

UNIVERSIDAD DE EXTREMADURA
ESCUELA DE INGENIERÍAS INDUSTRIALES

DISEÑO Y REALIZACIÓN A ESCALA DE UN SISTEMA DE APARCAMIENTO AUTÓNOMO

TRABAJO PRESENTADO PARA OPTAR AL TÍTULO DE
Grado en Ingeniería Electrónica y Automática (Rama Industrial)

POR

Alejandro Rubio González

BADAJOS, JUNIO DE 2016

TRABAJO FIN DE GRADO

Autor:

Alejandro Rubio González

Director o Directores:

Carlos J. García Orellana

Tribunal de evaluación:

José Luis Ausín Sánchez

J. Ignacio Suárez Marcelo

Juan Álvaro Fernández Muñoz

ÍNDICE

1	INTRODUCCIÓN.....	2
2	JUSTIFICACIÓN Y OBJETIVOS.....	4
3	MATERIALES Y MÉTODOS	5
3.1	Elementos Comerciales.....	5
3.1.1	Microcontrolador	5
3.1.2	Alimentación	9
3.1.3	Descripción del puente en H.....	11
3.1.4	Motores delantero y trasero.....	12
3.1.5	Descripción del encoder	14
3.1.6	Sensores de ultrasonido.....	16
3.2	Elementos desarrollados	21
3.2.1	Placa	21
3.2.2	Impresión de piezas en 3D.....	24
4	DISEÑO Y REALIZACIÓN DE SISTEMA DE CONTROL	30
4.1	Sistemas comunes.....	30
4.1.1	Implementación del Driver del Sensor de Ultrasonidos.....	30
4.1.2	Implementación del Driver de Dirección	32
4.1.3	Implementación del Driver del Motor de Propulsión.....	33
4.2	Descripción del sistema. Máquina de estados	34
4.3	Sistema de Control para el Paralelismo	36
4.4	Implementación del Algoritmo de Aparcamiento	39
5	RESULTADOS.....	41
6	CONCLUSIONES	47
7	BIBLIOGRAFÍA	49

1 INTRODUCCIÓN

El proyecto surge ante la necesidad de automatización de determinados procesos o tareas repetitivas, debido a la implantación de la tecnología para facilitar tareas de la vida cotidiana. Un campo importante sobre el que se está aplicando la automatización es el de la automoción, donde se lleva cada vez mayor grado sobre vehículos como coches, camiones o incluso otro tipo de transportes como trenes, tranvías, metros o aviones, donde existen tareas que realizan de forma autónoma los propios controladores de estos medios de transporte, facilitando el trabajo al conductor o incluso llegando a conseguir un mayor nivel de seguridad para los ocupantes, por evitarse situaciones de distracción del maquinista, que pudieran desembocar en peligrosos accidentes.

Por otro lado, dichos sistemas de ayuda electrónica, han ido evolucionando por la mejora de las técnicas de procesamiento y de la tecnología utilizada, consiguiéndose una mayor fluidez de los movimientos, así como la integración entre distintos dispositivos que llevan a cabo tareas paralelas.

En vehículos comerciales de gama media-alta, se implementan en los últimos años sistemas electrónicos de ayuda al conductor como sistemas de detección de obstáculos traseros y de ángulo muerto de espejos retrovisores, de alerta por posibilidad de somnolencia en carretera o detección y realización de la maniobra de aparcamiento. Este último sistema se presenta por la dificultad que tienen los conductores a la hora de maniobrar para conseguir un correcto aparcamiento en línea.

Por otro lado, los sistemas de aparcamiento ofrecidos por los fabricantes de vehículos suponen un coste elevado para el usuario. De esta necesidad parte la idea de una posible implantación del sistema de menor coste en un dispositivo de tamaño reducido, caso del coche de radio control objeto de este trabajo, como puente a un vehículo de mayores dimensiones.

En este trabajo de fin de grado se diseña, implementa y ajusta un sistema de aparcamiento automatizado en línea para un vehículo, detectándose el hueco y realizando la maniobra en el espacio disponible.

Al tratarse de un vehículo convencional, se define en el campo de la robótica como modelo de Ackerman [1], disponiendo el mismo de cuatro ruedas, donde las dos delanteras están dedicadas a la dirección y las traseras a la propulsión, montadas de forma paralela. Por definición, en el comportamiento en una curva, la rueda delantera interior gira un ángulo ligeramente superior a la rueda exterior, provocando que los ejes de prolongación de las ruedas delanteras se corten en el centro instantáneo de rotación, situado en un mismo punto del eje de prolongación de los ejes de las ruedas motrices. Esto elimina el deslizamiento que provoca los sobre virajes de la plataforma. Obviando las fuerzas centrífugas, los vectores de velocidad instantánea son tangentes a esta curva, por lo que las velocidades de movimiento del vehículo deberán evitar que las ruedas no resbalen. Por otro lado, el mayor problema que supone el modelo de Ackerman es la limitación en maniobrabilidad que supone su configuración.

El resto del Trabajo de Fin de Grado está organizado bajo la siguiente estructura.

El capítulo 2 justifica los objetivos por los cuales se lleva a cabo el estudio, y posterior trabajo, quedando relacionadas con las diferentes competencias del título cursado y por el que se presenta dicho trabajo de final de grado.

El capítulo 3 muestra los elementos comerciales y los desarrollados para el completo funcionamiento del sistema, quedando definidos y descritos tanto en características como en forma. Entre los elementos desarrollados se encuentra la placa donde se albergarán los componentes comerciales necesarios para su funcionamiento, así como la impresión de las piezas 3D para la acomodación de partes fijas sin ubicación previa. En el apartado de elementos comerciales se encuentran todas las partes restantes que conforman el sistema, siendo éstas de adquisición en tienda.

El capítulo 4 explica el sistema de control utilizado, incluyendo sistemas comunes, utilizados en todo momento por el microcontrolador para realizar el uso de los distintos motores y sensores. Posteriormente se describe la máquina de estados, el control de paralelismo, así como su estudio para el correcto ajuste del PID y por último, el algoritmo de aparcamiento con los estados por los que pasa el vehículo hasta completar el posicionamiento dentro del hueco del escenario.

El capítulo 5 muestra los resultados obtenidos en la realización del trabajo, donde se simula por ordenador la maniobra de aparcamiento, se establece la diferencia con un aparcamiento real mediante fotogramas y, por último, se compara con los datos tomados por el sensor lateral del vehículo sobre el escenario.

El capítulo 6 relaciona las conclusiones que se observan en relación con el proyecto realizado, así como posibles mejoras en el diseño del mismo o planteamientos futuros para el perfeccionamiento del control del sistema.

2 JUSTIFICACIÓN Y OBJETIVOS

El principal objetivo que se plantea en el presente trabajo es el desarrollo de un sistema autónomo que realice la búsqueda de aparcamiento en línea y la posterior maniobra del vehículo consiguiendo entrar y finalizar centrado en el hueco.

Con este proyecto se pretende desarrollar, mediante hardware y software, la resolución completa del aparcamiento, aplicando los conocimientos adquiridos tanto del hardware como del lenguaje de alto nivel utilizado en la plataforma de MBED.

Una parte importante del trabajo consiste en el uso e implementación de un sistema de control Proporcional - Integral – Derivativo aplicándose por tanto, el aprendizaje previo del mismo en la especialidad del título.

Además, es necesaria la implementación física de la circuitería donde se presentará el microcontrolador, por lo que se precisan conocimientos de electrónica, física y soldadura, así como de electrónica de potencia para el uso del convertidor DC-DC, el cual se explicará en capítulos posteriores.

Para el correcto ajuste de los parámetros de alimentación del sistema se emplean distintos instrumentos electrónicos de medida utilizados a lo largo de las prácticas en diversas asignaturas del título, que facilitan por ejemplo, la lectura del consumo en los bobinados de los motores, lo que permite comprobar los beneficios de la tecnología LiPo de las baterías.

Otro objetivo de este trabajo es el de evaluar el uso de los sensores de ultrasonidos de bajo coste, pudiéndose llevar a cabo una maniobra que requiere precisión de las medidas para el correcto aparcamiento del vehículo en el hueco.

Por último, se realizará la impresión en 3D de piezas necesarias para acoplar los distintos componentes del vehículo, necesitándose conocimientos de diseño en tres dimensiones. Este apartado favorece la creatividad del diseñador para encontrar la solución más óptima con la creación de los componentes necesarios en los programas de modelado 3D.

3 MATERIALES Y MÉTODOS

3.1 Elementos Comerciales

En este proyecto existen varios elementos básicos de electrónica necesarios para el correcto funcionamiento tanto del microcontrolador, de los motores de dirección y propulsión, como de los sensores necesarios para llevar a cabo los objetivos fijados.

Como base física del proyecto se utiliza un coche de radio control de juguete escala 1/16 de un Ferrari FFX, al que se le elimina tanto la alimentación por batería de 6 voltios, como toda la electrónica comercial que controla el vehículo, quedando tan solo para el uso del proyecto el chasis, el motor de propulsión, el motor de dirección y un pequeño encoder para el conocimiento del ángulo de giro.

Los componentes principales que se utilizarán se resumen en la siguiente lista:

- Microcontrolador MBED LPC1768.
- Batería tipo LiPo de 7,4 V.
- Conversor de voltaje DC-DC de Step-Down.
- Puente en H L293.
- Motor de dirección y de propulsión.
- Encoder.
- Sensores de proximidad por ultrasonidos.

3.1.1 Microcontrolador

Para el control del sistema, como centro de éste, se ha decidido utilizar un Microcontrolador MBED LPC1768 del fabricante NXP. Next eXPerience o NXP Semiconductors es una empresa líder en el sector fundada por la empresa holandesa Philips hace más de 50 años.

Los microcontroladores MBED son una serie de placas de microcontroladores ARM® diseñados para la creación rápida de prototipos. Este modelo en concreto, fue diseñado para el prototipo de todo tipo de dispositivos, en especial aquellos en los que se pretenda incluir Ethernet, USB y la flexibilidad de gran cantidad de periféricos o en los que se requiera el uso de memoria FLASH. La placa se ha diseñado para la incorporación en prototipados con PCB por puntos o para proyectos en Protoboard sin soldadura, por ello dispone de formato DIP (Dual In-line Package), incluyéndose, por otro lado, programador FLASH integrado mediante USB, agilizándose de esta forma el proceso de programación del dispositivo.

Se cuenta, por otro lado con 512 KB de memoria FLASH para el guardado del programa y toma de datos en ficheros internos, 35 KB de memoria RAM e interfaces de comunicación con el exterior

como Ethernet integrado, Host and Device USB, comunicación por bus CAN, SPI, I2C, ADC, DAC y otras interfaces de entrada/salida.

Como resumen, la placa de NXP LPC1768 MCU consta de las siguientes características principales:

- Procesador de alto rendimiento ARM® Cortex™ M3 Core.
- 96 MHz, 32KB de memoria RAM y 512KB de FLASH.
- Ethernet, USB Host/Device, 2xSPI, 2xI2C, 3xUART, CAN, 6xPWM, 6xADC, GPIO.

La placa en la que se integra el microcontrolador consta de las siguientes características:

- 40 pines de 0,1" en formato DIP, con dimensiones totales de 54x26 mm.
- Alimentación de 5V a través de USB o pines de entrada con rango 4,5-9 V.
- Programador FLASH drag 'n' drop USB integrado.

Las características del sitio web de MBED [2] para desarrolladores se resumen en:

- Compilador ligero online
- Uso de lenguaje de alto nivel C/C++ SDK.
- Librería de fácil acceso y comunidad de proyectos publicados.

Este sistema ha sido elegido por ser de reducido tamaño y formato DIP, lo cual favorece la integración en la placa PCB, bajo consumo, y de gran versatilidad de uso, teniendo variedad de métodos de comunicación con su alrededor. Dicho microcontrolador, como se ha comentado anteriormente, consta con un procesador ARM® Cortex™ M3 de 32 bits de alto rendimiento con una frecuencia de reloj de 96MHz, hecho que favorece la rapidez del sistema, evitando así la acumulación de tareas y la posibilidad del salto de *inputs* externos.

Una de las características de mayor interés para el proyecto se trata de que dicho componente cuenta con salidas PWM (Pulse Width Module) que son de gran utilidad para la utilización de motores, y una alimentación entre 4,5 y 9 V facilitándose la integración en el sistema, hecho que se estudia en los apartados siguientes.

El *pin-out* del microcontrolador está disponible en la página oficial del fabricante, el cual es de fundamental conocimiento para el uso de éste. En la Figura 3.1 se puede visualizar las diferentes configuraciones de las que dispone cada pin del MBED.

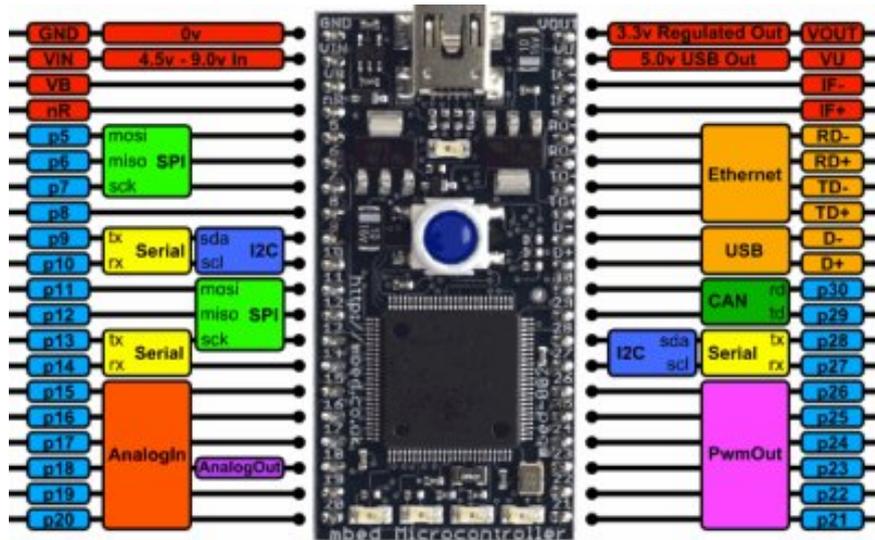


Figura 3.1 Esquema de conexionado de MBED NXP LPC1768.

La plataforma de MBED [2], mostrada en la Figura 3.2, proporciona una gran cantidad de bibliotecas, recursos y apoyo compartidos en una comunidad, así como el propio Manual de Usuario del sistema, facilitando la resolución de errores o la búsqueda de recursos necesarios para proyectos basados en microcontroladores ARM®.

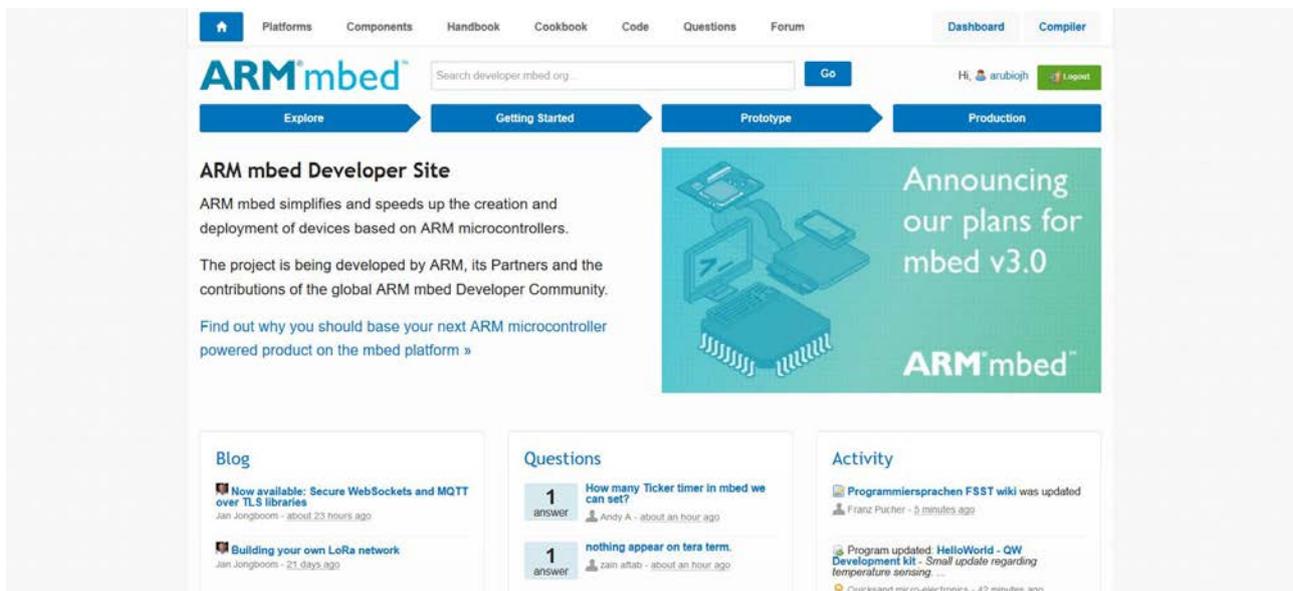


Figura 3.2 Página principal de la plataforma MBED.

