



Universidad
de La Laguna
Escuela Superior de
Ingeniería y Tecnología
Sección de Ingeniería Industrial



Universidad
de La Laguna

Escuela Superior de
Ingeniería y Tecnología

Escuela Superior de Ingeniería y Tecnología

Trabajo de Fin de Grado

DISEÑO Y DESARROLLO DE PLAYCAT,
UN CENTRO DE JUEGO ELECTRÓNICO
INTERACTIVO PARA GATOS

Titulación: Grado en Ingeniería Electrónica Industrial y Automática

Alumno: Mario Samir Batista Herrera

Julio, 2014

RESUMEN.....	2
ABSTRACT.....	2
CAPÍTULO 1. INTRODUCCIÓN.....	4
CAPÍTULO 2. ANTECEDENTES	8
CAPÍTULO 3. DESARROLLO ELECTRÓNICO	10
3.1 ESQUEMA GENERAL DEL DESARROLLO ELECTRÓNICO	10
3.2 ELEMENTOS ELECTRÓNICOS DEL PROTOTIPO.....	11
3.2.1 Motor.....	11
3.2.2 Arduino	13
3.2.3 Monster Moto Shield	15
3.2.4 Ethernet Shield.....	17
3.2.5 Router Wifi.....	19
3.2.6 Fuente de alimentación	21
3.2.7 Centralización de la electrónica.....	22
CAPÍTULO 4. DESARROLLO MECÁNICO.....	24
4.1 ESQUEMA GENERAL DEL DESARROLLO MECÁNICO.....	24
4.2 ELEMENTOS MECÁNICOS DEL PROTOTIPO	25
4.2.1 Base principal.....	25
4.2.2 Anclaje y protección del motor	26
4.2.3 Prolongación del eje del motor	29
4.2.4 Forrado de la primera parte del prototipo.....	31
4.2.5 Escalón.....	33
4.2.6 Tapa del motor.....	35
4.2.7 Recorrido de los cables del motor.....	37
CAPÍTULO 5. DESARROLLO INFORMÁTICO	39
5.1 LENGUAJES UTILIZADOS EN LA PROGRAMACIÓN	39
5.2 ENTORNO DE DESARROLLO DE ARDUINO.....	39
5.3 CÓDIGO DE PROGRAMACIÓN	41
5.3 CARGA DEL PROGRAMA	50
5.4 INTERFAZ WEB	51
CAPÍTULO 6. PRESUPUESTO DEL PROTOTIPO	53
CAPÍTULO 7. ESTUDIO DE VIABILIDAD ECONÓMICA.....	54
CAPÍTULO 8. CONCLUSIONES Y LÍNEAS FUTURAS	58
CAPÍTULO 9. BIBLIOGRAFÍA	62
ANEXO I. ESQUEMÁTICO DE ARDUINO UNO	63
ANEXO II. ESQUEMÁTICO DE MONSTER MOTO SHIELD.....	65
ANEXO III. DATASHEET VNH2SP30-E	67

Resumen

El objetivo del presente Trabajo de Fin de Grado es el diseño y desarrollo de PlayCat, un centro de juego electrónico interactivo para gatos. El prototipo debe hacer girar el juguete de un gato alrededor de la base del sistema. El sentido de giro y la velocidad podrán ser controlados por medio de un smartphone con conexión Wi-Fi.

El desarrollo del prototipo se divide en tres bloques. En primer lugar, se desarrollará la parte electrónica, que albergará la selección de un motor de corriente continua, su control mediante la plataforma de código abierto Arduino por medio de un puente de transistores de efecto campo, un módulo de ethernet y un router wifi. Todo el sistema estará alimentado por una fuente de alimentación común. En segundo lugar, se desarrollará mecánicamente el prototipo, etapa que contendrá, entre otras cosas, la construcción de la base, la sujeción del motor y el acoplamiento de una extensión al eje del motor. Finalmente se llevará a cabo un desarrollo informático, en el que se programará Arduino para que el motor pueda ser controlado de forma intuitiva desde una interfaz web por medio de un dispositivo wifi.

Abstract

The objective of this Degree Final Project is the design and development of PlayCat, an interactive electronic play center for cats. The prototype must rotate a cat's toy around the base of the system. The rotation direction and the movement speed must be controlled by a smartphone with Wi-Fi connection.

The prototype development is divided into three blocks. Firstly, the electronic part will be carried out. This part comprises the selection of a DC motor, its control with Arduino open source platform and with a field effect transistors bridge, an ethernet shield and a Wi-Fi router. All the system will be supplied with a computer's power source. Secondly, the mechanic part of the prototype will be developed. This phase includes the building of the base, the motor fastening and the link of an extension to the motor shaft.

Finally, the informatics development will be carried out. Arduino will be programmed to intuitively control the motor movements from a web interface with a Wi-Fi device.

Capítulo 1. Introducción

El objetivo del presente Trabajo Fin de Grado es el diseño y desarrollo de PlayCat, un centro de juego electrónico interactivo para gatos. PlayCat es un entretenimiento tanto para los gatos y como para los usuarios (ver figura 1.1). Por una parte el gato podrá jugar en todo momento con él, tanto en ausencia como en presencia del dueño, ya que se pondrá en marcha de forma automática si detecta que el gato se acerca. Por otro lado, el usuario podrá controlarlo desde cualquier dispositivo con conexión wifi si así lo desea.



Figura 1.1. Prototipo de PlayCat montado.

PlayCat funciona de una manera bastante sencilla. Lo único que el usuario debe hacer es escoger el juguete para gatos que le colgará a PlayCat, ya sea una pelota con cascabeles o luces, un peluche, cuerdas de colores, etc. Cuando PlayCat detecta que el gato se acerca, comienza a hacer que el juguete gire tanto en sentido horario como en sentido antihorario y a distintas velocidades, todo ello de forma aleatoria. Como se ha comentado antes, el usuario podría controlar los movimientos que realiza PlayCat en cualquier momento si se conecta a la misma red wifi que PlayCat, ya sea desde un smartphone, una desde tablet o desde un ordenador.

La idea de PlayCat surge de una necesidad principal, proporcionar al usuario un centro de juego automático para que su gato pueda entretenerse tanto en su ausencia como en su presencia. Normalmente un gato pasa muchas horas solo en casa, ya sea por cuestiones de trabajo, ocio u otras actividades que hagan que el dueño deba estar fuera del domicilio, y si el gato no tiene un sistema de entretenimiento del que no se canse, se entretendrá con el mobiliario de la casa o con las pertenencias del dueño.

Además otro punto a tener en cuenta, es que los gatos suelen ser muy activos por la noche si durante el día no han tenido un cierto nivel de actividad física, lo que conlleva a que se paseen por toda la casa o que incluso despierten a los dueños en mitad de la noche. Por ello, PlayCat también solventa este problema sin que los dueños tengan que preocuparse, haciendo que los gatos realicen un mayor nivel de actividad física durante el día para que el nivel de cansancio que tengan al final del día sea el suficiente como para que puedan dormir, o al menos, estar más tranquilos, durante toda la noche.

Una vez cubiertas estas necesidades, se pensó en la forma de hacer a PlayCat más atractivo para los usuarios. Por ello, se llevó a cabo la incorporación del control de PlayCat mediante cualquier dispositivo con conexión wifi. Desde un principio se ha considerado un diseño que no tuviera elementos peligrosos al descubierto que pudieran dañar al gato, ni que tuviera elementos eléctricos al alcance para evitar cualquier accidente.

El presente Trabajo Fin de Grado se basa fundamentalmente en el diseño de la electrónica, informática y mecánica necesaria para hacer que PlayCat cumpla con todas las especificaciones solicitadas. Dichas especificaciones son:

- Capacidad de detectar la presencia del gato.
- Ejecución de movimientos aleatorios, con limitaciones en el tiempo y velocidad de los giros.
- Parada del sistema si no se detecta al gato.
- Posibilidad de control mediante interfaz wifi.
- Seguridad y fiabilidad del sistema.
- Robustez física del sistema.

Para el diseño del prototipo, se ha tenido en cuenta la jerarquía de necesidades de Maslow [1], que indica que el nivel más importante a considerar es la funcionalidad, seguido de la fiabilidad, utilidad, competencia y creatividad.

Desde el punto de vista electrónico-informático, arduino ha sido un punto fundamental para el desarrollo del prototipo. Esta plataforma permite implementar un gran número de funciones y métodos de trabajo que, con otro tipo de electrónica, hubiera supuesto muchísimo más tiempo de desarrollo. Usar arduino evitó realizar un diseño electrónico más específico y un diseño informático a bajo nivel.

Además, PlayCat cuenta con la tecnología de los ultrasonidos para detectar la proximidad del gato, transistores MOSFET para la ampliación de corriente que alimentará a un motor de corriente continua, y comunicación wifi para el control mediante una interfaz web.

La realización de este Trabajo Fin de Grado se ha llevado a cabo en varias fases. En primer lugar, se han diseñado y probado separadamente las partes del prototipo de índole electrónico, informático y mecánico. A medida que estas partes estaban más o menos estructuradas se comenzó con el desarrollo físico del prototipo. En esta fase se fueron haciendo mejoras del diseño de partida, tanto de la parte electrónica, como de la mecánica e informática, para cumplir especificaciones que no se habían tenido en cuenta en un primer momento, o para incorporarle alguna mejora a lo diseñado en un principio, y hacer el diseño más intuitivo y fácil de controlar para el usuario.

Una vez que el prototipo se terminó, fue probado con diferentes gatos, lo que permitió evaluar de un modo realista el resultado obtenido y el interés económico-social que podría tener este producto.

En el esquema de la figura 1.2 se muestran las distintas partes que interactúan en el sistema de PlayCat.

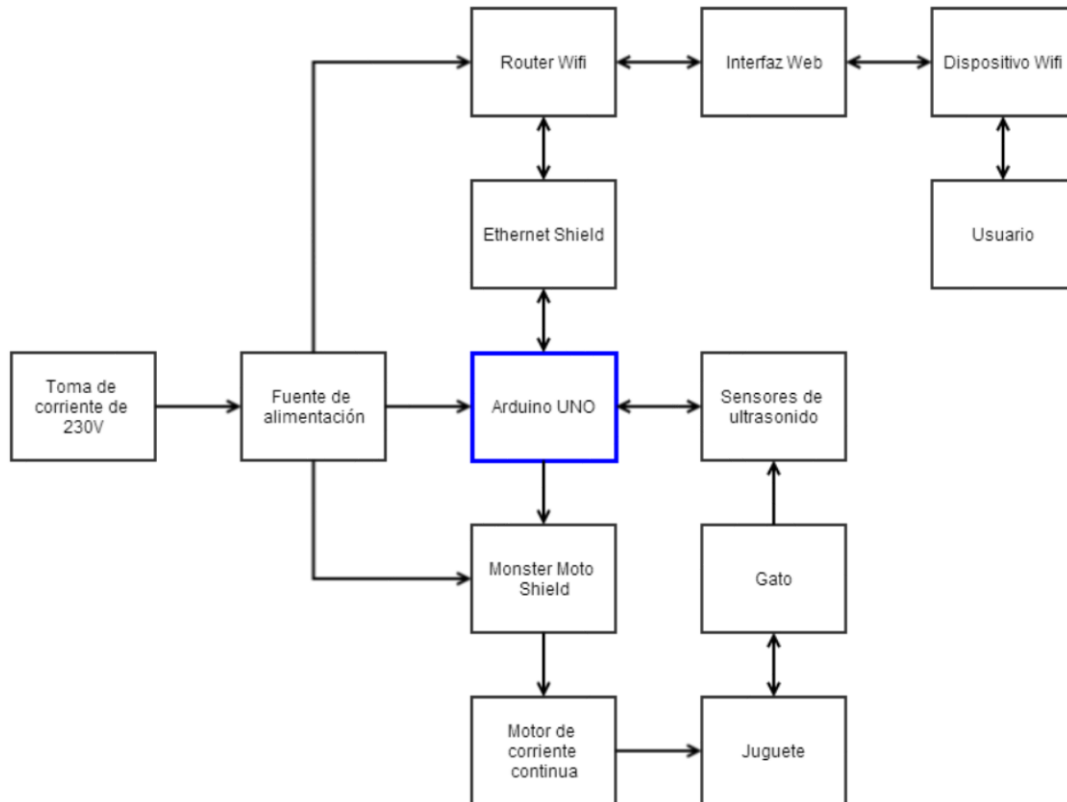


Figura 1.2. Esquema general de funcionamiento de PlayCat.

A lo largo de esta memoria se detalla la función de cada una de estas partes, desde el punto de vista de la electrónica, la informática y la mecánica, y se muestran todos los pasos que se siguieron para la creación del prototipo de PlayCat.

Por otro lado, PlayCat se ha presentado al concurso “Proyecto IDEAS”, organizado por la Fundación General de la Universidad de La Laguna (FGULL), consiguiendo el primer premio. Además, obtuvo un inesperado premio tecnológico para que se continúe su desarrollo durante 6 meses en el Parque Científico y Tecnológico de Tenerife. El Proyecto IDEAS permitió que el prototipo pudiera madurar más y adquirir un enfoque comercial para un posible futuro lanzamiento al mercado.

Capítulo 2. Antecedentes

En la actualidad hay infinidad de artículos de entretenimiento para gatos en el mercado, desde cualquier pelota con cascabeles o luces, rascadores, túneles o peluches, hasta centros de juego con plataformas a distintos niveles (ver figura 2.1).



Figura 2.1. Centro de juegos para gatos [2].

Existen juguetes más tecnológicos como DART (ver figura 2.2), que es un sistema rotatorio que cuenta con un puntero láser. Este sistema configura la velocidad de giro, se apaga pasados 15 minutos y funciona con tres pilas AA.



Figura 2.2. DART, puntero láser automático para gatos [3].

El mercado de este tipo de artículos está experimentando un crecimiento constante. El año pasado, los españoles gastaron en sus mascotas casi 124 millones de euros según un estudio realizado por Veterindustria (Asociación Empresarial Española de la Industria de Sanidad y Nutrición Animal) [4].

Otro dato de interés es que cada familia española gasta una media al año de 1500€ en los cuidados de su mascota (alimentación, limpieza, juguetes, veterinario...), mientras que en países europeos como Alemania o Francia esta cifra es sensiblemente más baja. El 80 % de las familias españolas que tienen gato lo consideran como una fuente de bienestar y aseguran que es un miembro más del núcleo familiar.

Estos datos incentivaron aún más la idea de desarrollar PlayCat, que aunque se enfrentaría con un mercado bastante competitivo, es un mercado muy amplio y con una alta facturación anual.

Capítulo 3. Desarrollo electrónico

3.1 Esquema general del desarrollo electrónico

La parte electrónica del prototipo se divide en cuatro bloques: la alimentación (toma de corriente, fuente de alimentación y la Monster Moto Shield), el control (arduino), la entrada de los sensores (sensores de ultrasonido) y la comunicación (Ethernet Shield y router wifi). En la figura 3.1 se puede observar cómo las distintas partes de la electrónica interactúan entre sí.

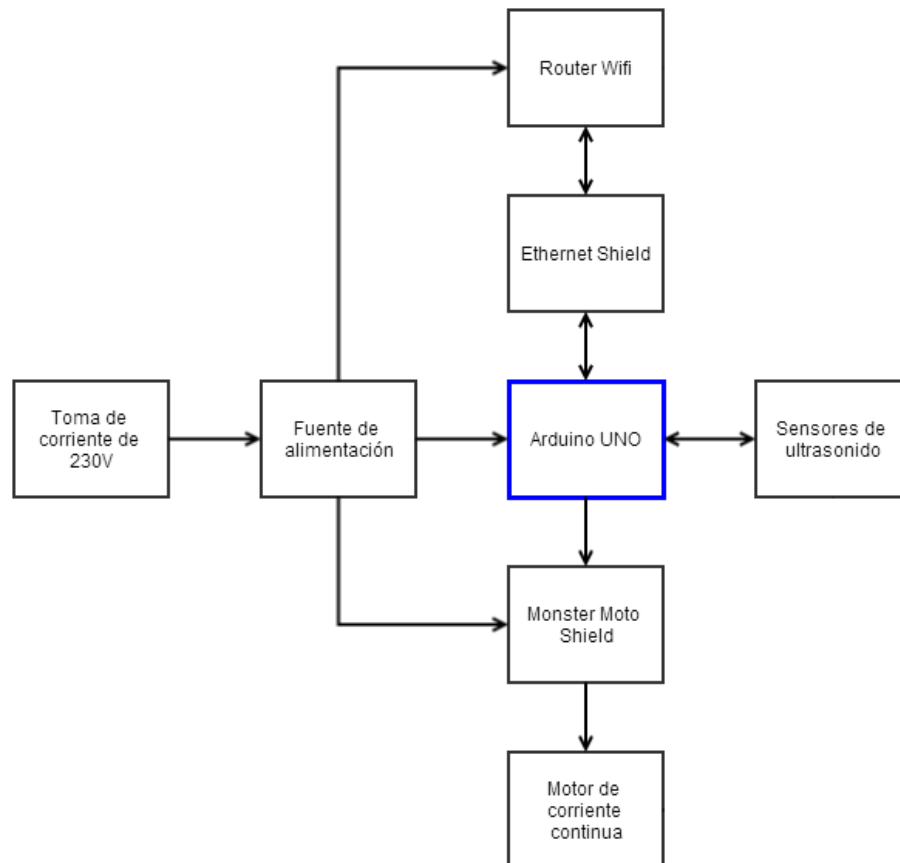


Figura 3.1. Esquema general de la electrónica del prototipo.

3.2 Elementos electrónicos del prototipo

3.2.1 Motor

El primer componente electrónico estudiado fue el motor. Era necesario escoger un tipo de motor para el cual tanto el control de la velocidad como el sentido de giro fuera sencillo de implementar. Se eligió un motor de corriente continua, ya que simplemente cambiando la polaridad, se consigue que cambie el sentido de giro del motor, y con el nivel de voltaje con el que se alimenta se modifica la velocidad a la que gira.

Además, el motor debe tener un cierto par motor, con el fin de poder cumplir las especificaciones requeridas. En este caso, dichas especificaciones se basaban en conseguir hacer girar una varilla roscada con forma de L que iría conectada al eje del motor y de la que estaría enganchado el juguete del gato. El caso más desfavorable sería en el que el gato consiguiera atrapar el juguete. Por lo tanto, el motor fue dimensionado para este caso crítico, en el que el motor debe realizar un par mayor.

Suponiendo que la fuerza del gato es totalmente horizontal, la situación de partida sería la reflejada en la figura 3.2.

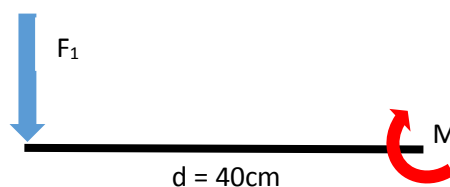


Figura 3.2. Diagrama de fuerzas presentes en la varilla que parte del eje del motor.

Si la masa del gato es de 6Kg, entonces la fuerza asociada a esa masa es:

$$F_1 = 6Kg \cdot 9,81 \frac{m}{s^2} = 58,56N$$

Haciendo el sumatorio de fuerzas se tiene que:

$$\begin{cases} \sum F = 0 \\ M = F_2 \cdot d \end{cases} \rightarrow F_1 = F_2 = \frac{M}{d} \rightarrow M = F_1 \cdot d = 58,56N \cdot 0,4m = 23,4 N \cdot m$$

Por lo que el motor deberá tener un par motor que cumpla que:

$$P_{motor} > 23,4N \cdot m$$

El motor seleccionado para el prototipo fue el de un limpiaparabrisas delantero de segunda mano, concretamente el de un Peugeot 306 (ver figura 3.3), el cual tiene un par motor de 25N·m, valor que es mayor al mínimo establecido en los cálculos (23,4N·m).

Los motores de limpiaparabrisas son bastante utilizados para hacer prototipos que requieran un par motor de estas dimensiones, debido a que son robustos, bastante asequibles y fáciles de encontrar en cualquier desguace de automóviles.



Figura 3.3 Motor DC con sujeción utilizado en el prototipo.

Este motor es de 12V y 45W, y tiene un consumo de 1,2A para girar a 40rpm y de 2,2A para girar a una velocidad de 62rpm [5]. Lo único que se le hizo al motor fue conectar ambos cables a un conector (ver figura 3.3), puesto que los cables que poseía eran demasiado cortos.

3.2.2 Arduino

Arduino es una plataforma de electrónica abierta para la creación de prototipos basada en software y hardware flexibles y fáciles de usar. Se creó para artistas, diseñadores, aficionados y cualquiera interesado en crear entornos u objetos interactivos.

Arduino puede recoger información del entorno a través de sus pines de entrada y alterar aquello que le rodea controlando luces, motores y otros actuadores. El microcontrolador en la placa arduino se programa mediante el lenguaje de programación arduino (basado en Wiring) y el entorno de desarrollo arduino (basado en Processing). Los proyectos de arduino pueden ejecutarse sin necesidad de que exista conexión a un ordenador, aunque puede existir esta comunicación para comunicarlo con diferentes tipos de software (ejemplos: Flash, Processing, MaxMSP...).

Las placas de arduino se pueden hacer a mano o comprarse montadas de fábrica, y el software se puede descargar de forma gratuita. Los ficheros de diseño de referencia (CAD) están disponibles bajo una licencia abierta, y el usuario es libre de adaptarlos a sus necesidades [6].

De la gama de modelos que presenta arduino, se ha escogido el arduino UNO (figura 3.4), ya que sus prestaciones son suficientes para poder implementar un primer prototipo de PlayCat. Éste es el modelo más difundido de la familia de arduino y en la tabla 3.1 aparecen las principales características de su versión rev3 (la más reciente).

Arduino da la posibilidad de alimentarlo por medio de un cable USB, lo cual es muy útil si se está programando constantemente, o por medio de un conector Jack, para un uso de arduino sin necesidad de tenerlo conectado al ordenador. En ningún caso estos cables se incluyen con la venta de arduino.

Se puede consultar el esquemático de Arduino UNO rev3 en el Anexo I de este proyecto.



Figura 3.4. Arduino UNO rev3 [7].

Microcontrolador	ATmega328
Voltaje de funcionamiento	5V
Alimentación (recomendada)	5 – 12V
Voltaje máximo de entrada (no recomendado)	20V
Número de pines entradas/salidas digitales	14 (de los cuales 6 dan salida PWM)
Número de entradas analógicas	6
Corriente DC por los pines I/O	40mA
Corriente DC para el pin 3.3V	50mA
Memoria flash	32KB (ATmega328) 0,5KB usados por el bootloader
SRAM	2KB (ATmega328)
EEPROM	1KB (ATmega328)
Velocidad de reloj	16MHz

Tabla 3.1. Características principales de arduino UNO rev3 [8].

Para el prototipo bajo desarrollo los 40mA de corriente de salida que son capaces de generar los pines del arduino UNO no son suficientes para hacer que el motor de corriente continua gire (ver apartado 3.2.1), por lo que es necesario amplificar esa corriente (hasta 2,2A para 62rpm). Para ello se hace uso de uno de los complementos poco difundidos con los que cuenta arduino, la Monster Moto Shield.

3.2.3 Monster Moto Shield

La Monster Moto Shield es uno de los módulos de la familia de arduino, y sus principales componentes son dos controladores VNH2SP30-E, compuestos por puentes de transistores MOSFET, que permiten hacer una gran amplificación de corriente y cuyo uso está muy difundido para el control de motores.

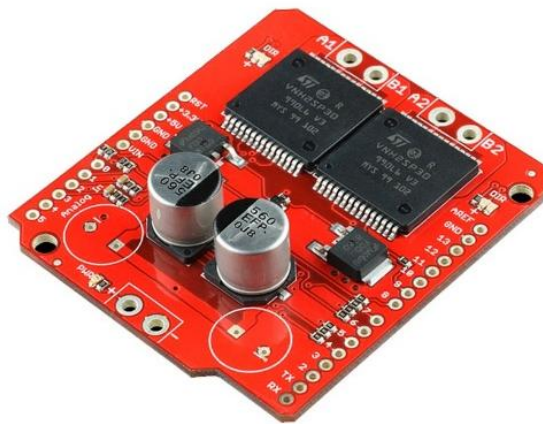


Figura 3.5. Monster Moto Shield con puentes MOSFET [9].

Esta placa lleva una alimentación de 12V, los cuales serán introducidos mediante los pines señalados por “PWR +” y “PWR –” en la placa (ver figura 3.5), y es capaz de generar salidas de hasta 30A. Un pequeño diodo LED situado a la izquierda de los pines de PWR comunicará si la placa está recibiendo alimentación. Además, la placa dispone de un pin analógico que va indicando la corriente que produce la placa.

Por otro lado, la placa cuenta también con 4 pines de salida para poder conectar motores: A1, B1, A2 y B2 (ver figura 3.5). Los dos primeros están pensados para el control de un motor, y los dos últimos para el control de un segundo motor. Sin embargo, aunque sólo cuente con dos salidas para conectar motores, se pueden colocar más de dos motores en paralelo si se desea que ambos motores giren a la misma velocidad, pero ello reduciría la corriente que puede recibir cada uno de ellos de forma independiente, y por tanto girarían a menos velocidad. Como en este caso sólo hay un solo motor, sólo se hará uso de los pines A1 y B1.

Para usos en los que esta placa tenga unos requisitos muy altos, es recomendable incorporarle un disipador de calor o un ventilador, y soldar directamente los cables a la placa en lugar de usar terminales de tornillo (aparte de los demás inconvenientes que puede conllevar un sistema de 30A). Sin embargo, está comprobado que para usos de hasta 6A, los chips apenas se calientan, por lo que para el caso de este prototipo caso no será necesaria una refrigeración extra para los chips, y se conectarán los cables a la placa mediante conectores con tornillo. La placa venía de fábrica sin conectores, así que se han adquirido tres conectores con tornillo (uno para la alimentación y los otros dos para las salidas A1-B1 y A2-B2) y se han soldado a la placa.

Esta placa, al igual que la mayoría de las destinadas para arduino, se inserta directamente sobre el arduino UNO, y los pines que no utilizará la Monster Moto Shield se podrán usar directamente desde dicha placa. Los pines de arduino que utiliza esta placa son los digitales que van desde el 0 hasta el 9, por lo que quedan libre del 10 al 13 de los digitales y las entradas analógicas de la 0 a la 5.

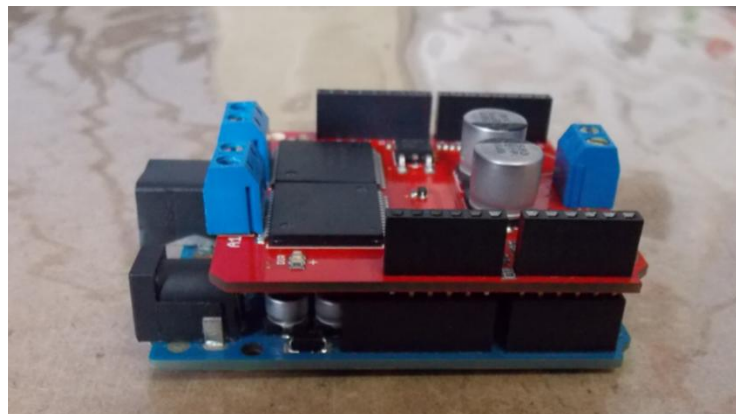


Figura 3.6. Monster Moto Shield con conectores montada sobre Arduino UNO [10].

Al conectar las placas de esta manera, los pines digitales 7 y 8 del Arduino corresponden a las salidas A1 y B1, respectivamente, de la Monster Moto Shield, mientras que los pines 4 y 9 corresponden a las salidas A2 y B2 respectivamente. Es decir, controlando los pines 7 y 8 (para un motor), y 4 y 9 (para otro motor), se puede hacer que los motores giren en un sentido u otro, o que se detengan (ver la tabla 3.2).

Pines de Arduino	Motor 1	
	D7	D8
Salidas de la Monster Moto Shield	A1	A2
Motor parado	0	0
Giro sentido horario	1	0
Giro sentido anti horario	0	1
Motor parado	1	1

Tabla 3.2. Configuración de los pines para el control del motor 1 [11].

Para el control del segundo motor, se hace uso de la misma lógica pero con los pines 4 y 9 de arduino. En el capítulo 5 de este proyecto, correspondiente al desarrollo informático, se explicará de forma más detallada cómo se programa arduino para controlar el comportamiento de los motores.

