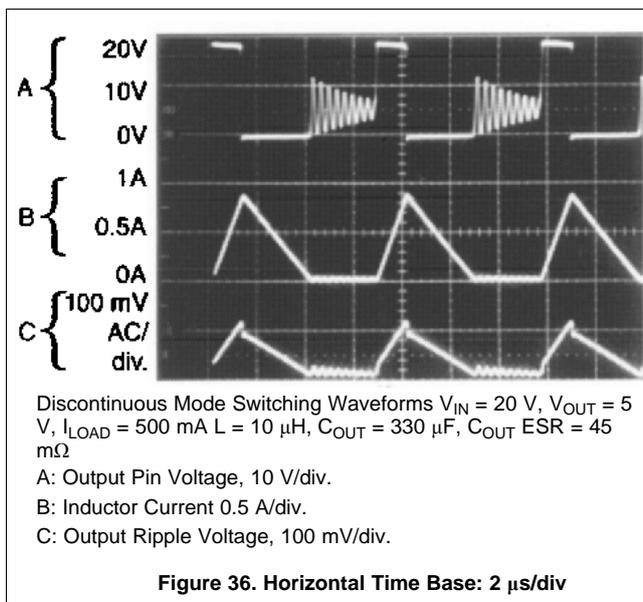
The RMS current rating requirement for the input capacitor of a buck regulator is approximately $\frac{1}{2}$ the DC load current. In this example, with a 3-A load, a capacitor with a RMS current rating of at least 1.5 A is required.

Figure 23 can be used to select an appropriate input capacitor. From the curves, locate the 50-V line and note which capacitor values have RMS current ratings greater than 1.5 A. Either a 470 μF or 680 μF , 50-V capacitor could be used.

For a through hole design, a 680- μF , 50-V electrolytic capacitor (Panasonic HFQ series or Nichicon PL series or equivalent) would be adequate. Other types or other manufacturers' capacitors can be used provided the RMS ripple current ratings are adequate.

For surface mount designs, solid tantalum capacitors can be used, but exercise caution with regard to the capacitor surge current rating (see *Input Capacitor (C_{IN})* in this data sheet). The TPS series available from AVX, and the 593D series from Sprague are both surge current tested.

9.2.2.3 Application Curves



10 Power Supply Recommendations

The LM2596 is designed to operate from an input voltage supply up to 40 V. This input supply must be well regulated and able to withstand maximum input current and maintain a stable voltage.

11 Layout

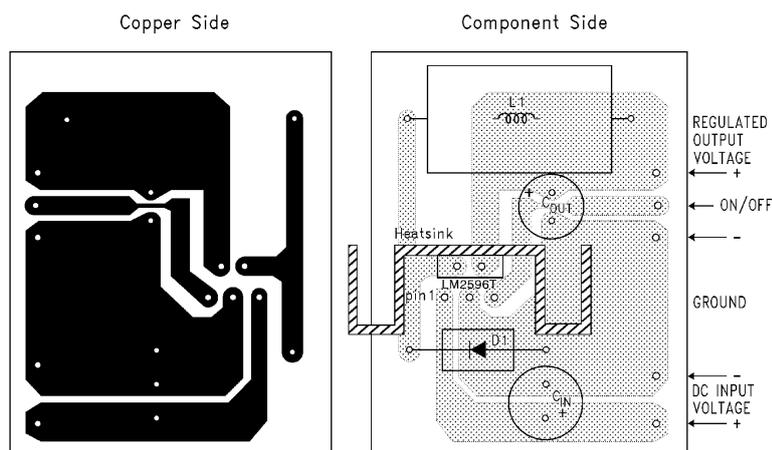
11.1 Layout Guidelines

As in any switching regulator, layout is very important. Rapidly switching currents associated with wiring inductance can generate voltage transients which can cause problems. For minimal inductance and ground loops, the wires indicated by heavy lines must be wide printed-circuit traces and must be kept as short as possible. For best results, external components must be placed as close to the switcher IC as possible using ground plane construction or single point grounding.

If open core inductors are used, take special care selecting the location and positioning of this type of inductor. Allowing the inductor flux to intersect sensitive feedback, IC groundpath and C_{OUT} wiring can cause problems.

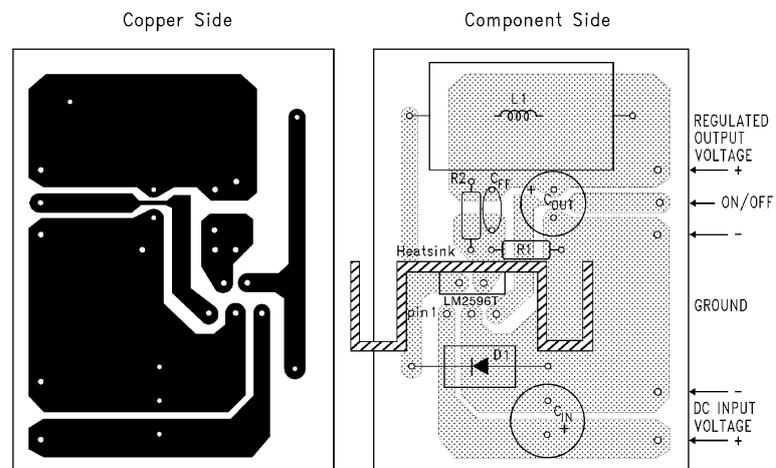
When using the adjustable version, take special care selecting the location of the feedback resistors and the associated wiring. Physically place both resistors near the IC, and route the wiring away from the inductor, especially an open-core type of inductor (see [Open-Core Inductors](#) for more information).

11.2 Layout Examples



C_{IN} — 470- μ F, 50-V, Aluminum Electrolytic Panasonic, *HFQ Series*
 C_{OUT} — 330- μ F, 35-V, Aluminum Electrolytic Panasonic, *HFQ Series*
D1 — 5-A, 40-V Schottky Rectifier, 1N5825
L1 — 47- μ H, L39, Renco, Through Hole
Thermalloy Heat Sink #7020

Figure 38. Typical Through-Hole PCB Layout, Fixed Output (1x Size), Double-Sided

Layout Examples (continued)


C_{IN} — 470- μ F, 50-V, Aluminum Electrolytic Panasonic, *HFQ Series*
 C_{OUT} —220- μ F, 35-V Aluminum Electrolytic Panasonic, *HFQ Series*
 D1—5-A, 40-V Schottky Rectifier, 1N5825
 L1—47- μ H, L39, Renco, Through Hole
 R_1 —1 k Ω , 1%
 R_2 —Use formula in Design Procedure
 C_{FF} —See [Table 6](#).
 Thermalloy Heat Sink #7020

Figure 39. Typical Through-Hole PCB Layout, Adjustable Output (1x Size), Double-Sided

11.3 Thermal Considerations

The LM2596 is available in two packages: a 7-pin TO-220 (T) and a 7-pin surface mount TO-263 (S).