Dicha plataforma incluye un kit de desarrollo basado en lenguajes de alto nivel C / C++ SDK (Software Development Kit), un soporte de desarrollo de placas o HDK (Hardware Development Kit), un compilador online y herramientas para desarrolladores, las cuales se comentaron anteriormente.

El Kit de Desarrollo de Software (SDK) es una plataforma de software C / C++ invocada por desarrolladores de todo el mundo para la creación de proyectos rápidos. MBED se ha preocupado de crear y probar el código de inicio, las bibliotecas y las API de los periféricos para que el usuario se dedique tan solo a la codificación de su proyecto. El Kit está disponible bajo la licencia de Apache 2.0, lo que permite su uso tanto en proyectos comerciales como en los personales. Por otro lado, el SDK ha sido diseñado para proporcionar suficiente abstracción de hardware, siendo de esta forma intuitivo y conciso, aunque lo suficientemente potente para construir proyectos complejos.

El Kit de Desarrollo de Hardware (HDK) es una plataforma que proporciona los archivos de diseño del subsistema completo de microcontroladores y firmware para el desarrollo de placas y productos personalizados beneficiados por el soporte nativo del SDK de MBED, el compilador online y la plataforma de desarrollo de MBED.

El compilador de MBED, Figura 3.3, es un IDE (Integrated Development Environment) gratuito para su uso con el hardware de la aplicación HDK. Detrás se encuentra la industria de ARM® con el uso del estándar C / C++, pre configurado y probado para generar el código rápido, de forma eficiente y sin ningún tipo de problema. La plataforma permite iniciar sesión en cualquier lugar para obtener acceso instantáneo al entorno de desarrollo a través de Windows, Mac o Linux, pudiéndose trabajar incluso a partir de archivos comprimidos o exportándose a diferentes herramientas.

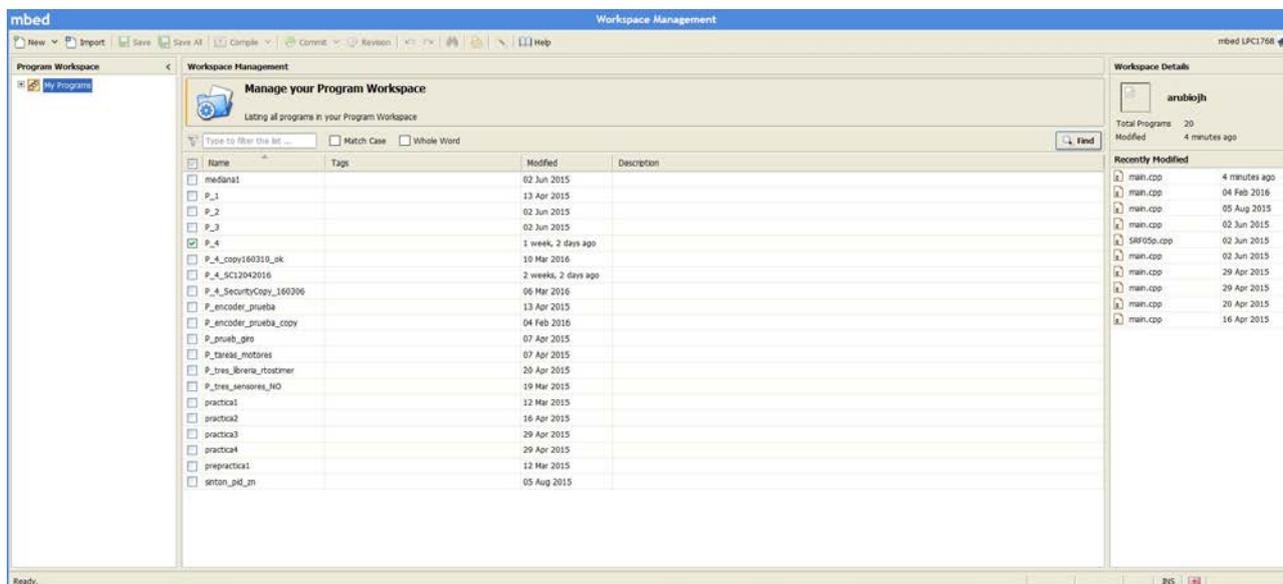


Figura 3.3 Compilador de MBED.

A pesar de que el compilador de MBED proporciona un espacio de trabajo privado, que está totalmente integrado con el sitio web del desarrollador MBED, se puede importar fácilmente las bibliotecas públicas y los ejemplos, así como publicar el código para colaborar fácilmente con otros usuarios. De esta forma se ha creado una comunidad de desarrolladores a nivel mundial que permite conseguir una reproducción de temas y respuestas en poco tiempo. Este sistema ha permitido crecer una comunidad activa y amigable de desarrolladores cualificados que están

ayudando colectivamente a los prototipos que se publican. Dichos desarrolladores también comparten miles de repositorios de código abierto, además de crear una extensa librería para los proyectos privados de los usuarios.

3.1.2 Alimentación

La alimentación de todo el sistema, desde el circuito hasta los motores y sensores, se realizará mediante una batería de 7,4 V de dos celdas de tipo LiPo (Litio y Polímero). Este tipo de baterías recargables se suele utilizar en sistemas eléctricos de radiocontrol, y es una opción muy viable respecto a otras baterías, ya que las baterías LiPo son más ligeras y con un tamaño reducido, teniendo una gran capacidad y una tasa de descarga alta. En contraprestación hay que tener precaución en el mal uso de éstas ya que utiliza un electrolito volátil que puede incendiarse o explotar.



Figura 3.4 Batería LiPo LRP de 7,4V 1.500 mAh.

La capacidad de la batería utilizada y mostrada en la Figura 3.4, es de 1.500 mAh y la descarga de 30 C, pudiendo suministrar picos de corriente de 30 veces su propia capacidad, evitándose cortes de tensión y por tanto eliminando la posibilidad de reseteo del microcontrolador, lo que ocasionaría un error en el correcto funcionamiento del sistema [3].

La carga de esta batería se realizará con un cargador apropiado para este tipo de tecnología con balanceo, el modelo elegido es un Dual Power Balance Charger B450AC, consiguiéndose un equilibrado de la tensión de todas las celdas de la misma. Por otro lado, este tipo de baterías permite una carga rápida de hasta 3C, es decir, de 3 veces su capacidad, siempre bajo supervisión por posible calentamiento.

Para la adecuación de la tensión de la batería a los 5 V estables con los que funciona la circuitería, se utiliza el convertor de voltaje DC-DC de Step - Down LM2596 de la Figura 3.2, correctamente regulado para tener el voltaje adecuado en la salida.



Figura 3.5 Conversor de voltaje DC-DC Step - Down LM2596.

Dicho ajuste se realiza mediante la resistencia regulable del convertor con ayuda de una fuente de tensión constante disponible en el laboratorio que permite establecer voltaje de salida e intensidad límite suministrados al convertor DC-DC. Para la correcta medida del voltaje que se pretende a la salida del convertor se utiliza un polímetro del laboratorio de electrónica, el cual nos dará una mayor exactitud en la toma del dato. Dicha resistencia variable se refleja en el esquema de la Figura 3.6, donde aparecen los componentes y conexiones que están presentes en la placa del convertor DC-DC del proyecto.

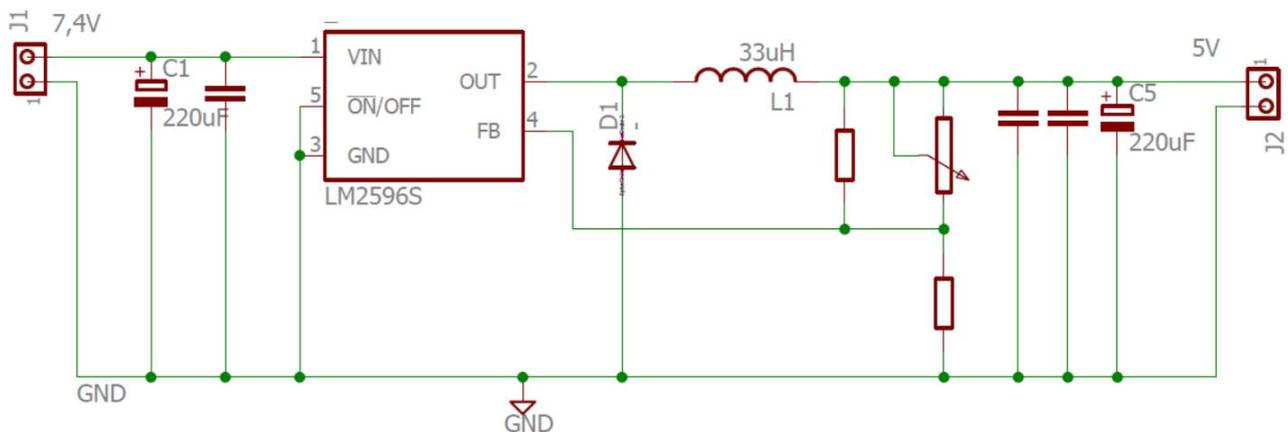


Figura 3.6 Esquema del convertor de voltaje DC-DC.

El conjunto de la batería de tecnología LiPo y el convertor de voltaje es una opción muy viable, debido a la gran estabilidad de tensión que proporcionan ambos sistemas, tanto la batería ante picos de corriente, como el convertor proporcionando en la salida el voltaje seleccionado estable ante cambios de tensión o de intensidad requeridos por el circuito.

Por otro lado, el convertor de voltaje DC-DC, según la hoja de características, mantendrá la tensión seleccionada siempre y cuando la fuente de alimentación externa no baje de los 5,5 voltios de entrada, caso que no se dará por utilizar este tipo de batería.

3.1.3 Descripción del puente en H

El Puente en H es utilizado en este proyecto para el control de los motores, tanto de dirección como de propulsión, optando por el modelo L 293 N, Figura 3.7, debido a que este componente tiene la capacidad de soportar una intensidad de salida de forma continua de 1 A, siendo por tanto óptimo por el consumo de intensidad que tendremos con los motores, a diferencia de su compañero de gama, el L293D, con diodos internos que ofrece menor intensidad de salida, según las hojas de características ofrecidas por el fabricante del componente.

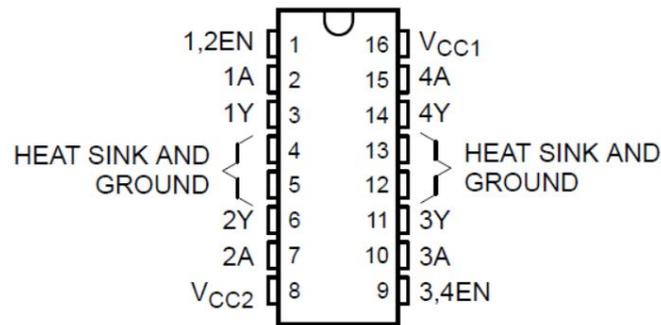


Figura 3.7 Puente en H L293 N.

Como inconveniente, se deberán colocar diodos externos dispuestos de la siguiente forma, para evitar picos de corriente inversa producida por los bobinados al tratar con motores, tal como se muestra en la Figura 3.8.

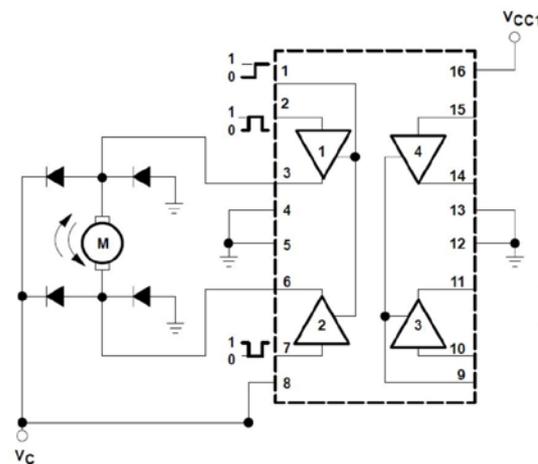


Figura 3.8 Esquema Puente en H. Conexión a motor DC.

Mediante el puente en H, se alimentan por una señal PWM cada motor, variando el porcentaje de voltaje medio que se suministra al bobinado. El motor, por ser un sistema mecánico con inercia, recibe una señal PWM pero es vista como el voltaje medio de la señal PWM dada. Según se varíe el porcentaje de esta señal, se tiene una mayor velocidad del motor trasero o una mayor rapidez de cambio de ángulo en la dirección.

Esta forma de control PWM se aplica directamente sobre los pines de entrada del L293N, pines 2 y 7 de la Figura 3.8. Por otro lado los ENABLE están colocados siempre en nivel alto, o activo, conectados al voltaje suministrado por el conversor, debido a que el control PWM se realiza sobre los descritos anteriormente.

3.1.4 Motores delantero y trasero

Los motores que se utilizan, son pequeño tamaño de corriente continua (DC), los cuales se deben alimentar entre 3 y 9 voltios. En este caso, se opta por alimentarlos a 5 voltios mediante el puente en H descrito anteriormente.

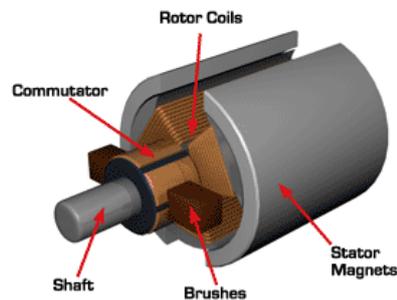


Figura 3.9 Partes de un motor DC.

El funcionamiento de estos motores de corriente continua (DC) o máquina de corriente continua se basa en la Ley de Lorentz, la cual establece la repulsión ejercida por los polos magnéticos de un imán presente en el estator cuando interactúan con los polos magnéticos de un electroimán bobinado que se encuentra en el eje del motor o rotor. [4]

Cuando la corriente eléctrica comienza a circular por la bobina del rotor, el campo electromagnético generado interactúa con el campo magnético creado por el imán permanente ocasionándose el giro. Dicha situación se produce cuando los polos del imán permanente y del bobinado coinciden, generándose un rechazo y un torque magnético o par de fuerza que provoca que el rotor comience a girar respecto al estator.

El sentido de giro del rotor sobre el estator viene dado por la combinación de las fuerzas provocadas por el sentido de la corriente y de acuerdo con la Fuerza de Lorentz, tal como se muestra en la Figura 3.10 simplificado de tan solo una espira en su bobinado.

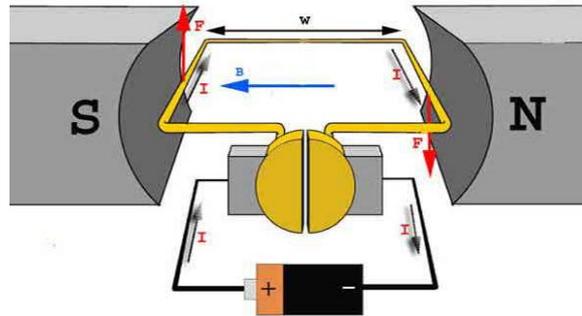


Figura 3.10 Esquema de motor de corriente continua.

Para el cambio de sentido de giro del motor tan solo es necesario cambiar la dirección de la corriente, provocándose por el cambio de los polos del motor sobre la batería.

El movimiento se termina produciendo por una parte clave del motor, la cual recibe el nombre de colector o conmutador, siendo visible en la Figura 3.9, al igual que las demás partes del motor. Esta parte se encuentra conectada a la corriente mediante las escobillas, y es necesaria para conmutar o cambiar constantemente el sentido de la circulación de la corriente eléctrica a través de la bobina del rotor cada medio ciclo de giro. De esta forma el polo norte del bobinado coincidirá con el polo norte fijo del imán permanente, así mismo, el polo sur, con el sur del imán. Al darse esta situación de coincidencia constante, se produce un rechazo continuamente entre ambos polos, lo que provoca el movimiento permanente del rotor siempre que esté conectado a la corriente eléctrica.



Figura 3.11 Motor DC de propulsión.

Para el proyecto, se debe tener en cuenta que motor de propulsión, Figura 3.11, estará puesto en funcionamiento durante todo el proceso de aparcamiento, realizando paradas y puestas en marcha, que provocarán picos en la demanda de corriente del circuito de alimentación. Por otro lado, este motor es de mayor tamaño, por lo que tendrá mayor bobinado y por ende mayor consumo de energía, llegando a un máximo de 0.7 A. Este parámetro es decisivo a la hora de la elección del puente en H que alimenta los motores.

Respecto al motor de dirección, se trata también de un motor en corriente continua (DC), en este caso el de la Figura 3.12, de menor tamaño, pero que se encontrará funcionando por pequeños

intervalos de tiempo, y por tanto, provocando picos de demanda en el arranque, propios de los motores al comenzar su movimiento.



Figura 3.12 Motor DC de dirección.

El giro en ambos motores se producirá mediante el uso de señales PWM dependiendo de las necesidades en cada momento, tal y como se describirá en el apartado 4.4.