Se puede consultar el esquemático de la Monster Moto Shield en el Anexo II de este proyecto y el datasheet del VNH2SP30-E en el Anexo III de este proyecto.

3.2.4 Ethernet Shield

Para comunicarse con el arduino mediante la interfaz web (ver figura 3.1) se utilizará una de las placas más versátiles e interesantes de arduino, la Ethernet Shield.



Figura 3.7. Ethernet Shield para Arduino UNO [12].

Esta placa permite a arduino comunicarse mediante una interfaz web, lo que ofrece un gran abanico de posibilidades para enviar y recibir datos. Además, esta placa cuenta con una entrada para tarjetas SD, que puede usarse para almacenar audio, imágenes o incluso parte del código que no sea posible almacenar en la memoria de arduino. Sin embargo, no se hará uso de tarjetas SD en este proyecto.

De igual forma que la Monster Moto Shield, la Ethernet Shield se encaja directamente sobre arduino, haciendo que algunos de sus pines sean destinados únicamente para la Ethernet Shield. En este caso, esos pines reservados para Ethernet Shield son los digitales del 10 al 13.

Por lo tanto en este proyecto se debe conectar al arduino UNO a la Ethernet Shield y a la Monster Moto Shield, quedando el conjunto como una torre de 3 placas (ver figura 3.8).

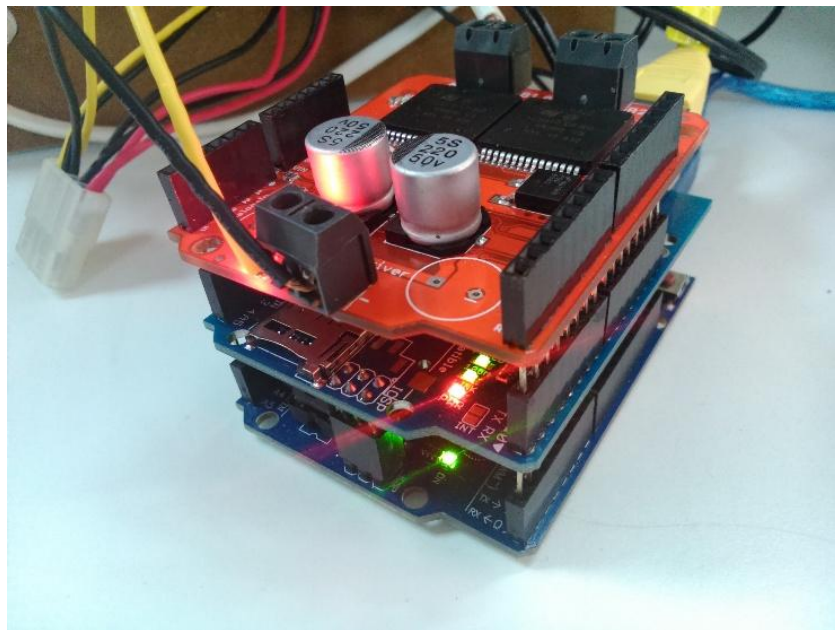


Figura 3.8. Ethernet Shield y Monster Moto Shield conectadas a Arduino UNO.

Para comunicar la Ethernet Shield se conecta mediante un cable Ethernet al router. En el capítulo 5 (capítulo correspondiente al desarrollo informático) se explicará de una forma más detallada el funcionamiento y configuraciones disponibles de la placa.

3.2.5 Router Wifi

Para comunicarse mediante una conexión wifi con arduino se tenían dos opciones: adquirir un módulo wifi en vez de la Ethernet Shield o conectar la Ethernet Shield a un router wifi. Como la segunda opción era mucho más económica, fue finalmente ésta la que se escogió.

Ya se disponía de un router wifi, pero por alguna razón funcionaba bien todo menos la red wifi, es decir, cualquier equipo se podía conectar al router mediante un cable de ethernet, pero no de forma inalámbrica. Después de ver que según la configuración todo parecía estar correcto, se llevó a cabo el desmontaje del router, y fue entonces cuando se encontró la causa del problema: el cable de la antena estaba roto. Visto esto, se volvió a unir los cables cuidadosamente mediante un conector (ver figura 3.10), y aprovechando que el router estaba desmontado, se realizó otra apertura en la parte frontal de la carcasa del router para sacar la antena, para que el cable estuviera más holgado, evitando así que se volviera a desconectar. También se cortó el cable de alimentación del router (ver figura 3.9), ya que se alimentará directamente de la fuente de alimentación común como el resto del sistema, como se explica en el apartado 3.2.6.

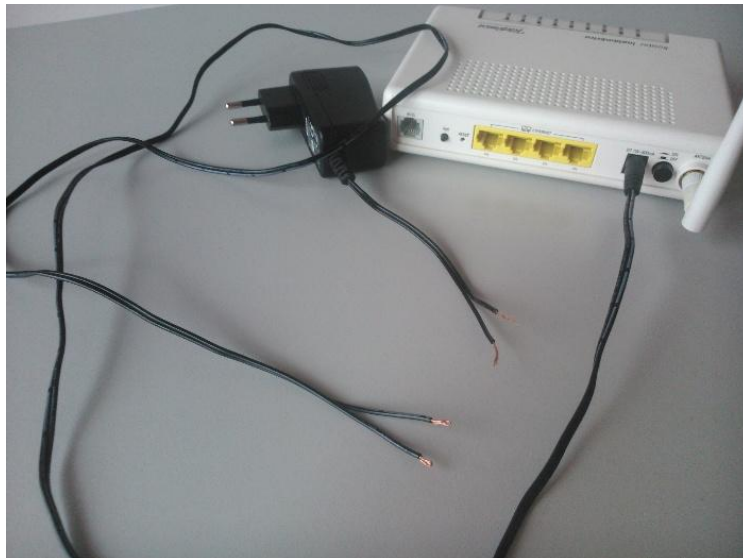


Figura 3.9. Corte del cable de alimentación del router wifi.

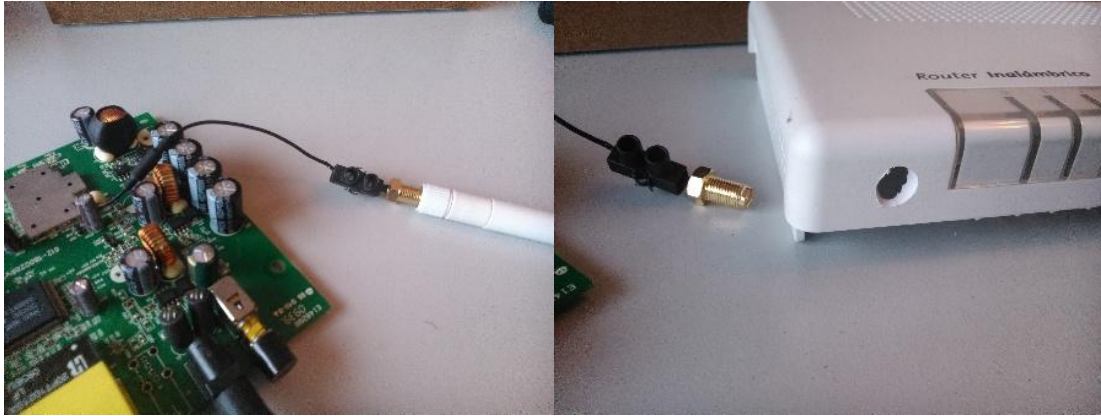


Figura 3.10. Reparación del cable de la antena (izquierda) y nuevo orificio en la carcasa del router para la disposición frontal de la antena wifi (derecha).



Figura 3.11. Antena wifi montada en la parte frontal del router.

Después de reparar el router, se accedió a él mediante un ordenador y un cable ethernet entrando al menú de configuraciones. Para ello, se introdujo la dirección IP en el navegador y se tecléo la clave que viene por defecto. Una vez dentro, se cambió el nombre que tiene por defecto la red wifi por “Miau”, y la contraseña por “tfgsamir”, para poder conectarse fácilmente desde cualquier dispositivo durante el periodo de desarrollo del prototipo.

Una vez configurado el router, se conecta a la Ethernet Shield mediante el cable de ethernet, y desde entonces se accede a él siempre vía wifi.

3.2.6 Fuente de alimentación

Como se señala en el esquema general de la introducción del proyecto (ver figura 1.2), tres son los componentes que necesitan alimentación en el sistema. Por una parte, el arduino UNO, que se conectará a una alimentación de 5 V. Por otro lado, la Monster Moto Shield, que necesita una tensión de alimentación de 12 V. Y por último, el router wifi, que demanda también una alimentación de 12 V.

Visto esto, se decidió alimentar todo el sistema con la fuente de alimentación convencional de un ordenador de sobremesa (ver figura 3.12). Una de las mejores características de estas fuentes es que producen alimentación a distintos voltajes, lo que soluciona el problema de tener un transformador para cada uno de los sistemas a alimentar, y permite apagar o encender todo el sistema simplemente con el apagado y encendido de la fuente de alimentación.

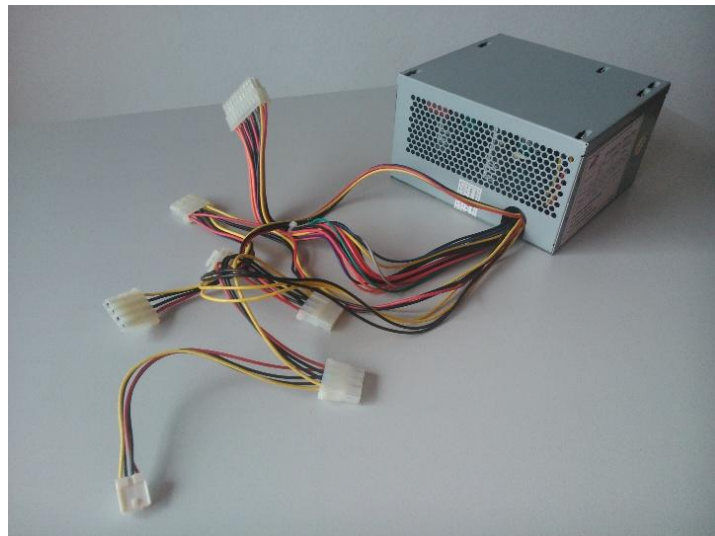


Figura 3.12. Fuente de alimentación utilizada para abastecer al sistema.

De la fuente de alimentación salen distintos conectores, que tienen como tarea alimentar a los distintos componentes que forman un ordenador, como la placa base, los discos duros, las unidades de lectura de discos, etc. Mirando la correspondencia de voltajes según el color del cable (ver figura 3.13), es posible identificar los conductores a partir de los que se pueden obtener los distintos voltajes necesarios para alimentar las diferentes partes del sistema.

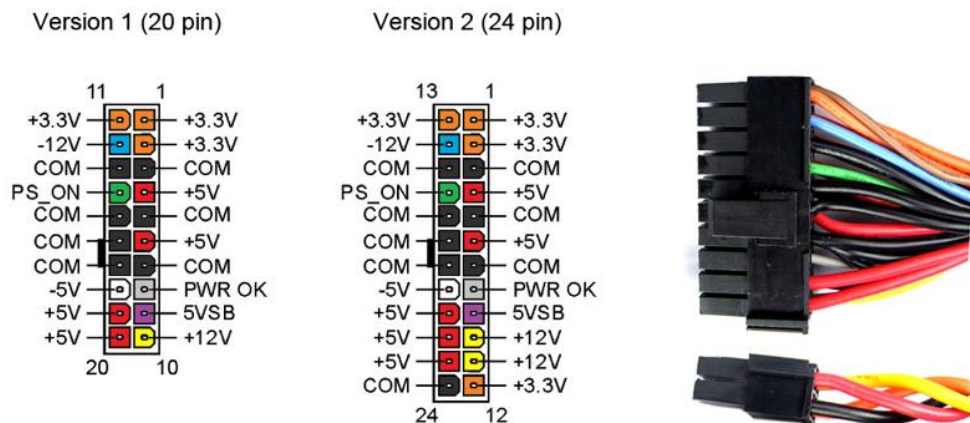


Figura 3.13. Pines del conector principal de una fuente de alimentación de PC [13].

Como se puede ver en el esquema de pines (ver figura 3.13), los 12 V se sacan de dos de los cables amarillos (uno para el router y otro para la Moto Monster Shield), los 5 V de uno de los cables rojos (para el arduino UNO) y las tierras de los cables negros.

Un aspecto bastante importante a tener en cuenta es que la fuente de alimentación de un ordenador de sobremesa no se pone en funcionamiento hasta que se pulsa el botón de encender. La reacción que hay asociada a esta acción es la conexión de la señal del pin identificado por PS_ON, el cable verde (ver figura 3.13) a tierra. Por ello, se realiza esta conexión en el propio conector mediante un cable, lo que implica que el sistema se encienda y se apague simplemente enchufando y desenchufando la fuente de alimentación de la toma de corriente.

3.2.7 Centralización de la electrónica

La electrónica del prototipo se decidió centralizar en un mismo lugar, para facilitar su transporte y evitar daños de la misma. Se colocaron todos los componentes en el interior de una caja de madera, que pertenecía a al bajo de un equipo de sonido. Fue bastante costoso introducir todos los componentes en la caja, debido al gran volumen de

la fuente y del router, por lo que se retiró la tapa de la fuente de alimentación. Esto no supone ningún riesgo eléctrico, ya que los componentes siguen estando protegidos la caja de madera, estando así aislados del exterior y del cualquier contacto directo o indirecto que pudiera producirse. Además de este modo se genera un flujo de aire que recorre la caja haciendo que todos los componentes tengan una mejor refrigeración.

Las partes que quedan accesibles desde el exterior de la caja son:

- El cable USB que se conecta con el arduino para poder hacer la programación del mismo desde el ordenador.
- El conector de la toma de corriente de la fuente de alimentación para encender y apagar el sistema.
- Los dos cables que salen de la Moto Monster Shield para alimentar el motor.
- La antena wifi del router, que sale por uno de los agujeros que tenía ya la caja.

Esta caja que alberga toda la electrónica siempre se encuentra fuera del alcance del gato para evitar cualquier problema, y el cable que va desde la Monster Moto Shield hasta el motor se colocó dentro de un tubo corrugado de PVC (ver figura 1.1).

Si fuera posible hacer un segundo prototipo de PlayCat toda la electrónica se colocaría junto al motor, y desde ese punto saldrían los cables que irían, debidamente protegidos, hasta una toma de corriente.



Figura 3.14. Router wifi y fuente de alimentación introducidos en la caja (izquierda) y toda la electrónica centralizada (derecha).

Capítulo 4. Desarrollo mecánico

4.1 Esquema general del desarrollo mecánico

Desde un comienzo se planeó que el prototipo fuera desmontable, para poder trabajar con él de una manera más cómoda y facilitar así su transporte. El prototipo está dividido en las siguientes partes (ver figura 4.1):

- La centralización de la electrónica, de la que salen los cables de alimentación que van a la toma de corriente y al motor.
- La base principal del prototipo.
- La base de sujeción y protección del motor.
- La extensión del eje del motor mediante una varilla roscada.
- Un escalón intermedio para ocultar los enganches de la base de sujeción del motor.
- Tapa de la caja del motor.
- Recubrimiento de moqueta de la base.

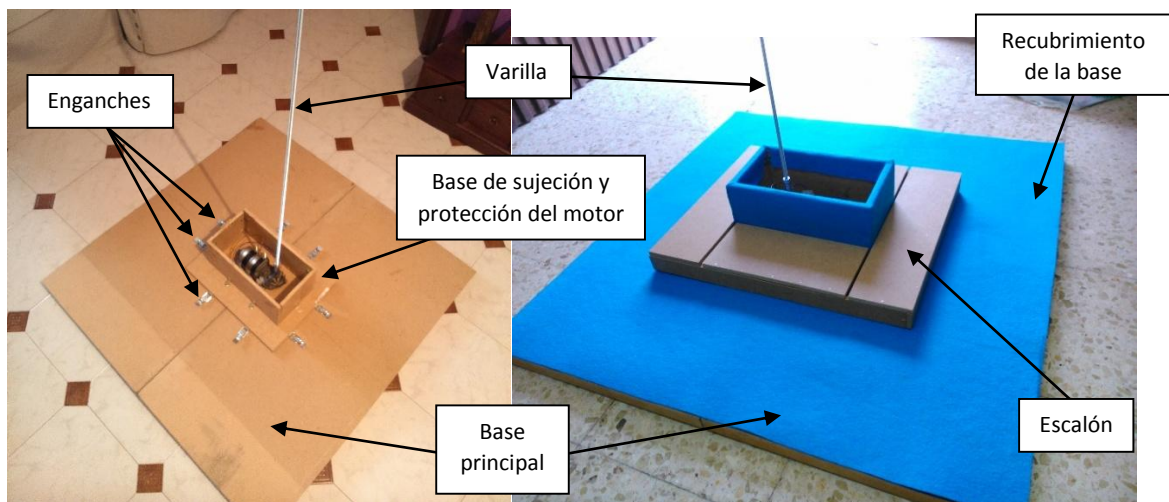


Figura 4.1. Explicación de las distintas partes del prototipo.

4.2 Elementos mecánicos del prototipo

4.2.1 Base principal

La base principal es un tablón de madera de dimensiones 1000x1000x40mm. Lo primero que se hizo fue cortarlo por la mitad, ya que resultaba muy difícil que una persona pudiera transportar un tablón de estas dimensiones y peso (20Kg). Luego se lijaron todos los bordes y se unieron ambas mitades por medio de dos bisagras rectas de hierro latonado (ver figura 4.2). Se escogió un tablón de estas dimensiones y peso por la estabilidad que le proporcionaría al prototipo, imposibilitando que el sistema volcara si el gato atrapara el juguete y tirara del mismo.



Figura 4.2. Una de las bisagras que unen las dos mitades de la base principal.

En un principio se colocaron las bisagras en la parte superior de la base, pero se comprobó que podrían suponer un posible obstáculo para el gato (aun estando la base forrada), por lo finalmente se colocaron en la parte inferior del tablón, la que está en contacto con el suelo. Al escoger esta opción, surgió un nuevo inconveniente: la base estaba más elevada por el centro que por los laterales. Para solucionar este problema se colocaron en las esquinas de la base cuatro fieltros adhesivos circulares de 50 milímetros de diámetro cuya altura es semejante al grosor de la parte central de la bisagra. De este modo se consigue que la base esté nivelada en toda su extensión.

4.2.2 Anclaje y protección del motor

El anclaje del motor a la base era un punto muy importante si se quería que el sistema no se tambaleara cuando estuviera en funcionamiento. Si el sistema debía ser desmontable, el motor debía tener su propia base, la base secundaria (ver figura 4.1), y sería ésta la que luego se fijaría a su vez a la base principal del sistema. Ambas uniones debían ser lo suficientemente seguras como para no se separaran o se desplazaran durante el uso de PlayCat.

Además, no se podía situar el motor directamente sobre la base secundaria. Si se hacía esto, el eje del motor no quedaba totalmente vertical, pues presenta un pequeño desnivel donde termina el cilindro y comienza la caja de engranajes (ver figura 4.3).



Figura 4.3. Señalización del reborde del motor (presente en toda su superficie).

Para solucionar este problema del reborde, se talló en la madera de la base del motor un pequeño orificio de superficie rectangular, tal que el reborde del motor entrara y

quedara más o menos encajado, haciendo ahora más difícil que el motor se desplazara a lo largo de su eje longitudinal. Una vez hecho esto, la sujeción del motor se llevó a cabo mediante dos abrazaderas metálicas de 60mm de diámetro. Se escogió esta medida para las abrazaderas dado que el diámetro de la parte cilíndrica del motor era de 63mm, y de esta forma, las abrazaderas se podían ajustar mediante 4 tornillos que irían de arriba abajo con sus correspondientes arandelas, tuercas y gomas anti vibración (ver figura 4.4).

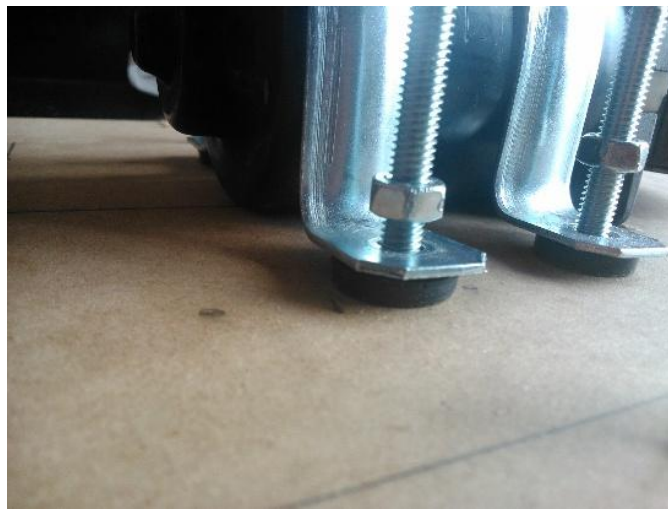


Figura 4.4. Disposición de las gomas anti vibración, abrazaderas y tuercas en la sujeción del motor.

Al colocar estos cuatro tornillos, se tuvo que realizar cuatro orificios en la base principal para que no quedara el hueco correspondiente a la altura de las cabezas de los tornillos entre ambas bases.

Una vez el motor se encontraba fijado a su base, se construyó una caja de madera a su alrededor a partir de tablones de 600x300x10mm con escuadras de hierro latonado en ángulo de 90 grados de 20x20mm (ver figura 4.5).

En un segundo prototipo, toda la electrónica se encontrará también en el interior de éste habitáculo, para reducir el número de componentes que forman PlayCat.



Figura 4.5. Motor anclado a su base y envuelto en una caja de madera.

Para unir esta base del motor a la base principal, se hizo uso de 8 enganches inoxidable (ver figura 4.6). Se fijaron 2 enchanches en cada uno de los lados de la base secundaria.

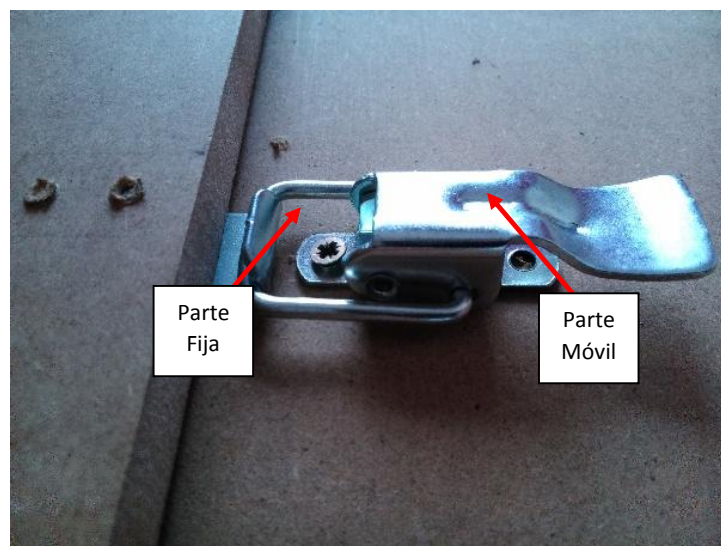


Figura 4.6. Uno de los 8 enganches que unen ambas bases.

En un primer momento, se situó la parte fija de los enganches (ver figura 4.6) en la parte superior de la base secundaria (como se encuentran en la figura 4.7), lo que fue un error, ya que debido al grosor de la tabla de la base secundaria, las parte fija y la parte móvil de los enganches no se encontraban a la misma altura, y esto hacía que los enganches no consiguieran bloquearse cuando la parte móvil (ver figura 4.7) se bajaba.

Desde que se tuvo constancia de este error, se retiraron los enganches y fueron colocados en la parte inferior de la base secundaria, donde ahora los enganches sí cumplían con su cometido por el hecho de que sus dos partes estaban a la misma altura (ver figura 23).

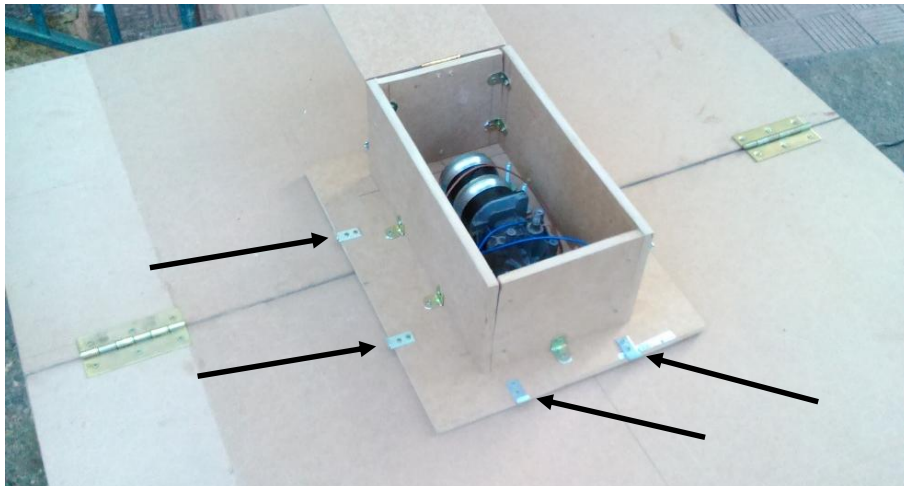


Figura 4.7. Incorrecta situación de la parte fija de los enganches.

4.2.3 Prolongación del eje del motor

Este fue uno de los puntos más problemáticos en el desarrollo físico del prototipo. Lo primero que se probó fue acoplar una estructura de tubos de PVC al eje del motor, pero la unión entre el eje metálico y el tubo de PVC ocasionó problemas de todo tipo, desde la rotura del PVC, hasta un mal resultado del pegamento usado para unir las partes de PVC. Visto esto, se optó por acoplar al eje del motor una varilla roscada de 1m de altura y de 8 mm de diámetro. El eje del motor también era roscado y tenía el mismo

diámetro que la varilla roscada, así que se unieron ambos mediante una tuerca métrica seguida de 3 tuercas para asegurar bien la varilla (ver figura 4.8). Esta unión es suficientemente resistente para soportar las fuerzas a las que se va a someter.



Figura 4.8. Eje roscado del motor (izquierda) y varilla roscada acoplada al eje del motor por medio de una tuerca métrica hexagonal seguida de tres tuercas (derecha).

En la parte superior de esta varilla de un metro de altura, se enlazaría otro tramo de varilla colocado perpendicularmente de 40cm de largo. Para realizar esta unión perpendicular entre dos varillas, no se encontró ningún complemento de unión de 90 grados, por ello se decidió unir ambas varillas mediante una unión por soldadura. Para que la unión fuera más resistente, se introdujo cada una de las puntas de las varillas en una tuerca métrica hexagonal (como la utilizada para unir el eje a la varilla) y fueron soldadas (ver figura 4.9).

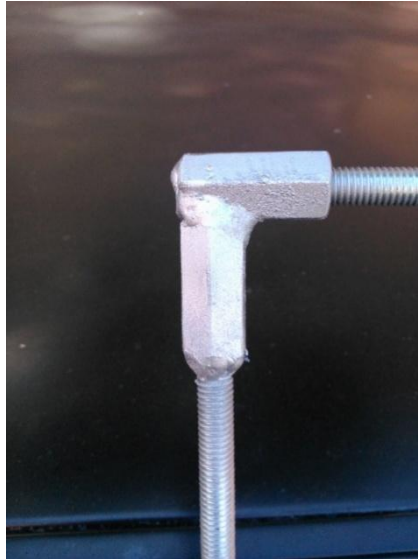


Figura 4.9. Extremos de las dos varillas introducidas en tuercas métricas hexagonales y soldadas.

4.2.4 Forrado de la primera parte del prototipo

Una vez se tenían construidas las estructuras del prototipo mencionadas en los apartados anteriores, se pasó al forrado de todas las partes con una moqueta de punzado azul. Para forrar todo el prototipo, se necesitaron 2 metros cuadrados de dicha moqueta.

En primer lugar, se llevó a cabo el forrado de la base principal. Esta parte del forrado debía tener la característica de ser desmontable, ya que al recoger la base principal (doblándola gracias a las bisagras expuestas en el apartado 4.2.1) la moqueta no podría encontrarse sujeta a la misma. Por ello se recortó a medida la parte correspondiente para la base principal, dejando un hueco rectangular en el centro correspondiente al tamaño de la base secundaria y de la parte fija de los enchanches situados en la base principal (ver figura 4.10).