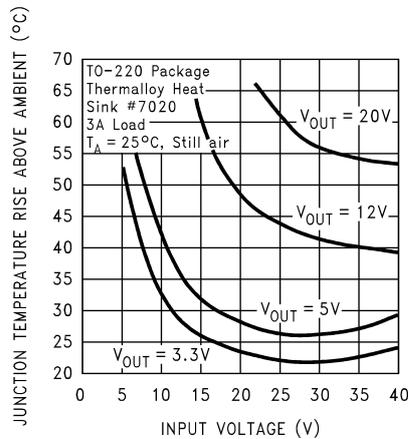
The TO-220 package requires a heat sink under most conditions. The size of the heat sink depends on the input voltage, the output voltage, the load current and the ambient temperature. [Figure 40](#) shows the LM2596T junction temperature rises above ambient temperature for a 3-A load and different input and output voltages. The data for these curves was taken with the LM2596T (TO-220 package) operating as a buck switching regulator in an ambient temperature of 25°C (still air). These temperature rise numbers are all approximate and there are many factors that can affect these temperatures. Higher ambient temperatures require more heat sinking.

The TO-263 surface mount package tab is designed to be soldered to the copper on a printed-circuit board (PCB). The copper and the board are the heat sink for this package and the other heat producing components, such as the catch diode and inductor. The PCB copper area that the package is soldered to must be at least 0.4 in², and ideally must have 2 or more square inches of 2-oz. (0.0028 in.) copper. Additional copper area improves the thermal characteristics, but with copper areas greater than approximately 6 in², only small improvements in heat dissipation are realized. If further thermal improvements are required, TI recommends double-sided, multilayer PCB with large copper areas and airflow.

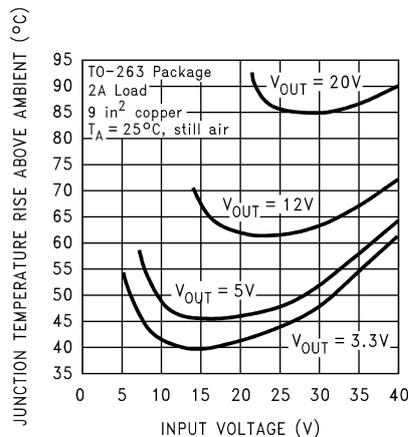
[Figure 41](#) shows the LM2596S (TO-263 package) junction temperature rise above ambient temperature with a 2-A load for various input and output voltages. This data was taken with the circuit operating as a buck switching regulator with all components mounted on a PCB to simulate the junction temperature under actual operating conditions. This curve can be used for a quick check for the approximate junction temperature for various conditions, but be aware that there are many factors that can affect the junction temperature. When load currents higher than 2 A are used, double-sided or multilayer PCB with large copper areas or airflow might be required, especially for high ambient temperatures and high output voltages.

For the best thermal performance, wide copper traces and generous amounts of PCB copper must be used in the board layout. (One exception to this is the output (switch) pin, which should **not** have large areas of copper.) Large areas of copper provide the best transfer of heat (lower thermal resistance) to the surrounding air, and moving air lowers the thermal resistance even further.

Package thermal resistance and junction temperature rise numbers are all approximate, and there are many factors that will affect these numbers. Some of these factors include board size, shape, thickness, position, location, and even board temperature. Other factors are trace width, total printed-circuit copper area, copper thickness, single- or double-sided multilayer board, and the amount of solder on the board. The effectiveness of the PCB to dissipate heat also depends on the size, quantity, and spacing of other components on the board, as well as whether the surrounding air is still or moving. Furthermore, some of these components such as the catch diode will add heat to the PCB and the heat can vary as the input voltage changes. For the inductor, depending on the physical size, type of core material, and the DC resistance, it could either act as a heat sink taking heat away from the board, or it could add heat to the board.

Thermal Considerations (continued)

CIRCUIT DATA FOR TEMPERATURE RISE CURVE TO-220 PACKAGE (T)

Capacitors	Through-hole electrolytic
Inductor	Through-hole, Renco
Diode	Through-hole, 5-A 40-V, Schottky
PCB	3-square inch, single-sided, 2-oz. copper (0.0028")

Figure 40. Junction Temperature Rise, TO-220

CIRCUIT DATA FOR TEMPERATURE RISE CURVE TO-263 PACKAGE (S)

Capacitors	Surface-mount tantalum, molded <i>D</i> size
Inductor	Surface-mount, Pulse Engineering, 68 μ H
Diode	Surface-mount, 5-A 40-V, Schottky
PCB	9-square inch, single-sided, 2-oz. copper (0.0028")

Figure 41. Junction Temperature Rise, TO-263

12 Device and Documentation Support

12.1 Custom Design with WEBENCH Tools

Click [here](#) to create a custom design using the LM2596 device with the WEBENCH® Power Designer.

1. Start by entering your V_{IN} , V_{OUT} and I_{OUT} requirements.
2. Optimize your design for key parameters like efficiency, footprint and cost using the optimizer dial and compare this design with other possible solutions from Texas Instruments.
3. WEBENCH Power Designer provides you with a customized schematic along with a list of materials with real time pricing and component availability.
4. In most cases, you will also be able to:
 - Run electrical simulations to see important waveforms and circuit performance,
 - Run thermal simulations to understand the thermal performance of your board,
 - Export your customized schematic and layout into popular CAD formats,
 - Print PDF reports for the design, and share your design with colleagues.
5. Get more information about WEBENCH tools at www.ti.com/webench.

12.2 Receiving Notification of Documentation Updates

To receive notification of documentation updates, navigate to the device product folder on ti.com. In the upper right corner, click on *Alert me* to register and receive a weekly digest of any product information that has changed. For change details, review the revision history included in any revised document.

12.3 Community Resources

The following links connect to TI community resources. Linked contents are provided "AS IS" by the respective contributors. They do not constitute TI specifications and do not necessarily reflect TI's views; see TI's [Terms of Use](#).

TI E2E™ Online Community *TI's Engineer-to-Engineer (E2E) Community*. Created to foster collaboration among engineers. At e2e.ti.com, you can ask questions, share knowledge, explore ideas and help solve problems with fellow engineers.

Design Support *TI's Design Support* Quickly find helpful E2E forums along with design support tools and contact information for technical support.

12.4 Trademarks

E2E is a trademark of Texas Instruments.

SIMPLE SWITCHER, WEBENCH are registered trademarks of Texas Instruments.

All other trademarks are the property of their respective owners.

12.5 Electrostatic Discharge Caution



These devices have limited built-in ESD protection. The leads should be shorted together or the device placed in conductive foam during storage or handling to prevent electrostatic damage to the MOS gates.

12.6 Glossary

[SLYZ022](#) — *TI Glossary*.

This glossary lists and explains terms, acronyms, and definitions.