3.1.5 Descripción del encoder

Este sensor es un encoder absoluto mecánico o codificador rotatorio mecánico, visible en la Figura 3.13, el cual es utilizado para conocer la posición de la dirección en cada momento. Este elemento cuenta con 4 pistas codificadas en un disco, leídas mediante contactos, creándose un código binario [1] [5].

Un encoder absoluto mecánico está formado por un circuito impreso de anillos grabados con un patrón. Con ello una fila de contactos lee el patrón, mostrado en la Figura 3.14, correspondiendo cada contacto a un anillo, de esta forma cuando los contactos rotan respecto al disco, se tocan o no las pistas, dependiendo de si hay alguna abertura en una posición concreta proporcionando una lectura u otra.



Figura 3.13 Aspecto físico de encoder mecánico.

Para que el sensor funcione, una corriente eléctrica es aplicada a la instalación de forma que cuando uno de los contactos toca alguno de los anillos, se cierra el circuito detectándose un 1 en caso de que se haya cerrado o un 0 cuando el circuito queda abierto. Esta combinación binaria produce el código usado para interpretar la ubicación de giro.

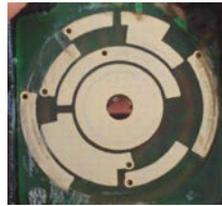


Figura 3.14 Patrón del circuito del encoder mecánico.

Este dispositivo por tanto, es utilizado para convertir la posición angular del eje de la dirección a un código digital Gray de 4 bits para su posterior procesado por parte del microcontrolador.

Tabla 3.1 Código Gray utilizado por el encoder

Código	Situación	Grados
1000	Derecha	29
1001		24,86
1011		20,72
1010		16,58
1110		12,44
1100		8,3
1101		4,16
1111	Centro	0
0111		-4,16
0101		-8,3
0100		-12,44
0110		-16,58
0010		-20,72
0011		-24,86
0001	Izquierda	-29

Partiendo del desconocimiento de la tabla de valores digitales a los que corresponde cada ángulo del encoder, es necesario en primer lugar realizar la comprobación de las soldaduras y el correcto contacto interno de los terminales del dispositivo. Tras esta verificación, se realiza un pequeño programa simple con el que se comprueba la tabla de codificación interna del encoder, leyendo la información mediante una conexión puerto serie por USB del MBED, e imprimiendo por pantalla la información deseada, tanto de la codificación binaria como del ángulo en el que se encuentra, completándose de esta forma cada una de las posiciones de la Tabla 3.1, por las que pasan los contactos internos del encoder.

La adecuación hardware del encoder se realiza mediante resistencias de Pull-Up de 10K Ω excepto la masa, tal como se muestra en el esquema de la Figura 3.15.

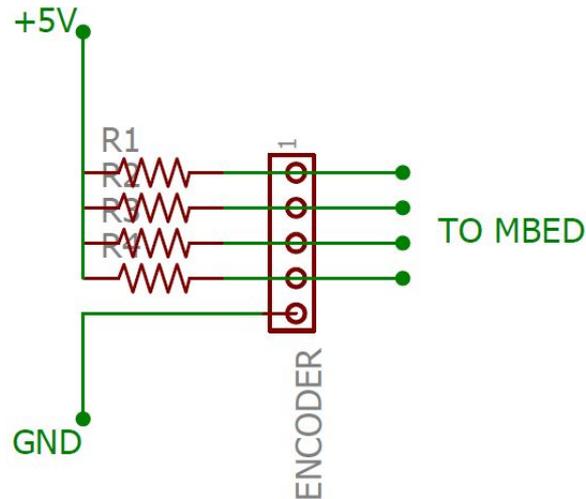


Figura 3.15 Esquema de conexionado del encoder.

Su conexión al microcontrolador se realiza mediante 4 entradas, en las que se leen señales digitales, tal como se ha descrito anteriormente. En el programa se describe la Tabla 3.1 para descifrar dicha codificación y tratar con ella como ángulos exactos para cada posición del encoder.

3.1.6 Sensores de ultrasonido

Para la medición de las distancias y conocimiento de la posición respecto a la pared a la que se encuentra el vehículo, se utilizan sensores de ultrasonidos modelo HC-SR04, como el de la Figura 3.16, que son compatibles con el modelo original SRF-04. Los sensores montados se caracterizan por ser de bajo consumo, fácil adecuación y bajo coste, convirtiéndolos en el tipo de sensor más popular en robótica móvil.



Figura 3.16 Sensor de ultrasonidos HC-SR04.

Estos sensores de rango activo o sensores de tiempo de vuelo se emplean para detección y evitación de obstáculos, tanto como para localización y modelado del entorno. [1][5] Para ello, se fundamentan en la velocidad de propagación del sonido, basándose en el principio de transmitir

un paquete de ondas de ultrasonido y medir el tiempo que tarda en reflejarse en algún objeto y volver al propio sensor, dándose la distancia resultante que recorren las ondas por la ecuación (3.1)

$$d = \frac{c \times t}{2} \quad (3.1)$$

Donde d es la distancia viajada por la onda, contando tan solo el camino de ida, t es el tiempo de vuelo, y c la velocidad de propagación de la onda, definida por la ecuación (3.2), siendo en el caso de los ultrasonidos la velocidad de propagación de las mismas dependiente del índice de calor específico γ , de la constante universal de los gases R y de la temperatura en grados Kelvin T . En situaciones normalizadas, esta velocidad es de 340 m/s.

$$c = \sqrt{\gamma \cdot R \cdot T} \quad (3.2)$$

El rango de medida de este modelo de sensor llega desde los 2-3 cm a un máximo de unos 3 metros, esquematizado sobre la Figura 3.17, siendo ésta una distancia mayor a la que se va a encontrar el vehículo de la pared, entrando igualmente el valor mínimo dentro del rango de trabajo, por lo que es completamente válido para el proyecto.

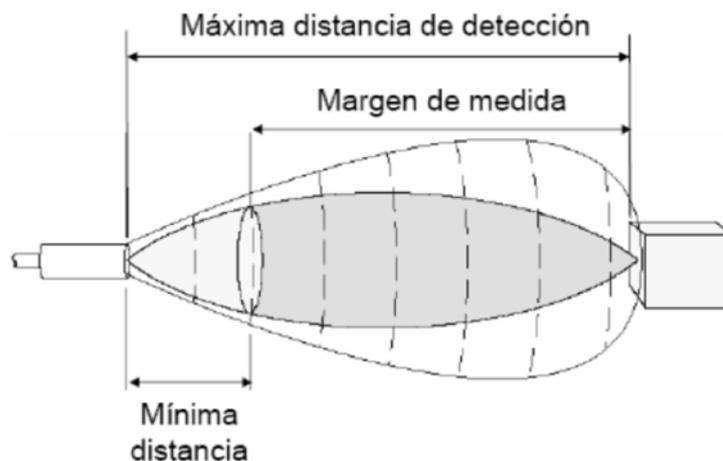


Figura 3.17 Esquema de distancias máxima y mínima del sensor.

Por otro lado, el consumo es de menos de 50 mA por cada uno de los tres sensores, conservándose consumo bajo respecto al amperaje demandado por los motores.

Para su conexión con el sistema, se necesitan dos pines de alimentación, conectándose a los 5 V proporcionados por el convertor DC-DC, y dos de comunicación con el MBED, TRIGGER, que debe ser salida digital y ECHO, recepción digital.

El funcionamiento de estos sensores requiere de un pulso de TRIGGER de unos 10 μ s para comenzar la medida. El sensor ultrasonido, una vez lanzado el pulso de TRIGGER, emite una ráfaga de 8 ciclos de ultrasonidos a una frecuencia de 40 kHz, activándose la señal de ECHO y comenzando el cronometraje hasta recibir el primer rebote en el objeto, momento el cual, la señal se vuelve a nivel bajo, siendo de esta forma proporcional a la distancia a la que se encuentre el objeto a medir, quedando descrito por el diagrama de tiempo de la Figura 3.18.

En el caso de que no se detecte nada, el sensor ultrasonido SRF04, pondrá automáticamente la señal de ECHO a nivel bajo tras pasar 30 ms.

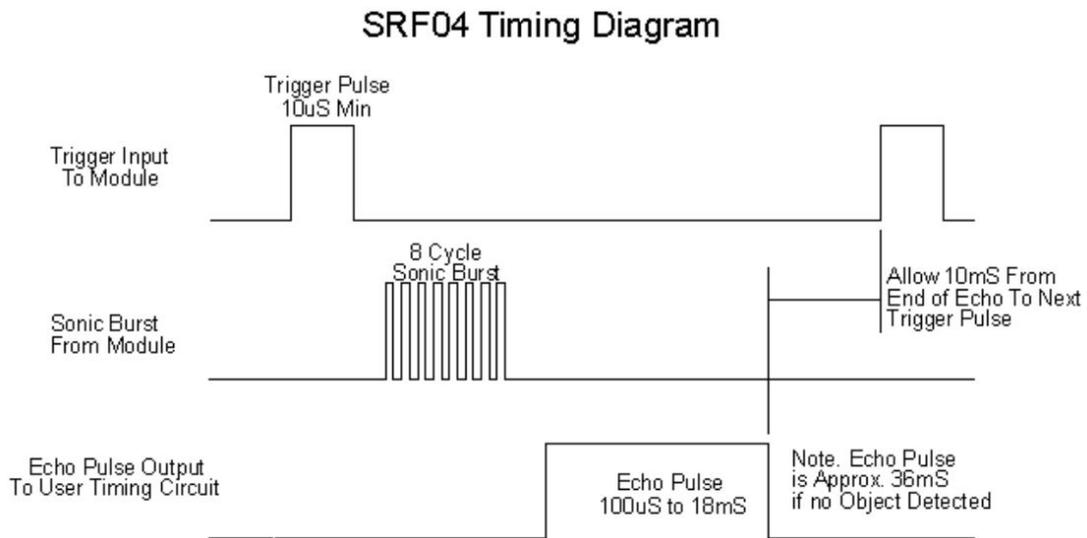


Figura 3.18 Diagrama de tiempos de funcionamiento del sensor SRF-04.

El circuito ha sido diseñado para ser de bajo coste, usando un PIC12C508 para el control de las funciones y realizar el pulso de los 40 kHz de los transductores piezoeléctricos, pudiendo ser el driver para la transmisión simplemente dirigido desde el PIC.

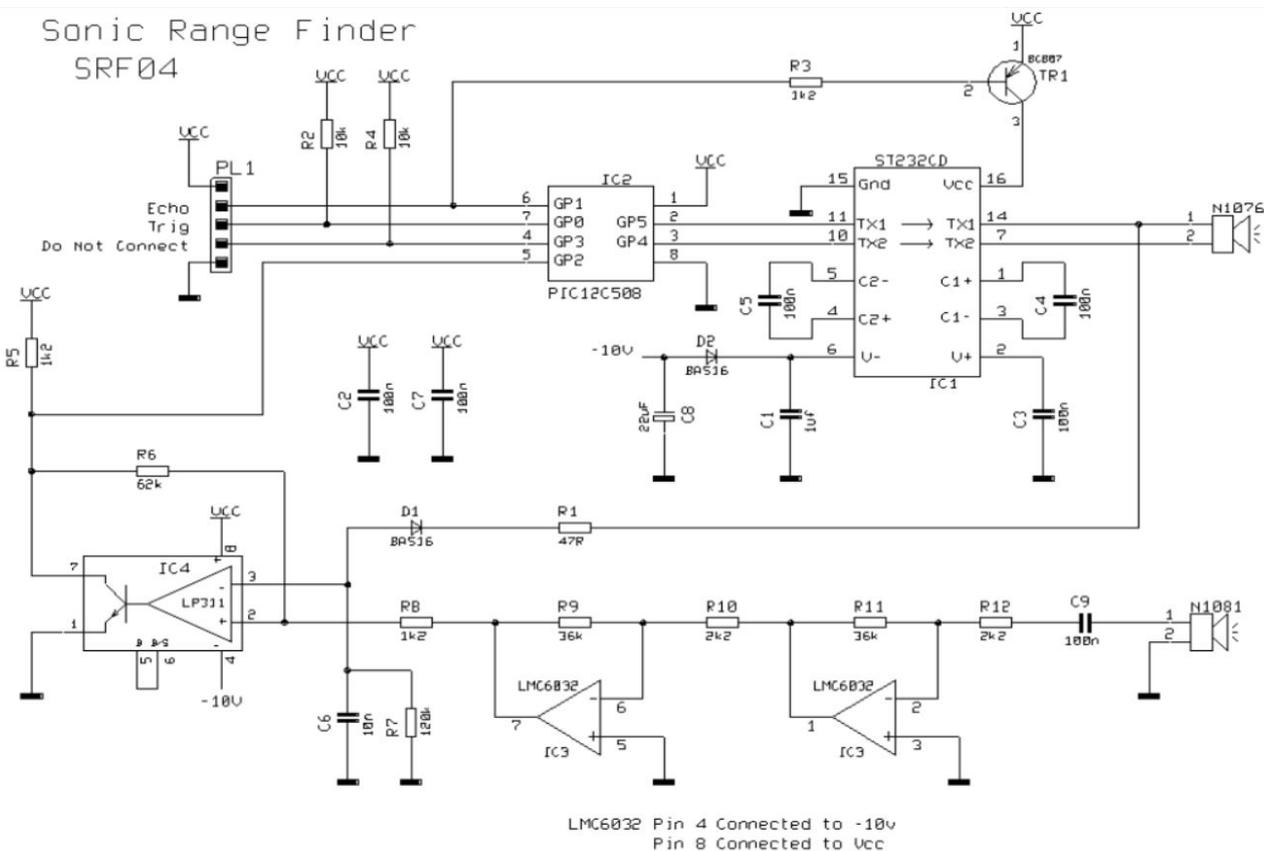


Figura 3.19 Esquema de sensor de ultrasonidos SRF-04.

La parte del esquema del sensor dedicada a la recepción se basa en un circuito de amplificador operacional de dos etapas, constando de componentes para adecuación de la señal como es el caso del condensador C8, que bloquea la posible carga residual de DC que pueda presentarse, encontrándose en el esquema de la Figura 3.19.

Cada etapa de ganancia se marca como 24 hasta un total de 576, donde se está cerca de la ganancia máxima disponible de 25 utilizando el LM1458, conocida su ganancia de ancho de banda de 1 MHz, operación realizada en la ecuación (3.3).

$$\frac{1.000.000}{40.000} = 25 \quad (3.3)$$

La salida del amplificador se alimenta en un comparador LM311, realizándose una pequeña retroalimentación provocando una cierta histéresis para proporcionar una salida estable.

La causa por la que éste sensor no es capaz de funcionar a una distancia muy pequeña es debida a que el receptor debe tomar el acoplamiento directo del transmisor que está situado justo a su lado.

Por otra parte, el transductor piezoeléctrico es un sistema mecánico que continua resonando algún tiempo después de terminarse de detectar el objeto, por lo que es muy difícil establecer la diferencia entre el timbre de acoplamiento directo y el eco de retorno.

Sobre el eco de retorno en un osciloscopio se puede observar como es mucho más grande en magnitud a corta distancia de la señal de acoplamiento cruzado, por lo que se ajusta el umbral de detección durante este tiempo de manera que solo el eco es detectable.

Posteriormente, el condensador C10 de 100 nF, se carga sobre -6 V. durante la ráfaga descargándose rápidamente a través de la resistencia R6 de 10 kΩ para restaurar la sensibilidad para ecos de mayor distancia. Este voltaje negativo para el amplificador operacional y el comparador se genera por un MAX232. Desafortunadamente, éste genera cierto ruido de alta frecuencia, por lo que se cierra mientras se escucha el eco, con ayuda del condensador C9 de 10uF.

Una vez se detecta el eco de la señal producida, el MAX232 se debe recargar, por lo que se espera al menos 10 ms entre ciclos de medición para recargar el voltaje necesario de nuevo.

En estos sensores se aplica alta frecuencia para lograr una mejoría en la detección de objetos de menor tamaño, ya que de esta forma la longitud de onda es menor que los obstáculos. Por otro lado, con esta solución se obtienen menores interferencias acústicas a pesar de disminuir el máximo alcance de medida.

Este proyecto puede realizarse con sensores de ultrasonidos por tratarse de superficies planas de fácil detección, así como de materiales no absorbentes de ondas sonoras. Por otro lado, en el escenario no existe ninguna superficie con gran ángulo de inclinación, ya que en este caso, la señal será reflectada en otro sentido alejándola del sensor y dificultando la medida, obteniéndose distancias erróneas.

Por último, se muestra visualmente la forma de la distribución del haz de ultrasonidos del HC-SR04, siendo cónica con el ancho en función de la superficie de los transductores. El patrón de haz

de los transductores, tomado de la hoja de características del fabricante, se muestra a en la Figura 3.20.