La moqueta de la base principal se fijaría a ella por medio de 8 tiras de velcro adhesivo colocadas en las esquinas y puntos medios de los lados de la base.

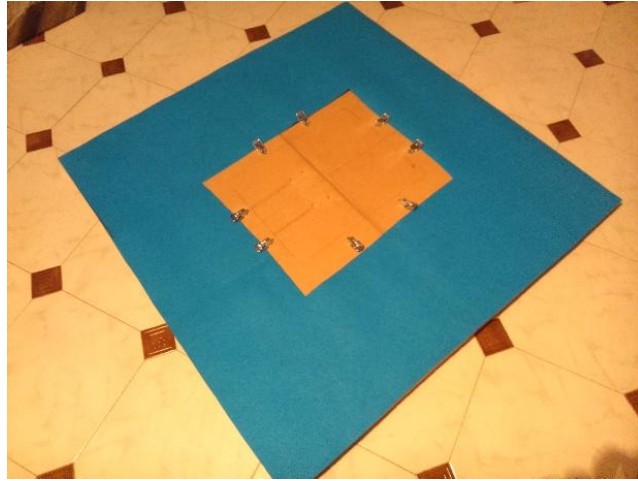


Figura 4.10. Moqueta de la base principal unida mediante tiras de velcro.

Después se pasó a forrar la caja que contenía el motor. En esta ocasión, al igual que para el forrado del resto del prototipo, la moqueta será fijada con cola (ver figura 4.11).



Figura 4.11. Proceso de forrado de la caja de protección del motor.

4.2.5 Escalón

Un problema que había presentado el incorporar una base secundaria era la aparición de los enganches alrededor de la misma (ver figura 4.1), por lo que se tenían que ocultar para que no supusieran un obstáculo para los gatos. Por ello, se construyó un escalón que estaría a una altura entre la base secundaria y la parte media de la caja de protección del motor.

Para la construcción de este escalón intermedio, se midieron todas las distancias, y luego se cortaron con un serrucho y se lijaron los trozos de madera que eran necesarios para construirlo (ver figura 4.12).



Figura 4.12. Partes de madera que formarán la estructura del escalón.

Para el montaje del escalón, se unieron todas las partes usando clavos, y se hizo de forma que la parte superior del cajón se dividiría en 4 partes. Dos de ellas tendrían 3 puntos de apoyo (ver figura 4.13), mientras que las otras dos, que quedarían en medio de las dos primeras, tendrían un solo punto de apoyo. A éstas últimas se les añadió un taco de madera a para que tuvieran un segundo punto de apoyo y así soportar el peso del gato

si éste se apoyara en el escalón (ver figura 4.13). Para la medida de éste taco de madera, se tuvo en cuenta que se apoyaría en la base secundaria.



Figura 4.13. Fabricación del escalón intermedio.

Una vez construido este escalón intermedio, se verificó que las medidas eran correctas y que encajaba perfectamente con la caja que protegía el motor (ver figura 4.14).

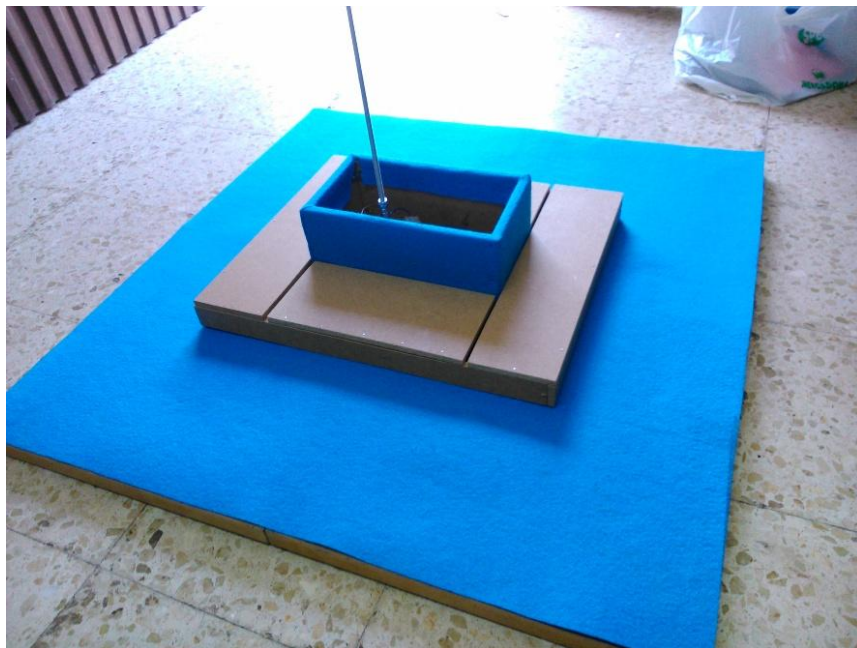


Figura 4.14. Comprobación de las dimensiones del escalón intermedio.

Finalmente, el escalón se forró con la misma moqueta que el resto del prototipo (ver figura 4.15).

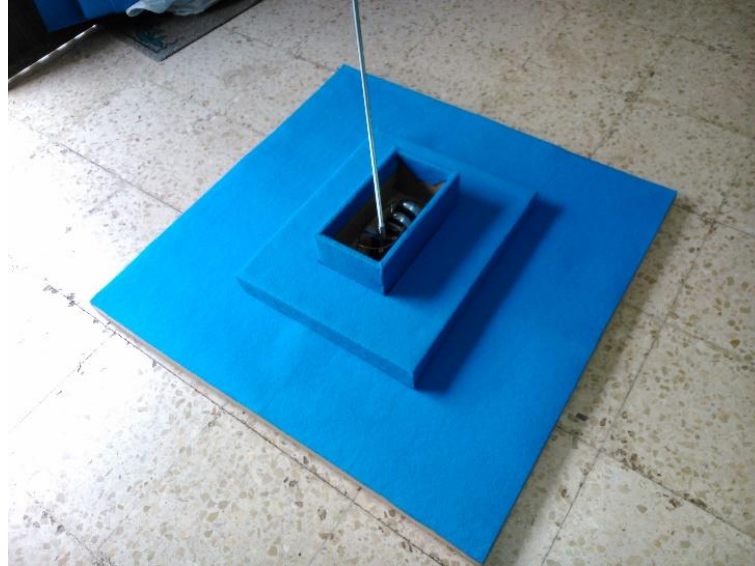


Figura 4.15. Escalón forrado y colocado en el prototipo.

4.2.6 Tapa del motor

La tapa que cerraría la caja del motor debía cumplir dos funciones. Por un lado, la de proteger el motor y aislarlo, y por otro lado, hacer que el motor no rotara, ya que las abrazaderas con las que fue anclado (ver apartado 4.2.2), no permitían su desplazamiento, pero sí dejaban algo de margen para que el motor girara si se aplicaba fuerza sobre la varilla en el sentido transversal del motor.

Por ello, la varilla roscada que partía desde el eje debía poder apoyarse en la tapa, y ésta debía impedir que siguiera desplazándose. A partir de esta idea, se diseñó una tapa que se ajustara debidamente al resto de la caja protectora del motor, pudiera ponerse y retirarse, e imposibilitara que el eje del motor saliera de su posición vertical.

Se decidió que la tapa se ajustaría al resto de la caja de protección del motor mediante dos topes de madera que encajarían internamente con las paredes de la caja, haciendo imposible que la tapa pudiera desplazarse en el eje transversal del motor.

Para que pudiera colocarse y retirarse, se realizó un corte de la forma de la varilla roscada que partía del eje del motor, que iba desde el borde de la tapa hasta donde se encontraría la varilla con la tapa colocada (ver figura 4.16).



Figura 4.16. Diseño de la tapa de la caja de protección del motor.

Una vez todas las partes fueron cortadas, lijadas y unidas mediante clavos, se forró con la misma moqueta que el resto del prototipo (ver figura 4.17).



Figura 4.17. Tapa de la caja del motor forrada y colocada.

4.2.7 Recorrido de los cables del motor

Una vez estaban construidas todas las partes del prototipo, quedaba definir el camino que seguirían los cables del motor desde su caja de protección hasta la centralización de la electrónica.

Para ello, en primer lugar se realizó un orificio en la parte baja de la caja de protección que permitiera el paso de los cables. Para asegurar que los cables no sufrieran tirones en los puntos de conexión, se les realizaron nudos a ambos lados de dicho orificio, centrando así la tensión en estos nudos en vez de en los conectores (ver figura 4.18).



Figura 4.18. Nudos interior y exterior de la caja del motor.

En la parte inferior de la caja de protección, los cables contaban con un conector, al cual se conectarían los cables que procederían de la centralización de la electrónica.

El siguiente obstáculo que debían superar los cables era el escalón intermedio. Para ello, se realizó un corte en la parte inferior del escalón que permitiera el paso de un tubo corrugado de PVC, el cual contendría los cables procedentes de la Monster Moto Shield (ver figura 4.19). Se decidió colocar los cables en el interior de un tubo corrugado de PVC para que ni el gato ni el usuario tuviera acceso a ellos, protegiéndolos de posibles contactos eléctricos.

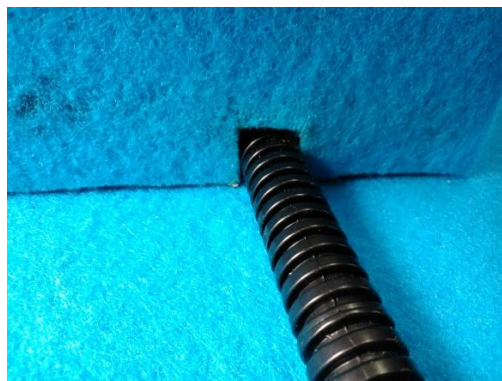


Figura 4.19. Colocación del tubo corrugado de PVC a través del escalón del prototipo.

Capítulo 5. Desarrollo informático

5.1 Lenguajes utilizados en la programación

Como se explicó en el apartado 3.2.2, la programación de arduino está basada en Wiring. Este tipo de programación permite hacer uso de un conjunto de funciones, variables y constantes que encapsulan el funcionamiento del hardware, facilitando el uso del mismo. Además, permite el uso de las características de lenguaje C++ dentro del ambiente de desarrollo, lo que permite crear funciones, punteros, clases y objetos e incluso usar el lenguaje máquina.

Arduino se suele programar fundamentalmente en lenguaje C++, pero cuando se quiere contar con más utilidades, debe añadirse parte del código fuente en otros lenguajes de programación. Para este prototipo, se utilizará el lenguaje C++ para la base del programa, el HTML para crear la interfaz web y establecer la comunicación con ella, y el JavaScript para poder obtener los datos del slider de la interfaz.

5.2 Entorno de desarrollo de arduino

Como se comentó en el apartado 3.2.2, arduino cuenta con su propio entorno de desarrollo, el cual puede ser descargado directamente de forma gratuita desde su página web [14].

Una vez descargado e instalado, se debe configurar el entorno. En primer lugar, se debe seleccionar la tarjeta que se programará, en este caso, un arduino UNO (ver figura 5.1).

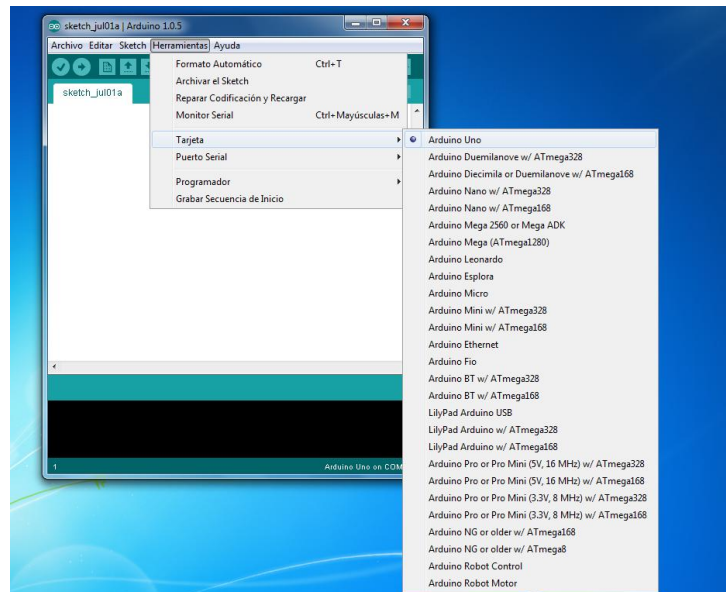


Figura 5.1. Selección de la tarjeta en el entorno de desarrollo de arduino.

En segundo lugar, se debe seleccionar el puerto serial en el que se encuentra conectado el arduino UNO (ver figura 5.2).

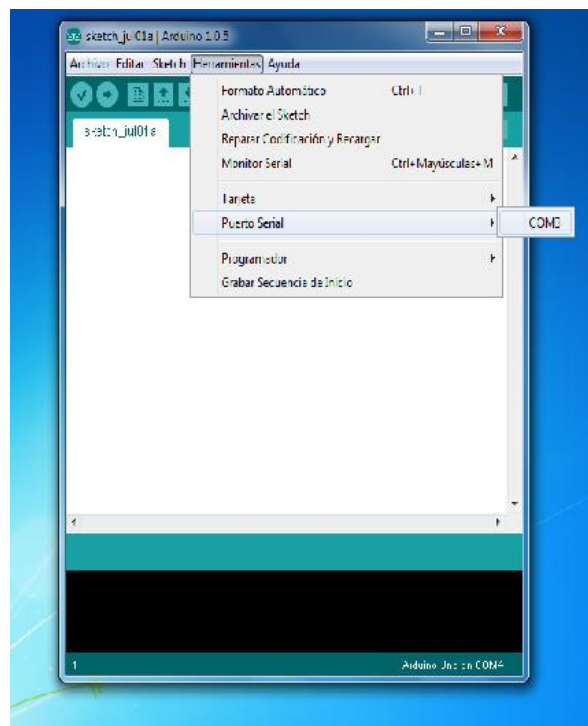


Figura 5.2. Selección del puerto serial.

Y por último, se debe seleccionar el programador, que en este caso será el AVRISP mkII (ver figura 5.3).

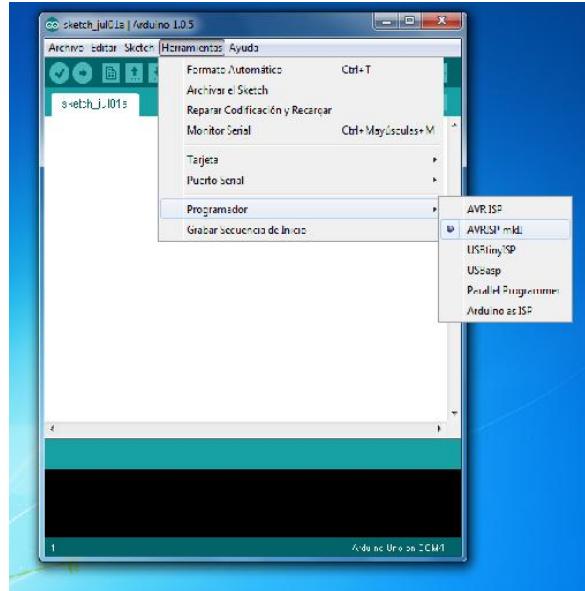


Figura 5.3. Selección del programador.

5.3 Código de programación

Una vez el entorno se encontraba configurado, se pasó a la programación del código. A medida que se fue programando, se le fueron añadiendo comentarios para que se entendiera qué realizaba cada instrucción en futuras modificaciones.

En un principio se pensó en una interfaz basada en botones, en los que el usuario seleccionaría, por medio de 7 botones, la velocidad de giro para un sentido determinado. Sin embargo, tras acabar ese diseño, se optó por tener el control de PlayCat mediante un slider, lo que suponía un gran cambio en la programación. A continuación se muestra el código utilizado para programar el arduino del prototipo con sus respectivos comentarios.

```
// Pines usados por la Ethernet Shield: 10 - 13
// Pines usador por la Monster Moto Shield: 0 - 9
// Pines libres: A0 - A5 (entradas analógicas)

#include <SPI.h> // Librería para comunicar el arduino con dispositivos SPI
#include <Ethernet.h> // Librería para utilizar la Ethernet Shield
byte mac[] = {0xDE,0xAD,0xBE,0xEF,0xFE,0xED}; // Definición de la dirección MAC
IPAddress ip(192,168,1,39); // Definición de la dirección IP del arduino
EthernetServer servidor(80); // Definición de un servidor TCP en el puerto 80

/* Etiquetas para la Monster Moto Shield */

#define BRAKEVCC 0 // Etiqueta para frenar el motor a VCC
#define CW 1 // Etiqueta para que el motor gire en sentido horario
#define CCW 2 // Etiqueta para que el motor gire en sentido anti horario
#define BRAKEGND 3 // Etiqueta para frenar el motor a GND

String readString = String(30); // Cadena para la comunicación
String estado; // Variable para guardar el estado del motor
int pwm = 0; // Variable para introducir la señal PWM en la Monster Moto Shield
int nivel = 0; // Variable para mostrar el nivel de velocidad del motor

/*Declaración de los pines de la Monster Moto Shield */

int inApin[2] = {7, 4}; // Pines de giro horario de ambos motores
int inBpin[2] = {8, 9}; // Pines de giro antihorario de ambos motores
int pwmpin[2] = {5, 6}; // Pines de las entradas PWM de ambos motores
int cspin[2] = {2, 3}; // CS: Entradas analógicas para medir las corrientes
int enpin[2] = {0, 1}; // EN: Salida analógica para cambiar el estado
```

```
/* Configuración inicial del programa */
```

```
void setup(){  
  Ethernet.begin(mac, ip); // Inicialización con las direcciones asignadas  
  servidor.begin(); // Comunicación con el servidor  
  /* Configuración de los pines de la Monster Moto Shield como salidas */  
  for (int i=0; i<2; i++){  
    pinMode(inApin[i], OUTPUT); // Se declara como salida  
    pinMode(inBpin[i], OUTPUT); // Se declara como salida  
    pinMode(pwmpin[i], OUTPUT); // Se declara como salida  
  }  
  motorOff(); // Se para el motor mediante la función motorOff  
  estado = "Parado"; // Se guarda el estado del motor  
  nivel = 0; // Se establece que el nivel de velocidad es 0  
}
```

```
/* Configuración del bucle del programa */
```

```
void loop(){  
  EthernetClient cliente= servidor.available();  
  if(cliente){  
    boolean lineaenblanco = true;  
    while(cliente.connected()){  
      if(cliente.available()){ // Si el cliente está conectado  
        char c = cliente.read(); // Se lee la petición HTTP carácter a carácter  
        if(readString.length()<30){  
          readString.concat(c); /* Se almacenan los caracteres en la cadena  
readString */  
        }  
  
        if(c=='\n' && lineaenblanco){ //Si la petición HTTP ha finalizado
```

```
/* Si se ha pasado el parámetro SPEED estamos ante una petición AJAX
por un cambio en el valor del slider */
int PARAM = readString.indexOf("SPEED=");
if (PARAM != -1) {
    int slider = readString.substring(PARAM+6).toInt(); /* Se le asigna a
readString la cadena que manda el slider de la interfaz. Después se convierte a
entero */

    // Adaptar el valor de velocidad del slider a lo que espera el motor
    int a = 0; // Se crea una variable para guardar el sentido de giro
    if (slider == 4){ // Configuraciones para motor parado si el slider está
en el centro
        motorOff(); // Se para el motor mediante la función motorOff
        estado = "Parado"; // Se guarda el estado del motor
        nivel = 0; // Se guarda el nivel de velocidad del motor
    }

    else if (slider < 4){ /* Configuraciones para girar en CCW si el slider
está a la izquierda del centro */
        a = CCW; // Se establece el sentido de giro
        estado = "Girando en CCW"; // Se guarda el estado del motor
        if (slider == 3){
            pwm = 70; // Se guarda el nivel de la señal PWM
            nivel = 1;} // Se guarda el nivel de velocidad
        else if (slider == 2){
            pwm = 90; // Se guarda el nivel de la señal PWM
            nivel = 2;} // Se guarda el nivel de velocidad
        else if (slider == 1){
            pwm = 100; // Se guarda el nivel de la señal PWM
            nivel = 3;} // Se guarda el nivel de velocidad
        else if (slider == 0){
            pwm = 150; // Se guarda el nivel de la señal PWM
```

```
nivel = 4;} // Se guarda el nivel de velocidad
}

else if (slider > 4){ // Configuraciones para girar en CW si el slider está
a la derecha del centro
    a = CW; // Se establece el sentido de giro
    estado = "Girando en CW"; // Se guarda el estado del motor
    if (slider == 5){
        pwm = 65; // Se guarda el nivel de la señal PWM
        nivel = 1;} // Se guarda el nivel de velocidad
    else if (slider == 6){
        pwm = 80; // Se guarda el nivel de la señal PWM
        nivel = 2;} // Se guarda el nivel de velocidad
    else if (slider == 7){
        pwm = 95; // Se guarda el nivel de la señal PWM
        nivel = 3;} // Se guarda el nivel de velocidad
    else if (slider == 8){
        pwm = 145; // Se guarda el nivel de la señal PWM
        nivel = 4;} // Se guarda el nivel de velocidad
    }
    motorGo(0, a, pwm); /* Se asigna al motor el sentido (a) y la velocidad
de giro (pwm) mediante la función motorGo */

// Respuesta HTTP a la petición AJAX
cliente.println("HTTP/1.1 200 OK"); // Cabecera HTTP estándar
cliente.println("Content-Type: text/plain");
cliente.println();
cliente.println(String(nivel) + ":" + estado); /* Se devuelve el nivel de
velocidad y el estado mediante la cadena, separados por ":" */
cliente.stop(); // Cierre del conexión con el cliente
readString="";
continue;
```

```

}
cliente.println("HTTP_HE/1.1 200 OK"); // Cabecera HTTP estándar
cliente.println("Content-Type: text/html");
cliente.println();

/* Página Web en HTML */
// Si la petición no es AJAX, mostrar la página de la interfaz de usuario

cliente.println("<html>"); // Inicialización código HTML
cliente.println("<head>"); // Inicialización cabecera
cliente.println("<title>Miau Wifi</title>"); // Título de la pestaña

/* Código JavaScript para recoger los datos del slider y hacer las peticiones por
AJAX */
cliente.println("<script language=\"javascript\" type=\"text/javascript\">"
    // Función para hacer una petición Get HTTP por Ajax
    "function ajaxGet(url, onreadystatechange) {"
    "xmlHttp = new XMLHttpRequest();"
    "xmlHttp.onreadystatechange = onreadystatechange;"
    "xmlHttp.open(\"GET\", url, true);"
    "xmlHttp.send( null );"
    "}"
    // Función invocada si se produce un cambio en el slider
    "function sliderOnChange(value) {"
    // Hacer la petición GET para actualizar los valores
    "ajaxGet('./?SPEED='+value, function (){"
    "if (xmlHttp.readyState == 4 && xmlHttp.status == 200) {"
    // Si la petición GET tiene éxito, recuperar el estado del motor
    "result = xmlHttp.responseText.split(\":\");"
    // Se asigna el primer valor de la cadena como "nivel"
    "document.getElementById(\"nivel\").innerHTML=result[0];"
    // Se asigna el segundo valor de la cadena como "estado"

```

```

        "document.getElementById(\"estado\").innerHTML=result[1];"
        "  });"
        "}"
        "</script>");

    cliente.println("</head>"); // Finalización cabecera

    cliente.print("<body bgcolor='blue'>"); /* Se establece el color de fondo
de la interfaz web */

    cliente.println("<center>"); // Texto centrado

    cliente.print("<font size='80' font color='orange'>"); /* Tamaño y color del
título de la web */

    cliente.println("<h1>PlayCat Wifi</h1>"); // Título de la Web

    cliente.print("<font size='80' font color='orange'>"); /* Tamaño y color del
título de "Velocidad" */

    cliente.println("<label for=fader>Velocidad</label>"); /* Se muestra
"Velocidad" */

    cliente.print("<br>"); // Se baja una línea

    cliente.print("<font size='5' font color='white'>"); /* Se establece el
tamaño y color de los elementos indicadores de la velocidad */

    cliente.print("100% <= "); // Límite de la izquierda

    cliente.println("<input type=range min=0 max=8 value=4 id=fader
onchange=\"sliderOnChange(this.value)\">"); // Implementación del slider

    cliente.print(" => 100%"); // Límite de la derecha

    cliente.print("<br>");

    cliente.print("|"); // Signo para mostrar dónde está el cero

    cliente.print("<br>");

    cliente.print("0%"); // Se muestra la ubicación del nivel 0 de velocidad

    cliente.print("<br>");

    cliente.print("<br>");

    cliente.println("</center>"); // Finalización de texto centrado

    cliente.println("<font size='70' font color='white'>"); /* Tamaño y color del
"Estado de PlayCat" */

    cliente.print("Estado de PlayCat: "); // Se muestra "Estado de PlayCat"

```



```
        cliente.print("<font size='70' font color='orange'>"); /* Tamaño y color del
estado del estado */
        cliente.print("<span id=\"estado\">"); /* Se obtiene estado del
identificador */
        cliente.print(estado); // Se muestra el estado
        cliente.print("</span>"); // Se cierra el identificador
        cliente.print("<br>");
        cliente.println("<font size='70' font color='white'>"); /* Color y tamaño del
“Nivel de Velocidad” */
        cliente.print("Nivel de Velocidad: "); // Se muestra “Nivel de Velocidad”
        cliente.print("<font size='70' font color='orange'>"); /* Color y tamaño del
nivel */
        cliente.print("<span id=\"nivel\">"); // Se obtiene nivel del identificador
        cliente.print(nivel); // Se muestra nivel
        cliente.print("</span>"); // Se cierra el identificador
        cliente.println("<center>");
        cliente.print("<br>");
        cliente.print("<br>");
        cliente.print("<br>");
        cliente.print("<a href='http://twitter.com/SamirBatista'
target='_blank'>@SamirBatista</a>"); /* Se habilita un enlace a la cuenta de
Twitter */
        cliente.println("</html>"); // Finalización del código HTML
        cliente.stop(); // Cierre de la conexión con el cliente
        readString="";
    }
}
}
}
}
```

```
void motorOff(){ // Función para frenar el motor
```

```
    for (int i=0; i<2; i++) //Para los pines inA, inB y pwm de cada motor
```

```

{
    digitalWrite(inApin[i], LOW); // Se pone el pinA a baja
    digitalWrite(inBpin[i], LOW); // Se pone el pinB a baja
    analogWrite(pwmpin[i], 0); // Se pone el pwmpin a baja
}
}

```

/* motorGo() establecerá a un motor una dirección de giro a una determinada velocidad, y continuará con esa configuración hasta que se establezca otra o se use la función motorOff para pararlo.

motor: debe ser 0 or 1, seleccionará cuál de los dos motores será controlado

direct: debe estar entre 0 y 3, y tendrá el siguiente resultado:

0: frenar a VCC

1: giro horario

2: giro anti horario

3: frenar a GND

pwm: debe estar en un valor entre 0 y 1023. Cuanto mayor sea el número, más rápido girará el motor */

```

void motorGo(uint8_t motor, uint8_t direct, uint8_t pwm){ /* Función para poner
el motor en marcha (qué motor, qué dirección, qué velocidad) */

```

```

    if (motor <= 1) // Si el motor es el cero o el uno
    {
        if (direct <=4) // Y si la dirección tiene un valor entre 0 y 4
        {
            if (direct <=1){ // Configuración pin inA en función del parámetro direct
                digitalWrite(inApin[motor], HIGH); // Se pone a alta
            }
            else{
                digitalWrite(inApin[motor], LOW); // Se pone a baja
            }
        }
    }
}

```

```
    }  
    if ((direct==0)||direct==2){ /* Configuración pin inB en función del  
parámetro direct */  
        digitalWrite(inBpin[motor], HIGH); // Se pone a alta  
    }  
    else{  
        digitalWrite(inBpin[motor], LOW); // Se pone a alta  
    }  
    analogWrite(pwmpin[motor], pwm); // Se le asigna la señal PWM al motor  
}  
}  
}
```

5.3 Carga del programa

Una vez se terminó de diseñar el código, fue cargado en arduino. Para ello es fundamental que el arduino UNO esté conectado al ordenador mediante un cable USB y que el entorno de desarrollo esté configurado como se indicaba en el apartado 5.2.