13 Mechanical, Packaging, and Orderable Information

The following pages include mechanical, packaging, and orderable information. This information is the most current data available for the designated devices. This data is subject to change without notice and revision of this document. For browser-based versions of this data sheet, refer to the left-hand navigation.



www.ti.com

PACKAGE OPTION ADDENDUM

3-Oct-2018

PACKAGING INFORMATION

Orderable Device	Status (1)	Package Type	Package Drawing	Pins	Package Qty	Eco Plan (2)	Lead/Ball Finish (6)	MSL Peak Temp (3)	Op Temp (°C)	Device Marking (4/5)	Samples
LM2596S-12/NOPB	ACTIVE	DDPAK/ TO-263	KTT	5	45	Pb-Free (RoHS Exempt)	CU SN	Level-3-245C-168 HR		LM2596S -12 P+	Samples
LM2596S-3.3	NRND	DDPAK/ TO-263	KTT	5	45	TBD	Call TI	Call TI		LM2596S -3.3 P+	
LM2596S-3.3/NOPB	ACTIVE	DDPAK/ TO-263	KTT	5	45	Pb-Free (RoHS Exempt)	CU SN	Level-3-245C-168 HR		LM2596S -3.3 P+	Samples
LM2596S-5.0	NRND	DDPAK/ TO-263	KTT	5	45	TBD	Call TI	Call TI		LM2596S -5.0 P+	
LM2596S-5.0/NOPB	ACTIVE	DDPAK/ TO-263	KTT	5	45	Pb-Free (RoHS Exempt)	CU SN	Level-3-245C-168 HR		LM2596S -5.0 P+	Samples
LM2596S-ADJ/NOPB	ACTIVE	DDPAK/ TO-263	KTT	5	45	Pb-Free (RoHS Exempt)	CU SN	Level-3-245C-168 HR	-40 to 125	LM2596S -ADJ P+	Samples
LM2596SX-12/NOPB	ACTIVE	DDPAK/ TO-263	KTT	5	500	Pb-Free (RoHS Exempt)	CU SN	Level-3-245C-168 HR		LM2596S -12 P+	Samples
LM2596SX-3.3/NOPB	ACTIVE	DDPAK/ TO-263	KTT	5	500	Pb-Free (RoHS Exempt)	CU SN	Level-3-245C-168 HR		LM2596S -3.3 P+	Samples
LM2596SX-5.0/NOPB	ACTIVE	DDPAK/ TO-263	KTT	5	500	Pb-Free (RoHS Exempt)	CU SN	Level-3-245C-168 HR		LM2596S -5.0 P+	Samples
LM2596SX-ADJ	NRND	DDPAK/ TO-263	KTT	5	500	TBD	Call TI	Call TI	-40 to 125	LM2596S -ADJ P+	
LM2596SX-ADJ/NOPB	ACTIVE	DDPAK/ TO-263	KTT	5	500	Pb-Free (RoHS Exempt)	CU SN	Level-3-245C-168 HR	-40 to 125	LM2596S -ADJ P+	Samples
LM2596T-12/LF03	ACTIVE	TO-220	NDH	5	45	Green (RoHS & no Sb/Br)	CU SN	Level-1-NA-UNLIM		LM2596T -12 P+	Samples
LM2596T-12/NOPB	ACTIVE	TO-220	NDH	5	45	Green (RoHS & no Sb/Br)	CU SN	Level-1-NA-UNLIM		LM2596T -12 P+	Samples
LM2596T-3.3/LF03	ACTIVE	TO-220	NDH	5	45	Green (RoHS & no Sb/Br)	CU SN	Level-1-NA-UNLIM		LM2596T -3.3 P+	Samples
LM2596T-3.3/NOPB	ACTIVE	TO-220	NDH	5	45	Green (RoHS & no Sb/Br)	CU SN	Level-1-NA-UNLIM		LM2596T -3.3 P+	Samples
LM2596T-5.0	NRND	TO-220	NDH	5	45	TBD	Call TI	Call TI		LM2596T -5.0 P+	
LM2596T-5.0/LF03	ACTIVE	TO-220	NDH	5	45	Green (RoHS & no Sb/Br)	CU SN	Level-1-NA-UNLIM		LM2596T -5.0 P+	Samples



www.ti.com

PACKAGE OPTION ADDENDUM

3-Oct-2018

Orderable Device	Status (1)	Package Type	Package Drawing	Pins	Package Qty	Eco Plan (2)	Lead/Ball Finish (6)	MSL Peak Temp (3)	Op Temp (°C)	Device Marking (4/5)	Samples
LM2596T-5.0/NOPB	ACTIVE	TO-220	NDH	5	45	Green (RoHS & no Sb/Br)	CU SN	Level-1-NA-UNLIM		LM2596T -5.0 P+	Samples
LM2596T-ADJ	NRND	TO-220	NDH	5	45	TBD	Call TI	Call TI	-40 to 125	LM2596T -ADJ P+	
LM2596T-ADJ/LF02	ACTIVE	TO-220	NEB	5	45	Green (RoHS & no Sb/Br)	CU SN	Level-1-NA-UNLIM		LM2596T -ADJ P+	Samples
LM2596T-ADJ/NOPB	ACTIVE	TO-220	NDH	5	45	Green (RoHS & no Sb/Br)	CU SN	Level-1-NA-UNLIM	-40 to 125	LM2596T -ADJ P+	Samples

(1) The marketing status values are defined as follows:

ACTIVE: Product device recommended for new designs.

LIFEBUY: TI has announced that the device will be discontinued, and a lifetime-buy period is in effect.

NRND: Not recommended for new designs. Device is in production to support existing customers, but TI does not recommend using this part in a new design.

PREVIEW: Device has been announced but is not in production. Samples may or may not be available.

OBSOLETE: TI has discontinued the production of the device.

(2) **RoHS:** TI defines "RoHS" to mean semiconductor products that are compliant with the current EU RoHS requirements for all 10 RoHS substances, including the requirement that RoHS substance do not exceed 0.1% by weight in homogeneous materials. Where designed to be soldered at high temperatures, "RoHS" products are suitable for use in specified lead-free processes. TI may reference these types of products as "Pb-Free".

RoHS Exempt: TI defines "RoHS Exempt" to mean products that contain lead but are compliant with EU RoHS pursuant to a specific EU RoHS exemption.

Green: TI defines "Green" to mean the content of Chlorine (Cl) and Bromine (Br) based flame retardants meet JS709B low halogen requirements of <=1000ppm threshold. Antimony trioxide based flame retardants must also meet the <=1000ppm threshold requirement.

(3) **MSL, Peak Temp.** - The Moisture Sensitivity Level rating according to the JEDEC industry standard classifications, and peak solder temperature.

(4) There may be additional marking, which relates to the logo, the lot trace code information, or the environmental category on the device.

(5) Multiple Device Markings will be inside parentheses. Only one Device Marking contained in parentheses and separated by a "-" will appear on a device. If a line is indented then it is a continuation of the previous line and the two combined represent the entire Device Marking for that device.

(6) **Lead/Ball Finish** - Orderable Devices may have multiple material finish options. Finish options are separated by a vertical ruled line. Lead/Ball Finish values may wrap to two lines if the finish value exceeds the maximum column width.

Important Information and Disclaimer: The information provided on this page represents TI's knowledge and belief as of the date that it is provided. TI bases its knowledge and belief on information provided by third parties, and makes no representation or warranty as to the accuracy of such information. Efforts are underway to better integrate information from third parties. TI has taken and continues to take reasonable steps to provide representative and accurate information but may not have conducted destructive testing or chemical analysis on incoming materials and chemicals. TI and TI suppliers consider certain information to be proprietary, and thus CAS numbers and other limited information may not be available for release.