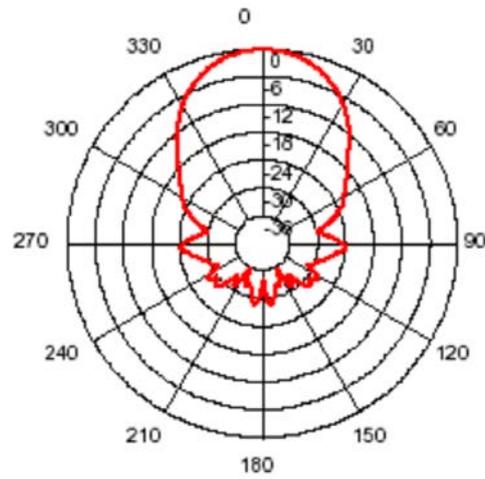


Figura 3.20 Patrón de haz de ultrasonidos.

3.2 Elementos desarrollados

3.2.1 Placa

El diseño de la placa donde se sitúa el microcontrolador se realiza en primer lugar en la herramienta informática EAGLE (Easily Applicable Graphical Layout Editor), que se trata de un programa de diseño de diagramas y PCB's con autoenrutador. Este programa es conocido por disponer de licencia gratuita y ser de fácil uso, teniendo editor tanto de esquemas como de diseño de placas de circuito impreso.

Con el uso de esta herramienta se obtiene la mejor combinación por cercanía a las conexiones de los distintos elementos que se sitúan alrededor del microcontrolador, siendo la solución la mostrada en la Figura 3.21.

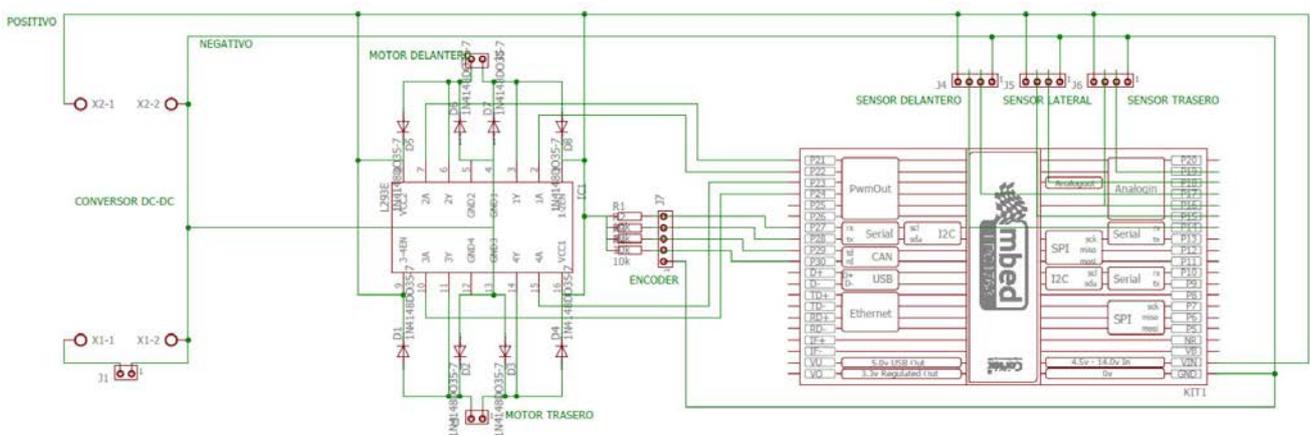


Figura 3.21 Diseño de la placa en herramienta EAGLE.

De esta misma herramienta se extrae el diseño de la placa de circuito impreso del sistema, realizado con la misma filosofía de optimización que el esquemático anterior. Debido al cruce de pistas que se observa en la Figura 3.21, ha sido necesario el diseño de la PCB en dos caras.

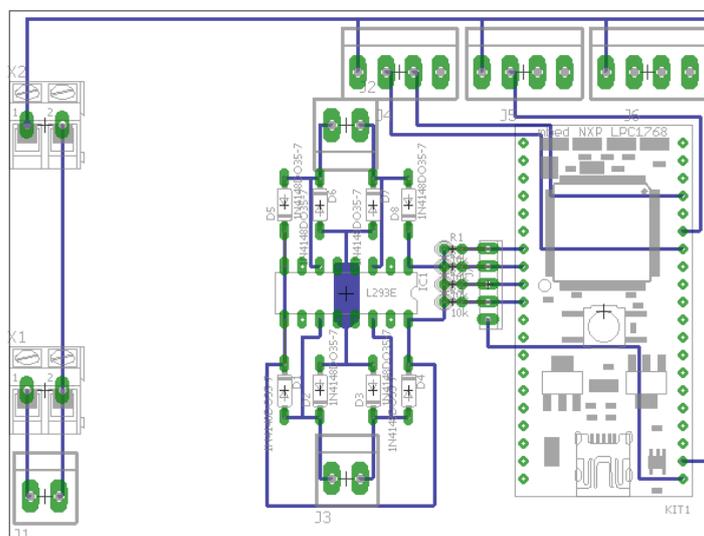


Figura 3.22 Diseño de la cara inferior de la PCB en la herramienta EAGLE.

La cara inferior de la PCB se representa en la Figura 3.22 donde las pistas aparecen de color azul, encontrándose la mayoría de las conexiones de alimentación negativa. En el centro de la imagen se puede observar la zona de disipación de calor del L293N, dispuesta con una mayor superficie de cobre bajo el mismo.

El resto de las pistas necesarias se encuentran en la cara superior, en la Figura 3.23, donde se representan en color rojo. Estas pistas forman la minoría del conexionado del circuito, ya que se trata tan solo de los pocos cruces producidos en la capa inferior.

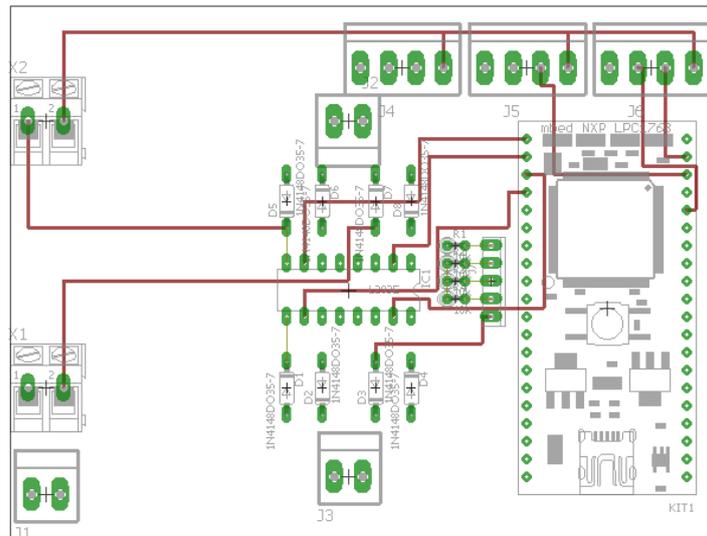


Figura 3.23 Diseño de la cara superior de la PCB en la herramienta EAGLE.

Posteriormente se procede con el montaje que se realiza sobre una placa de prototipado microperforada de puntos de soldadura, presentándose los componentes anteriormente diseñados en la herramienta EAGLE, tal como se muestra en la Figura 3.24, tomando como centro de partida el microcontrolador MBED. Una vez correctamente distribuida y optimizando en mayor medida el espacio, son soldados todos los componentes en su posición definitiva, estañando las pistas necesarias mediante unión de los puntos que se precisan de la placa o por medio de la utilización de puentes de conductores, buscándose en todo momento la menor distancia entre los puntos de unión para evitar posibles pérdidas, enganches con otros elementos móviles del vehículo e incluso roturas.

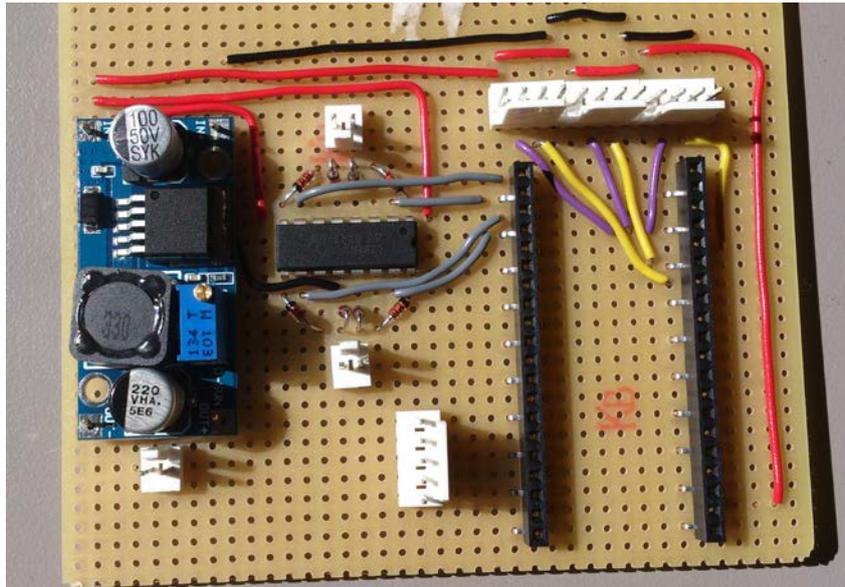


Figura 3.24 Presentación de componentes en placa.

Para las conexiones de los dos motores, los tres sensores y el encoder, se utilizan respectivamente conectores de dos, cuatro y cinco pines, así como, para la situación del microcontrolador y el puente en H en la placa se decide utilizar zócalos consiguiendo de esta forma facilidad de cambio en caso de fallo de alguno de los componentes, y evitar fatiga de los componentes al soldarlos a la placa.

Por otro lado, la energía se distribuye a lo largo de la placa mediante el convertidor DC-DC descrito en apartados anteriores, y desde el que se suministran los 5 V a cada elemento, siendo éstos y en este orden: el Puente en H, los tres sensores de ultrasonidos, el encoder y, por último, el MBED, el cual es alimentado mediante los pines V_{in} y GND disponibles en el mismo. El resto del conexionado que se utiliza para cada uno de los componentes al microcontrolador MBED se resume en la siguiente tabla:

Tabla 3.2 Tabla de conexionado a MBED

Elemento	Pin elemento	Pin MBED	Tipo de uso
Ultrasonido Delantero	Trigger	P17	Digital Out
	Echo	P14	Digital In
Ultrasonido Lateral	Trigger	P18	Digital Out
	Echo	P15	Digital In
Ultrasonido Trasero	Trigger	P19	Digital Out
	Echo	P16	Digital In
Puente en H	Motor dirección	P21	PWM
		P22	PWM
	Motor propulsión	P23	PWM
		P24	PWM
Encoder	Bit 0	P27	Digital In
	Bit 1	P28	Digital In
	Bit 2	P29	Digital In
	Bit 3	P30	Digital In

Por último, la placa se sitúa sobre el modelo a escala mediante tornillos con arandelas aislantes para evitar algún posible contacto con cualquier otro elemento o pista cercana.

3.2.2 Impresión de piezas en 3D

Para la correcta adecuación de los sensores y la batería en el modelo de radio control del vehículo que se ha elegido para el prototipo, se ha decidido imprimir en 3D las piezas necesarias. Para ello es necesario una impresora 3D, siendo esta un dispositivo que utiliza la tecnología de impresión en tres dimensiones para crear objetos, los cuales se construyen capa a capa sobre una plataforma o cama caliente, conociéndose este tipo de proceso como fabricación o manufactura aditiva.

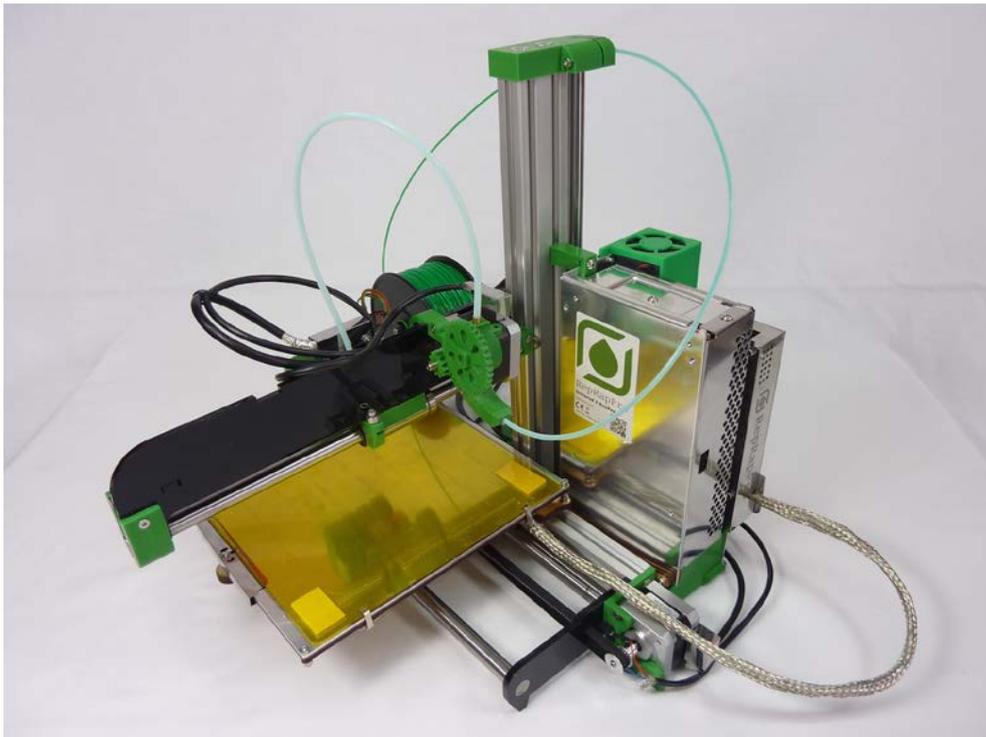


Figura 3.25 Impresora 3D RepRapPro Ormerod 2.

El proceso de fabricación aditiva es un avance importante en la fabricación de objetos. Esto se debe a que el procedimiento actual de fabricación, generalmente implica la reducción o la alteración del material para llegar al objeto deseado, así mismo estos procedimientos requieren una cantidad significativa de maquinaria y un número considerable de personal cualificado. El funcionamiento de la impresión en tres dimensiones comienza con la creación de un diseño, más conocido como modelo, el cual es un archivo con la representación digital del objeto que se va a imprimir. Este archivo es enviado y leído por el software interno de la impresora, enviando instrucciones a la máquina, que comienza a fabricar el objeto por capas.

La impresión en tres dimensiones, implica el uso de un archivo digital o modelo, creado mediante el uso de un software de modelado, siendo la herramienta elegida para esta labor el programa Inventor, de la conocida empresa Autodesk.

Para el correcto y mayor optimizado diseño de las piezas a imprimir, en primer lugar se ha necesitado realizar la medición de los tamaños de los sensores de ultrasonidos, así como de la batería con un escalímetro digital, disponible en el laboratorio de electrónica, tal como se muestra

en la Figura 3.26, para conseguir una perfecta unión de las piezas sin necesidad de la utilización de ningún tipo de adhesivo o goma que las sujeten.

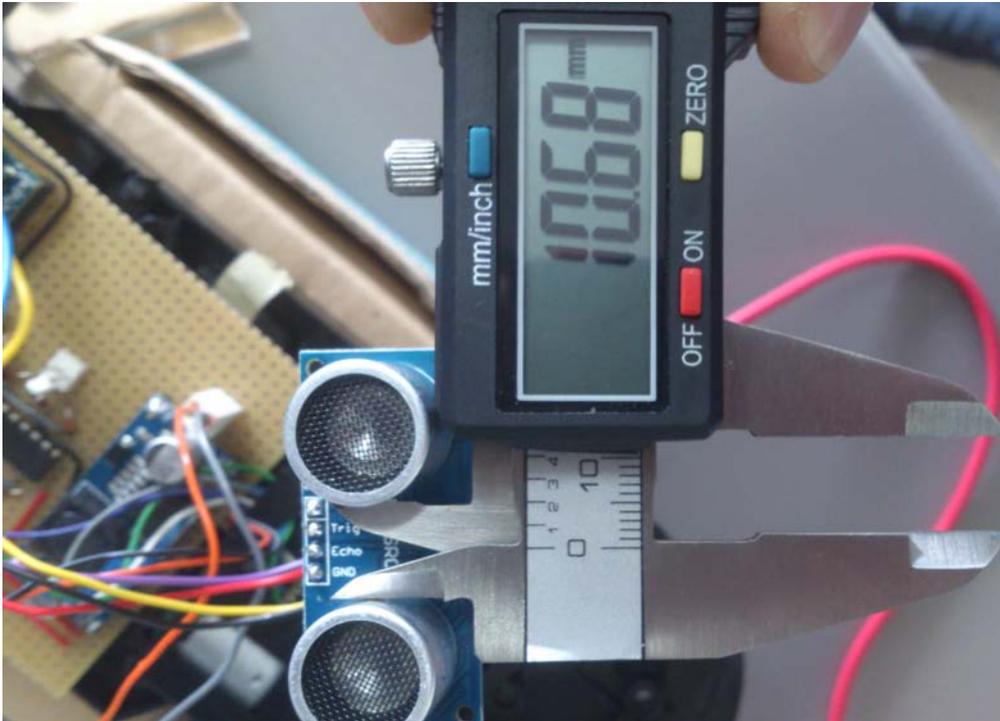


Figura 3.26 Toma de medidas de los sensores de ultrasonidos.