Una vez comprobado esto, se carga el programa en la tarjeta (ver figura 5.4). Al seleccionar la opción de cargar, el programa compilará el código, y si se detecta algún fallo de sintaxis, lo mostrará mediante un mensaje en la parte inferior del entorno de desarrollo. Si se carga sin errores, arduino estará listo para la ejecución del programa.



Figura 5.4. Carga del código en arduino.

5.4 Interfaz web

Una vez el programa está cargado en arduino, se deberá encender el prototipo conectando la fuente de alimentación a una toma de corriente. Pasados unos segundos, el router comenzará a generar una red wifi que aparecerá con el nombre de “Miau”. Se deberá conectar el dispositivo wifi con el que se quiera controlar PlayCat a dicha red mediante la clave “tfgsamir”.

Una vez el dispositivo wifi haya establecido la conexión con el router del prototipo, se abrirá el navegador desde el dispositivo wifi. Una vez el navegador esté abierto, se introducirá en la barra de direcciones la dirección IP que le fue configurada a arduino (ver figura 5.5).

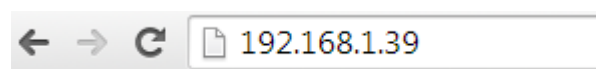


Figura 5.5. Dirección IP de arduino introducida en la barra de dirección del navegador Google Chrome.

Por último, se debe entrar dicha dirección y se cargará automáticamente la interfaz que se ha diseñado en el código del apartado 5.3 (ver figura 5.6).



Figura 5.6. Interfaz web de PlayCat.

Como se observa en la figura 5.6, la interfaz cuenta con un slider, el cual se puede situar en 9 posiciones, cada una de ellas correspondiente a cada una de las velocidades programadas. Cuando el slider se encuentra centrado (situado en la posición que se muestra en la figura 45), el motor estará parado. Si el slider se desplaza hacia la izquierda, el motor girará en sentido anti horario a 4 velocidades distintas de menor a mayor, y si se desplaza hacia la derecha, el motor girará en sentido horario a 4 velocidades distintas de menor a mayor.

A medida que se desplaza el slider a lo largo de su recorrido, se actualiza automáticamente tanto el “Estado de PlayCat”, que puede indicar “Parado”, “Girando en CCW” o “Girando en CW”, como el “Nivel de Velocidad”, que tomará valores enteros entre 0 y 4.

Capítulo 6. Presupuesto del prototipo

Artículo	Precio (€)	Unidades	Total
Arduino UNO R3 ATmega328	8,46	1	8,46
Ethernet Shield	7,50	1	7,50
Monster Moto Shield	18,88	1	18,88
Router Wifi	13,05	1	13,05
Fuente de alimentación 12V 15A / 180W	18,40	1	18,40
Motor DC limpiaparabrisas Peugeot 306	15,00	1	15,00
Cable H07V-K 1,5mm ²	0,11	6	0,66
Cable Ethernet RJ45 90 cm	1,00	1	1,00
Tubo corrugado PVC 16/2,5m	1,22	1	1,22
Conector de dos entradas para soldar 1,5mm ²	0,50	3	1,50
Bloque de 8 conectores	0,75	1	0,75
Tablón de Madera 1000x1000x40 mm	8,51	1	8,51
Tablón de Madera 600x300x10 mm	2,45	3	7,35
30 Tornillos tirafondo 3,5x16 SPAX	2,22	1	2,22
Escuadra de hierro latonado en ángulo 20x20mm	0,16	14	2,24
Bisagra recta hierro latonado	1,79	2	3,58
4 fieltros adhesivos 50mm beige	1,45	1	1,45
Varilla roscada a Zinc diam 8mm 1m	1,18	1	1,18
4 tuercas métricas hexagonal 8mm	2,05	1	2,05
2 varillas roscadas DM8 20 CM + 4 Tuercas + 4 Arandelas	3,10	1	3,10
Abrazadera pared D60mm	0,50	2	1,00
Moqueta punzado azul 1m ²	3,35	2	6,70
Velcro adhesivo 1m	1,20	1	1,20
Enganche inoxidable + tornillos	1,25	8	10,00
Cola para pegar	2,50	1	2,50
Mano de obra (hora)	19,50	24	468
		Total (€)	607,50

Por lo que ejecución del prototipo han tenido un coste total de 607,50 €.

Capítulo 7. Estudio de viabilidad económica

A partir del interés social que ha tenido PlayCat, se decidió desarrollar un estudio de viabilidad económica para ver qué ganancias se podrían producir al comercializar PlayCat. Algo fundamental para ello, es hacer un diseño específico de la electrónica y fabricar la estructura de PlayCat a partir de plástico (ver capítulo 8). Para esto último, se contratarían los servicios de ProtoRapid, una empresa de Barcelona que se dedica a la fabricación rápida de piezas y prototipos. Se les envía un diseño de del producto y responden enviando el presupuesto.

Haciendo una aproximación, tanto los costes como los beneficios que implicaría la fabricación del producto serían los mostrados en la tabla 8.1.

Electrónica	25,00 €
Fuente de alimentación	10,00 €
Motor DC	10,00 €
Estructura plástica más envío	20,00 €
Moqueta y otros	5,00 €
Mano de obra	12,00 €
Coste de producción	82,00 €
Beneficios (40 %)	24,60 €
IVA (21 %)	22,39 €

Total	128,99 €
Precio final	129,95 €

Tabla 8.1. Desglose de costes por unidad.

Además, habría unos costes iniciales por parte del molde de inyección, que tendrá un coste aproximado de 6000€, las herramientas necesarias para el montaje del producto (400€) y los materiales para la producción de 60 unidades (4200€). Además será necesario alquilar un pequeño taller para el montaje del producto (unos 400 € mensuales).

Teniendo en cuenta esos costes, y viendo la producción que se podría llevar a cabo por montador (ver tabla 8.2), podemos hacer una proyección de los beneficios netos que produciría PlayCat. Dichos beneficios se encontrarían en una franja limitada por las proyecciones pesimista y optimista.

Producción diaria del montador	3
Producción mensual del montador	60
Producción mensual con 2 montadores	126
Producción mensual con 3 montadores	192
Producción mensual con 4 montadores	258

Tabla 8.2. Producción en función del número de montadores.

En la proyección pesimista, se ha supuesto que los 8 primeros meses se tendría contratado un solo montador, el cual montaría 60 unidades al mes, y los 16 siguientes meses trabajarían dos montadores, que producirían 126 unidades mensuales. Por otro lado, en la proyección optimista, se consideró que se contrataría un montador durante los 4 primeros meses, pasando a ser 2 montadores durante los 4 meses siguientes, 3 montadores en los 4 meses siguientes y 4 montadores en el segundo año.

En la tabla 8.3 se muestran los beneficios que se obtienen teniendo contratados a 1, 2, 3 y 4 montadores si se venden todas las unidades que producen.

Beneficios mensuales 1 montador	1.133,84 €
Beneficios mensuales 2 montadores	2.821,06 €
Beneficios mensuales 3 montadores	4.508,29 €
Beneficios mensuales 4 montadores	6.195,51 €

Tabla 8.3. Beneficios mensuales obtenidos según el número de montadores contratados.

En la figura 8.1 se muestra la cantidad de unidades que serían vendidas tanto para la proyección pesimista como para la optimista.

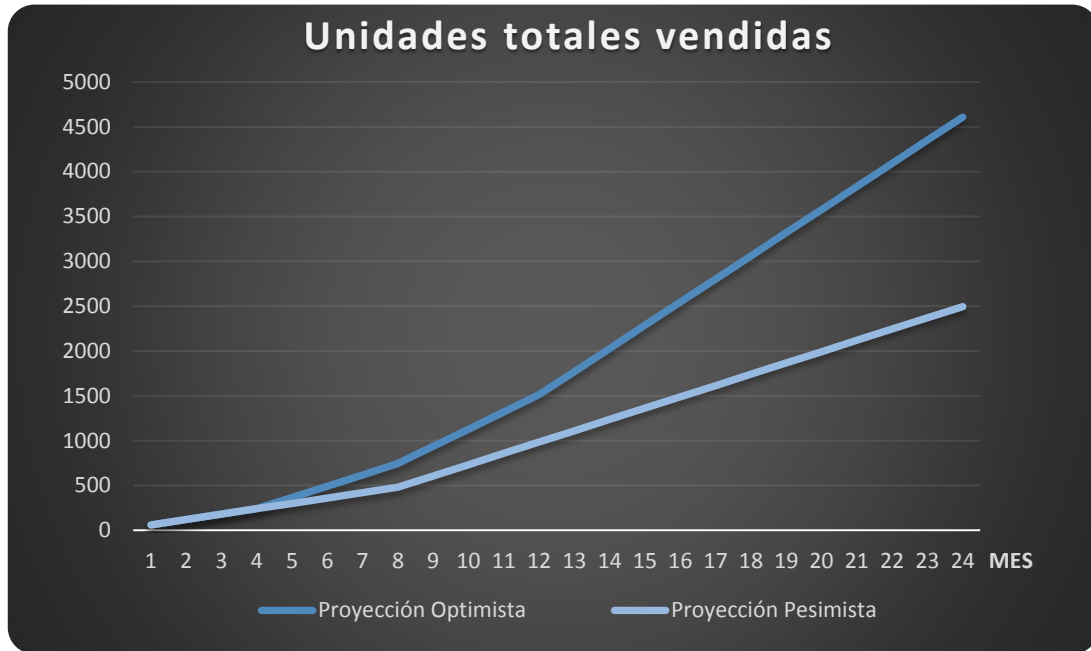


Figura 8.1. Estimación de unidades vendidas en los dos primeros años.

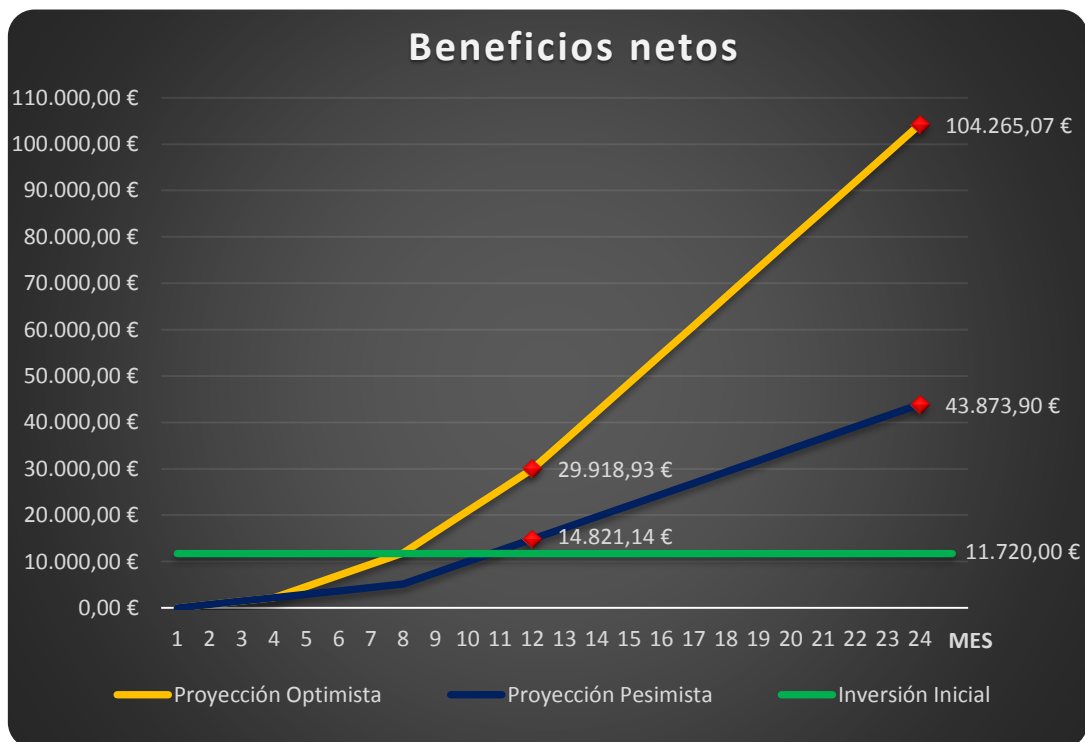


Figura 8.2. Beneficios netos en los 24 primeros meses.

Como se observa en la figura 8.2, la inversión inicial se recupera en 8 meses en el caso de la proyección pesimista, mientras que en la optimista, en 11 meses.

Por otro lado, la comercialización de este producto podría generar unos beneficios entre 15.000€ y 30.000€ en los 12 primeros meses, y pasados dos años, entre 40.000 y 100.000€.

Viendo los beneficios que podría generar la comercialización de PlayCat (ver figura 8.1), se puede deducir que es un proyecto perfectamente viable si se ejecuta de la forma adecuada y se da a conocer el producto.

Capítulo 8. Conclusiones y líneas futuras

El objetivo principal de este proyecto fue diseñar un centro de juegos para gatos que pudiera ser controlado con un smartphone. El juguete se basa en un mecanismo capaz de rotar del que se cuelga algo atractivo para los gatos. En el caso de que el gato consiguiera atrapar el objeto colgado, el mecanismo puede ejercer un mayor par y conseguir quitárselo. Por ello, el diseño estuvo centrado en un motor, la forma de alimentarlo y controlarlo, y cómo realizar ese control mediante un smartphone.

El motor seleccionado fue un motor de corriente continua, controlado mediante la plataforma de código abierto Arduino. Aunque Arduino no tenía la capacidad de alimentar directamente al motor, debido a que demandaba más corriente de la que Arduino podía producir, se pudo llevar a cabo por medio del uso de puentes de transistores de efecto campo. La comunicación con el smartphone se hizo por medio de tecnología wifi. Esta comunicación fue posible por la interacción de un router wifi con arduino por medio de la Ethernet Shield. Todos los componentes del sistema se han alimentado con una fuente de alimentación de un ordenador.

Como resultado final, el prototipo montado es capaz de realizar todas las funciones que han sido proyectadas. Desde cualquier dispositivo con wifi se puede controlar tanto el sentido de giro como la velocidad a la que se mueve. Además, su diseño visual es bastante impactante, dado que se decidió forrarlo en su totalidad para hacerlo más vistoso. Otro detalle muy importante que se tuvo en cuenta fue el de no dejar ningún elemento que pudiera suponer un obstáculo para el gato, ni ningún elemento eléctrico que pudiera estar al alcance del mismo.

En este proyecto se han aplicado conocimientos adquiridos en numerosas asignaturas del Grado en Ingeniería Electrónica Industrial y Automática. Algunas de ellas han estado presentes de forma más directa, como Informática, Fundamentos de Ingeniería Electrónica, Fundamentos de Ingeniería Eléctrica, Electrónica Digital, Mecánica de Máquinas y Automatización Industrial. Y otras de forma más indirecta, como

Fundamentos Matemáticos, Física I, Procesos de Fabricación y Ciencia y Tecnología de Materiales entre otras. Además, se tuvieron que adquirir conocimientos que no estaban presentes en el plan de estudios del Grado, como el uso y programación de Arduino y sus complementos, programación en HTML o procesos para el diseño y construcción física de prototipos.

Paralelamente al desarrollo del prototipo para el Trabajo de Fin de Grado, PlayCat también experimentó una madurez comercial durante varias sesiones en el Proyecto IDEAS. Durante esta etapa, se trabajó sobre la propuesta de valor que poseía el producto, las necesidades que cubriría, los canales por los que se daría a conocer, los segmentos de clientes a los que estaría destinado y una estructura de costes e ingresos entre otros puntos. En junio de 2014, PlayCat fue presentado en el Foro Ideas junto al resto de proyectos que se habían formado en el Proyecto IDEAS [15], donde obtuvo el primer premio. Además, obtuvo otro un premio por parte del Parque Científico y Tecnológico de Tenerife, en el que se premiaba con 6 meses de alojamiento y asesoramiento en sus viveros para continuar con el desarrollo de PlayCat.

Visto el interés social que ha causado el producto y la oportunidad que se le ha dado, se seguirá con su desarrollo. En primer lugar, el prototipo experimentará una segunda fase en la que se incorporarán nuevas funciones, como la detención de presencia mediante sensores de ultrasonidos o el diseño de una aplicación móvil para su control (en lugar de la interfaz web con la que cuenta actualmente) que sea capaz de informar al usuario en cualquier momento y desde cualquier lugar si su gato se encuentra jugando con PlayCat y el tiempo que ha estado jugando durante el día. El uso de esta aplicación conllevará una suscripción de pago mensual de bajo coste, que supondrá una fuente directa de ingresos.

Cuando estas nuevas incorporaciones sean totalmente funcionales, se llevará a cabo el diseño de un segundo prototipo. En este caso sí se hará un diseño específico para el prototipo, el cual dispondrá, por un lado, de una fuente de alimentación de menor tamaño que proporcione exactamente la potencia que demanda el sistema, y por otro lado, de una

placa base en la que se encontrará centralizada toda la electrónica. Algunos de los componentes principales de esta placa base serán el microcontrolador que mejor se ajuste a sus necesidades, un puente de transistores de efecto campo para la ampliación de corriente, terminales para la conexión de la fuente de alimentación, del motor y de los sensores de ultrasonido, y un módulo que permita conectarse directamente al router del usuario por medio de una conexión wifi. Además, la estructura pasará a ser de plástico en lugar de madera, reduciendo así costes y haciéndola más ligera.

Conclusions and future lines

The main goal of this project was to design a cats' play center which could be controlled with a smartphone. The play center is based on a rotation mechanism holding something attractive for cats. In the case that the cat got the hanging object, the mechanism should improve its power and take it off. Therefore the design was focused on the motor, its supply its control, and how to make this control from a smartphone.

The selected motor was a DC Motor, controlled with the Arduino open source platform. Arduino didn't have the capacity to supply the motor directly, because the motor needs more current than the produced by Arduino. This current was produced with field effect transistor bridges. The communication with the smartphone was made with Wi-Fi technology. This communication was possible by the interaction of a Wi-Fi router with Arduino through the Ethernet Shield. All the system components were supplied with a computer's power source.

The final prototype is able to carry out all the planned functionalities. The rotation direction and the movement velocity can be controlled from any Wi-Fi device. Furthermore its visual design is very shocking, because the prototype has been totally covered for being more attractive. The design does not contain any element that could be an obstacle for the cat or any reachable electric element.

In this project knowledge acquired in several subjects of the Automatic and Industrial Electronic Engineering Degree has been applied. Directly, like Computer Science, Electronic Engineering, Electric Engineering, Digital Electronic, Machine Mechanism and Industrial Automation, and indirectly, like Mathematics, Physics, Manufacturing Processes and Material Technology and Science. Additionally, it was necessary to acquire new knowledge: Arduino use and programming, HTML programming and design and building techniques for prototypes.

PlayCat prototype has experienced commercial improvements through some sessions of “Proyecto IDEAS”. During this phase some important aspects were analyzed: the product value, the needs that it covered the marketing channels, the customers’ profiles, costs and benefits, among others. In June 2014 Playcat won the first prize of the “Foro Ideas” and a prize of Parque Científico y Tecnológico de Tenerife (accommodation and advice for 6 months in their offices to continue the development of Playcat) [15].

Due to the social interest originated by the prototype and the given opportunity, its development will continue. In the next development phase new functions will be incorporated: presence detection with ultrasonic sensors and the design of a mobile phone app to control it (instead of the current web interface). This app will inform the user everywhere and every time if his cat is playing with PlayCat and for how long. The use of this app will have a monthly cost, an additional benefit.

For the second prototype a specific electronic design will be carried out. The prototype will have the power supply source that exactly generates the power needed by the system. All the electronic systems will be placed in the same electronic board. The microcontroller and the field effect transistor bridge that best achieve the system requirements will be chosen, new terminals for the motor, supply and ultrasonic sensors connections will be added, and a Wi-Fi module for connecting directly with the user router will be analyzed. Additionally, the structure will be built with plastic instead of wood, to reduce the cost and the weight.

Capítulo 9. Bibliografía

- [1] Jerarquía de necesidades de Maslow. www.alquimistas.evilnolo.com/2006/01/30/el-diseno-y-su-jerarquia-de-necesidades-aplicadas-al-nintendo-revolution/
- [2] Figura de centro de juego para gatos. www.img.mx.class.posot.com/es_mx/2014/05/27/Centro-De-Juego-Y-Rascadero-Para-Gatos-20140527103027.jpg
- [3] Figura de DART. www.thinkgeek.com/product/dcd0
- [4] Datos de gastos de los españoles en sus mascotas. www.consumer.es/web/es/mascotas/perros/convivencia-y-psicologia/convivencia/2013/10/17/218036.php
- [5] Datos del Motor Limpiaparabrisas. www.spanish.alibaba.com/product-gs/zd1003-wiper-motor-for-ford-12v-dc-motor-721333715.html?s=p
- [6] Información de Arduino. www.arduino.cc/es
- [7] Figura de Arduino UNO. www.seeedstudio.com
- [8] Características de Arduino UNO. www.ardumania.es/ardutienda/es/arduino/11-arduino-uno.html
- [9] Figura de la Monster Moto Shield. www.robot-domestici.it
- [10] Figura de la Monster Moto Shield con Arduino UNO. www.labdegaragem.com
- [11] Tabla de configuración de los pines de la Monster Moto Shield. www.labdegaragem.com/profiles/blogs/tutorial-como-utilizar-o-monster-motor-shield
- [12] Figura de la Ethernet Shield. www.miniinthebox.com
- [13] Figura de los pines de la fuente de alimentación de un PC. www.luisllamas.es/2014/01/fuente-de-alimentacion-casera-con-fuente-atx-1-de-2
- [14] Arduino website. www.arduino.cc/Main/Software
- [15] Vídeo de la presentación en el foro IDEAS 2014 (PlayCat a partir del minuto 31). www.vimeo.com/99790539



Universidad
de La Laguna
Escuela Superior de
Ingeniería y Tecnología
Sección de Ingeniería Industrial



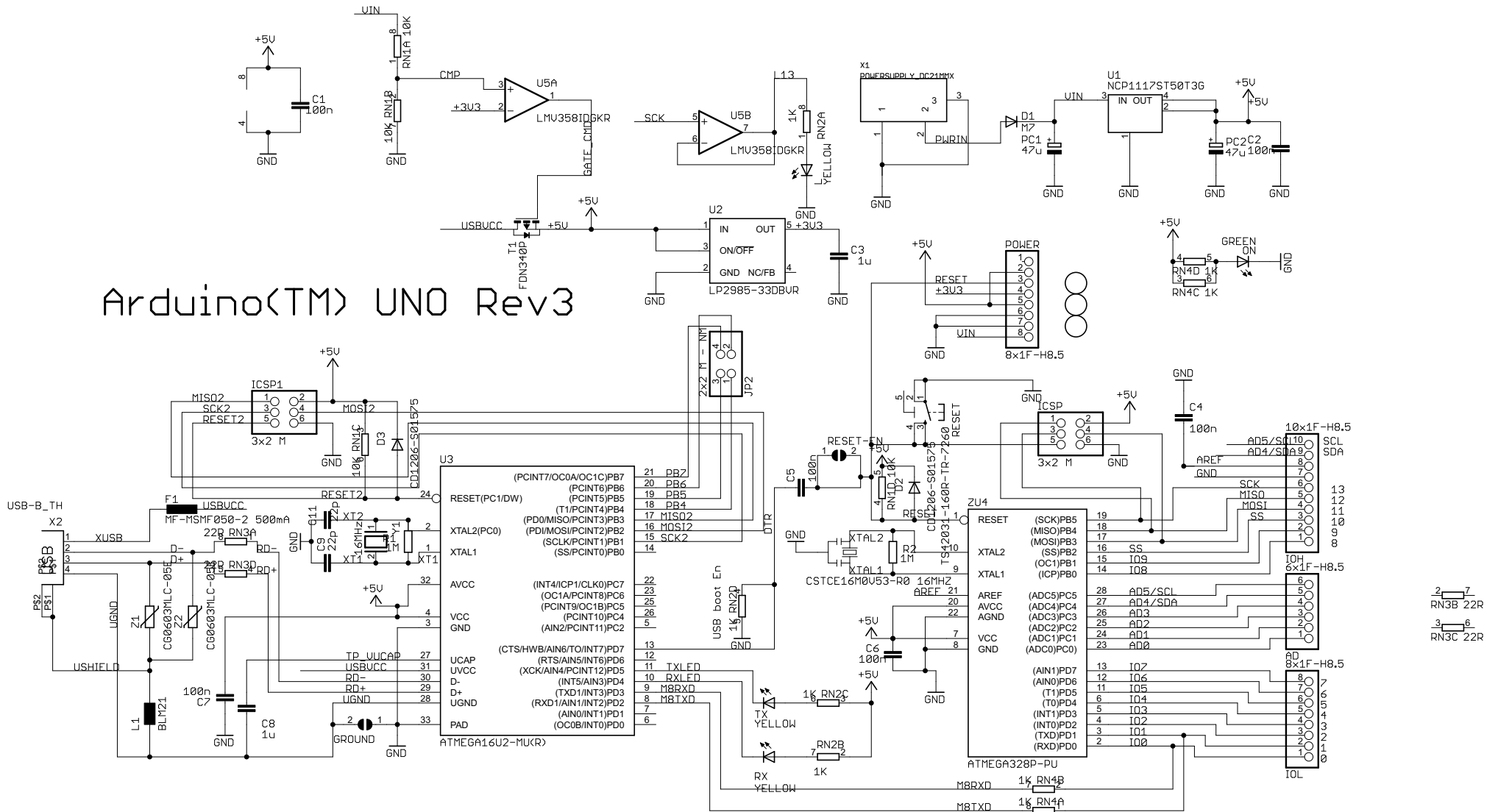
Universidad
de La Laguna

Escuela Superior de
Ingeniería y Tecnología

Escuela Superior de Ingeniería y Tecnología

ANEXO I

ESQUEMÁTICO DE ARDUINO UNO



Reference Designs ARE PROVIDED "AS IS" AND "WITH ALL FAULTS. Arduino 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. Arduino 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." Arduino 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.

ARDUINO is a registered trademark.