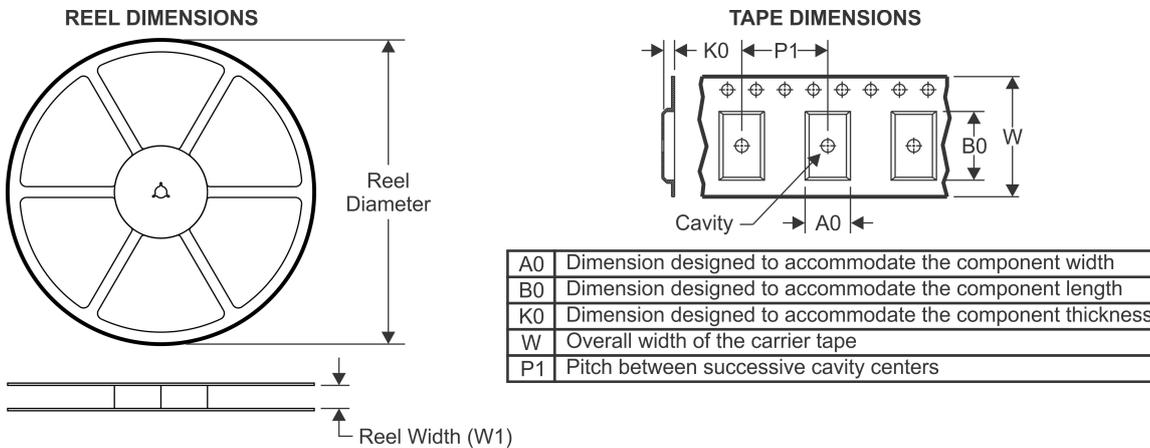
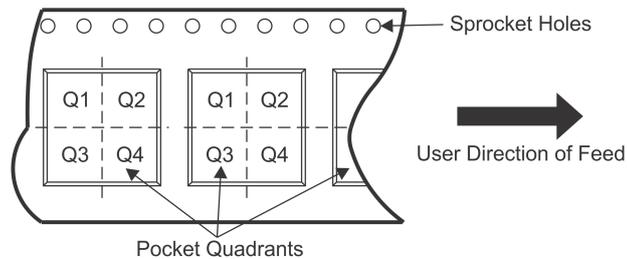


www.ti.com

PACKAGE OPTION ADDENDUM

3-Oct-2018

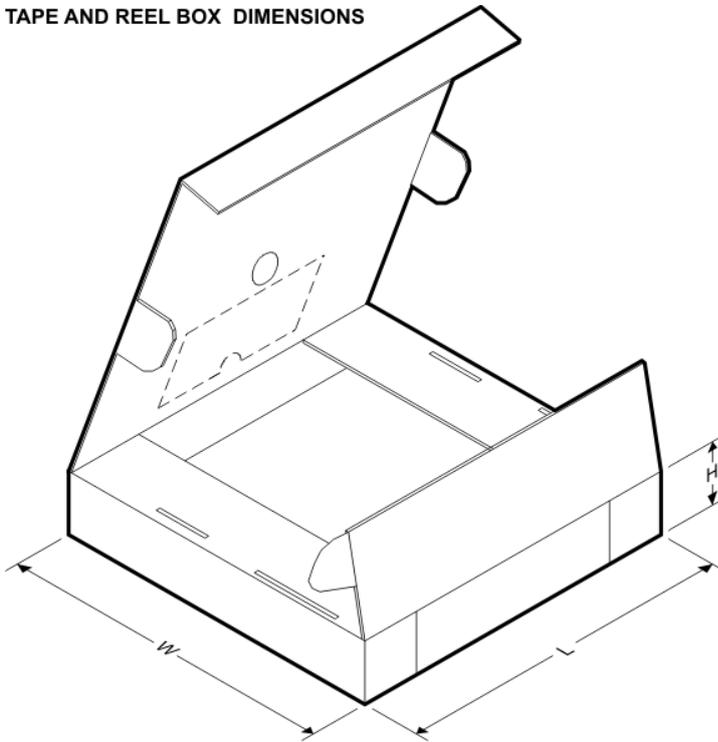
In no event shall TI's liability arising out of such information exceed the total purchase price of the TI part(s) at issue in this document sold by TI to Customer on an annual basis.

TAPE AND REEL INFORMATION

QUADRANT ASSIGNMENTS FOR PIN 1 ORIENTATION IN TAPE


*All dimensions are nominal

Device	Package Type	Package Drawing	Pins	SPQ	Reel Diameter (mm)	Reel Width W1 (mm)	A0 (mm)	B0 (mm)	K0 (mm)	P1 (mm)	W (mm)	Pin1 Quadrant
LM2596SX-12/NOPB	DDPAK/TO-263	KTT	5	500	330.0	24.4	10.75	14.85	5.0	16.0	24.0	Q2
LM2596SX-3.3/NOPB	DDPAK/TO-263	KTT	5	500	330.0	24.4	10.75	14.85	5.0	16.0	24.0	Q2
LM2596SX-5.0/NOPB	DDPAK/TO-263	KTT	5	500	330.0	24.4	10.75	14.85	5.0	16.0	24.0	Q2
LM2596SX-ADJ	DDPAK/TO-263	KTT	5	500	330.0	24.4	10.75	14.85	5.0	16.0	24.0	Q2
LM2596SX-ADJ/NOPB	DDPAK/TO-263	KTT	5	500	330.0	24.4	10.75	14.85	5.0	16.0	24.0	Q2