La primera de las piezas diseñadas en esta herramienta gráfica se trata del soporte para el sensor de ultrasonidos delantero, adoptando la forma que se muestra en la Figura 3.27.

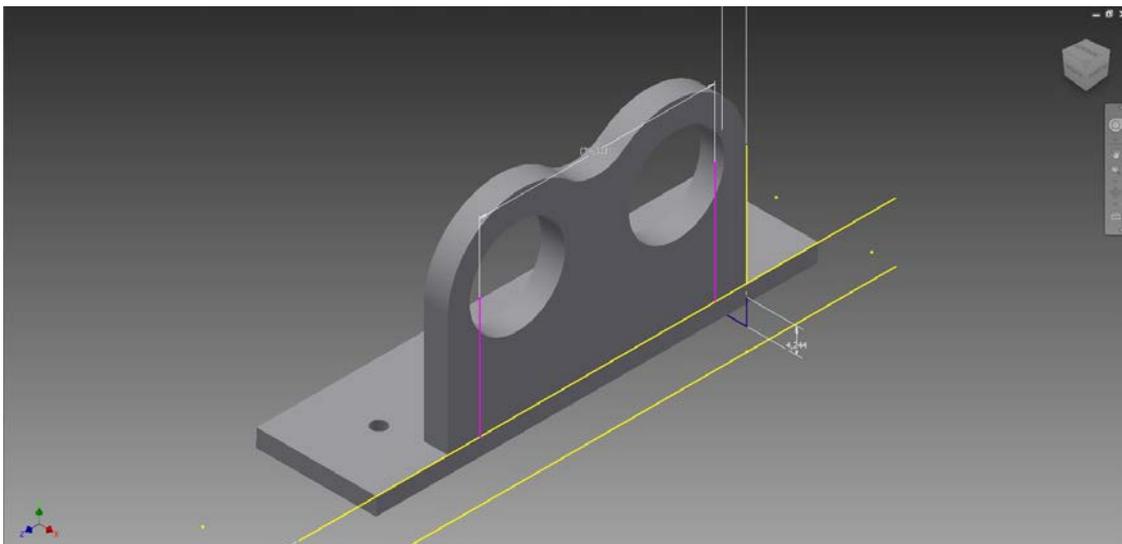


Figura 3.27 Modelo 3D del soporte delantero en la herramienta Inventor.

Respecto a la pieza de la Figura 3.28, se trata del soporte posterior para el sensor trasero y lateral derecho del vehículo. Esta pieza permite que el sensor lateral se encuentre justo en el eje trasero del vehículo permitiendo una mejor toma de datos. Además de esta pieza se realizó la impresión de dos arandelas separadoras de 2 cm. para conseguir la altura final necesaria para los sensores respecto al chasis del vehículo.

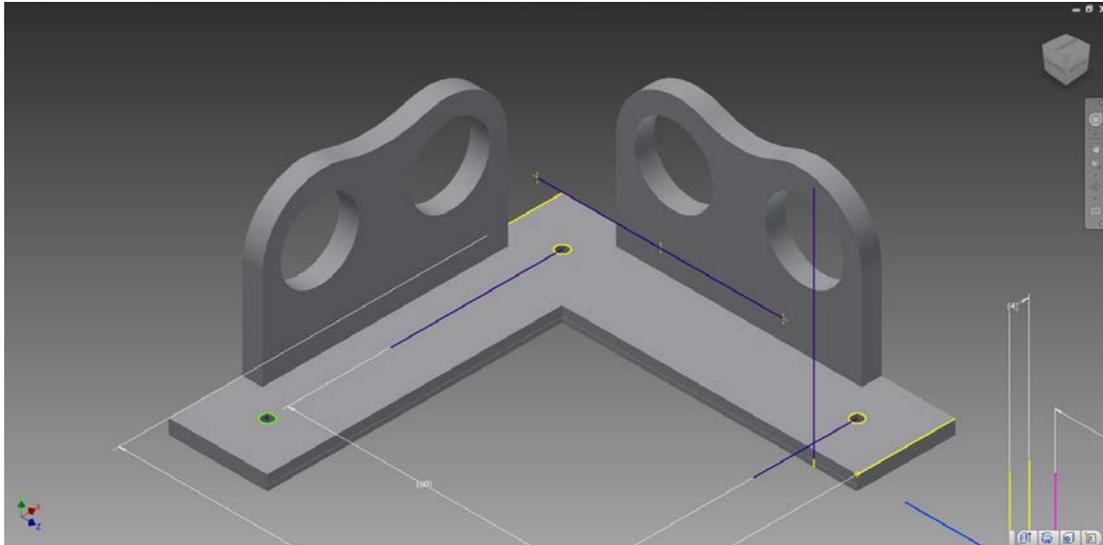


Figura 3.28 Modelo 3D del soporte posterior en la herramienta Inventor.

Por último, se realiza el diseño de la “cama” en la que se sitúa la batería, pudiéndose observar en la Figura 3.29.

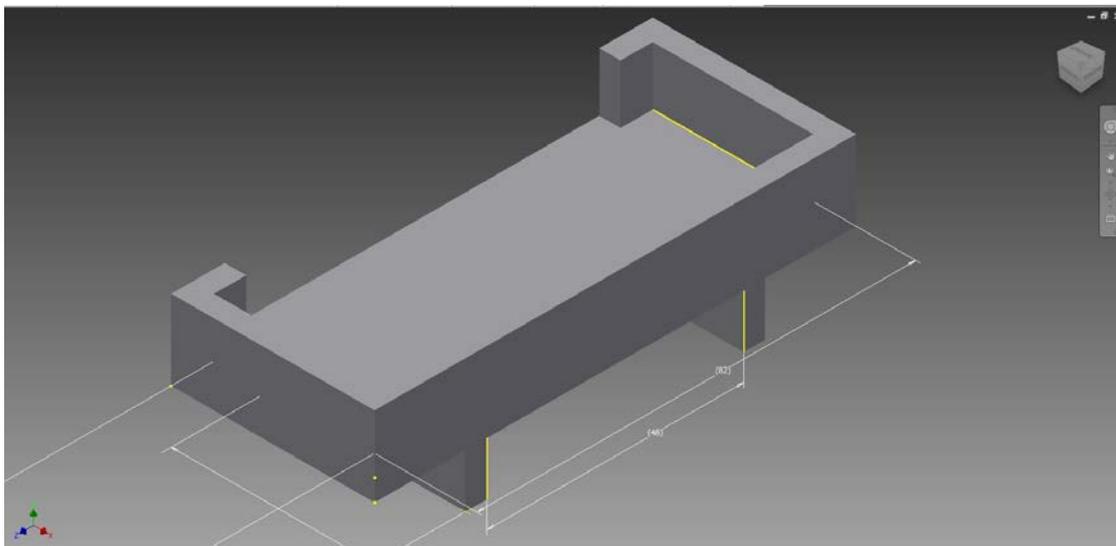


Figura 3.29 Modelo 3D de la cama para la batería en la herramienta Inventor.

También es posible obtener un modelo 3D digitalizando un objeto con el uso de un escáner 3D que crea un patrón digital utilizando el modelo real de la misma manera que un escáner 2D, crea un archivo digital de un documento en papel.

Crear un modelo usando un software de diseño o un escáner permite obtener un archivo en tres dimensiones que contiene información significativa incluyendo las dimensiones y el tamaño del objeto. Este objeto se puede almacenar como un archivo STL (abreviatura de estéreo litografía). El archivo STL es leído por el software que maneja la impresora (Generador de código G), que divide el modelo 3D en capas horizontales minuciosamente finas, de la misma manera que una resonancia magnética divide el cuerpo humano en “rebanadas”. Este software interno procede a enviar instrucciones a la impresora, que comienza a crear el objeto a partir de la base hasta que alcanza la parte superior del mismo.

El funcionamiento de la impresora 3D es complejo, con más de una docena de diferentes tecnologías existentes actualmente dedicadas a este tipo de impresión. A pesar de sus diferencias, estas tecnologías funcionan todas con el mismo principio, con la diferencia principal de la forma de estratificar el material.

La impresora 3D que se ha utilizado es la denominada Ormerod 2 de la Figura 3.25, del fabricante inglés RepRapPro, la que se ha montado y ajustado para su correcto funcionamiento. En ella se ha realizado la impresión de las piezas con un polímero termoplástico y biodegradable con el nombre de PLA.

La RepRapPro Ormerod 2 es la actualización de la antigua Ormerod, la que ha sido diseñada con carácter educativo siendo útil para crear prototipos rápidos y ahorrar meses en el ciclo de diseño. Esta impresora tiene la capacidad de imprimir en varios colores y plásticos con el proceso conocido como *fused filament fabrication* (FFF), pudiéndose crear casi cualquier forma modelada en ordenador. El software utilizado por la misma funciona con cualquier tipo de plataforma en PC, contándose con Ethernet, USB e incluso microSD para impresión independiente

En la Figura 3.30 se puede observar como la Ormerod 2 imprime por capas los modelos diseñados anteriormente para los soportes delantero y trasero.

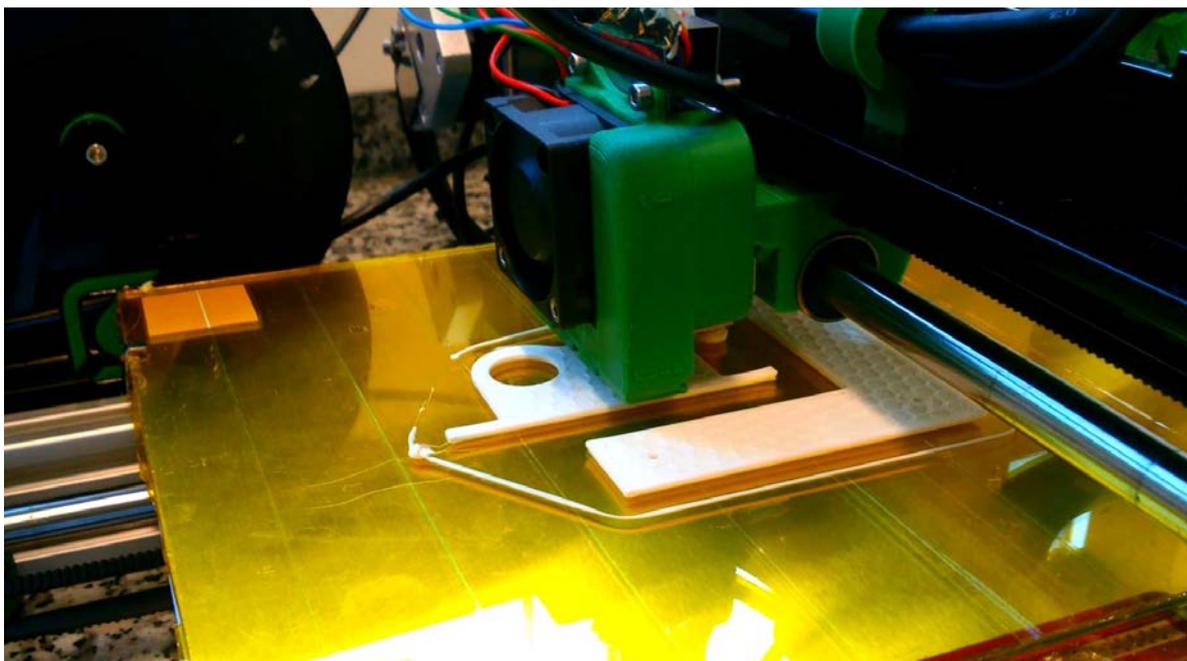


Figura 3.30 Impresión de soportes para sensores ultrasonidos en impresora 3D.

En una impresora 3D casera, como en este caso se utiliza, el material más comúnmente empleado es el plástico derretido o semilíquido, depositándose en el cabezal de impresión para el comienzo de la creación del modelo. Dicha impresión puede realizarse en diferentes densidades consiguiéndose un acabado de mayor o menor calidad, y por lo tanto ahorrando material plástico en este último caso. Esta situación es visible en la Figura 3.31 donde se muestra la “cama” de la batería impresa en 3D con ambas calidades, siendo la pieza de la izquierda la más frágil y económica.

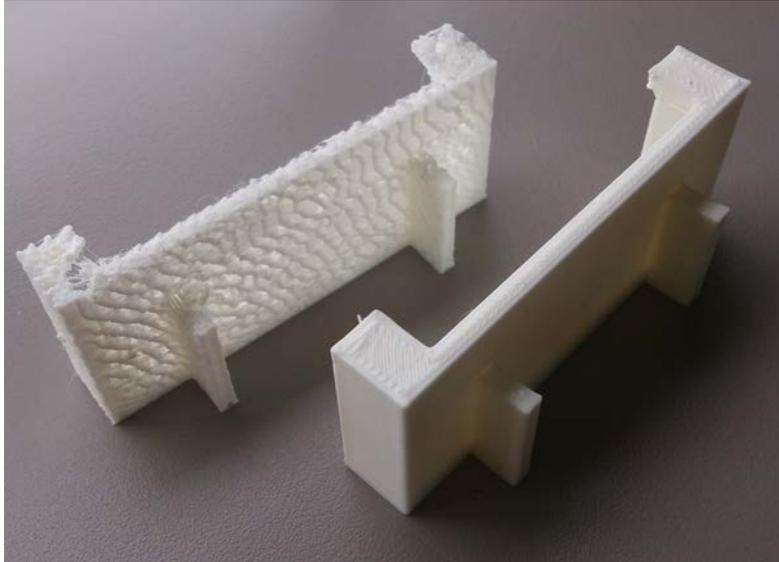


Figura 3.31 Cama de batería en diferente calidad de impresión 3D.

Una vez que el proceso de impresión se ha completado, los objetos que se han producido pueden tener pequeñas manchas o rayas, pudiendo ser pulidos o pintados como finalización, por tanto, las piezas impresas se han finalizado con el uso de una pulidora rotativa, con herramientas para lijar los bordes con malos acabados así como ajustar al radio correcto de los taladros, como se puede observar en la Figura 3.32.

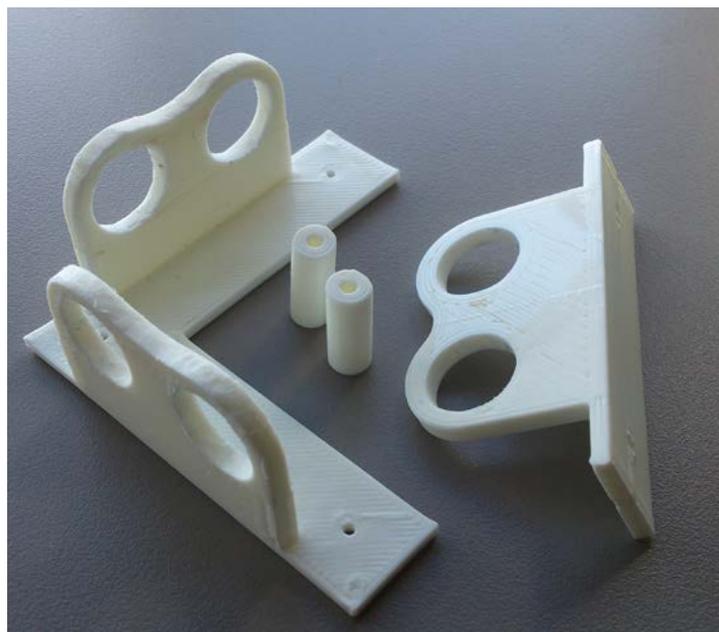


Figura 3.32 Impresiones 3D para los tres sensores de ultrasonidos.

Una vez ensamblados los soportes de los sensores mediante tornillos al chasis, así como el de la batería sin necesidad de ningún tipo de unión, se incluyen los sensores de ultrasonidos en sus ubicaciones y la batería en su lugar, tomando el vehículo el aspecto que se puede apreciar en la Figura 3.33, con todos los elementos correctamente integrados en el prototipo. El resultado final incluso provoca agrado visual debido a la forma de los sensores en similitud con posibles focos delanteros y traseros del vehículo.