Use of the ARDUINO name must be compliant with <http://www.arduino.cc/en/Main/Policy>



Universidad
de La Laguna
Escuela Superior de
Ingeniería y Tecnología
Sección de Ingeniería Industrial



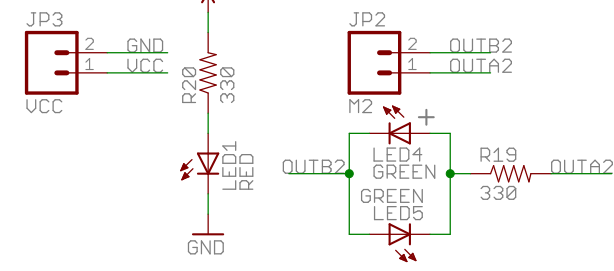
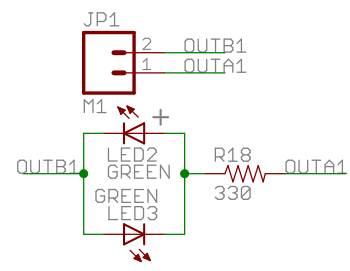
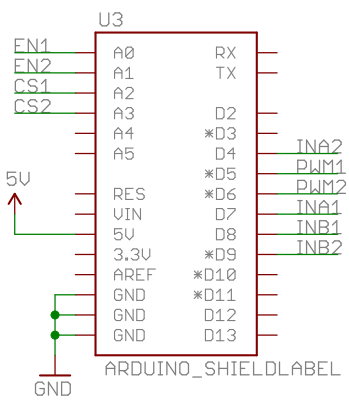
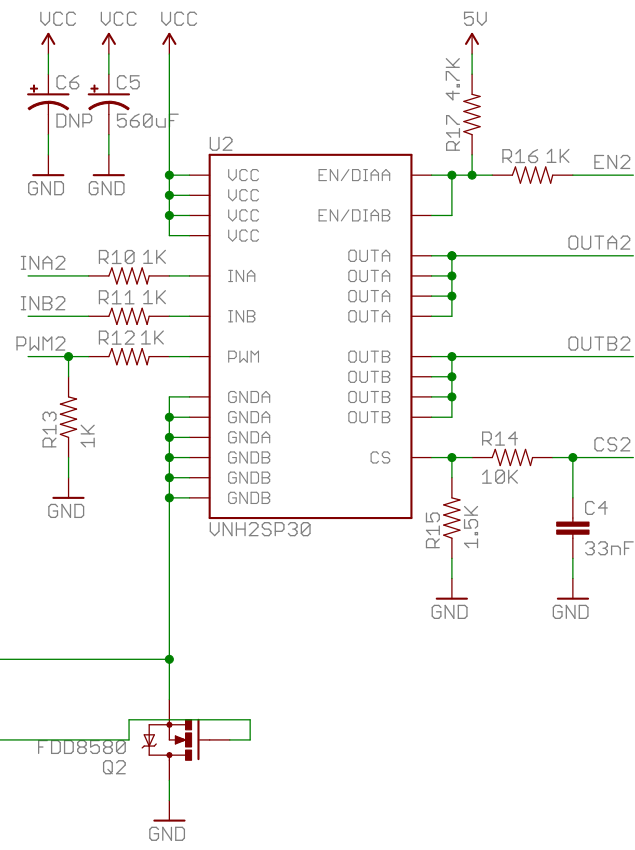
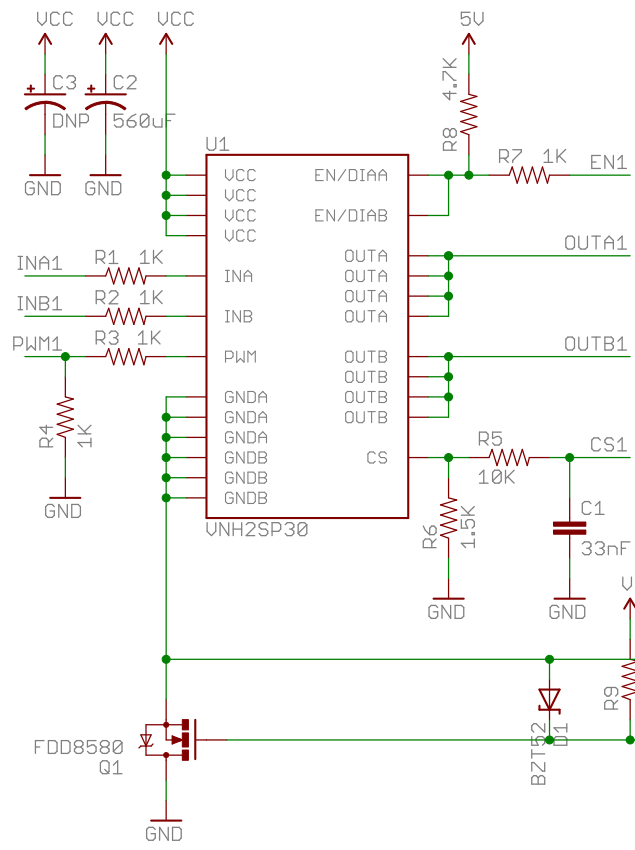
Universidad
de La Laguna

Escuela Superior de
Ingeniería y Tecnología

Escuela Superior de Ingeniería y Tecnología

ANEXO II

ESQUEMÁTICO DE MONSTER MOTO SHIELD



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

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



Universidad
de La Laguna
Escuela Superior de
Ingeniería y Tecnología
Sección de Ingeniería Industrial



Universidad
de La Laguna

Escuela Superior de
Ingeniería y Tecnología

Escuela Superior de Ingeniería y Tecnología

ANEXO III

DATASHEET VNH2SP30-E

Automotive fully integrated H-bridge motor driver

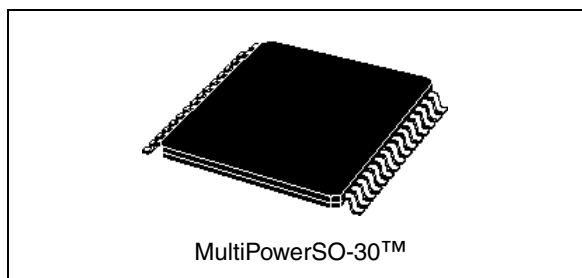
Features

Type	$R_{DS(on)}$	I_{out}	V_{CCmax}
VNH2SP30-E	19m Ω max (per leg)	30A	41V

- 5V logic level compatible inputs
- Undervoltage and overvoltage shut-down
- Overvoltage clamp
- Thermal shut down
- Cross-conduction protection
- Linear current limiter
- Very low stand-by power consumption
- PWM operation up to 20 kHz
- Protection against loss of ground and loss of V_{CC}
- Current sense output proportional to motor current
- Package: ECOPACK[®]

Description

The VNH2SP30-E is a full bridge motor driver intended for a wide range of automotive applications. The device incorporates a dual monolithic high side driver and two low side switches. The high side driver switch is designed using STMicroelectronic's well known and proven proprietary VIPower™ M0 technology which permits efficient integration on the same die of a true Power MOSFET with an intelligent signal/protection circuitry.



The low side switches are vertical MOSFETs manufactured using STMicroelectronic's proprietary EHD ('STripFET™') process. The three die are assembled in the MultiPowerSO-30 package on electrically isolated leadframes. This package, specifically designed for the harsh automotive environment offers improved thermal performance thanks to exposed die pads. Moreover, its fully symmetrical mechanical design allows superior manufacturability at board level. The input signals IN_A and IN_B can directly interface to the microcontroller to select the motor direction and the brake condition. The $DIAG_A/EN_A$ or $DIAG_B/EN_B$, when connected to an external pull-up resistor, enable one leg of the bridge. They also provide a feedback digital diagnostic signal. The normal condition operation is explained in [Table 12: Truth table in normal operating conditions on page 14](#). The motor current can be monitored with the CS pin by delivering a current proportional to its value. The speed of the motor can be controlled in all possible conditions by the PWM up to 20 kHz. In all cases, a low level state on the PWM pin will turn off both the LS_A and LS_B switches. When PWM rises to a high level, LS_A or LS_B turn on again depending on the input pin state.

Table 1. Device summary

Package	Order codes	
	Tube	Tape and Reel
MultiPowerSO-30	VNH2SP30-E	VNH2SP30TR-E

Contents

- 1 Block diagram and pin description 5**
- 2 Electrical specifications 8**
 - 2.1 Absolute maximum ratings 8
 - 2.2 Electrical characteristics 9
 - 2.3 Electrical characteristics curves 16
- 3 Application information 20**
 - 3.1 Reverse battery protection 21
- 4 Package and PCB thermal data 25**
 - 4.1 PowerSSO-30 thermal data 25
 - 4.1.1 Thermal calculation in clockwise and anti-clockwise operation in steady-state mode 26
 - 4.1.2 Thermal resistances definition (values according to the PCB heatsink area) 26
 - 4.1.3 Thermal calculation in transient mode 26
 - 4.1.4 Single pulse thermal impedance definition (values according to the PCB heatsink area) 26
- 5 Package and packing information 29**
 - 5.1 ECOPACK® packages 29
 - 5.2 MultiPowerSO-30 package mechanical data 29
 - 5.3 Packing information 31
- 6 Revision history 32**

List of tables

Table 1.	Device summary	1
Table 2.	Block description	5
Table 3.	Pin definitions and functions	6
Table 4.	Pin functions description	7
Table 5.	Absolute maximum ratings	8
Table 6.	Power section	9
Table 7.	Logic inputs (INA, INB, ENA, ENB)	9
Table 8.	PWM	10
Table 9.	Switching ($V_{CC} = 13V$, $R_{LOAD} = 0.87W$, unless otherwise specified)	10
Table 10.	Protection and diagnostic	10
Table 11.	Current sense ($9V < V_{CC} < 16V$)	11
Table 12.	Truth table in normal operating conditions	14
Table 13.	Truth table in fault conditions (detected on OUTA)	14
Table 14.	Electrical transient requirements	15
Table 15.	Thermal calculation in clockwise and anti-clockwise operation in steady-state mode	26
Table 16.	Thermal parameters	28
Table 17.	MultiPowerSO-30 mechanical data	30
Table 18.	Document revision history	32

List of figures

Figure 1.	Block diagram	5
Figure 2.	Configuration diagram (top view)	6
Figure 3.	Current and voltage conventions	8
Figure 4.	Definition of the delay times measurement	11
Figure 5.	Definition of the low side switching times	12
Figure 6.	Definition of the high side switching times	12
Figure 7.	Definition of dynamic cross conduction current during a PWM operation.	13
Figure 8.	On state supply current.	16
Figure 9.	Off state supply current.	16
Figure 10.	High level input current.	16
Figure 11.	Input clamp voltage.	16
Figure 12.	Input high level voltage	16
Figure 13.	Input low level voltage	16
Figure 14.	Input hysteresis voltage	17
Figure 15.	High level enable pin current	17
Figure 16.	Delay time during change of operation mode	17
Figure 17.	Enable clamp voltage	17
Figure 18.	High level enable voltage	17
Figure 19.	Low level enable voltage	17
Figure 20.	PWM high level voltage	18
Figure 21.	PWM low level voltage	18
Figure 22.	PWM high level current.	18
Figure 23.	Overshoot shutdown	18
Figure 24.	Undervoltage shutdown	18
Figure 25.	Current limitation.	18
Figure 26.	On state high side resistance vs Tcase	19
Figure 27.	On state low side resistance vs Tcase	19
Figure 28.	Turn-On delay time	19
Figure 29.	Turn-Off delay time	19
Figure 30.	Output voltage rise time	19
Figure 31.	Output voltage fall time	19
Figure 32.	Typical application circuit for DC to 20 kHz PWM operation short circuit protection	20
Figure 33.	Behavior in fault condition (How a fault can be cleared).	21
Figure 34.	Half-bridge configuration.	22
Figure 35.	Multi-motors configuration	22
Figure 36.	Waveforms in full bridge operation	23
Figure 37.	Waveforms in full bridge operation (continued)	24
Figure 38.	MultiPowerSO-30™ PC board	25
Figure 39.	Chipset configuration	25
Figure 40.	Auto and mutual Rthj-amb vs PCB copper area in open box free air condition	25
Figure 41.	MultiPowerSO-30 HSD thermal impedance junction ambient single pulse	27
Figure 42.	MultiPowerSO-30 LSD thermal impedance junction ambient single pulse	27
Figure 43.	Thermal fitting model of an H-bridge in MultiPowerSO-30	28
Figure 44.	MultiPowerSO-30 package outline	29
Figure 45.	MultiPowerSO-30 suggested pad layout	30
Figure 46.	MultiPowerSO-30 tube shipment (no suffix)	31
Figure 47.	MultiPowerSO-30 tape and reel shipment (suffix "TR")	31

1 Block diagram and pin description

Figure 1. Block diagram

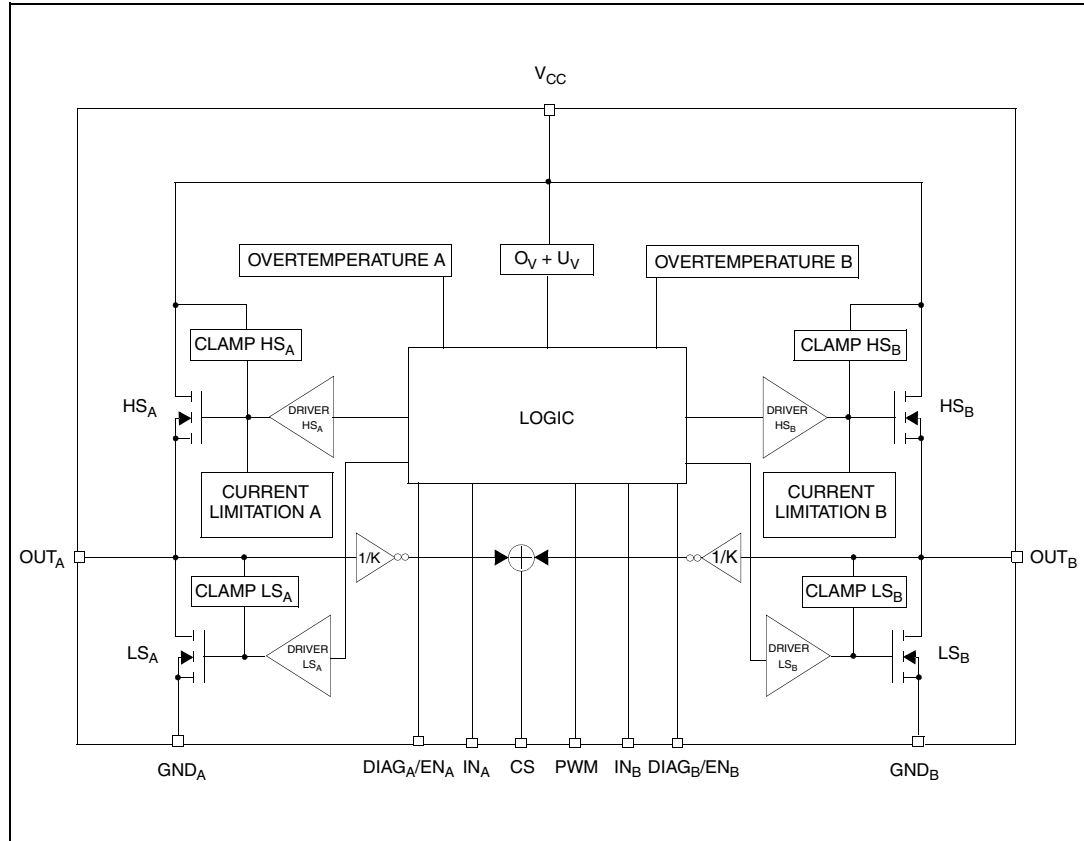


Table 2. Block description

Name	Description
Logic control	Allows the turn-on and the turn-off of the high side and the low side switches according to the truth table
Overvoltage + undervoltage	Shuts down the device outside the range [5.5V..16V] for the battery voltage
High side and low side clamp voltage	Protects the high side and the low side switches from the high voltage on the battery line in all configurations for the motor
High side and low side driver	Drives the gate of the concerned switch to allow a proper $R_{DS(on)}$ for the leg of the bridge
Linear current limiter	Limits the motor current by reducing the high side switch gate-source voltage when short-circuit to ground occurs
Overtemperature protection	In case of short-circuit with the increase of the junction's temperature, shuts down the concerned high side to prevent its degradation and to protect the die
Fault detection	Signals an abnormal behavior of the switches in the half-bridge A or B by pulling low the concerned EN_x /DIAG _x pin

Figure 2. Configuration diagram (top view)

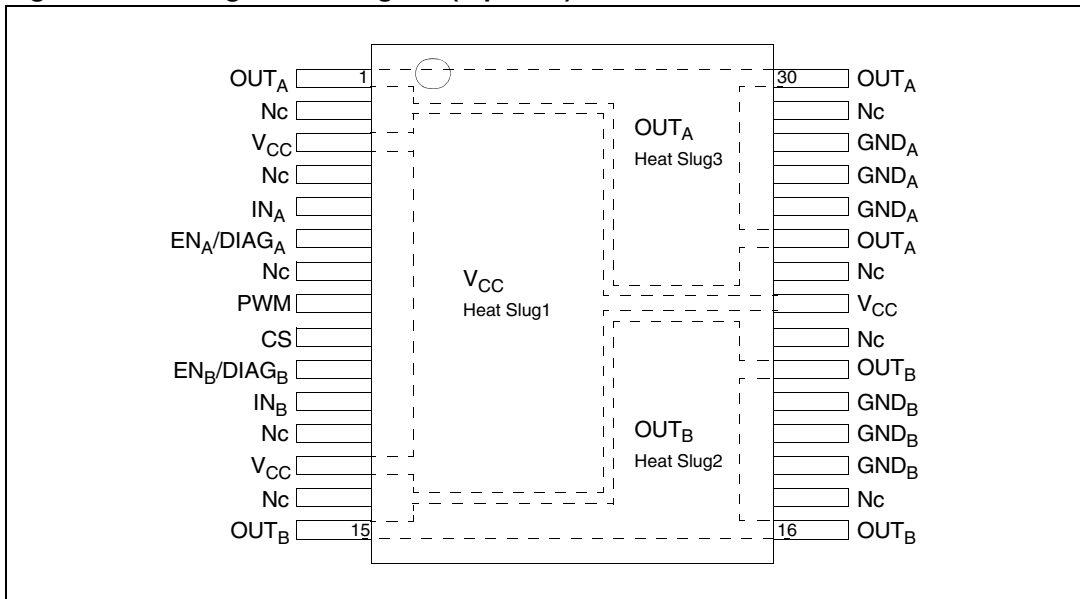


Table 3. Pin definitions and functions

Pin No	Symbol	Function
1, 25, 30	OUT _A , Heat Slug3	Source of high side switch A / Drain of low side switch A
2, 4, 7, 12, 14, 17, 22, 24, 29	NC	Not connected
3, 13, 23	V _{CC} , Heat Slug1	Drain of high side switches and power supply voltage
6	EN _A /DIAG _A	Status of high side and low side switches A; open drain output
5	IN _A	Clockwise input
8	PWM	PWM input
9	CS	Output of current sense
11	IN _B	Counter clockwise input
10	EN _B /DIAG _B	Status of high side and low side switches B; open drain output
15, 16, 21	OUT _B , Heat Slug2	Source of high side switch B / Drain of low side switch B
26, 27, 28	GND _A	Source of low side switch A ⁽¹⁾
18, 19, 20	GND _B	Source of low side switch B ⁽¹⁾

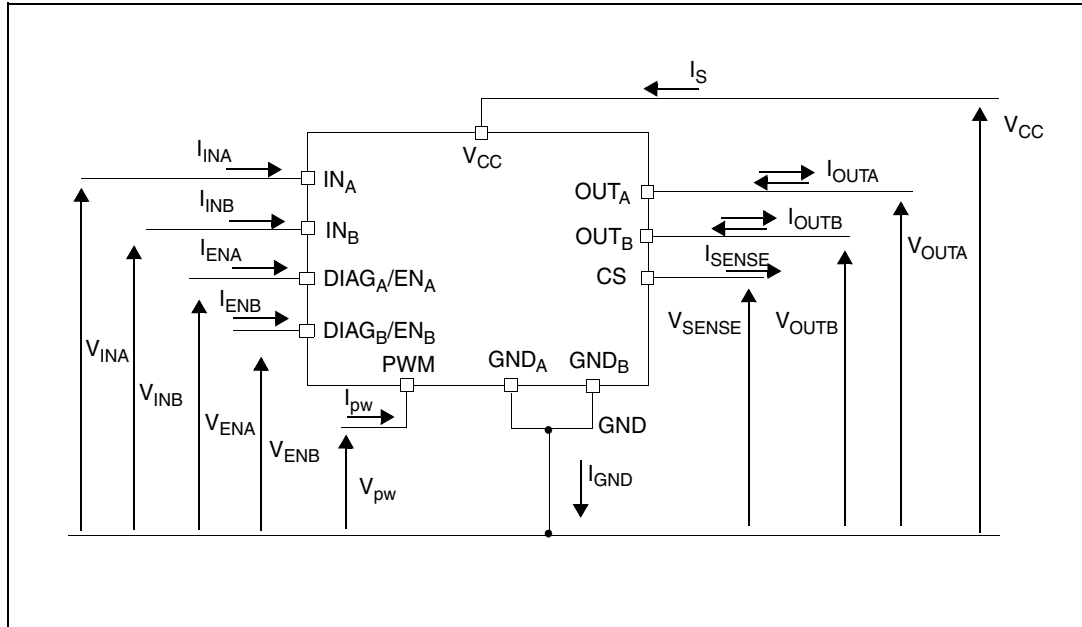
1. GND_A and GND_B must be externally connected together.

Table 4. Pin functions description

Name	Description
V_{CC}	Battery connection
GND_A, GND_B	Power grounds; must always be externally connected together
OUT_A, OUT_B	Power connections to the motor
IN_A, IN_B	Voltage controlled input pins with hysteresis, CMOS compatible. These two pins control the state of the bridge in normal operation according to the truth table (brake to V_{CC} , brake to GND, clockwise and counterclockwise).
PWM	Voltage controlled input pin with hysteresis, CMOS compatible. Gates of low side FETs are modulated by the PWM signal during their ON phase allowing speed control of the motor.
$EN_A/DIAG_A, EN_B/DIAG_B$	Open drain bidirectional logic pins. These pins must be connected to an external pull up resistor. When externally pulled low, they disable half-bridge A or B. In case of fault detection (thermal shutdown of a high side FET or excessive ON state voltage drop across a low side FET), these pins are pulled low by the device (see truth table in fault condition).
CS	Analog current sense output. This output sources a current proportional to the motor current. The information can be read back as an analog voltage across an external resistor.

2 Electrical specifications

Figure 3. Current and voltage conventions



2.1 Absolute maximum ratings

Table 5. Absolute maximum ratings

Symbol	Parameter	Value	Unit
V_{CC}	Supply voltage	+41	V
I_{max}	Maximum output current (continuous)	30	A
I_R	Reverse output current (continuous)	-30	
I_{IN}	Input current (IN_A and IN_B pins)	± 10	mA
I_{EN}	Enable input current ($DIAG_A/EN_A$ and $DIAG_B/EN_B$ pins)	± 10	
I_{pw}	PWM input current	± 10	
V_{CS}	Current sense maximum voltage	-3/+15	V
V_{ESD}	Electrostatic discharge ($R = 1.5k\Omega$, $C = 100pF$)		
	– CS pin	2	kV
	– logic pins	4	kV
	– output pins: OUT_A , OUT_B , V_{CC}	5	kV
T_j	Junction operating temperature	Internally limited	$^{\circ}C$
T_c	Case operating temperature	-40 to 150	
T_{STG}	Storage temperature	-55 to 150	

2.2 Electrical characteristics

$V_{CC} = 9V$ up to 16 V; $-40^{\circ}C < T_J < 150^{\circ}C$, unless otherwise specified.

Table 6. Power section

Symbol	Parameter	Test conditions	Min	Typ	Max	Unit
V_{CC}	Operating supply voltage		5.5		16	V
I_S	Supply current	Off state with all Fault Cleared & $EN_x=0$ $I_{N_A} = I_{N_B} = PWM = 0$; $T_J = 25^{\circ}C$; $V_{CC} = 13V$ $I_{N_A} = I_{N_B} = PWM = 0$ Off state: $I_{N_A} = I_{N_B} = PWM = 0$		12	30 60	μA μA mA
		On state: I_{N_A} or $I_{N_B} = 5V$, no PWM			10	mA
R_{ONHS}	Static high side resistance	$I_{OUT} = 15A$; $T_J = 25^{\circ}C$			14	m Ω
		$I_{OUT} = 15A$; $T_J = -40$ to $150^{\circ}C$			28	
R_{ONLS}	Static low side resistance	$I_{OUT} = 15A$; $T_J = 25^{\circ}C$			5	
		$I_{OUT} = 15A$; $T_J = -40$ to $150^{\circ}C$			10	
V_f	High side free-wheeling diode forward voltage	$I_f = 15A$		0.8	1.1	V
$I_{L(off)}$	High side off state output current (per channel)	$T_J = 25^{\circ}C$; $V_{OUTX} = EN_x = 0V$; $V_{CC} = 13V$			3	μA
		$T_J = 125^{\circ}C$; $V_{OUTX} = EN_x = 0V$; $V_{CC} = 13V$			5	
I_{RM}	Dynamic cross-conduction current	$I_{OUT} = 15A$ (see Figure 7)		0.7		A

Table 7. Logic inputs (I_{N_A} , I_{N_B} , EN_A , EN_B)

Symbol	Parameter	Test conditions	Min	Typ	Max	Unit
V_{IL}	Input low level voltage	Normal operation ($DIAG_x/EN_x$ pin acts as an input pin)			1.25	V
V_{IH}	Input high level voltage		3.25			
V_{IHYST}	Input hysteresis voltage		0.5			
V_{ICL}	Input clamp voltage	$I_{IN} = 1mA$	5.5	6.3	7.5	
		$I_{IN} = -1mA$	-1.0	-0.7	-0.3	
I_{INL}	Input low current	$V_{IN} = 1.25V$	1			μA
I_{INH}	Input high current	$V_{IN} = 3.25V$			10	
V_{DIAG}	Enable output low level voltage	Fault operation ($DIAG_x/EN_x$ pin acts as an output pin); $I_{EN} = 1mA$			0.4	V

Table 8. PWM

Symbol	Parameter	Test conditions	Min	Typ	Max	Unit
V_{pwl}	PWM low level voltage				1.25	V
I_{pwl}	PWM pin current	$V_{pw} = 1.25V$	1			μA
V_{pwh}	PWM high level voltage		3.25			V
I_{pwh}	PWM pin current	$V_{pw} = 3.25V$			10	μA
$V_{pwhhyst}$	PWM hysteresis voltage		0.5			V
V_{pwcl}	PWM clamp voltage	$I_{pw} = 1mA$	$V_{CC} + 0.3$	$V_{CC} + 0.7$	$V_{CC} + 1.0$	
		$I_{pw} = -1mA$	-6.0	-4.5	-3.0	
C_{INPWM}	PWM pin input capacitance	$V_{IN} = 2.5V$			25	pF

Table 9. Switching ($V_{CC} = 13V$, $R_{LOAD} = 0.87\Omega$, unless otherwise specified)

Symbol	Parameter	Test conditions	Min	Typ	Max	Unit
f	PWM frequency		0		20	kHz
$t_{d(on)}$	Turn-on delay time	Input rise time < $1\mu s$ (see Figure 6)			250	μs
$t_{d(off)}$	Turn-off delay time	Input rise time < $1\mu s$ (see Figure 6)			250	
t_r	Rise time	(see Figure 5)		1	1.6	
t_f	Fall time	(see Figure 5)		1.2	2.4	
t_{DEL}	Delay time during change of operating mode	(see Figure 4)	300	600	1800	
t_{rr}	High side free wheeling diode reverse recovery time	(see Figure 7)		110		ns
$t_{off(min)}^{(1)}$	PWM minimum off time	$9V < V_{CC} < 16V$; $T_j = 25^\circ C$; $L = 250\mu H$; $I_{OUT} = 15A$			6	μs

1. To avoid false Short to Battery detection during PWM operation, the PWM signal must be low for a time longer than $6\mu s$.

Table 10. Protection and diagnostic

Symbol	Parameter	Test conditions	Min	Typ	Max	Unit
V_{USD}	Undervoltage shut-down				5.5	V
	Undervoltage reset			4.7		
V_{OV}	Overvoltage shut-down		16	19	22	
I_{LIM}	High side current limitation		30	50	70	A
V_{CLP}	Total clamp voltage (V_{CC} to GND)	$I_{OUT} = 15A$	43	48	54	V
T_{TSD}	Thermal shut-down temperature	$V_{IN} = 3.25V$	150	175	200	$^\circ C$
T_{TR}	Thermal reset temperature		135			
T_{HYST}	Thermal hysteresis		7	15		

Table 11. Current sense ($9V < V_{CC} < 16V$)

Symbol	Parameter	Test conditions	Min	Typ	Max	Unit
K_1	I_{OUT}/I_{SENSE}	$I_{OUT} = 30A; R_{SENSE} = 1.5k\Omega;$ $T_j = -40 \text{ to } 150^\circ\text{C}$	9665	11370	13075	
K_2	I_{OUT}/I_{SENSE}	$I_{OUT} = 8A; R_{SENSE} = 1.5k\Omega;$ $T_j = -40 \text{ to } 150^\circ\text{C}$	9096	11370	13644	
$dK_1 / K_1^{(1)}$	Analog sense current drift	$I_{OUT} = 30A; R_{SENSE} = 1.5k\Omega;$ $T_j = -40 \text{ to } 150^\circ\text{C}$	-8		+8	%
$dK_2 / K_2^{(1)}$	Analog sense current drift	$I_{OUT} > 8A; R_{SENSE} = 1.5k\Omega;$ $T_j = -40 \text{ to } 150^\circ\text{C}$	-10		+10	
I_{SENSE0}	Analog sense leakage current	$I_{OUT} = 0A; V_{SENSE} = 0V;$ $T_j = -40 \text{ to } 150^\circ\text{C}$	0		65	μA

1. Analog sense current drift is deviation of factor K for a given device over (-40°C to 150°C and $9V < V_{CC} < 16V$) with respect to its value measured at $T_j = 25^\circ\text{C}, V_{CC} = 13V$.