TAPE AND REEL BOX DIMENSIONS

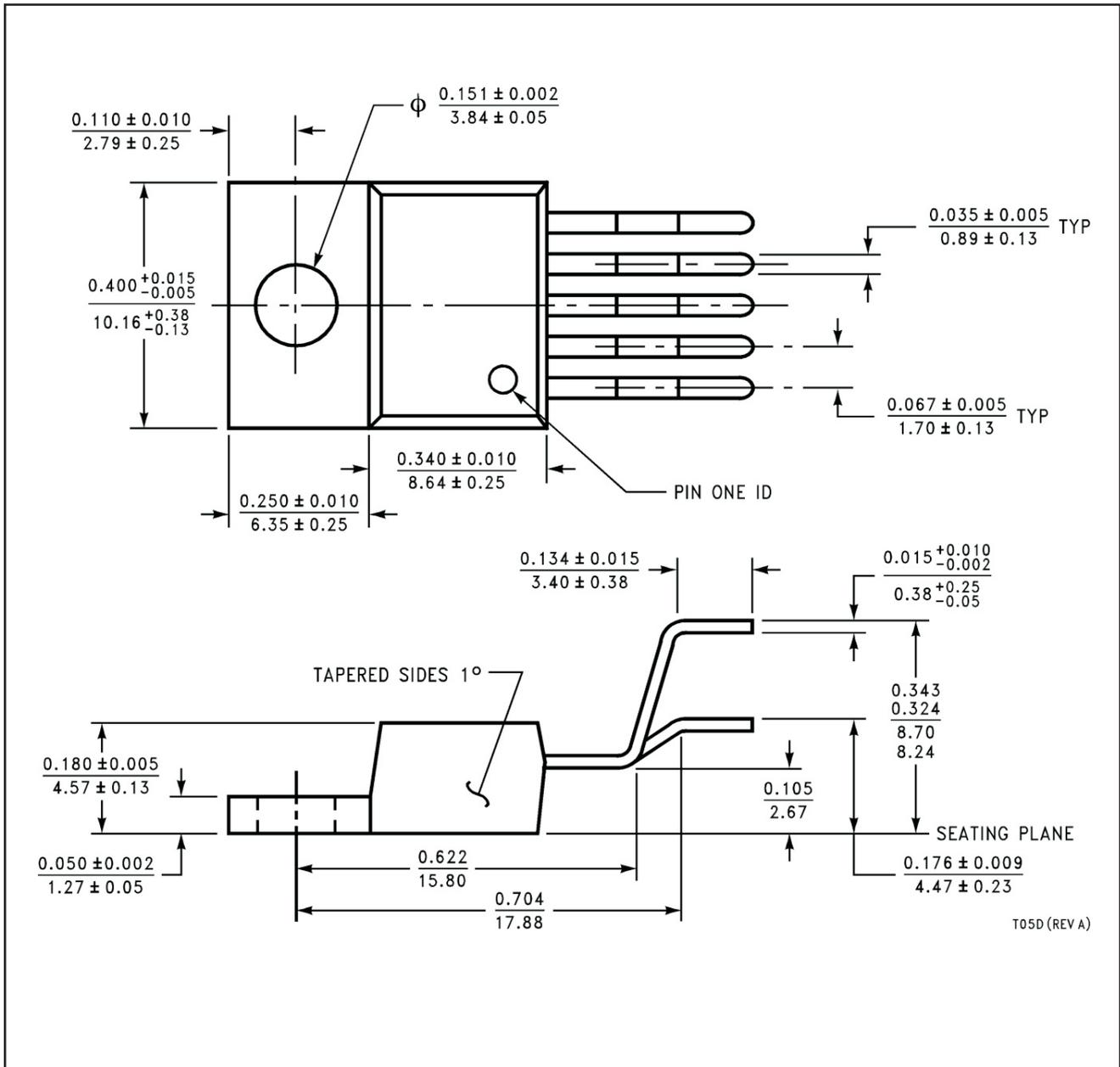


*All dimensions are nominal

Device	Package Type	Package Drawing	Pins	SPQ	Length (mm)	Width (mm)	Height (mm)
LM2596SX-12/NOPB	DDPAK/TO-263	KTT	5	500	367.0	367.0	45.0
LM2596SX-3.3/NOPB	DDPAK/TO-263	KTT	5	500	367.0	367.0	45.0
LM2596SX-5.0/NOPB	DDPAK/TO-263	KTT	5	500	367.0	367.0	45.0
LM2596SX-ADJ	DDPAK/TO-263	KTT	5	500	367.0	367.0	45.0
LM2596SX-ADJ/NOPB	DDPAK/TO-263	KTT	5	500	367.0	367.0	45.0