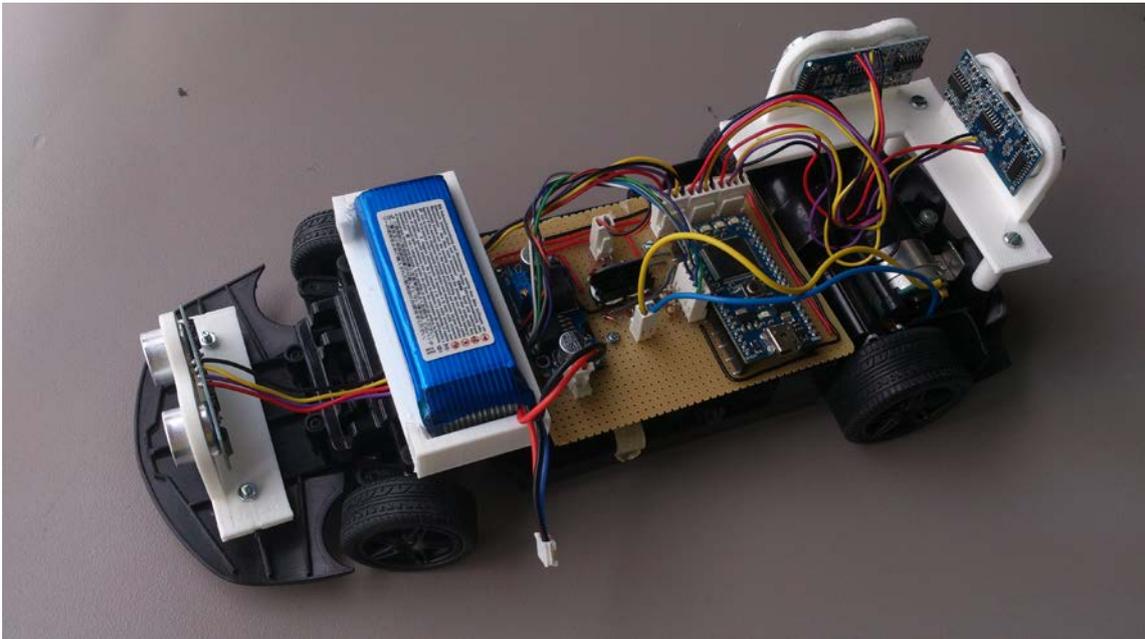


Figura 3.33 Impresiones 3D integrados en el prototipo.

4 DISEÑO Y REALIZACIÓN DE SISTEMA DE CONTROL

El diseño del sistema de control para el automatizado del aparcamiento del vehículo prototipo consta de dos partes claramente diferenciadas: el sistema principal de control y los drivers u objetos que controlan cada una de las funciones de los periféricos del microcontrolador.

El software ha sido desarrollado en C++, encontrándose en el programa principal el flujo de proceso o flujo de estados, por los que va sucediendo el vehículo, al pasar por cada una de las circunstancias que produce el escenario sobre el que actúa el sistema.

Los objetos que controlan cada periférico, son principalmente los drivers del motor de propulsión, del encoder y su motor para la dirección, así como los tres sensores de proximidad. Cada uno de los dispositivos se encuentra en un objeto distinto, siendo activados desde el programa principal.

En otra clase paralela, se encuentra también el objeto del controlador PID, el cual es llamado y realizado su declaración de variables en el programa principal, consiguiendo de esta forma el paralelismo del vehículo sobre el escenario.

4.1 Sistemas comunes

En este apartado aparecen los sistemas comunes al programa en todos los estados o etapas por las que va pasando el vehículo hasta completar su aparcamiento, es decir, estos objetos o sistemas comunes son los de los drivers utilizados tanto en el sistema de control del paralelismo como en el algoritmo de aparcamiento, y forman parte de la base para el correcto funcionamiento del sistema.

Al ser la base del sistema, estos drivers, tienen que tener una correcta implementación, ya que un error en los mismos, podría provocar desajustes en el vehículo, y por tanto, un comportamiento erróneo.

4.1.1 Implementación del Driver del Sensor de Ultrasonidos

El driver de los sensores de ultrasonidos es el central y más complejo de los tres objetos a tratar en este apartado, debido a su gran responsabilidad en la realización de una correcta medida para más tarde tratar en el bucle principal del programa. Así mismo, no debe molestar en el fluido funcionamiento del completo sistema, por estar constantemente ocupado el microcontrolador con el objeto presente.

Como base del driver, se toma la biblioteca disponible del sensor ultrasonido SRF05, la cual se encuentra en la página web de MBED [2], a la que posteriormente se le realizarán las modificaciones correspondientes para adaptarse al sistema a tratar en este trabajo.

La librería comienza configurando las interrupciones que se producirán cuando se active la señal de ECHO, así como cuando caiga el flanco de dicha señal. Para ello y como modificación de la librería, se realiza una tarea para controlar el RTOS Timer mediante el uso de punteros, los cuales activaran cada sensor para el comienzo de la medida. No obstante, se tendrá un Ticker que

realizará la función de Start cada cierto intervalo de tiempo, teniéndose en cuenta el tiempo necesario para la toma de datos del sensor.

La primera tarea que se establece es la de envío del tren de pulsos de TRIGGER durante un tiempo de 20 μ s, momento tras el cual se desactiva dicha señal. Siguiendo a dicha tarea, la siguiente resetea y comienza a realizar su función el contador, el cual se detiene al recibir el flanco de bajada del ECHO.

Posteriormente, para la realización del cálculo de la distancia, el HC-SR04 devuelve la señal de ECHO, que es proporcional a la distancia a la que viaja la onda de ultrasonidos, con lo que, es necesario operar para obtener la distancia, como indica la ecuación (4.1).

$$d = \frac{c \cdot t}{2} = \frac{340 \cdot t}{2} \quad (4.1)$$

Al realizar la medida del tiempo de los ultrasonidos en μ s y necesitar obtenerse la distancia en centímetros, se realiza la conversión de unidades obteniéndose en el divisor una constante de 58, tal como se refleja en la ecuación (4.2)

$$Dis\ tan\ cia(cm) = \frac{\mu s}{58} \quad (4.2)$$

Las modificaciones realizadas a continuación son referentes a la adaptación de la librería disponible para su uso en el proyecto que se plantea.

La primera de ellas es relativa al uso de un objeto Mutex, que es utilizado como testigo, el cual funciona rotándose entre los tres sensores. La sistemática que se utiliza comienza con la obtención del Mutex o testigo por parte de uno de los tres sensores, quedando inutilizado para el resto de sensores, momento tras el cual realiza la medición devolviendo la medida de distancia y dejando libre de nuevo el Mutex para el siguiente sensor. Dicho nuevo sensor que tome el Mutex realizará el mismo ciclo hasta dejarlo libre, siendo esta la dinámica de funcionamiento.

Otra modificación es la de la creación de funciones de Start y Stop para el caso en que se quieran parar los sensores de ultrasonidos, como en la situación de encontrarse en entornos cerrados. Un ejemplo de esta situación es cuando se encuentra el vehículo en hueco donde los ecos de unos sensores pueden afectar a la medida veraz de ultrasonidos que está tomando los datos en ese momento.

Por último, se realiza el cálculo de la mediana para 3 tomas de distancias, creándose de esta forma lo que se conoce en robótica como filtro a nivel de datos [1]. Este tipo de filtro de datos de los sensores se realiza a bajo nivel por tratar sobre los datos capturados directamente, siendo solo posible bajo datos similares, es decir, procedentes de sensores equivalentes o del mismo sensor realizado en distintos instantes de tiempo.

El objetivo básico buscado es la eliminación o reducción de ruidos en el dato a tratar, siendo una manera sencilla de filtro, aunque en contraprestación se lleve un aumento de carga computacional del sistema.

Una mediana estadística se define como el número central de un grupo de números ordenados por tamaño, donde si la cantidad de términos es par, la mediana es el promedio de los dos números centrales, y si es impar es el valor central del conjunto de los números.

En este caso, se realiza una mediana de 3 datos mediante con la ayuda de un vector donde se van almacenando cada una de las medidas hasta completar el mismo. Una vez completado el vector, se ordenan los elementos de menor a mayor valor, eligiéndose posteriormente el elemento central, el cual tendrá mayor probabilidad de no sobresalir respecto a la toma de una muy mayor o muy menor medida errónea. De esta manera se evita la trata con valores erróneos no coincidentes o muy alejados del valor real de la distancia a la que se encuentra el vehículo dentro del escenario.

4.1.2 Implementación del Driver de Dirección

Este objeto es utilizado para controlar la dirección del vehículo, produciendo el giro mediante un motor DC o corriente continua y detectando el ángulo proporcionado por el encoder descrito en apartados anteriores.

Para comenzar, se define el bus por el que circulará la información de la posición del encoder de la dirección, así como, la salida hacia el motor que produce el movimiento de la misma. La tabla de la codificación del encoder es utilizada para asignar un ángulo a cada una de las 15 posiciones disponibles resumidas en la siguiente Tabla 4.1.

Tabla 4.1 Código Gray del encoder en el objeto de dirección

Posición	Código binario	Hexadecimal	Grados	Situación
1	1000	0x8	29	Derecha
2	1001	0x9	24,86	
3	1011	0xB	20,72	
4	1010	0xA	16,58	
5	1110	0xE	12,44	
6	1100	0xC	8,3	
7	1101	0xD	4,16	
8	1111	0xF	0	Centro
9	0111	0x7	-4,16	
10	0101	0x5	-8,3	
11	0100	0x4	-12,44	
12	0110	0x6	-16,58	
13	0010	0x2	-20,72	
14	0011	0x3	-24,86	
15	0001	0x1	-29	Izquierda

Una vez definida la tabla sobre la que se basa el movimiento de la dirección y sus límites, se establecen los límites de movimiento mínimo del motor de DC para evitar que se produzcan tanto movimientos demasiado rápidos, los cuales llevan a oscilaciones, como de poca fuerza del motor provocando la incapacidad de giro de los engranajes del sistema de dirección.

El control utilizado sobre la dirección para llevar de un ángulo a otro es una operación lineal, por la que si la posición a la que se quiere llevar la dirección es cercana a la posición actual del encoder, se requiere poca velocidad del motor, utilizándose un PWM bajo, evitando así sobre oscilaciones sobrepasando la posición requerida, y por el contrario, si la posición a la que se quiere llevar la dirección es lejana a la actual del encoder, se suministra mayor velocidad llegando a establecerse el ángulo requerido en menor tiempo.

Por otro lado, el objeto recibe la información del ángulo que se requiere en cualquier instante de tiempo mediante una tarea programada o Ticker, el cual es comprobado cada 20 ms., el cual devuelve el ángulo establecido al motor con realimentación mediante el sensor encoder.

Se ha optado por este tipo de control en lugar de un PID por simplicidad tanto de implementación, como de control y ajuste del mismo, ya que este sistema requiere poca complejidad al ser sencillo por constar de pocas variables que tener en cuenta, practicándose de esta forma la realización de un control lineal manual para este pequeño sistema.

4.1.3 Implementación del Driver del Motor de Propulsión

Este último driver es utilizado para el motor de propulsión. Se ha tomado como base el driver controlador para Puente en H, estableciéndose en él las normas a aplicar sobre el PWM de las salidas del MBED correspondientes que actúan sobre el Puente en H, y a su vez sobre el Motor trasero del vehículo.

El objeto se define mediante dos pines, los cuales se pondrán en activo o inactivo con el PWM y la frecuencia del mismo con el que se pretenda controlar el Puente en H, así como, el sentido de giro que se quiera establecer sobre el motor de propulsión para el avance o retroceso del vehículo.

Por ello, se establece el mismo periodo para ambos pines de salida del MBED, y un reseteo de los mismos a nivel bajo, provocando que el motor pase a reposo, por si guardan valores anteriores, tal como se muestra en la Figura 4.1, donde aparece el código principal del driver del motor de propulsión donde se comentan las partes más relevantes.

```

#include "Motor2.h"           // Librerías
#include "mbed.h"

Motor2::Motor2(PinName pwm0, PinName pwm1):
    _pwm0(pwm0), _pwm1(pwm1) {
    // Condiciones iniciales para los valores de PWM
    _pwm0.period(0.01);
    _pwm1.period(0.01);
    _pwm0 = 0;
    _pwm1 = 0;
    minsp=0.15;              // Valor minimo muy lento para cuando se quiera precision de movimiento.
}

void Motor2::minspeed(float s)
{
    minsp=s;
}

void Motor2::speed(float speed) { // Función para establecer valor del PWM de velocidad
    if (speed == 0) {
        _pwm1=0;
        _pwm0=0;
        return;
    }

    if (abs(speed) < minsp) {
        speed = minsp*((speed >0) ? 1 : -1);
    }
    if (speed>0) {           // Avance o retroceso segun el signo
        _pwm1=abs(speed);
        _pwm0=0;
    } else {
        _pwm0=abs(speed);
        _pwm1=0;
    }
}

```

Figura 4.1 Código del driver para motor de propulsión.

Según la Tabla 4.2 siguiente se observan los diferentes estados en los que se puede encontrar el motor de propulsión del vehículo, según la activación de pines.

Tabla 4.2 Tabla de verdad de estados del motor según activación de pines

Pin 0	Pin 1	Estado
0	0	Reposo
0	1	Avance
1	0	Retroceso
1	1	Reposo

La principal restricción que se establece en este objeto, es la creación del límite inferior, consiguiéndose que no se llegue al PWM que provoca par mínimo, y por tanto, que no se produzca giro del motor por falta de fuerza, y en el caso de que se quiera colocar un valor inferior al que se ha comentado, el propio objeto reescribe el mismo a dicho límite.

Desde el programa principal se le da este valor de PWM al objeto, donde se procesará, comprobando su signo para asignar a cada salida el valor que le corresponda consiguiéndose de esta forma el giro del motor en el sentido que se pretende.

4.2 Descripción del sistema. Máquina de estados

El programa principal es la base sobre la que se sustenta todo el sistema, siendo así el corazón del mismo. Está formado por una serie de estados sobre los que se va encontrando el vehículo a lo largo de su trayectoria por el escenario.

Para darse el paso de un estado a otro se deben cumplir una serie de condiciones únicas, volviendo al mismo estado en caso de no cumplirse. Este tipo de sistema se conoce con el nombre de máquina de estados finitos secuenciales o autómatas de Moore [7], donde las salidas se obtienen directamente de las variables de estado internos, o en cualquier caso dependen únicamente de estas.

El sistema de control se divide principalmente en dos partes del proceso, siendo la primera el sistema de control para el paralelismo con la pared para el posterior y segunda parte, la maniobra del aparcamiento del vehículo.

El diagrama de flujo del programa principal, representado mediante un gráfico de transición de estados, es una forma de planteamiento del autómatas de Moore. Este termina con la detección del espacio donde realizar el aparcamiento, tal como se representa en la Figura 4.2.

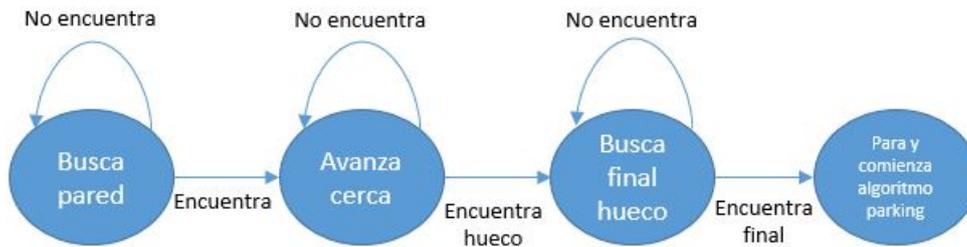


Figura 4.2 Diagrama de flujo principal.

Por otro lado, una vez encontrado el hueco donde realizar el aparcamiento, se pasa al algoritmo de parking, el cual se puede observar en la Figura 4.3.

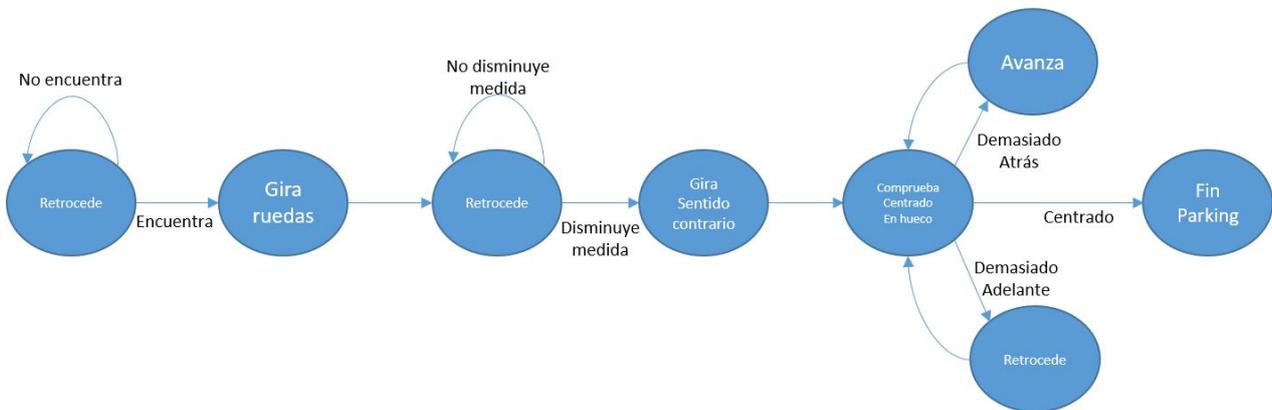


Figura 4.3 Diagrama de flujo maniobra de aparcamiento.

Ambos procesos se realizan de forma continuada con tan solo una pausa en su ejecución para la diferenciación entre las dos fases descritas anteriormente, donde se dirige el vehículo y se implementa las instrucciones para el algoritmo de movimiento del mismo, aunque se encuentra todo el sistema sobre la misma máquina de estados.