Figure 4. Definition of the delay times measurement

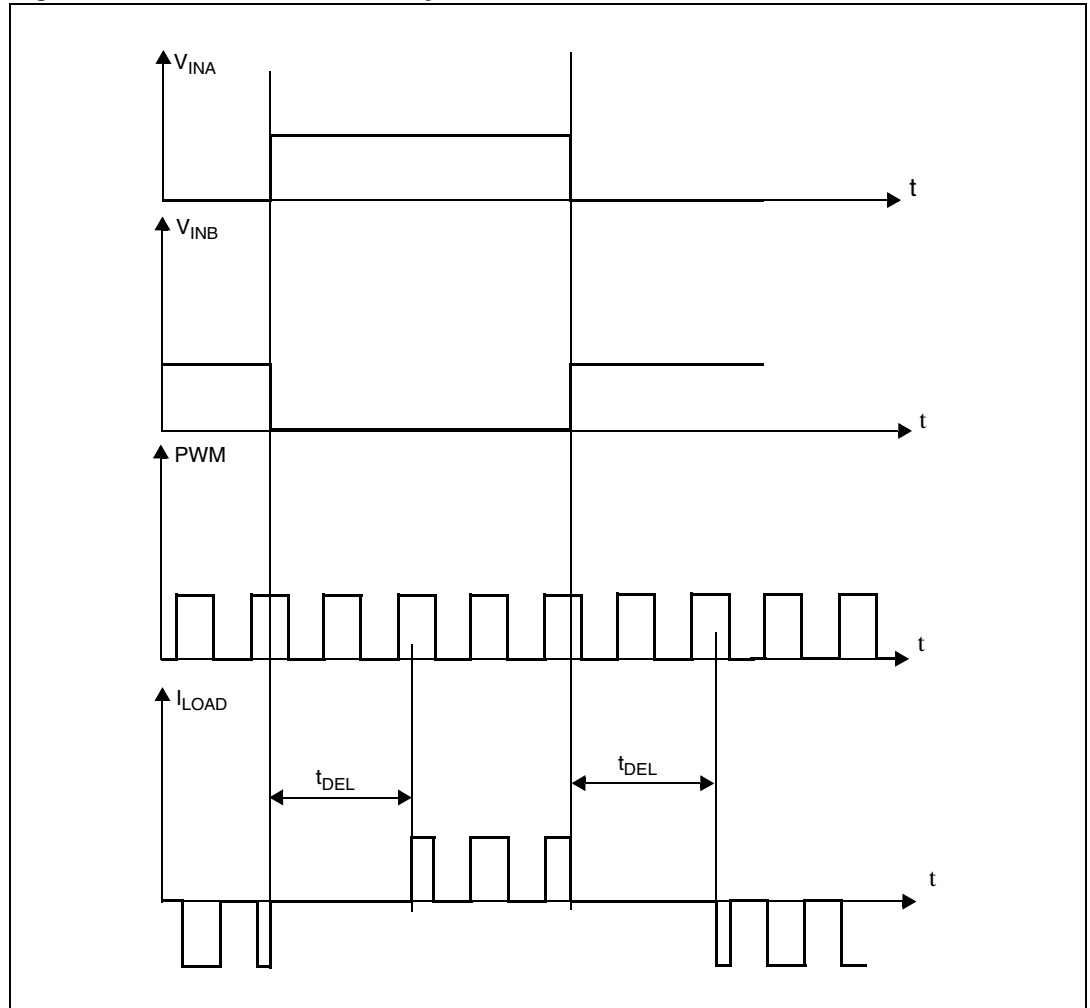


Figure 5. Definition of the low side switching times

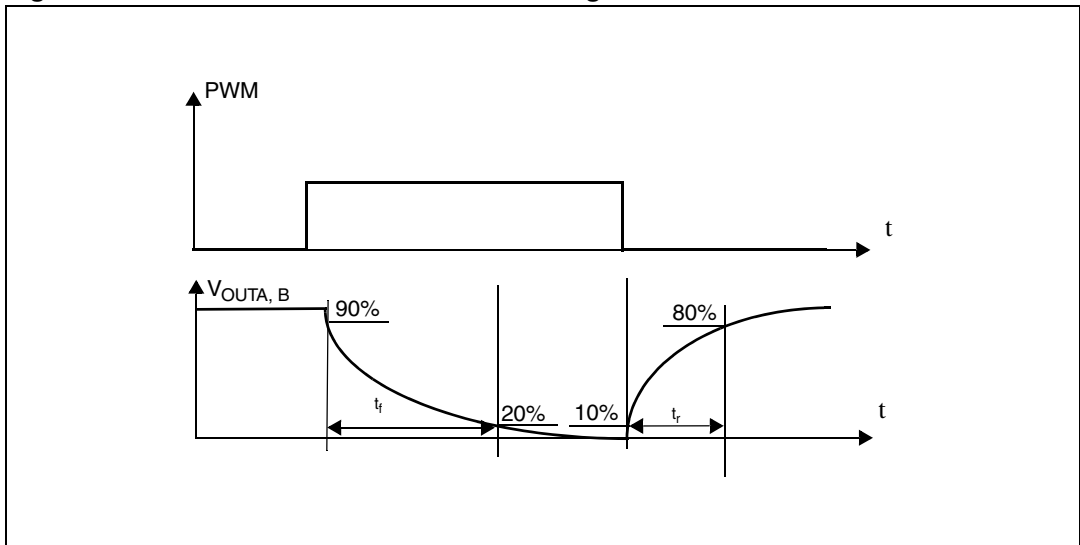


Figure 6. Definition of the high side switching times

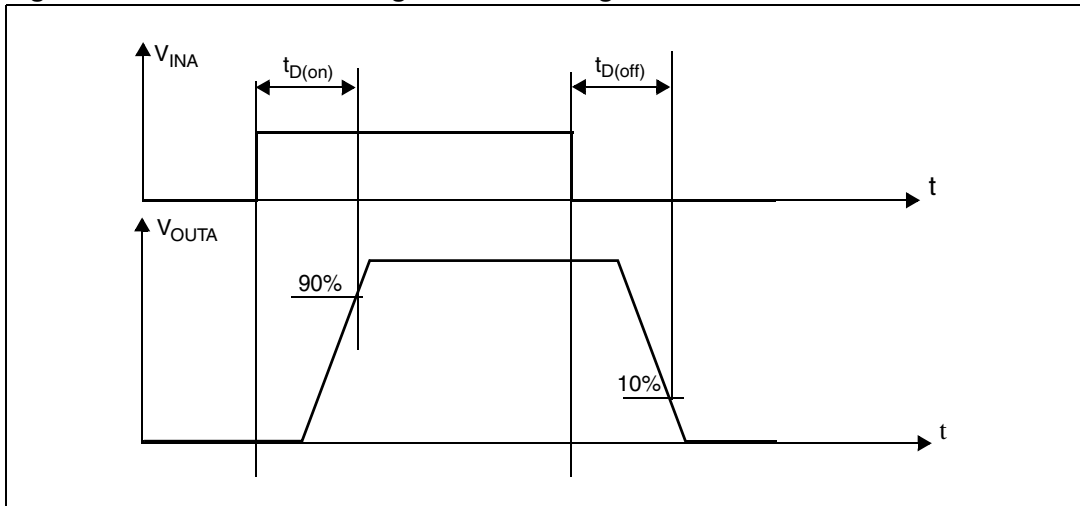


Figure 7. Definition of dynamic cross conduction current during a PWM operation

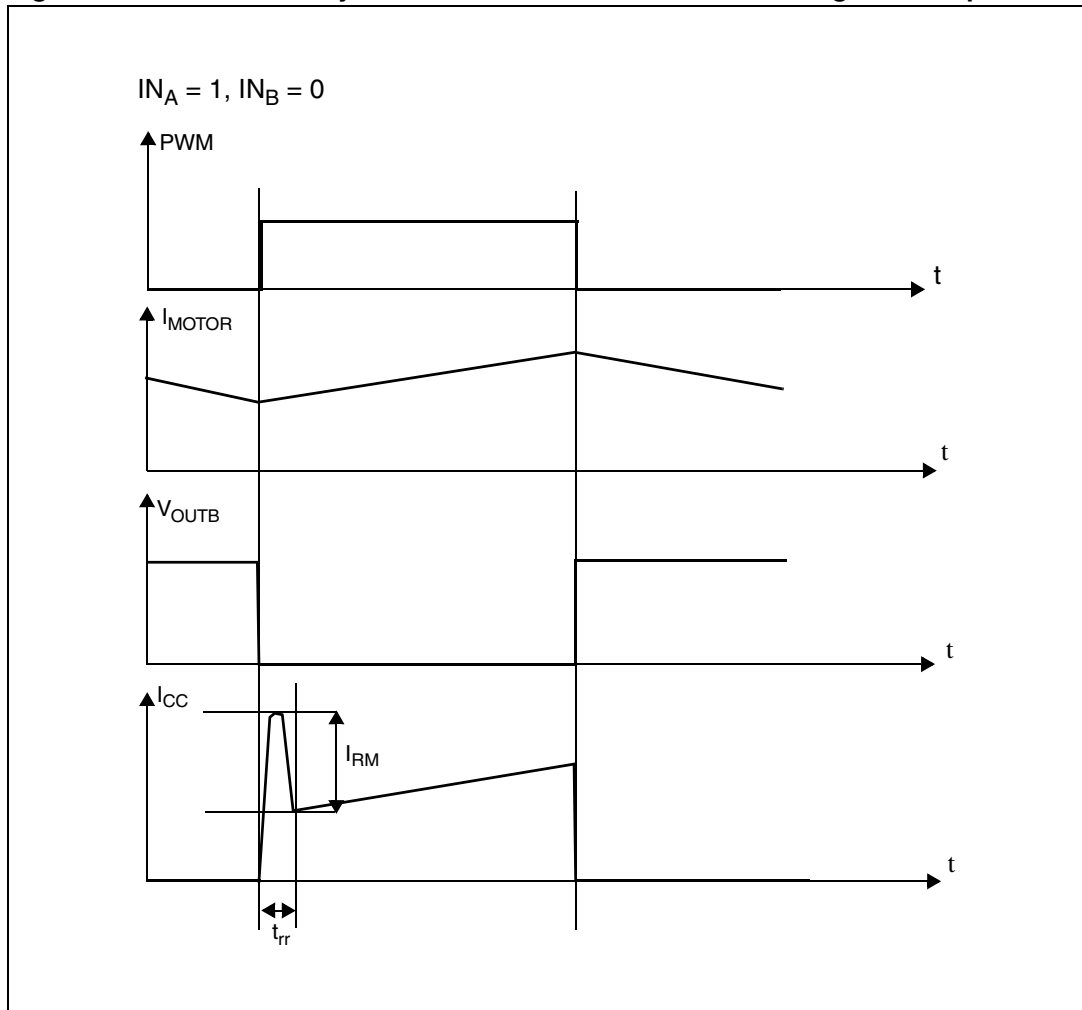


Table 12. Truth table in normal operating conditions

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

Table 13. Truth table in fault conditions (detected on OUT_A)

IN _A	IN _B	DIAG _A /EN _A	DIAG _B /EN _B	OUT _A	OUT _B	CS
1	1	0	1	OPEN	H	High Imp.
	0				L	
0	1				H	I _{OUTB} /K
	0					L
X	X				0	OPEN
	1				1	H
	0	1	L	High Imp.		

↑
Fault Information
↑
Protection Action

Note: Notice that saturation detection on the low side power MOSFET is possible only if the impedance of the short-circuit from the output to the battery is less than 100mΩ when the device is supplied with a battery voltage of 13.5V.

Table 14. Electrical transient requirements

ISO T/R - 7637/1 Test pulse	Test Level I	Test Level II	Test Level III	Test Level IV	Test levels delays and impedance
1	-25V	-50V	-75V	-100V	2ms, 10Ω
2	+25V	+50V	+75V	+100V	0.2ms, 10Ω
3a	-25V	-50V	-100V	-150V	0.1μs, 50Ω
3b	+25V	+50V	+75V	+100V	
4	-4V	-5V	-6V	-7V	100ms, 0.01Ω
5	+26.5V	+46.5V	+66.5V	+86.5V	400ms, 2Ω

ISO T/R - 7637/1 test pulse	Test levels result I	Test levels result II	Test levels result III	Test levels result IV
1	C	C	C	C
2				
3a				
3b				
4				
5 ⁽¹⁾		E	E	E

1. For load dump exceeding the above value a centralized suppressor must be adopted.

Class	Contents
C	All functions of the device are performed as designed after exposure to disturbance.
E	One or more functions of the device are not performed as designed after exposure to disturbance and cannot be returned to proper operation without replacing the device.