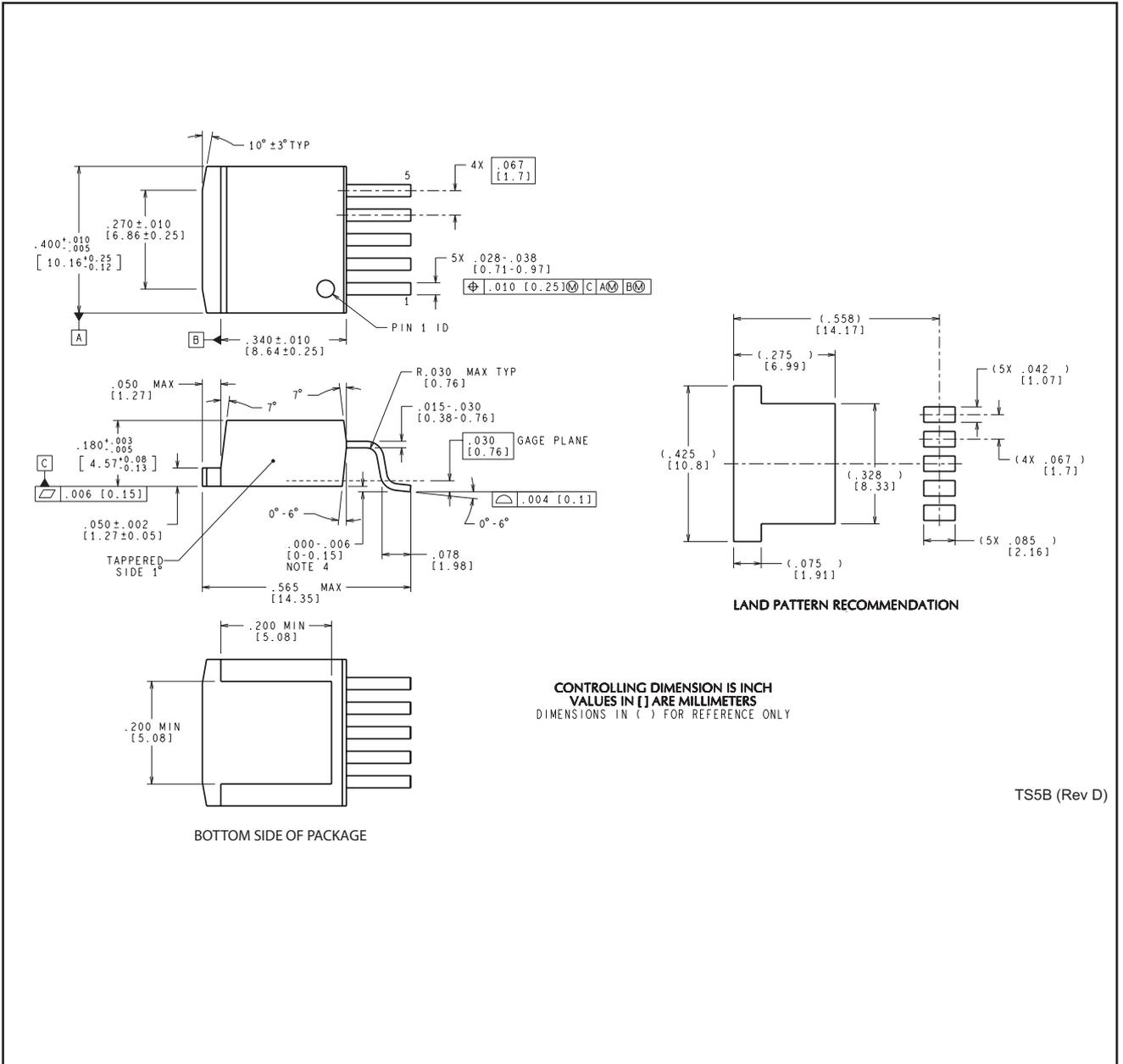
MECHANICAL DATA

NDH0005D



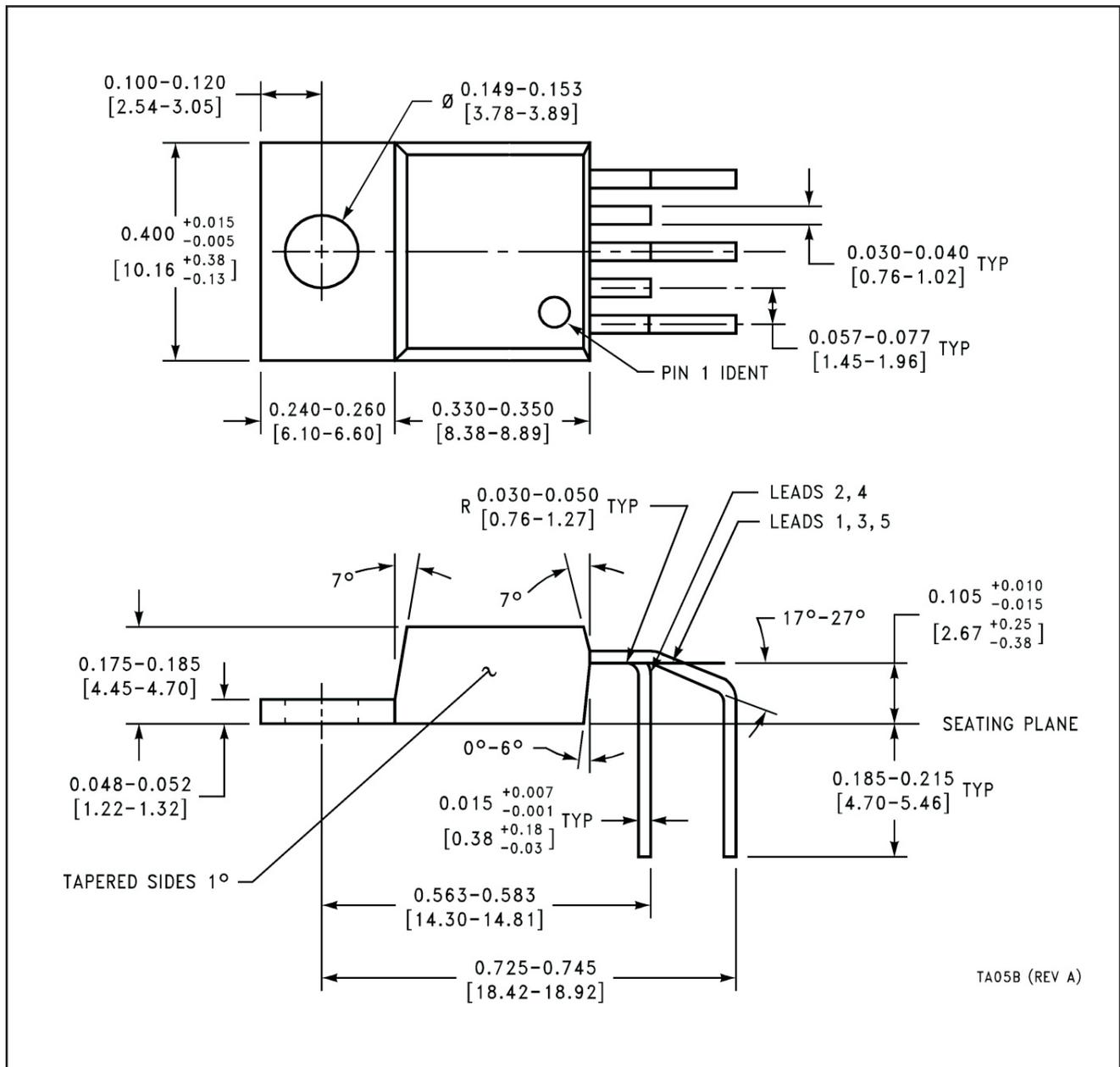
MECHANICAL DATA

KTT0005B



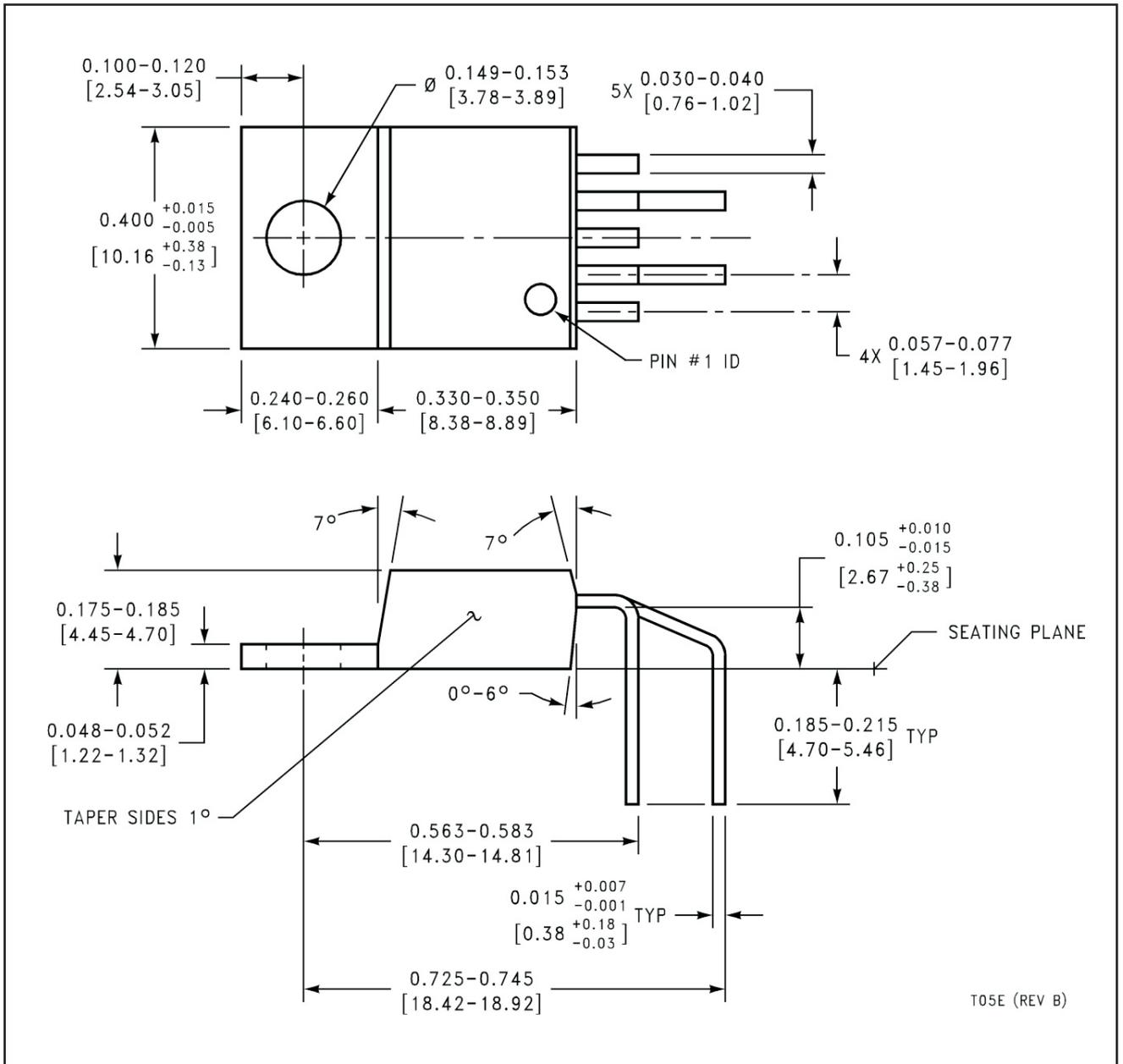
MECHANICAL DATA

NEB0005B



MECHANICAL DATA

NEB0005E



IMPORTANT NOTICE AND DISCLAIMER

TI PROVIDES TECHNICAL AND RELIABILITY DATA (INCLUDING DATASHEETS), DESIGN RESOURCES (INCLUDING REFERENCE DESIGNS), APPLICATION OR OTHER DESIGN ADVICE, WEB TOOLS, SAFETY INFORMATION, AND OTHER RESOURCES "AS IS" AND WITH ALL FAULTS, AND DISCLAIMS ALL WARRANTIES, EXPRESS AND IMPLIED, INCLUDING WITHOUT LIMITATION ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT OF THIRD PARTY INTELLECTUAL PROPERTY RIGHTS.

These resources are intended for skilled developers designing with TI products. You are solely responsible for (1) selecting the appropriate TI products for your application, (2) designing, validating and testing your application, and (3) ensuring your application meets applicable standards, and any other safety, security, or other requirements. These resources are subject to change without notice. TI grants you permission to use these resources only for development of an application that uses the TI products described in the resource. Other reproduction and display of these resources is prohibited. No license is granted to any other TI intellectual property right or to any third party intellectual property right. TI disclaims responsibility for, and you will fully indemnify TI and its representatives against, any claims, damages, costs, losses, and liabilities arising out of your use of these resources.

TI's products are provided subject to TI's Terms of Sale (www.ti.com/legal/termsofsale.html) or other applicable terms available either on ti.com or provided in conjunction with such TI products. TI's provision of these resources does not expand or otherwise alter TI's applicable warranties or warranty disclaimers for TI products.

Mailing Address: Texas Instruments, Post Office Box 655303, Dallas, Texas 75265
Copyright © 2018, Texas Instruments Incorporated