Dicha máquina de estados completa se implementa en el código mediante una función *switch* en la cual se va realizando el cambio de estado según se vayan cumpliendo unas u otras condiciones definidas en funciones y captadas del escenario por el vehículo. La función *switch* con todos los estados por los que va evolucionando el vehículo junto a la descripción de la acción realizada en cada uno de ellos se muestra a continuación en la Figura 4.4.

```

        switch(estado) {
// ESTADO 0 // Busca la pared del escenario
    case b_pared:
        estadocero();
        break;
// ESTADO 1 // Busca el paralelismo con la pared
    case p_pared :
        estadouno();
        break;
// ESTADO 2 // Encuentra el hueco y continua paralelo al mismo
    case p_pared_long:
        estadodos();
        break;
// ESTADO 3 // Detecta el final del hueco de aparcamiento
    case b_fin_hueco:
        estadotres();
        break;
// ESTADO 4 // Para al detectar el hueco
    case para_hueco:
        estadocuatro();
        break;
// ESTADO 5 // Retrocede para iniciar maniobra de aparcamiento
    case retro_p_hueco:
        estadocinco();
        break;
// ESTADO 6 // Gira para meter la parte trasera del vehiculo en el hueco
    case gira_park:
        estadoseis();
        break;
// ESTADO 7 // Retrocede recto en diagonal en el hueco
    case retro_park:
        estadosiete();
        break;
// ESTADO 8 // Gira hacia el otro lado para completar el aparcamiento
    case gira_otro_park:
        estadoocho();
        break;
// ESTADO 9 // Centra el vehiculo dentro del hueco
    case centra_park:
        estadonueve();
        break;
// ESTADO 10 // Rutina de fin de parking con parpadeo de LED's
    case fin_park:
        estadodiez();
        break;

```

Figura 4.4 Código de la máquina de estados junto con descripción de estados.

A continuación, en los dos siguientes apartados se explica el paso por los distintos estados y las condiciones que se han de cumplir para conseguir la fluida maniobra del vehículo en el espacio de aparcamiento.

4.3 Sistema de Control para el Paralelismo

En este sistema se establece el control necesario a realizar para obtener la búsqueda y posterior paralelismo sobre el escenario en el que se encuentra el vehículo, siendo el control PID el más adecuado para llevar a cabo este cometido.

El controlador a utilizar se trata de un Proporcional-Integral-Derivativo [6], es decir, un lazo de control por realimentación, controlando la variable de proceso y chequeando que cumpla el set

point que se le ha asignado a la misma, ajustando la variable de proceso mediante un cambio en la salida del controlador, la cual llevara a la variable de proceso al set point elegido.

El sistema arranca con la búsqueda de la pared, momento en el que tras acercarse a la pared en el ESTADO 0, el sistema de control es aplicado en el ESTADO 1, donde, se definen la variable a controlar, siendo esta la distancia lateral del vehículo, y el *set point* de la distancia a la que se quiere llevar el vehículo, aplicando el resultado de las operaciones a la dirección.

Para asegurar la correcta colocación paralela del vehículo sobre el escenario en el que se encuentra, se establece una variable que será activada al encontrarse sobre la distancia fijada con el *set point*, variable sin la cual no se encontrará el hueco.

Para el cálculo de este control por PID, se utiliza la herramienta SIMULINK® de MATLAB®, con la que mediante un modelado del sistema analógico del sistema sobre el que se quiere actuar, se calculan los valores para cada parte del control.

Es necesario basar el modelo en la distancia lateral de *set point* a la que el vehículo se encontrará de la pared. Para ello y como parte esencial, entra en juego tanto el ángulo de la dirección, como la velocidad a la que se desplaza el coche, calculándose de esta forma la velocidad con la que se acerca o aleja de la pared, pudiéndose obtener posteriormente la distancia buscada. La ecuación (4.3) establece la relación de los parámetros indicados anteriormente.

$$V_y = \frac{dy}{dt} = v \cdot \text{sen}(\alpha) \quad (4.3)$$

Una vez que se tiene la expresión se realiza el modelo en la herramienta SIMULINK® tal como se representa en la Figura 4.5, donde se fija la velocidad constante del vehículo para un PWM concreto al que funcionará en el momento del control del paralelismo sobre el escenario.

Para el conocimiento de dicha velocidad constante se realiza la media sobre varias medidas tomadas según el tiempo que tarda el vehículo en realizar un espacio concreto. Para ello será necesario que el motor de propulsión siempre provoque el movimiento con la misma velocidad sobre el coche, esperándose en el caso de que se encuentre en un estado de transición hasta dicha constante.

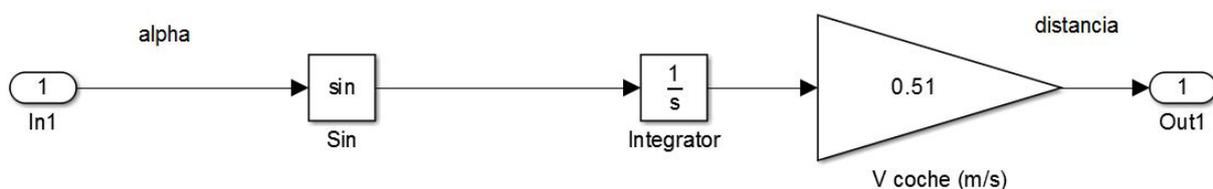


Figura 4.5 Modelo vehículo en SIMULINK® según ecuación (4.4).

Una vez que se tiene la variable de control para el acercamiento a la pared en la Figura 4.5, se incorpora al resto del sistema representado en la Figura 4.6, donde se establece el lazo de realimentación negativo característico de tipo de control utilizado en este caso, así como el propio controlador PID.

Debido a que el modelo que se ha creado con la ecuación (4.4) es ideal, se realizan una serie de modificaciones sobre el lazo para obtener resultados más reales. Para ello, se incorpora un bloque de *Retraso* o *Delay*, debido a que el sistema no tiene una respuesta completamente instantánea, por tratarse de un sistema mecánico. Posteriormente, se incorpora un bloque de *Saturación* mediante el que se establece el ángulo máximo de giro en ambos sentidos, acercándose de esta forma a la respuesta dada por la dirección del vehículo.

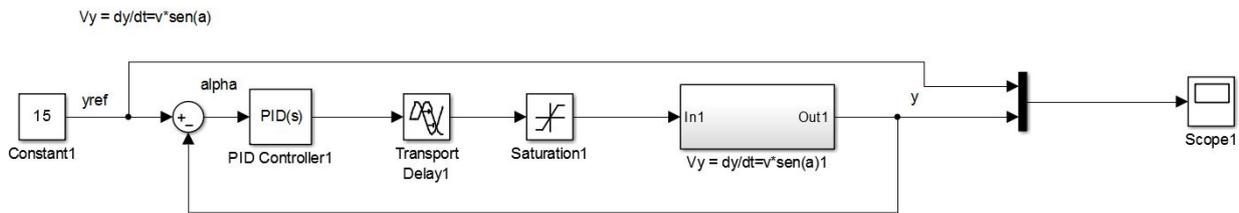


Figura 4.6 Modelo controlador en SIMULINK®.

Por último, es necesario colocar a la entrada del sistema la constante que referencia el *set point*, mostrándose tanto esta referencia, como la salida del sistema mediante un visor disponible en la librería de componentes del SIMULINK®.

El controlador que es utilizado en SIMULINK®, con forma en Paralelo, se define por la ecuación (4.4), expresada a continuación.

$$CO_{SIMULINK} = P + I \frac{1}{s} + D \frac{N}{1 + N \frac{1}{s}} \tag{4.4}$$

Los valores obtenidos de PID paralelo en SIMULINK® son los representados en la siguiente Tabla 4.3.

Tabla 4.3 Valores de PID en obtenidos en SIMULINK®

	Valor
P	1,4329
I	0,0095
D	0,5850

Estos valores se deben convertir a la función matemática del control PID usada por el objeto en MBED. La forma se expresa según la ecuación (4.5) siguiente.

$$CO_{MBED} = K_c \left(e(t) + \frac{1}{T_i} \int e(t) dt + T_d \frac{dPV}{dt} \right) \quad (4.5)$$

Tras la transformación de una forma matemática a la otra, se obtienen los valores representados en la Tabla 4.4.

Tabla 4.4 Valores de PID para su utilización en MBED

	Valor
P	1,4329
I	76,920
D	0,4082

4.4 Implementación del Algoritmo de Aparcamiento

Una vez que se detecta el hueco donde realizar la maniobra de aparcamiento, el PID queda desactivado y se continúa en el ESTADO 2, manteniendo la dirección en ángulo 0 hasta detectarse el fin del hueco con el correspondiente paso al ESTADO 3.

En este momento se continúa avanzando cierto tiempo en el ESTADO 4, para situarse correctamente, necesitando de esta forma menor espacio para realizar la maniobra dentro del hueco. Antes de continuar con cualquier otro movimiento, se realiza una breve pausa de unos segundos para visualizar la separación del algoritmo de aparcamiento respecto de la separación del paralelismo con la pared del escenario. Una vez correctamente situado para la realización del algoritmo de aparcamiento, comienza la maniobra con el paso al siguiente estado.

En el ESTADO 5, el motor de propulsión recibe la orden de comienzo de su movimiento de retroceso con una trayectoria rectilínea hasta detectarse de nuevo el final del hueco, lo que provoca el paso al ESTADO 6 donde se le asigna a la dirección el máximo ángulo hacia su derecha para enfocar la parte trasera del vehículo dentro del aparcamiento. En este estado el motor de propulsión continúa su movimiento de retroceso.

El vehículo permanece girando hasta detectar cierta distancia trasera, la cual ha sido determinada para una correcta trayectoria del mismo en la maniobra, basada en la trayectoria que realiza el coche en la posición de ángulo máximo de dirección, las dimensiones de longitud y anchura del vehículo.

En este momento, se accede al ESTADO 7, en el cual el prototipo endereza la dirección y se desplaza retrocediendo hacia dentro del hueco en diagonal, hasta detectar la distancia trasera mínima que indica que se está dentro del hueco.

En este momento el vehículo realiza el giro hacia la izquierda la dirección, para enderezar el vehículo, consiguiendo colocarse paralelo a la línea de la pared y por lo tanto del fondo del hueco.

Esto se produce en el ESTADO 8, tomándose como referencia la pared trasera para el final del movimiento.

Cuando la parte trasera del vehículo llega a escasos centímetros de la pared de referencia en el hueco, se endereza la dirección del vehículo de nuevo, en el ESTADO 9, estado en el cual se centrará el vehículo avanzando en el hueco hasta igualar la distancia de los sensores delantero y trasero.

En este momento, se procede al paso del siguiente, el ESTADO 10, donde se realiza una rutina de parpadeo de todas las luces del MBED indicando que se ha realizado el parking con éxito.

Esta trayectoria de aparcamiento se basa en la distancia medida desde la parte trasera del vehículo al muro trasero del hueco, sobre las dimensiones del prototipo y las restricciones del mismo dadas por el giro máximo que permite el mayor ángulo posible de la dirección.

5 RESULTADOS

En este apartado se muestran los resultados obtenidos después del estudio y posterior realización del trabajo, consiguiéndose resultados favorables, ya que el vehículo, tras ser ajustados correctamente sus parámetros, ha realizado un correcto aparcamiento en el hueco propuesto con el escenario.

La primera prueba de ello se obtiene tras la simulación de la trayectoria ideal que debería seguir el vehículo en la maniobra de aparcamiento una vez que se encuentra en paralelo con la pared. Dicha simulación se ha realizado mediante codificación en la herramienta MATLAB®, con la que se ha obtenido la gráfica de la Figura 5.1, donde se puede visualizar la correcta trayectoria a realizar, pudiéndose comparar posteriormente con la real que realiza el vehículo.

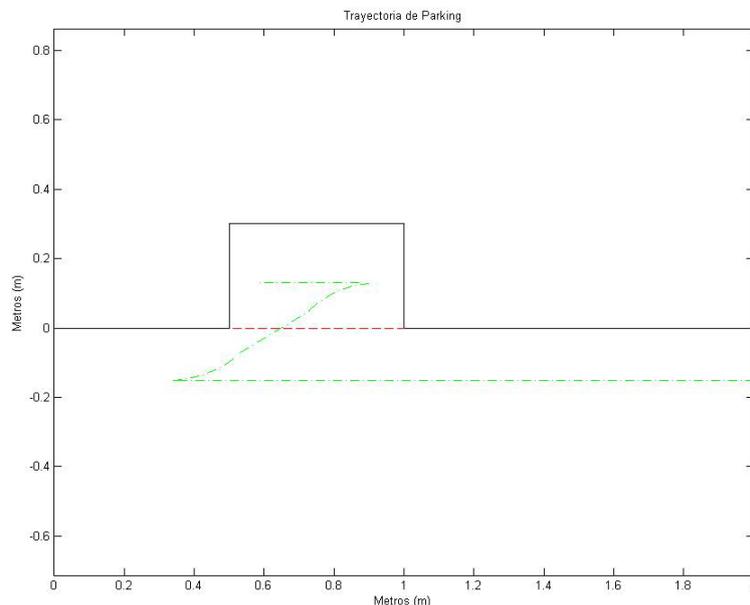


Figura 5.1 Trayectoria ideal de maniobra de parking en MATLAB®.

En esta trayectoria matemática se puede observar un paralelismo ideal, en el que el vehículo traza una trayectoria perfectamente rectilínea, seguida de una maniobra limpia dentro del hueco, consiguiéndose un aparcamiento centrado.

Al comparar con la maniobra real de aparcamiento obtenida mediante fotogramas mientras el vehículo la realizaba, se asemeja en gran medida a la obtenida en primer lugar por la trayectoria ideal de la maniobra en la Figura 5.1 obtenida por simulación informática.

En la Figura 5.2 se muestra el modelo una vez ha conseguido encontrar la pared y mantenerse posteriormente paralelo a ella con alguna ligera oscilación propia de la pequeña variación en las medidas del sensor lateral. En este estado el vehículo está buscando el principio del hueco en el que aparcarse posteriormente.

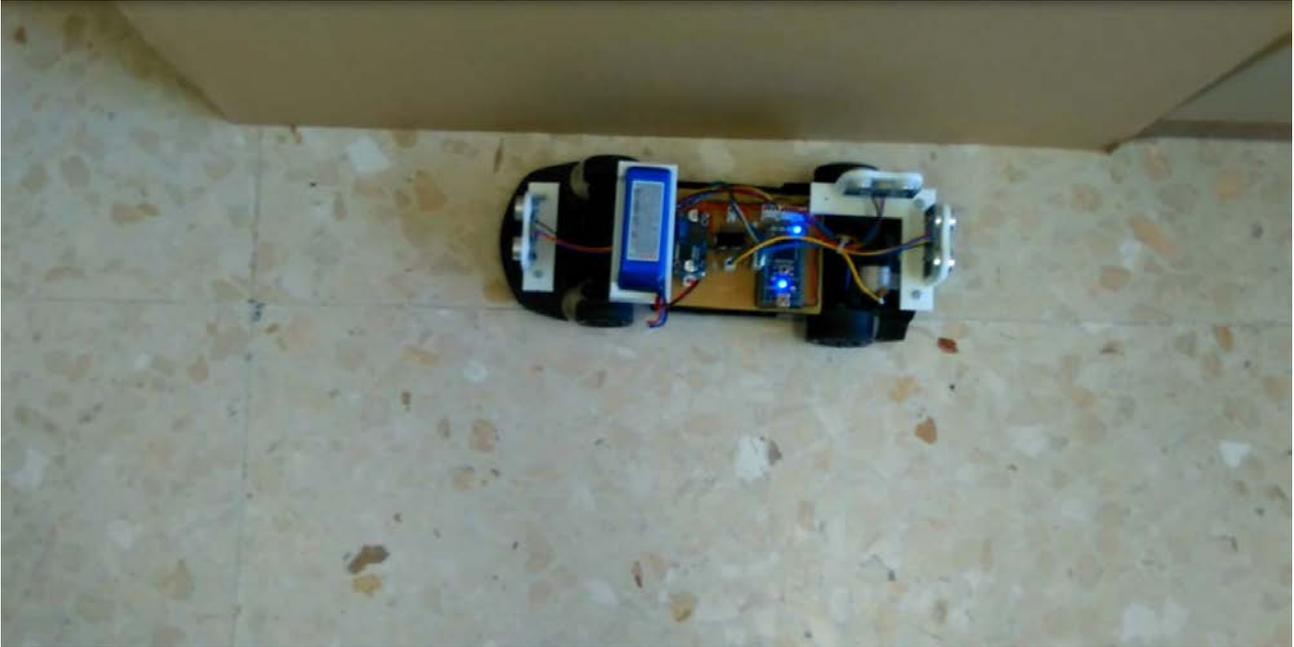


Figura 5.2 Trayectoria real del vehículo en paralelo a la pared.

Una vez que el vehículo ha encontrado el principio del hueco, continua paralelo a la pared, siguiendo la trayectoria lineal que había conseguido anteriormente al estar paralelo al escenario, situación que se muestra en la Figura 5.3, se dispone a buscar el final del mismo para posteriormente iniciar la maniobra de aparcamiento.