2.3 Electrical characteristics curves

Figure 8. On state supply current

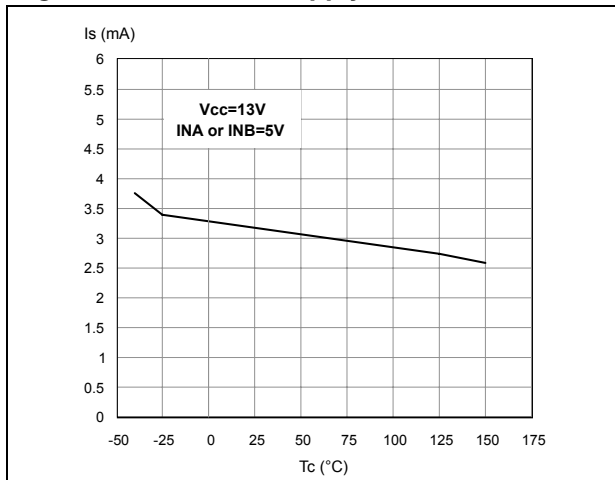


Figure 9. Off state supply current

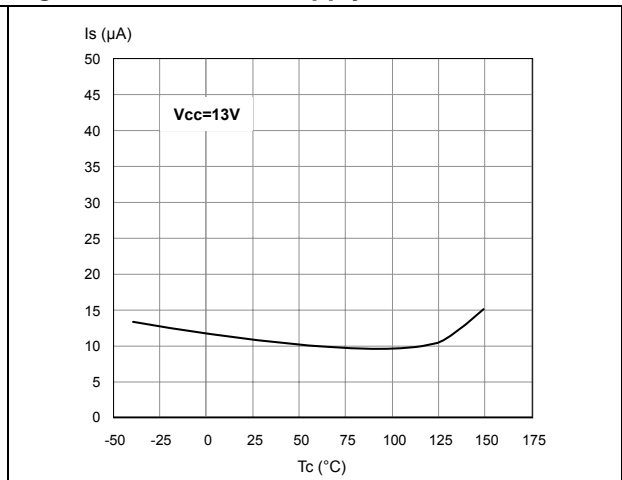


Figure 10. High level input current

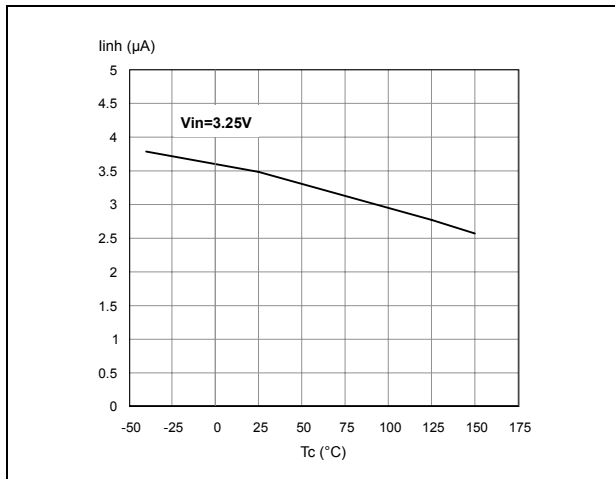


Figure 11. Input clamp voltage

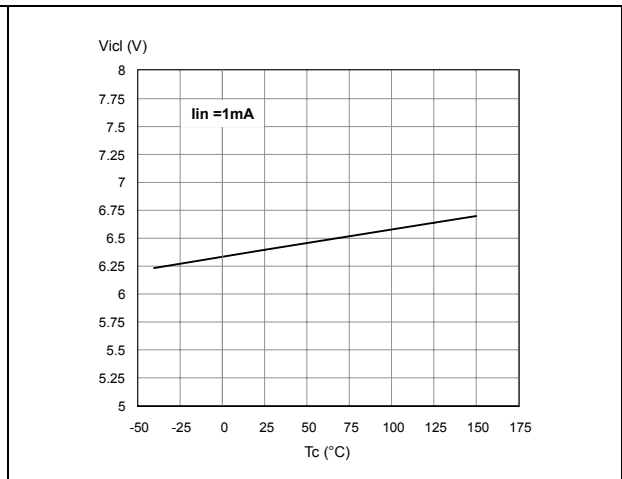


Figure 12. Input high level voltage

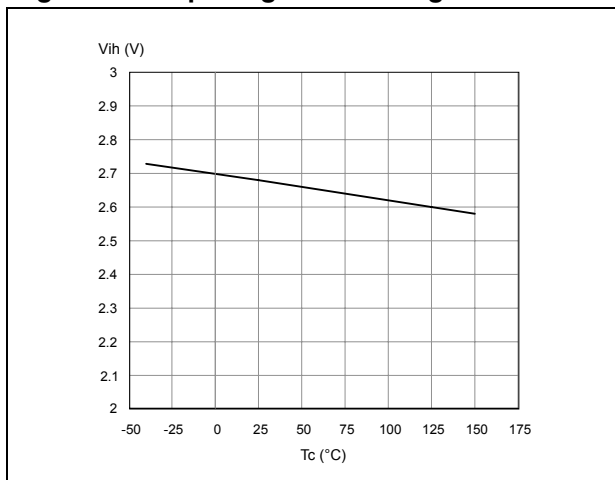


Figure 13. Input low level voltage

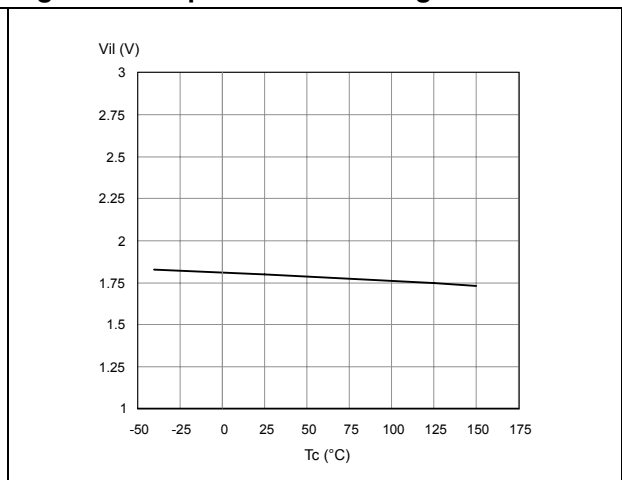


Figure 14. Input hysteresis voltage

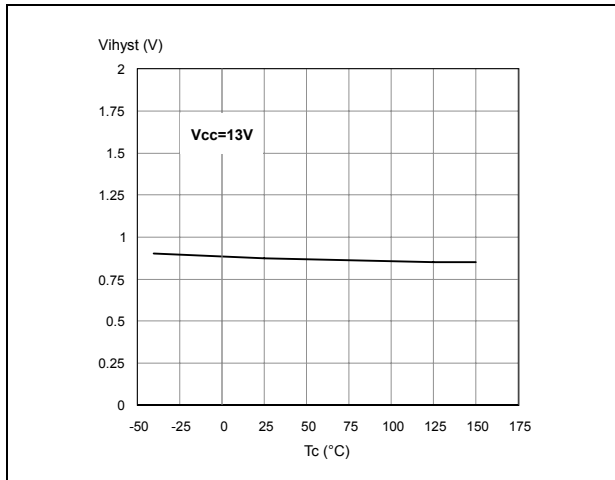


Figure 15. High level enable pin current

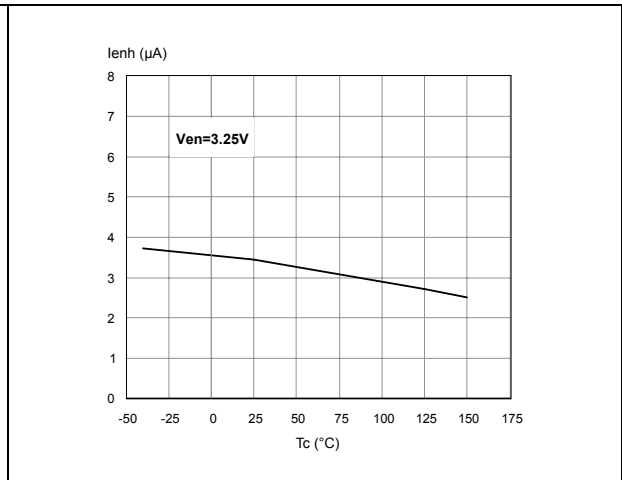


Figure 16. Delay time during change of operation mode

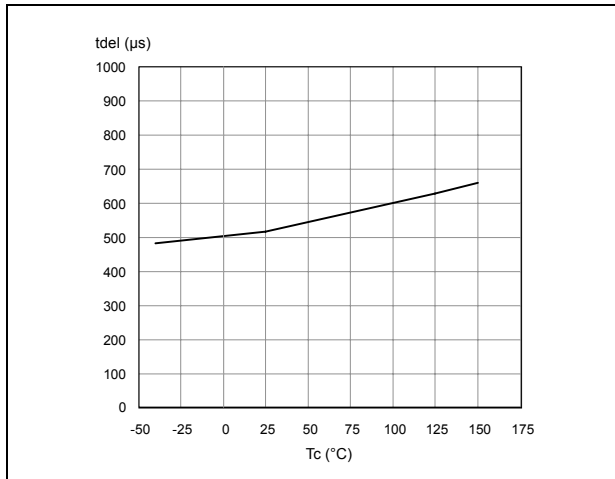


Figure 17. Enable clamp voltage

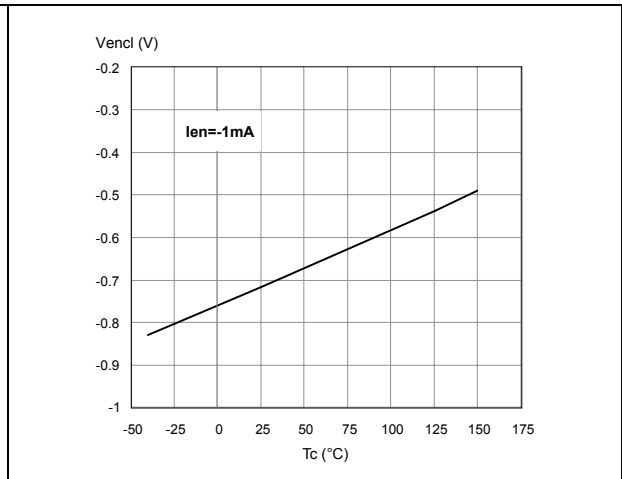


Figure 18. High level enable voltage

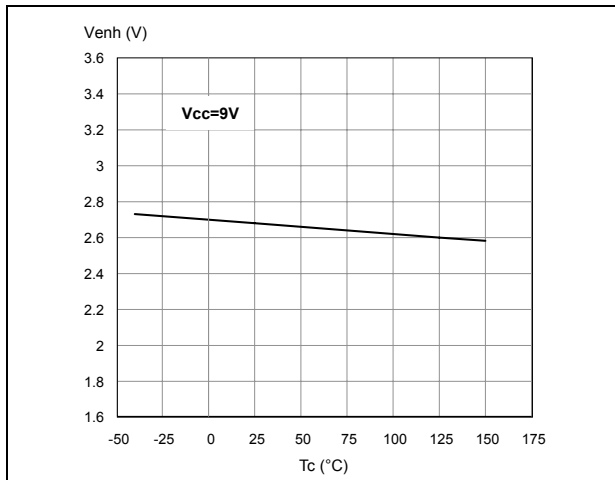


Figure 19. Low level enable voltage

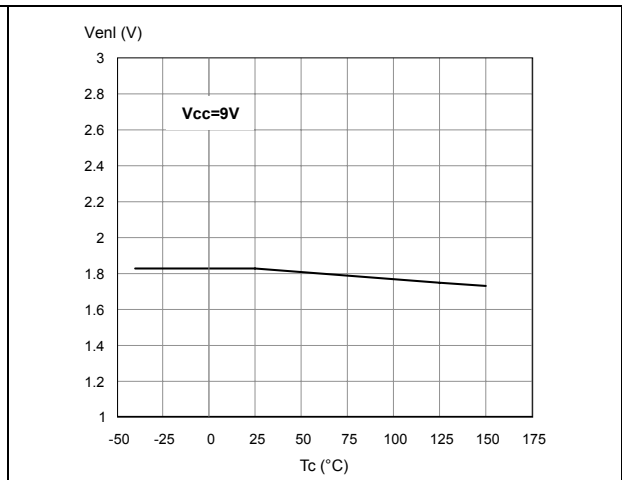


Figure 20. PWM high level voltage

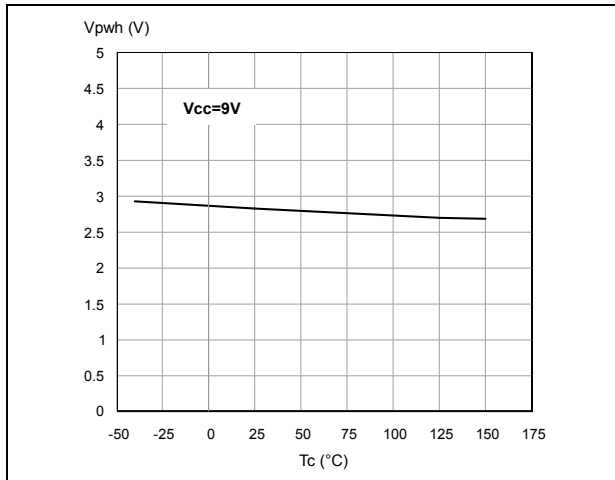


Figure 21. PWM low level voltage

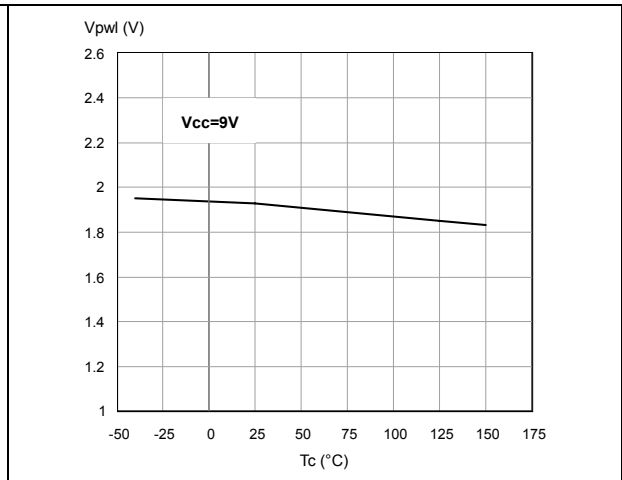


Figure 22. PWM high level current

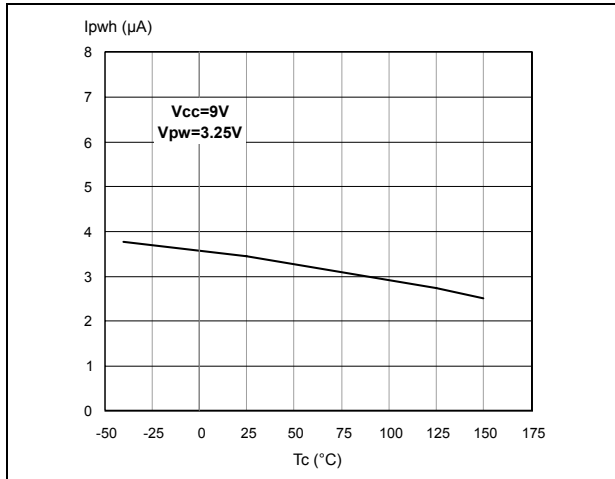


Figure 23. Overvoltage shutdown

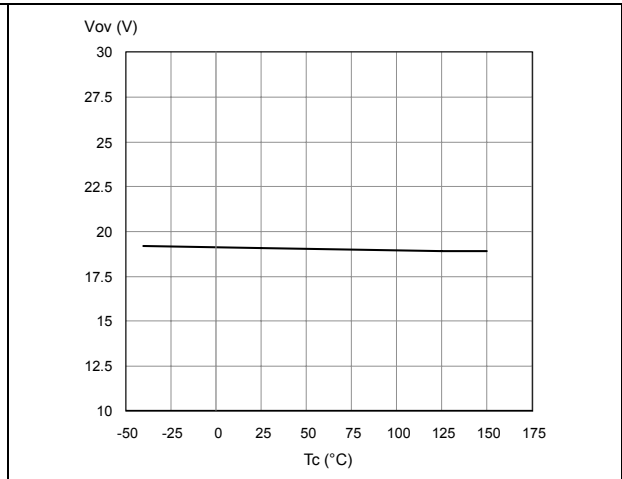


Figure 24. Undervoltage shutdown

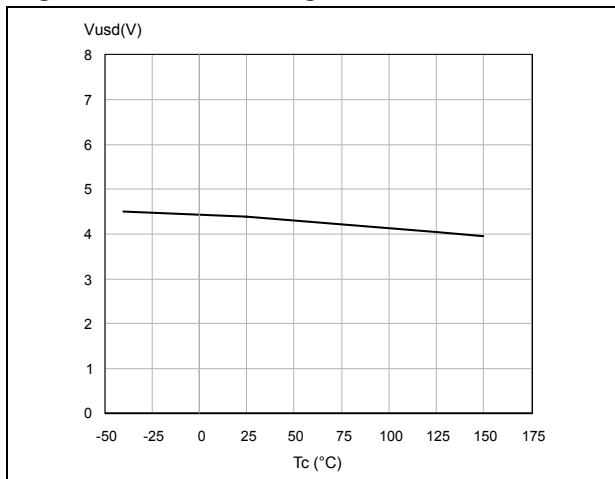


Figure 25. Current limitation

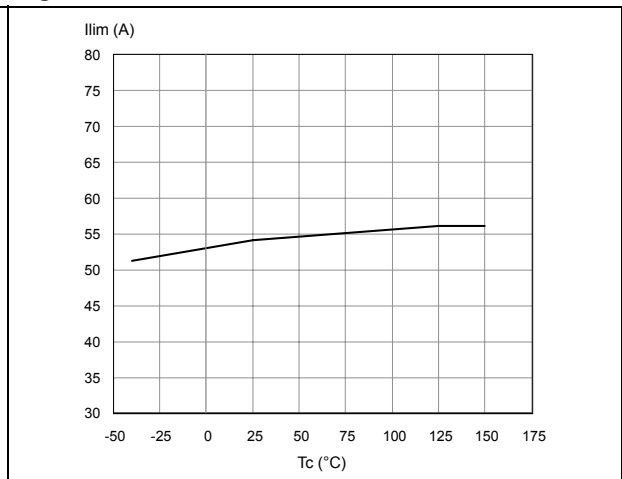


Figure 26. On state high side resistance vs T_{case}

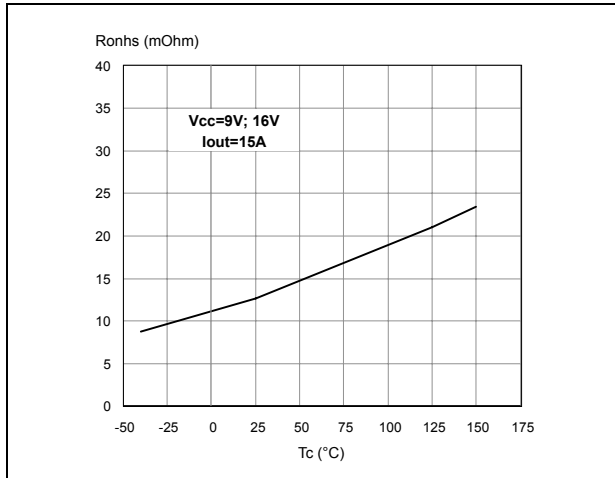


Figure 27. On state low side resistance vs T_{case}

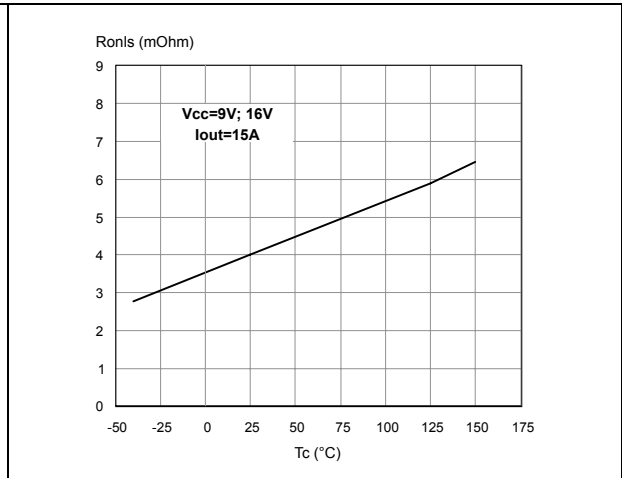


Figure 28. Turn-On delay time

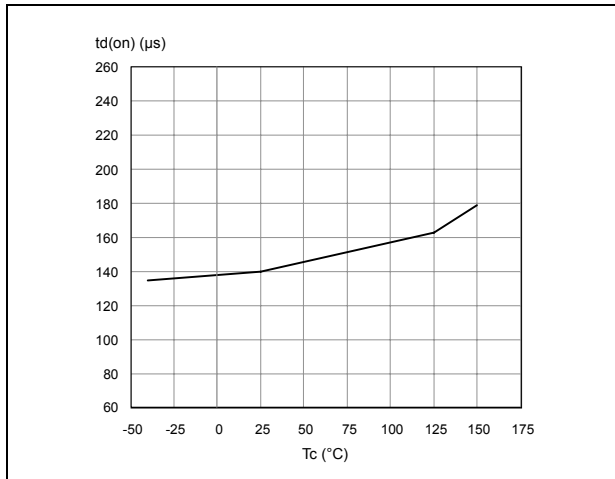


Figure 29. Turn-Off delay time

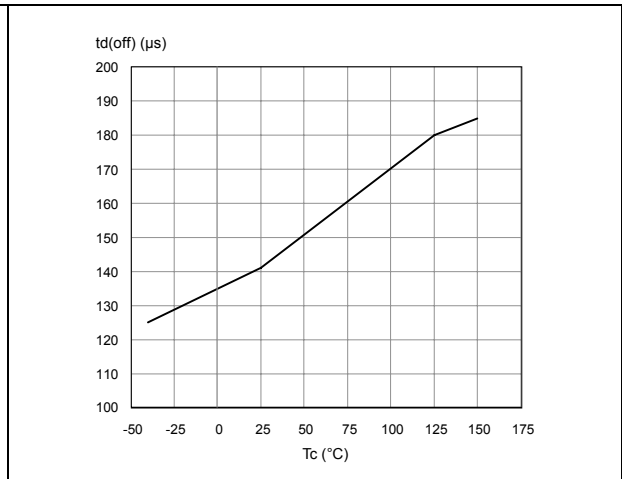


Figure 30. Output voltage rise time

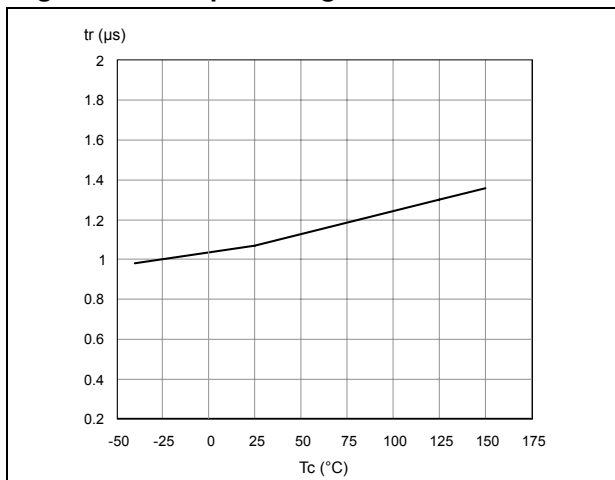
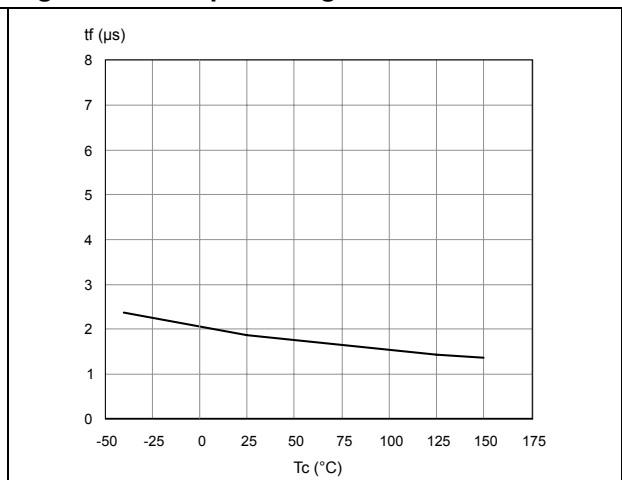


Figure 31. Output voltage fall time

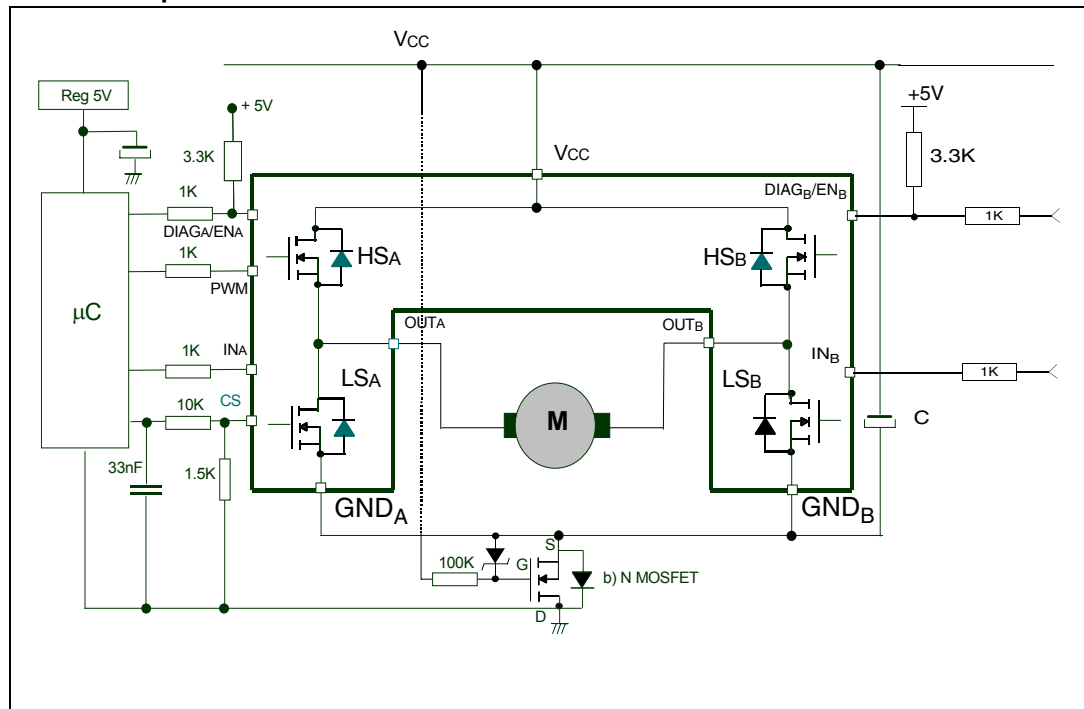


3 Application information

In normal operating conditions the DIAG_X/EN_X pin is considered as an input pin by the device. This pin must be externally pulled high.

PWM pin usage: in all cases, a “0” on the PWM pin will turn off both LS_A and LS_B switches. When PWM rises back to “1”, LS_A or LS_B turn on again depending on the input pin state.

Figure 32. Typical application circuit for DC to 20 kHz PWM operation short circuit protection



Note: The value of the blocking capacitor (C) depends on the application conditions and defines voltage and current ripple onto supply line at PWM operation. Stored energy of the motor inductance may fly back into the blocking capacitor, if the bridge driver goes into tri-state. This causes a hazardous overvoltage if the capacitor is not big enough. As basic orientation, 500µF per 10A load current is recommended.

In case of a fault condition the DIAG_X/EN_X pin is considered as an output pin by the device. The fault conditions are:

- overtemperature on one or both high sides
- short to battery condition on the output (saturation detection on the low side power MOSFET)

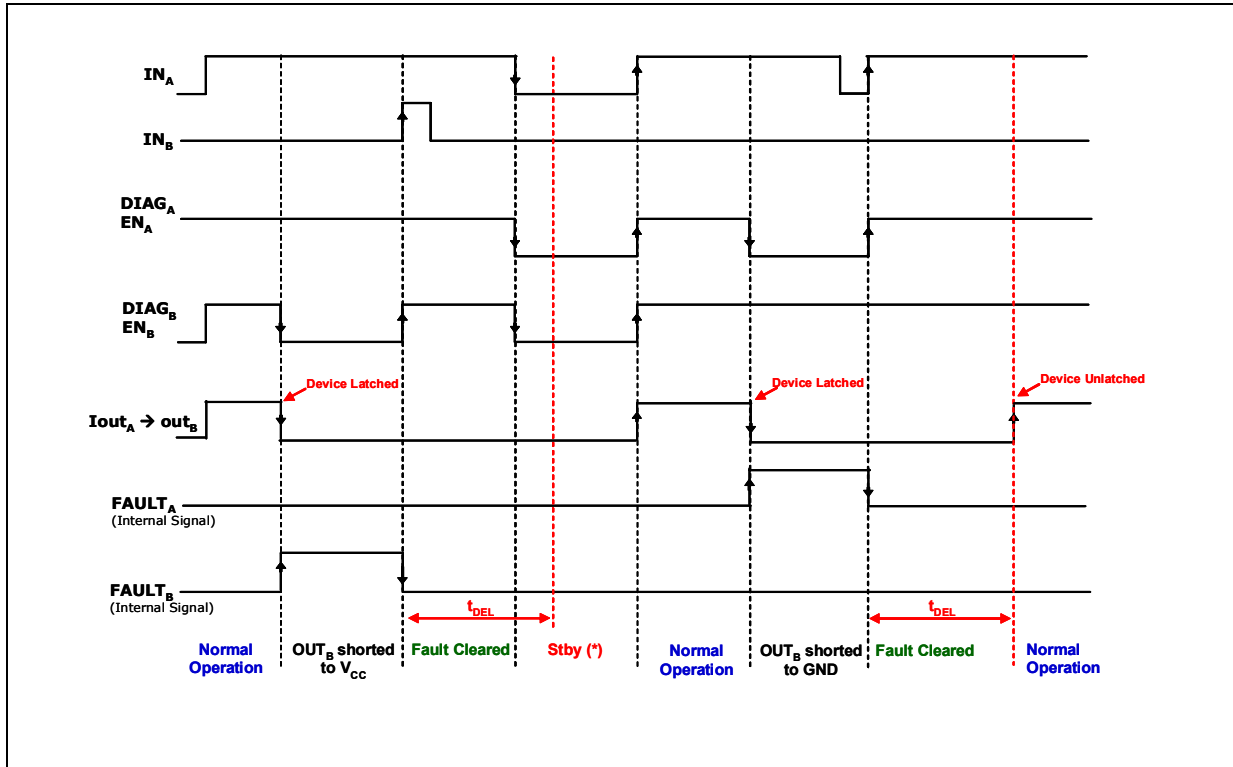
Possible origins of fault conditions may be:

- OUT_A is shorted to ground → overtemperature detection on high side A.
- OUT_A is shorted to V_{CC} → low side power MOSFET saturation detection.

When a fault condition is detected, the user can know which power element is in fault by monitoring the IN_A , IN_B , $DIAG_A/EN_A$ and $DIAG_B/EN_B$ pins.

In any case, when a fault is detected, the faulty leg of the bridge is latched off. To turn on the respective output (OUT_X) again, the input signal must rise from low to high level.

Figure 33. Behavior in fault condition (How a fault can be cleared)



Note: In case of the fault condition is not removed, the procedure for unlatching and sending the device in Stby mode is:

- Clear the fault in the device (toggle : IN_A if $EN_A=0$ or IN_B if $EN_B=0$)
- Pull low all inputs, PWM and Diag/EN pins within t_{DEL} .

If the Diag/En pins are already low, PWM=0, the fault can be cleared simply toggling the input. The device will enter in stby mode as soon as the fault is cleared.

3.1 Reverse battery protection

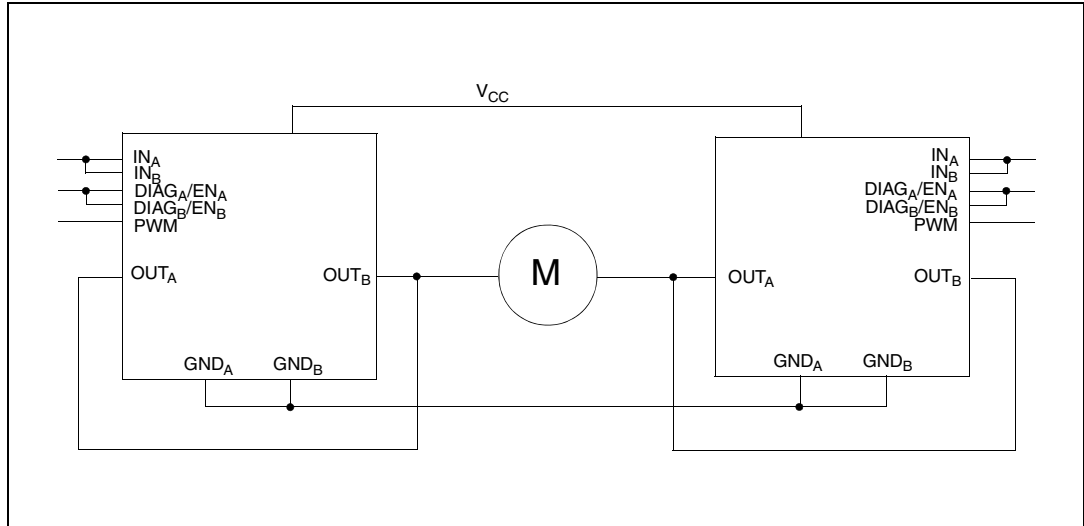
Three possible solutions can be considered:

1. a Schottky diode D connected to V_{CC} pin
2. an N-channel MOSFET connected to the GND pin (see [Figure 32: Typical application circuit for DC to 20 kHz PWM operation short circuit protection on page 20](#))
3. a P-channel MOSFET connected to the V_{CC} pin

The device sustains no more than -30A in reverse battery conditions because of the two body diodes of the power MOSFETs. Additionally, in reverse battery condition the I/Os of VNH2SP30-E are pulled down to the V_{CC} line (approximately -1.5V). A series resistor must

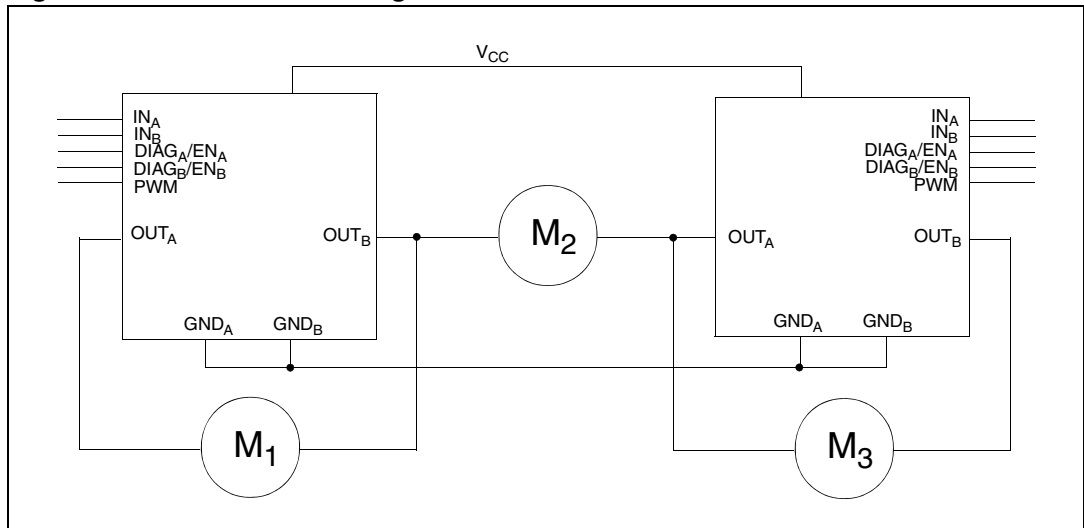
be inserted to limit the current sunk from the microcontroller I/Os. If I_{Rmax} is the maximum target reverse current through μC I/Os, the series resistor is:

Figure 34. Half-bridge configuration



Note: The VNH2SP30-E can be used as a high power half-bridge driver achieving an On resistance per leg of 9.5mΩ.

Figure 35. Multi-motors configuration



Note: The VNH2SP30-E can easily be designed in multi-motors driving applications such as seat positioning systems where only one motor must be driven at a time. $DIAG_X/EN_X$ pins allow to put unused half-bridges in high impedance.

Figure 36. Waveforms in full bridge operation

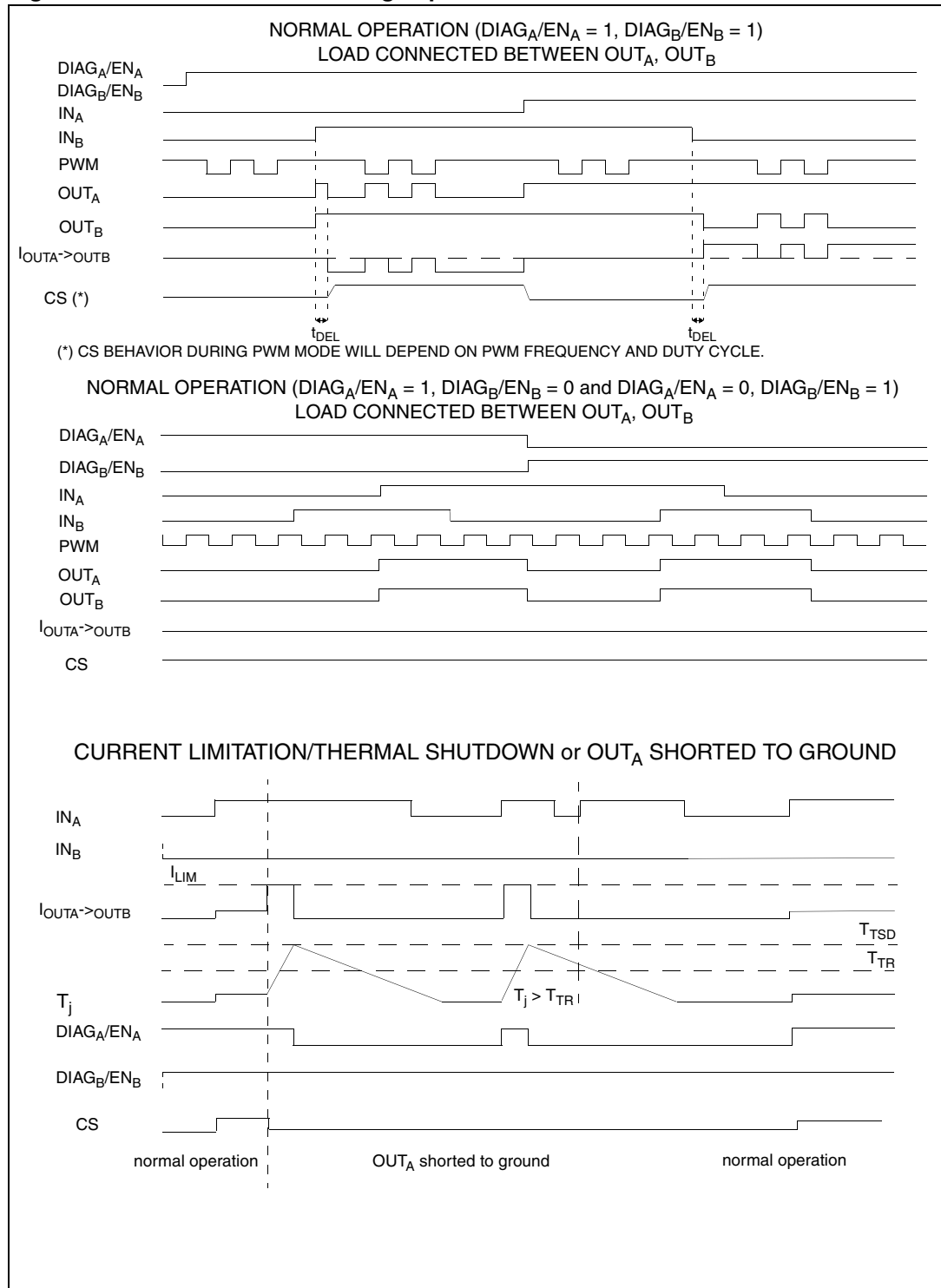
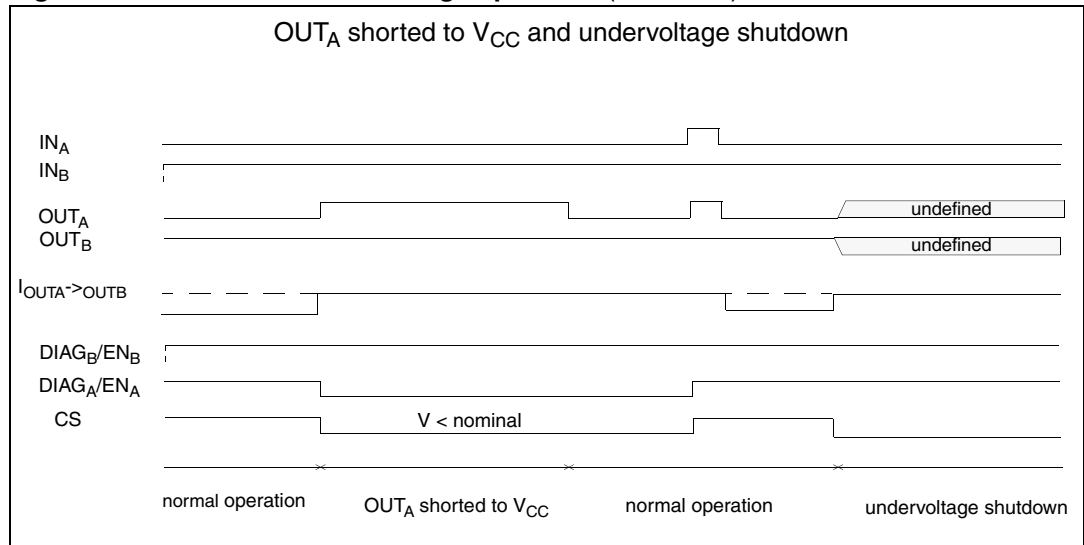


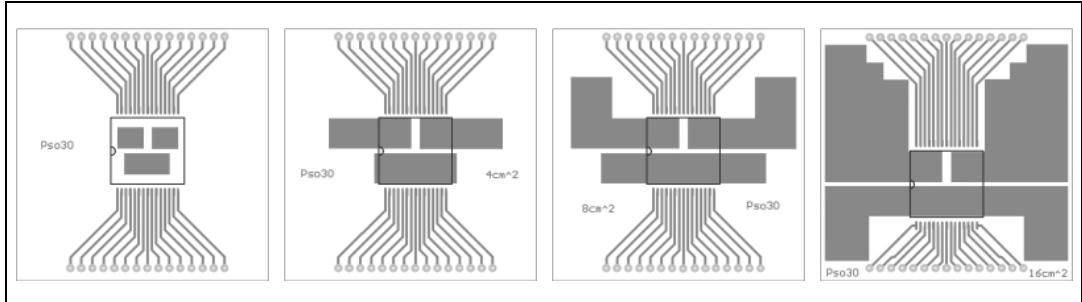
Figure 37. Waveforms in full bridge operation (continued)



4 Package and PCB thermal data

4.1 PowerSSO-30 thermal data

Figure 38. MultiPowerSO-30™ PC board



Note: Layout condition of R_{th} and Z_{th} measurements (PCB FR4 area = 58mm x 58mm, PCB thickness = 2mm. Cu thickness = 35 μ m, Copper areas: from minimum pad layout to 16cm²).