DOCUMENTO II



PLANOS

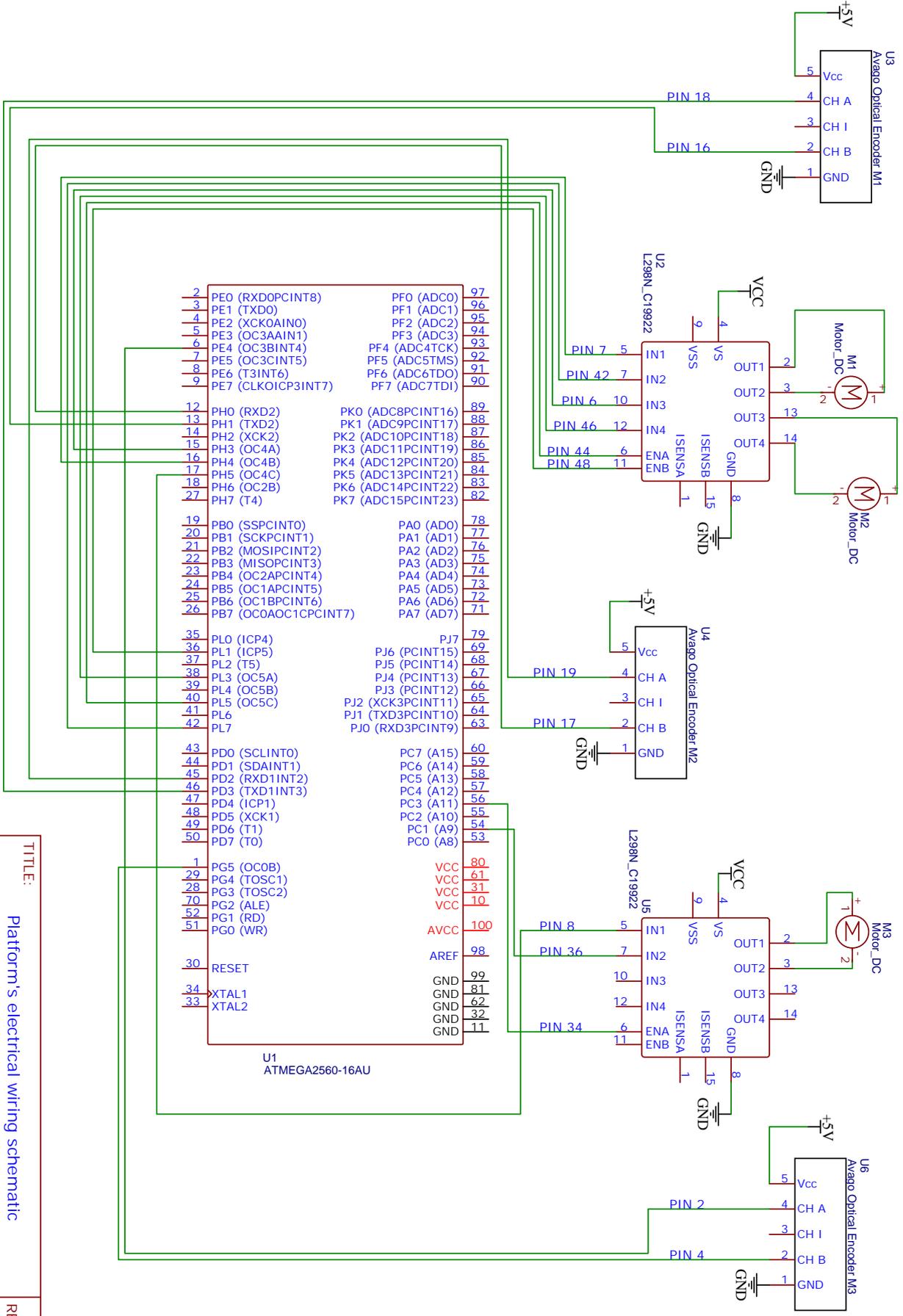


Lista de planos

Plano 1. Diagrama electrónico de la plataforma

Plano 2. Esquema para la PCB de la plataforma

Plano 3. Diseño de la PCB de la plataforma



2	PE0 (RXD0PCINT8)	PF0 (ADC0)	97
3	PE1 (TXD0)	PF1 (ADC1)	96
4	PE2 (XCK0AIN0)	PF2 (ADC2)	95
5	PE3 (OC3AINT1)	PF3 (ADC3)	94
6	PE4 (OC3BINT4)	PF4 (ADC4TCK)	93
7	PE5 (OC3CINT5)	PF5 (ADC5TMS)	92
8	PE6 (T3INT6)	PF6 (ADC6TDO)	91
9	PE7 (CLKOICP3INT7)	PF7 (ADC7TDI)	90
12	PH0 (RXD2)	PK0 (ADC8PCINT16)	89
13	PH1 (TXD2)	PK1 (ADC9PCINT17)	88
14	PH2 (XCK2)	PK2 (ADC10PCINT18)	87
15	PH3 (OC4A)	PK3 (ADC11PCINT19)	86
16	PH4 (OC4B)	PK4 (ADC12PCINT20)	85
17	PH5 (OC4C)	PK5 (ADC13PCINT21)	84
18	PH6 (OC2B)	PK6 (ADC14PCINT22)	83
27	PH7 (T4)	PK7 (ADC15PCINT23)	82
19	PB0 (SSPCINT0)	PA0 (AD0)	78
20	PB1 (SCKPCINT1)	PA1 (AD1)	77
21	PB2 (MOSPCINT2)	PA2 (AD2)	76
22	PB3 (MISOPCINT3)	PA3 (AD3)	75
23	PB4 (OC2APCINT4)	PA4 (AD4)	74
24	PB5 (OC1APCINT5)	PA5 (AD5)	73
25	PB6 (OC1BPCINT6)	PA6 (AD6)	72
26	PB7 (OCOAOIC1PCINT7)	PA7 (AD7)	71
35	PL0 (ICP4)	PJ7 (PCINT15)	79
36	PL1 (ICP5)	PJ6 (PCINT14)	69
37	PL2 (T5)	PJ5 (PCINT13)	68
38	PL3 (OC5A)	PJ4 (PCINT12)	67
39	PL4 (OC5B)	PJ3 (PCINT11)	66
40	PL5 (OC5C)	PJ2 (XCK3PCINT11)	65
41	PL6	PJ1 (TXD3PCINT10)	64
42	PL7	PJ0 (RXD3PCINT9)	63
43	PD0 (SCLINT0)	PC7 (A15)	60
44	PD1 (SDAINT1)	PC6 (A14)	59
45	PD2 (RXD1INT2)	PC5 (A13)	58
46	PD3 (TXD1INT3)	PC4 (A12)	57
47	PD4 (ICP1)	PC3 (A11)	56
48	PD5 (XCK1)	PC2 (A10)	55
49	PD6 (T1)	PC1 (A9)	54
50	PD7 (T0)	PC0 (A8)	53
1	PG5 (OC0B)	VCC	80
29	PG4 (TOSC1)	VCC	61
28	PG3 (TOSC2)	VCC	31
70	PG2 (ALE)	VCC	10
52	PG1 (RD)	AVCC	100
51	PG0 (WR)	AREF	98
30	RESET	GND	99
34	XTAL1	GND	81
33	XTAL2	GND	62
		GND	32
		GND	11

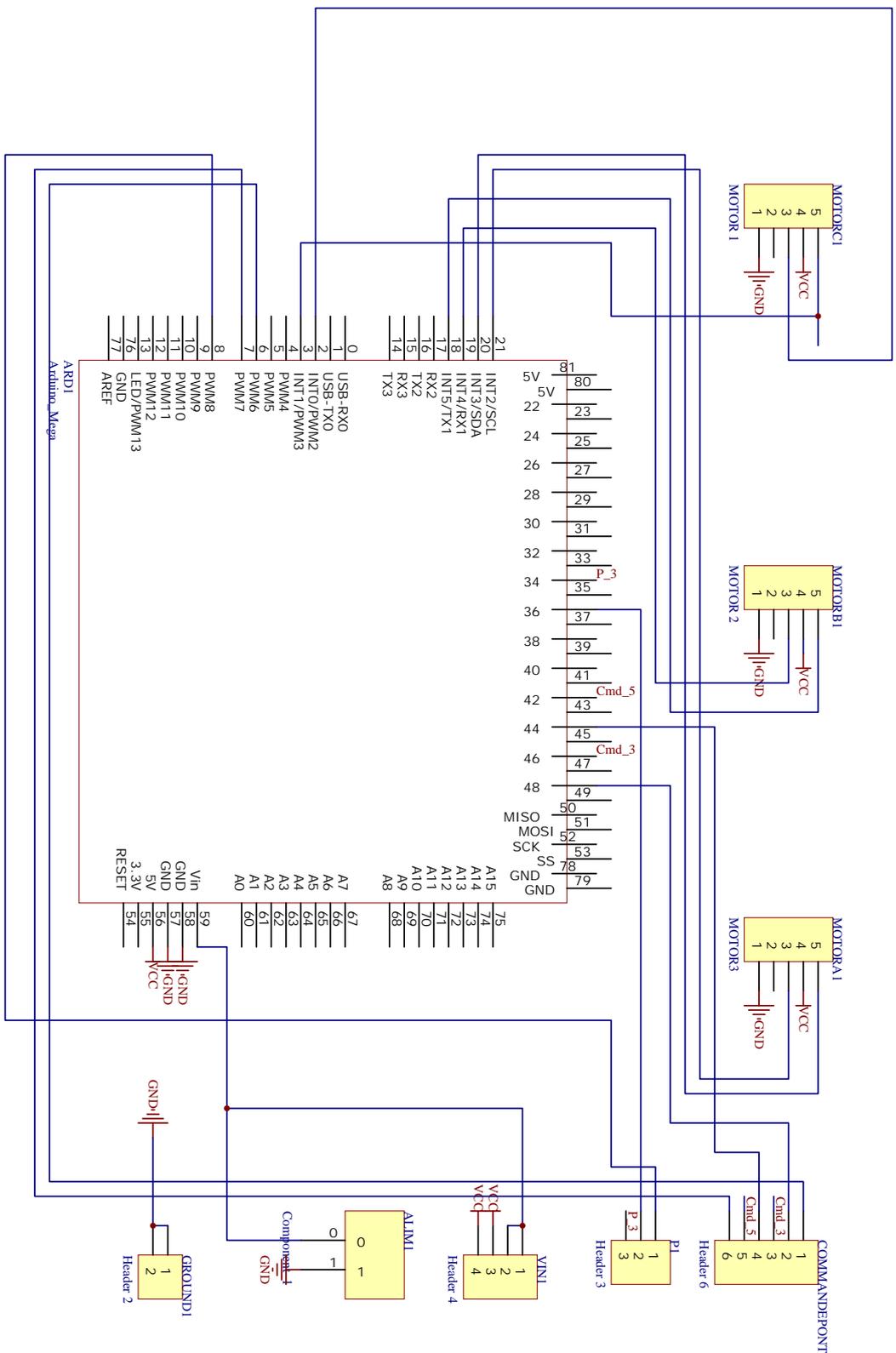
U1
ATMEGA2560-16AU

TITLE:
Platform's electrical wiring schematic

Company: Eurobot 2019
Date: 18-06-2019
Drawn By: Daniel Horcajo

REV: 1.2
Sheet: 1/1





TITLE: **PCB Base roulante**

REV: 1.4



Company: **Eurobot 2019**

Date: **21-03-2019**

Drawn By: **Said Lourhaoui**

Sheet: 1/1

Carte