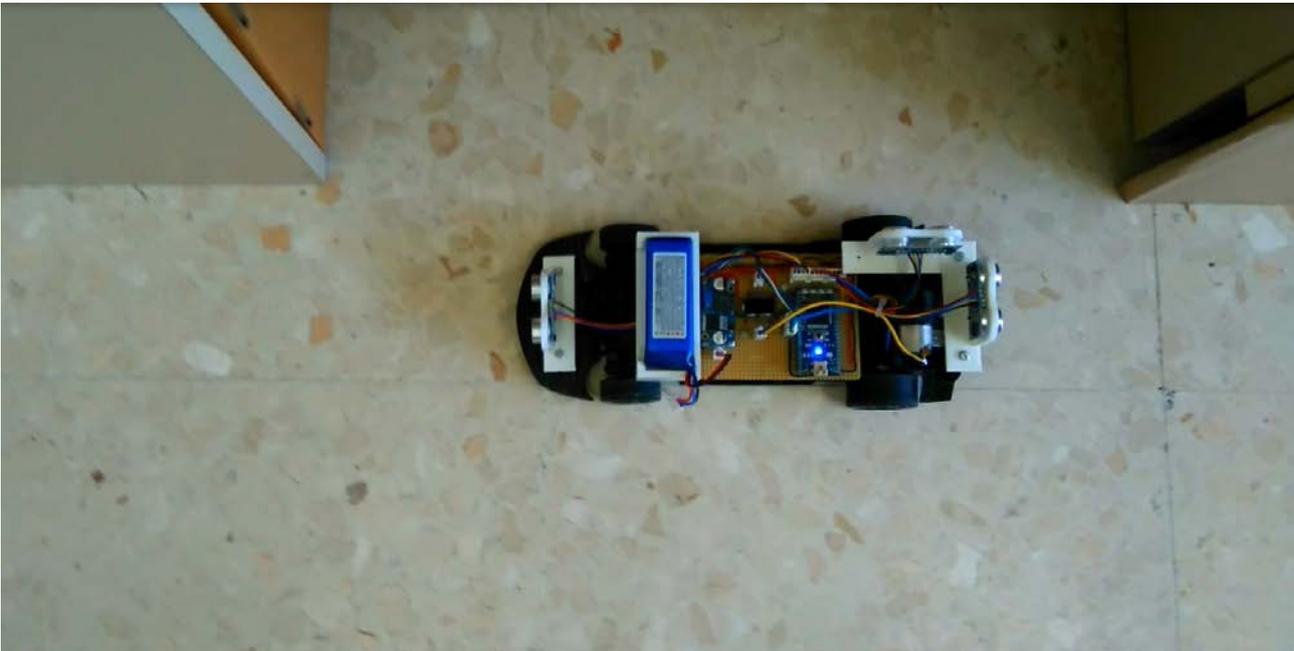


Figura 5.3 Trayectoria real del vehículo encontrando el hueco.

En la Figura 5.4, el vehículo ya ha encontrado el hueco completo, avanzando el suficiente espacio para encontrarse en una posición en la que la parte trasera del vehículo haya sobrepasado el final del hueco, encontrándose en un lugar más apropiado para realizar la maniobra aprovechando en mayor medida el espacio disponible en el aparcamiento. En este momento, el vehículo comienza la maniobra, girando por completo la dirección y retrocediendo.

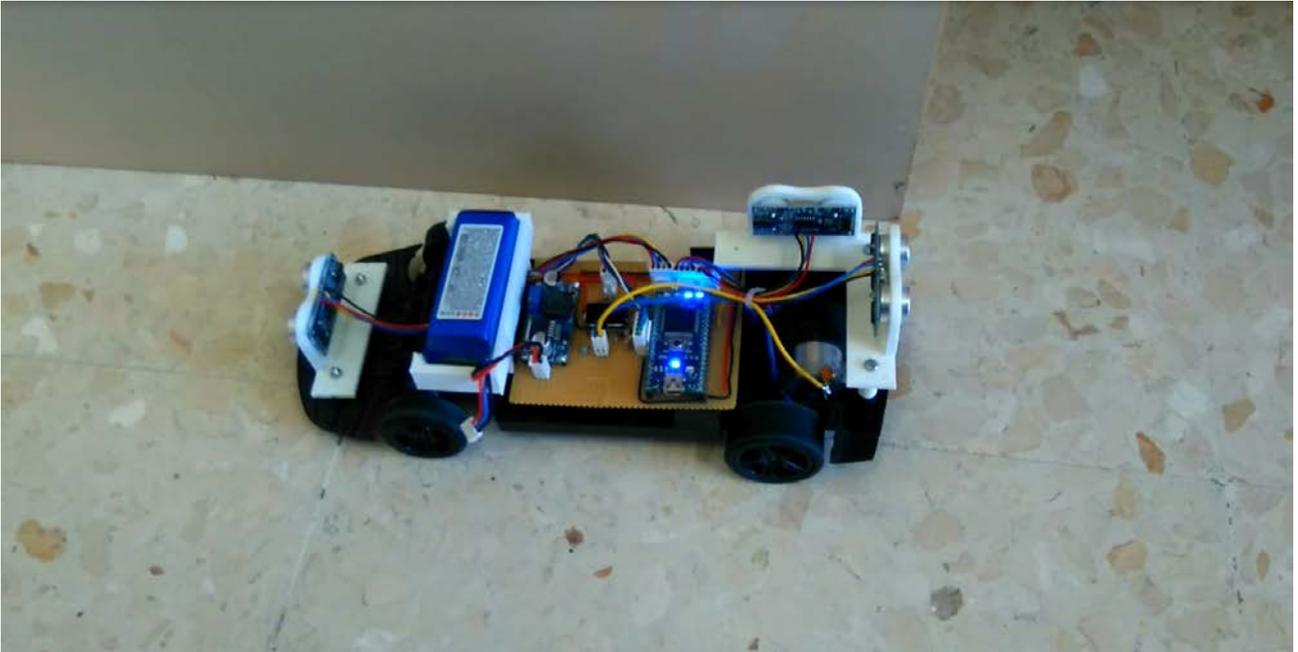


Figura 5.4 Trayectoria real del vehículo iniciando maniobra.



Figura 5.5 Trayectoria real del vehículo entrando en el hueco.

Continuando a la escena anterior, en la Figura 5.5, el vehículo ha ido introduciendo su parte trasera en el espacio, encontrándose entrando recto en diagonal en el hueco hasta tener el eje trasero en el interior del mismo.

En la Figura 5.6, el vehículo se encuentra completando el aparcamiento, girando la dirección para terminar en el interior del hueco paralelo a la pared del fondo del mismo, momento en el que el modelo deja el eje de la dirección paralelo al eje trasero.



Figura 5.6 Trayectoria real del vehículo terminando la maniobra.

Finalmente, en la Figura 5.7, el eje de la dirección del vehículo se coloca paralelo al eje de las ruedas traseras, centrándose el modelo en el hueco, avanzando para mantener la misma distancia en los sensores delantero y trasero, finalizando la operación de aparcamiento automático, lo que se muestra mediante el parpadeo de todas las luces azules del microcontrolador MBED. Una vez que se llega al fin del aparcamiento, los sensores de ultrasonidos dejan de realizar medidas y todos los motores quedan parados sin realizar ninguna acción más.



Figura 5.7 Trayectoria real del vehículo una vez finalizada la maniobra.

Para completar la comparación entre la trayectoria real realizada por el vehículo del trabajo y la simulada por ordenador, se realiza la toma de datos del sensor de ultrasonidos lateral del coche con el fin de obtener de esta forma una gráfica con lo que capta el vehículo a lo largo de su recorrido por el escenario.

Dicha toma de medidas se representa en la gráfica de la Figura 5.8, sobre la que se puede visualizar como el vehículo reconoce la pared a la que debe permanecer paralelo, así como el hueco sobre el que posteriormente realiza la maniobra.

La gráfica se divide en 3 partes claramente diferenciadas explicadas en la descripción del software:

- Búsqueda de la pared y posterior paralelismo con el escenario. Esta franja de la gráfica de la Figura 5.8 se encuentra entre el comienzo de la maniobra hasta el segundo 14, pudiéndose apreciar como el lateral del vehículo comienza a acercarse a la pared hasta mantenerse paralelo a la misma.
- Búsqueda del hueco y paralelo al mismo. Comenzando esta zona aproximadamente en el segundo 14 de la Figura 5.8, momento en el cual encuentra el hueco, apreciándose el escalón que produce el aumento de la distancia hasta volver a encontrar la pared con su consiguiente disminución de distancia sobre el segundo 18.
- Realización de la maniobra de aparcamiento. Esta parte de la gráfica de la Figura 5.8 se distingue una vez encontrado el hueco, momento en el que el vehículo realiza una parada una vez terminado el hueco alrededor del segundo 19. Posteriormente se realiza el retroceso para comenzar la maniobra de aparcamiento alrededor del segundo 20 donde se puede apreciar de nuevo un aumento de la distancia al introducir la parte trasera del vehículo de nuevo en el hueco. Por último, desde este momento hasta el

final de la captura de datos, se puede visualizar una ligera curva en forma de “S” correspondiente a la medida del fondo del hueco al ir realizando los diferentes giros de dirección para completar la maniobra.

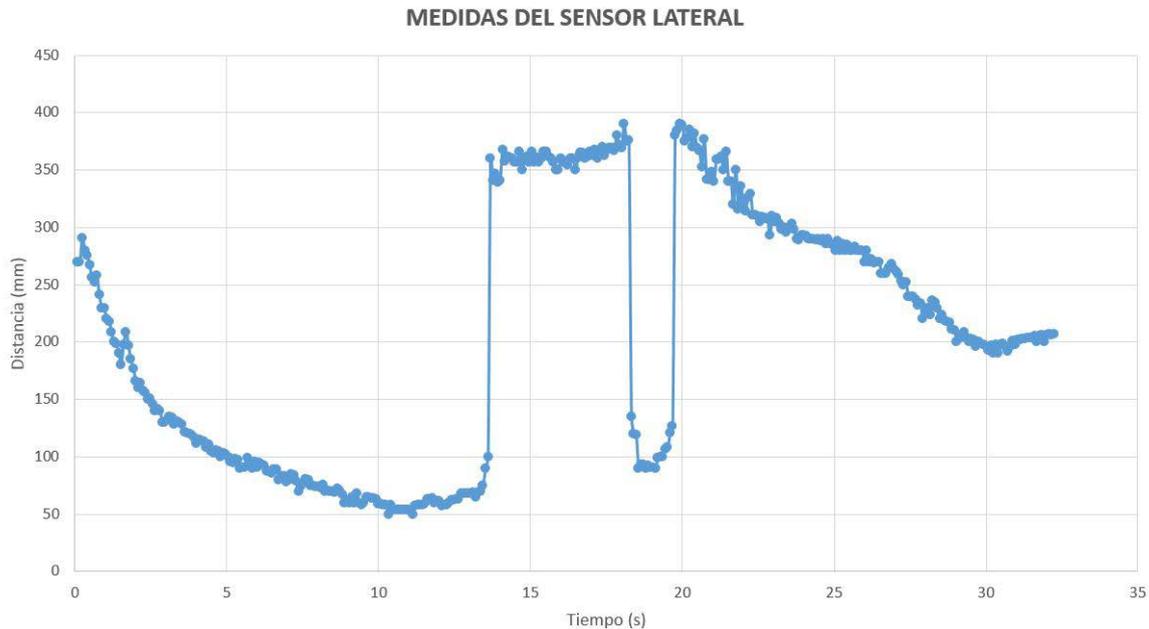


Figura 5.8 Captura de datos del sensor lateral a lo largo del recorrido.

Una vez que se comprueba que el vehículo realiza correctamente todas las tareas, se procede a realizar una estadística basada en la repetición de la tarea un determinado número de veces comprobando así la probabilidad de éxito que se obtiene y por consiguiente la estabilidad del sistema. Tras colocar el vehículo en la misma situación de partida en el comienzo del escenario sobre el que se realiza el paralelismo y posterior maniobra en el hueco detectado, se obtiene una probabilidad bastante elevada y satisfactoria de un 86,66 % de conseguir finalizar el aparcamiento con una correcta posición dentro del hueco.

Esta estadística está basada sobre 30 pruebas en las que se parte de la misma posición obteniéndose una trayectoria muy similar en todas las repeticiones, con lo que se obtiene un resultado positivo tras la realización de las mismas.

6 CONCLUSIONES

En este último capítulo se muestran las conclusiones obtenidas a lo largo de la realización del proyecto sobre el que se ha trabajado. De ellas se deducen posibles cambios futuros para el perfeccionamiento del sistema actual.

- Se ha realizado un sistema de aparcamiento autónomo, en un vehículo a escala, con un porcentaje de éxito de un 86,66 % por lo que se concluye que el resultado del mismo es satisfactorio.
- Se concluye que los sensores de ultrasonido HC-SR04 son válidos para este uso en proyectos pequeños que requieren precisión, siempre y cuando se lleve un procesamiento de los datos mediante software para eliminar posibles ruidos e interferencias que se produzcan en la lectura de los mismos.
- Se ha comprobado la facilidad de diseño e impresión en 3D con las herramientas disponibles actualmente, siendo de gran potencial, rapidez y utilidad el conocimiento y uso de este tipo de tecnología para la creación de pequeñas piezas necesarias en cualquier proyecto. Por otro lado, la resistencia de las piezas es suficientemente elevada para dicho uso, permitiendo incluso algún uso que requiera esfuerzos de torsión.
- Se ha observado que la alimentación de todo el sistema, formada por la batería de LiPo y el convertor DC-DC, es adecuada para el funcionamiento de motores en combinación con el microcontrolador, debido a que los motores demandan en sus arranques picos de intensidad, pudiendo producir micro cortes en la alimentación por la imposibilidad de suministro de la batería, provocando el reseteo del microcontrolador y por tanto el reinicio del sistema. Sin embargo, la batería elegida es capaz de suministrar 30 veces la propia capacidad de la misma, así como el convertor DC-DC, permite el paso de estos picos de intensidad sin aumentar su temperatura en exceso.
- Se ha comprobado que el microcontrolador MBED LPC 1768 utilizado tiene la suficiente potencia para realizar el trabajo sin desbordarse, con el consiguiente fallo, a pesar de poderse mejorar en el aspecto de la multitarea, pudiendo llevar a cabo de esta manera un mayor número de acciones sin llegar al colapso.

Este trabajo ha permitido concluir las posibles mejoras que se pueden realizar sobre el modelo para su mayor optimización son:

- Realización de una placa PCB de doble cara con el plano obtenido en Eagle, con el que se consigue un circuito más optimizado, con mayor compacidad, donde los componentes tienen la misma situación que en la placa actual pero con menor posibilidad de error por mala conexión de las pistas y de las soldaduras. Se tendría una mayor disipación de calor del componente que más sufre en éste circuito, tratándose del puente en H donde se tienen constantemente picos de intensidad producidos por los motores de propulsión y de dirección.

- Cambio de los sensores de ultrasonidos utilizados para el proyecto debido a las interferencias y baja precisión de estos, siendo necesario un mayor procesamiento de los datos para obtener medidas reales con las que trabajar. Este cambio se realizaría por otros sensores de mayor precisión como pueden ser sensores de infrarrojos o incluso de mayor calidad, sensores de distancia por láser.
- Modificación del encoder y del motor DC de la dirección por un servo electrónico, el cual tiene mayor precisión de giro por incluir una mayor gama de ángulos que puede tomar la dirección del vehículo, por lo tanto menor error en cada corrección de la misma. Por otro lado un servo electrónico, tan solo necesita una señal PWM con lo que el control del ángulo de la dirección en cada momento es más sencillo.
- Colocación de otro sensor lateral en el tren delantero para conseguir un mejor paralelismo, con lo que se podrá contrastar la información del tren delantero y trasero, dado que actualmente, contando tan solo con el ultrasonidos del tren trasero, se tiene cierto retraso en la lectura de la distancia real a la pared, por la distancia existente entre la dirección y el punto de medida de la distancia.
- Elección de un microcontrolador de mayor potencia para tener mayor soltura a la hora de realizar multitarea de diversas acciones en el mismo momento, debido a que el LPC 1768 utilizado queda en cierto modo restringido en esta característica.

7 BIBLIOGRAFÍA

- [1] F.Torres, J.Pomares, P. Gil, S.T. Puente, R. Aracil, “Robots y sistemas sensoriales”, Pearson Education, 2002.
- [2] Página oficial de MBED, < <https://www.mbed.com/en/>>, Consultada el día 20/01/2016
- [3] E. Alonso, “Ponte las Pilas”, *AUTOTEC MODELISMO RC*, Año XIV Nº177, p. 56 - p. 57.
- [4] Sanz Feito, J., “Máquinas eléctricas”, Prentice Hall, Madrid, 2002.
- [5] A. Barrientos, L.F. Peñin, C. Balaguer, R. Aracil, “Fundamentos de robotica”, McGraw Hill, 2007.
- [6] K. Ogata, “Modern Control Engineering”, Prentice Hall, Fifth Edition, 2010.
- [7] García Moreno, E., “Automatización de Procesos Industriales”, UPV, 1999