Figure 39. Chipset configuration

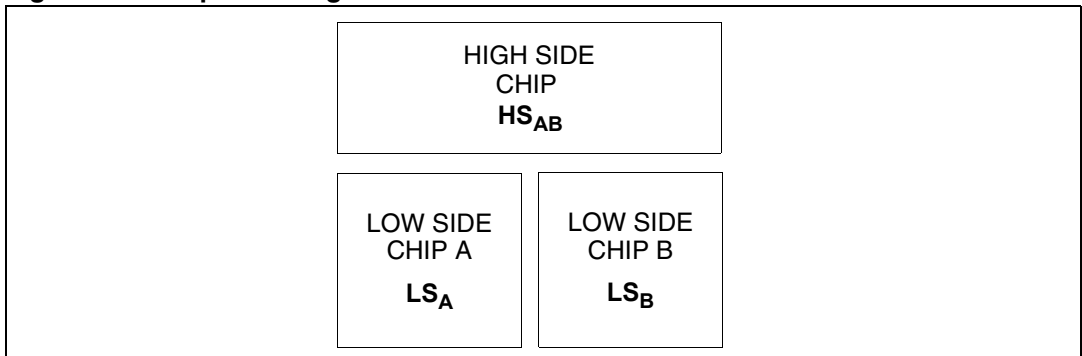
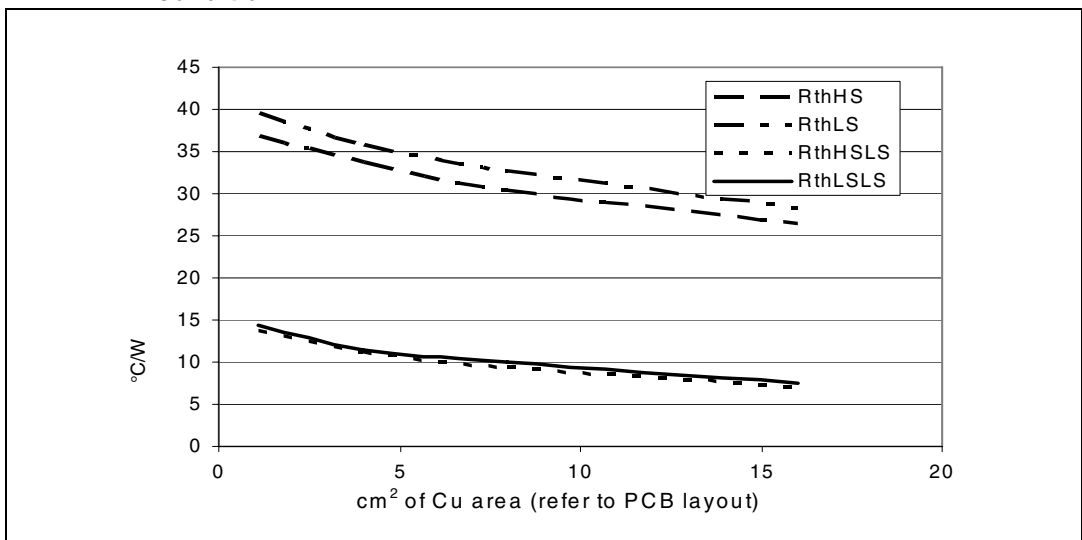


Figure 40. Auto and mutual $R_{thj-amb}$ vs PCB copper area in open box free air condition



4.1.1 Thermal calculation in clockwise and anti-clockwise operation in steady-state mode

Table 15. Thermal calculation in clockwise and anti-clockwise operation in steady-state mode

HS _A	HS _B	LS _A	LS _B	T _{jHSAB}	T _{jLSA}	T _{jLSB}
ON	OFF	OFF	ON	$P_{dHSA} \times R_{thHS} + P_{dLSB} \times R_{thHSLs} + T_{amb}$	$P_{dHSA} \times R_{thHSLs} + P_{dLSB} \times R_{thLSLs} + T_{amb}$	$P_{dHSA} \times R_{thHSLs} + P_{dLSB} \times R_{thLS} + T_{amb}$
OFF	ON	ON	OFF	$P_{dHSB} \times R_{thHS} + P_{dLSA} \times R_{thHSLs} + T_{amb}$	$P_{dHSB} \times R_{thHSLs} + P_{dLSA} \times R_{thLS} + T_{amb}$	$P_{dHSB} \times R_{thHSLs} + P_{dLSA} \times R_{thLSLs} + T_{amb}$

4.1.2 Thermal resistances definition (values according to the PCB heatsink area)

R_{thHS} = R_{thHSA} = R_{thHSB} = High Side Chip Thermal Resistance Junction to Ambient (HS_A or HS_B in ON state)

R_{thLS} = R_{thLSA} = R_{thLSB} = Low Side Chip Thermal Resistance Junction to Ambient

R_{thHSLs} = R_{thHSALSB} = R_{thHSBLSA} = Mutual Thermal Resistance Junction to Ambient between High Side and Low Side Chips

R_{thLSLs} = R_{thLSALSB} = Mutual Thermal Resistance Junction to Ambient between Low Side Chips

4.1.3 Thermal calculation in transient mode^(a)

$$T_{jHSAB} = Z_{thHS} \times P_{dHSAB} + Z_{thHSLs} \times (P_{dLSA} + P_{dLSB}) + T_{amb}$$

$$T_{jLSA} = Z_{thHSLs} \times P_{dHSAB} + Z_{thLS} \times P_{dLSA} + Z_{thLSLs} \times P_{dLSB} + T_{amb}$$

$$T_{jLSB} = Z_{thHSLs} \times P_{dHSAB} + Z_{thLSLs} \times P_{dLSA} + Z_{thLS} \times P_{dLSB} + T_{amb}$$

4.1.4 Single pulse thermal impedance definition (values according to the PCB heatsink area)

Z_{thHS} = High Side Chip Thermal Impedance Junction to Ambient

Z_{thLS} = Z_{thLSA} = Z_{thLSB} = Low Side Chip Thermal Impedance Junction to Ambient

Z_{thHSLs} = Z_{thHSABLSA} = Z_{thHSBLSB} = Mutual Thermal Impedance Junction to Ambient between High Side and Low Side Chips

Z_{thLSLs} = Z_{thLSALSB} = Mutual Thermal Impedance Junction to Ambient between Low Side Chips

a. Calculation is valid in any dynamic operating condition. P_d values set by user.

Equation 1: pulse calculation formula

$$Z_{TH\delta} = R_{TH} P \delta + Z_{THtp}(1 - \delta)$$

where $\delta = t_p/T$

Figure 41. MultiPowerSO-30 HSD thermal impedance junction ambient single pulse

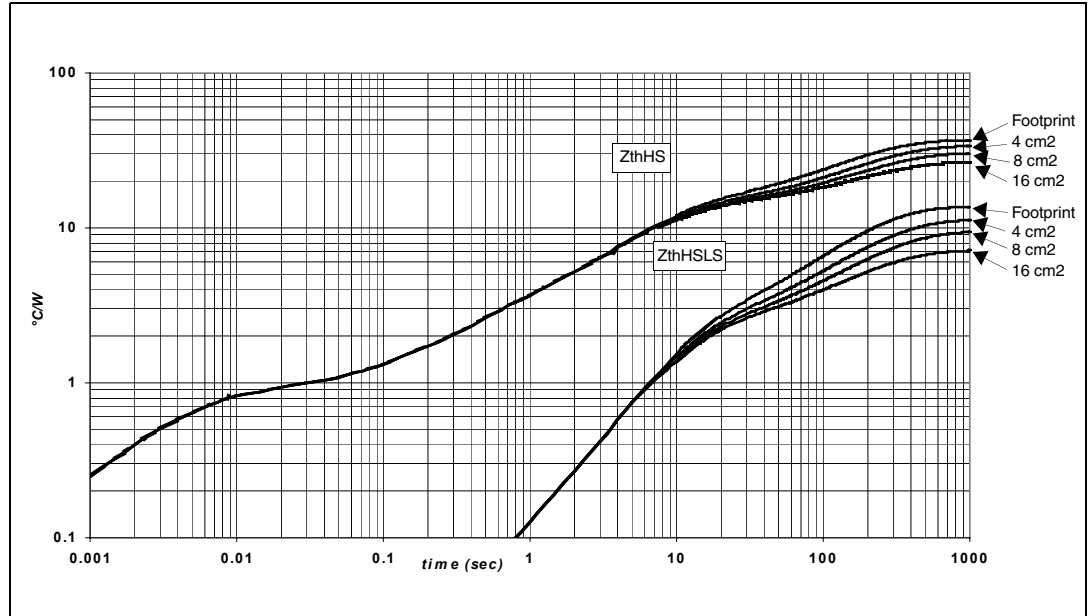


Figure 42. MultiPowerSO-30 LSD thermal impedance junction ambient single pulse

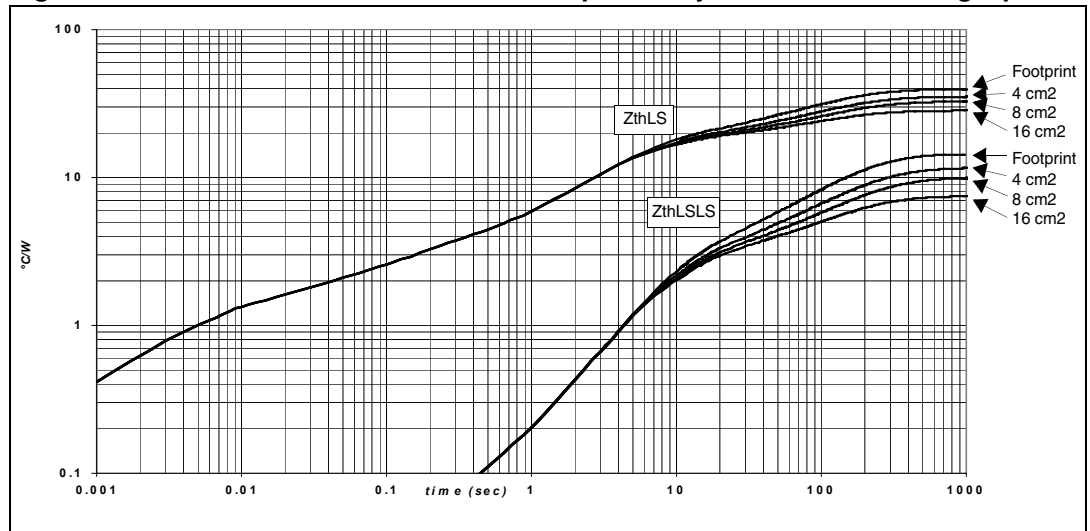


Figure 43. Thermal fitting model of an H-bridge in MultiPowerSO-30

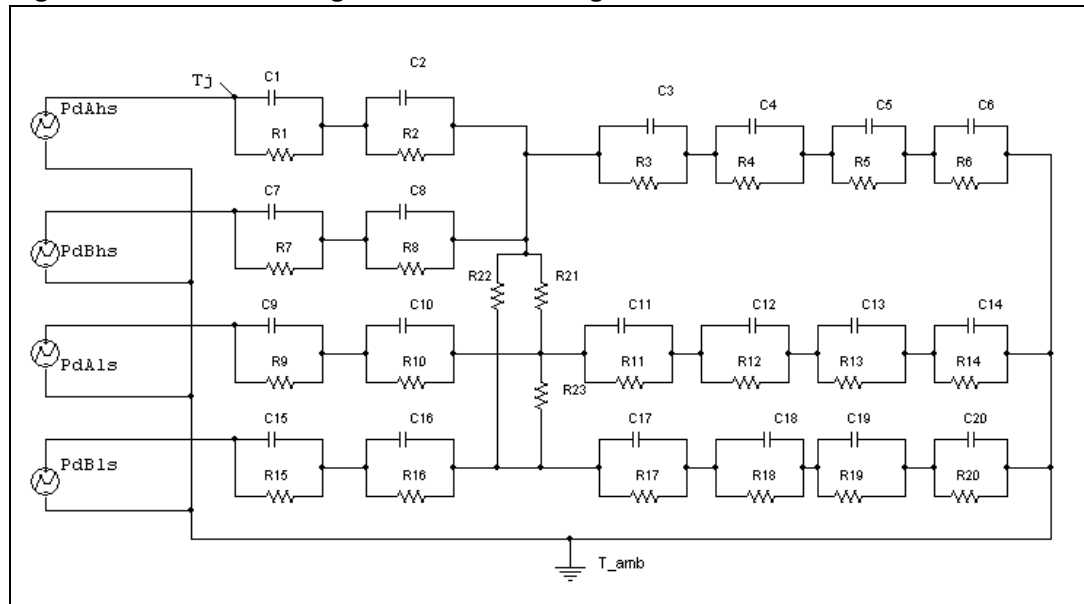


Table 16. Thermal parameters⁽¹⁾

Area/island (cm ²)	Footprint	4	8	16
R1 = R7 (°C/W)	0.05			
R2 = R8 (°C/W)	0.3			
R3 (°C/W)	0.5			
R4 (°C/W)	1.3			
R5 (°C/W)	14			
R6 (°C/W)	44.7	39.1	31.6	23.7
R9 = R15 (°C/W)	0.2			
R10 = R16 (°C/W)	0.4			
R11 = R17 (°C/W)	0.8			
R12 = R18 (°C/W)	1.5			
R13 = R19 (°C/W)	20			
R14 = R20 (°C/W)	46.9	36.1	30.4	20.8
R21 = R22 = R23 (°C/W)	115			
C1 = C7 (W.s/°C)	0.005			
C2 = C8 (W.s/°C)	0.008			
C3 = C11 = C17 (W.s/°C)	0.01			
C4 = C13 = C19 (W.s/°C)	0.3			
C5 (W.s/°C)	0.6			
C6 (W.s/°C)	5	7	9	11
C9 = C15 (W.s/°C)	0.003			
C10 = C16 (W.s/°C)	0.006			
C12 = C18 (W.s/°C)	0.075			
C14 = C20 (W.s/°C)	2.5	3.5	4.5	5.5

1. The blank space means that the value is the same as the previous one.

5 Package and packing information

5.1 ECOPACK® packages

In order to meet environmental requirements, ST offers these devices in ECOPACK® packages. ECOPACK® packages are lead-free. The category of Second Level Interconnect is marked on the package and on the inner box label, in compliance with JEDEC Standard JESD97. The maximum ratings related to soldering conditions are also marked on the inner box label.

ECOPACK is an ST trademark. ECOPACK specifications are available at www.st.com.

5.2 MultiPowerSO-30 package mechanical data

Figure 44. MultiPowerSO-30 package outline

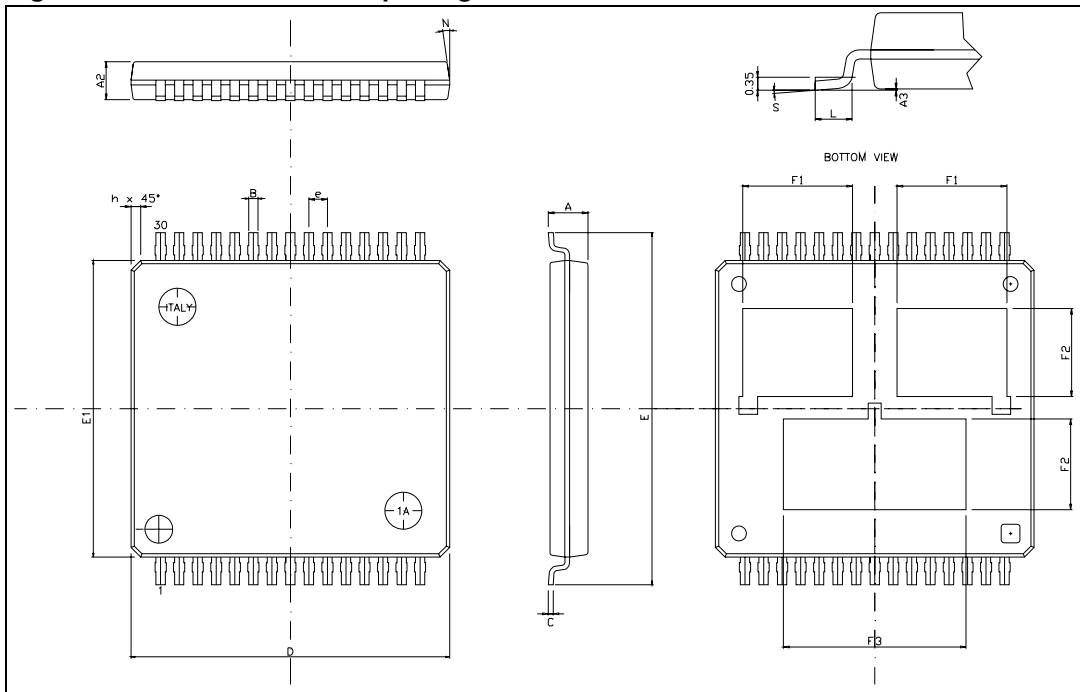
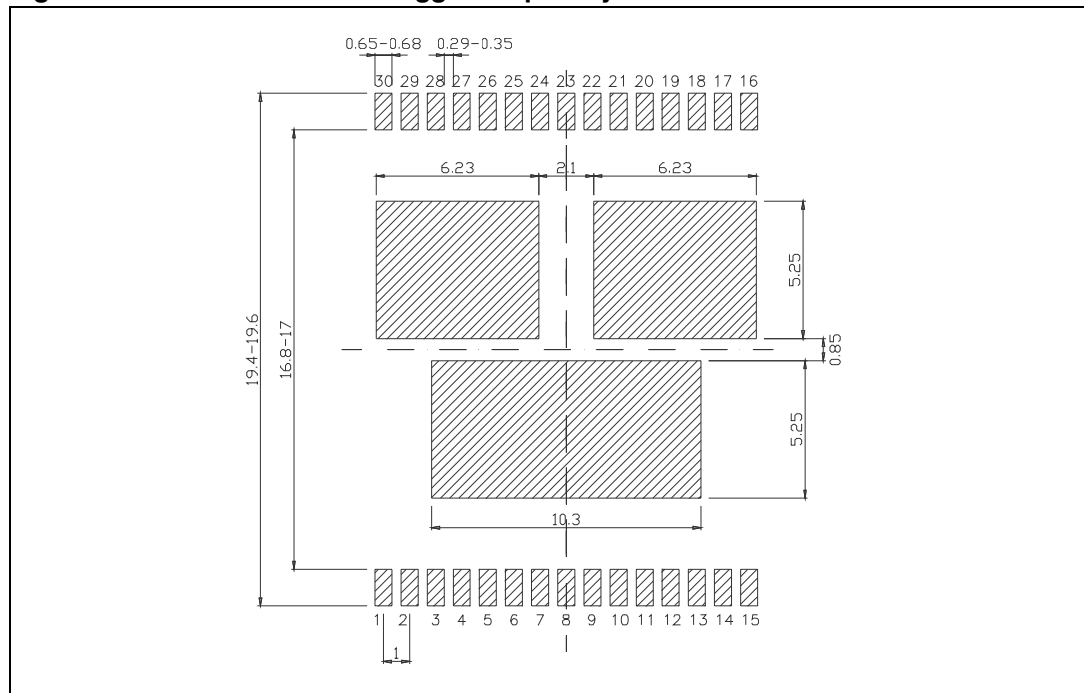


Table 17. MultiPowerSO-30 mechanical data

Symbol	Millimeters		
	Min	Typ	Max
A			2.35
A2	1.85		2.25
A3	0		0.1
B	0.42		0.58
C	0.23		0.32
D	17.1	17.2	17.3
E	18.85		19.15
E1	15.9	16	16.1
e		1	
F1	5.55		6.05
F2	4.6		5.1
F3	9.6		10.1
L	0.8		1.15
N			10deg
S	0deg		7deg

Figure 45. MultiPowerSO-30 suggested pad layout



5.3 Packing information

Note: The devices can be packed in tube or tape and reel shipments (see the [Device summary on page 1](#) for packaging quantities).

Figure 46. MultiPowerSO-30 tube shipment (no suffix)

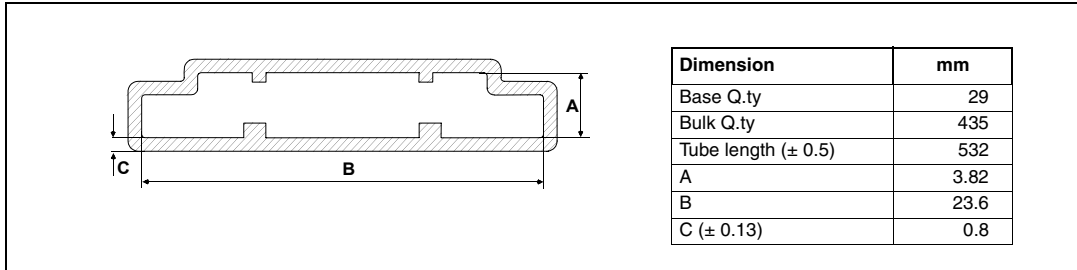
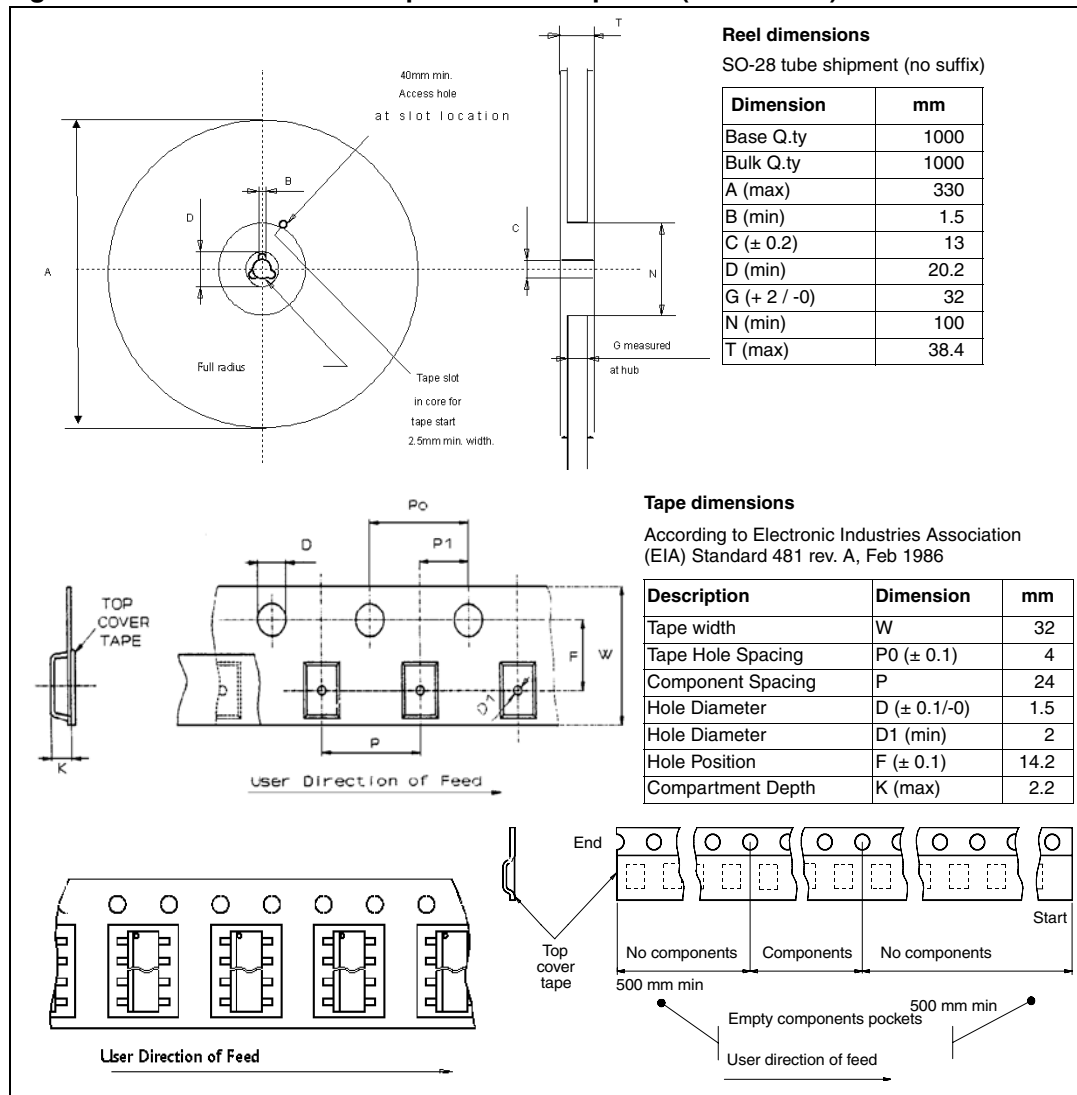


Figure 47. MultiPowerSO-30 tape and reel shipment (suffix “TR”)



6 Revision history

Table 18. Document revision history

Date	Revision	Description of changes
Sep-2004	1	First issue
Dec- 2004	2	Inserted $t_{off(min)}$ test condition modification and note Modified I_{RM} figure number
Feb-2005	3	Minor changes
Apr-2005	4	Public release
01-Sep-2006	5	Document converted into new ST corporate template. Added table of contents, list of tables and list of figures Removed figure number from package outline <i>on page 1</i> Changed <i>Features on page 1</i> to add ECOPACK® package Added <i>Section 1: Block diagram and pin description on page 5</i> Added <i>Section 2.2: Electrical characteristics on page 9</i> Added “low” and “high” to parameters for I_{INL} and I_{INH} in <i>Table 7 on page 9</i> Inserted note in <i>Figure 32 on page 20</i> Added vertical limitation line to left side arrow of $t_{D(off)}$ to <i>Figure 7 on page 13</i> Added <i>Section 4.1: PowerSSO-30 thermal data on page 25</i> Added <i>Section 5: Package and packing information on page 29</i> Added <i>Section 5.3: Packing information on page 31</i> Updated disclaimer (last page) to include a mention about the use of ST products in automotive applications
15-May-2007	6	Document reformatted and converted into new ST template.
06-Feb-2008	7	Corrected Heat Slug numbers in <i>Table 3: Pin definitions and functions</i> .
02-Oct-2008	8	Added new information in <i>Table 6: Power section</i> Added <i>Figure 33: Behavior in fault condition (How a fault can be cleared)</i>

Please Read Carefully:

Information in this document is provided solely in connection with ST products. STMicroelectronics NV and its subsidiaries ("ST") reserve the right to make changes, corrections, modifications or improvements, to this document, and the products and services described herein at any time, without notice.

All ST products are sold pursuant to ST's terms and conditions of sale.

Purchasers are solely responsible for the choice, selection and use of the ST products and services described herein, and ST assumes no liability whatsoever relating to the choice, selection or use of the ST products and services described herein.

No license, express or implied, by estoppel or otherwise, to any intellectual property rights is granted under this document. If any part of this document refers to any third party products or services it shall not be deemed a license grant by ST for the use of such third party products or services, or any intellectual property contained therein or considered as a warranty covering the use in any manner whatsoever of such third party products or services or any intellectual property contained therein.

UNLESS OTHERWISE SET FORTH IN ST'S TERMS AND CONDITIONS OF SALE ST DISCLAIMS ANY EXPRESS OR IMPLIED WARRANTY WITH RESPECT TO THE USE AND/OR SALE OF ST PRODUCTS INCLUDING WITHOUT LIMITATION IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE (AND THEIR EQUIVALENTS UNDER THE LAWS OF ANY JURISDICTION), OR INFRINGEMENT OF ANY PATENT, COPYRIGHT OR OTHER INTELLECTUAL PROPERTY RIGHT.

UNLESS EXPRESSLY APPROVED IN WRITING BY AN AUTHORIZED ST REPRESENTATIVE, ST PRODUCTS ARE NOT RECOMMENDED, AUTHORIZED OR WARRANTED FOR USE IN MILITARY, AIR CRAFT, SPACE, LIFE SAVING, OR LIFE SUSTAINING APPLICATIONS, NOR IN PRODUCTS OR SYSTEMS WHERE FAILURE OR MALFUNCTION MAY RESULT IN PERSONAL INJURY, DEATH, OR SEVERE PROPERTY OR ENVIRONMENTAL DAMAGE. ST PRODUCTS WHICH ARE NOT SPECIFIED AS "AUTOMOTIVE GRADE" MAY ONLY BE USED IN AUTOMOTIVE APPLICATIONS AT USER'S OWN RISK.

Resale of ST products with provisions different from the statements and/or technical features set forth in this document shall immediately void any warranty granted by ST for the ST product or service described herein and shall not create or extend in any manner whatsoever, any liability of ST.

ST and the ST logo are trademarks or registered trademarks of ST in various countries.

Information in this document supersedes and replaces all information previously supplied.

The ST logo is a registered trademark of STMicroelectronics. All other names are the property of their respective owners.

© 2008 STMicroelectronics - All rights reserved

STMicroelectronics group of companies

Australia - Belgium - Brazil - Canada - China - Czech Republic - Finland - France - Germany - Hong Kong - India - Israel - Italy - Japan - Malaysia - Malta - Morocco - Singapore - Spain - Sweden - Switzerland - United Kingdom - United States of America

www.st.com