



ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA (ICAI)
INGENIERO ELECTROMECAÁNICO
Especialidad Electrónica

Embedded System Controller for a Longboard

Autor: Valeriano Torres Ordóñez
Director: Sally Wood

Madrid
Julio 2016

Valeriano
Torres
Ordóñez

Embedded System Controller for a Longboard





Índice de la memoria

<i>Parte I Memoria</i>	3
<i>Capítulo 1 Introducción</i>	5
1.1 Estudio de los trabajos existentes / tecnologías existentes	5
1.2 Motivación del proyecto.....	6
1.3 Objetivos.....	7
1.4 Metodología / Solución desarrollada.....	8
1.5 Recursos / herramientas empleadas.....	10
<i>Capítulo 2 Solución desarrollada</i>	13
2.1 Introducción.....	13
2.2 Acondicionamiento de los componentes electrónicos	13
2.3 Máquina de estados	19
2.4 Diseños propios	25
2.5 Montaje.....	30
2.6 Aplicación para Smartphone.....	36
2.7 Comunicación Arduino - Aplicación.....	38
<i>Capítulo 3 Resultados/Experimentos</i>	41
<i>Capítulo 4 Conclusiones</i>	45
<i>Capítulo 5 Futuros desarrollos</i>	47
5.1 Propuesta componentes a futuro.....	48
<i>Parte II Código fuente</i>	49



<i>Capítulo 1</i>	<i>Código fuente: Arduino.....</i>	<i>51</i>
<i>Parte III</i>	<i>Datasheets</i>	<i>69</i>
<i>Capítulo 1</i>	<i>Datasheets principales.....</i>	<i>71</i>



Índice de ilustraciones

Ilustración 1: Esquema de la placa de circuito impreso diseñada para el proyecto.	14
Ilustración 2: Esquema del acondicionamiento del sensor delantero mediante divisor de tensión.....	15
Ilustración 3: Acondicionamiento del pin receptor del módulo Bluetooth mediante un divisor de tensión.....	18
Ilustración 4: Diagrama de transiciones de la lógica implementada en el microcontrolador.	20
Ilustración 5: Captura de pantalla de la aplicación SCU en el modo de regulación desde el Smartphone. En ella se puede apreciar la barra de desplazamiento.	21
Ilustración 6: Vistas del diseño en 3D de la cubierta del sensor delantero.	27
Ilustración 7: Resultado final de la cubierta del sensor delantero.	27
Ilustración 8: Vista superior e inferior de la placa de circuito impreso empleada, con los elementos electrónicos y pines debidamente soldados.	28
Ilustración 9: Placa de plástico acrílico.	29
Ilustración 10: Vista del sensor delantero una vez unido a la tabla. También se pueden apreciar el lijado y los agujeros mencionados en el paso anterior.	31
Ilustración 11: Vista superior de la cubierta del sensor delantero una vez instalada.	32
Ilustración 12: LEDs encendidos y ubicación del LED delantero.	33
Ilustración 13: Vista inferior de los componentes unidos a la tabla por medio de la placa acrílica.	35



Ilustración 14: Logo de la aplicación (SCU).....	36
Ilustración 15: Ejemplo de bloques de código de la aplicación desarrollada.....	37
Ilustración 16: Capturas de pantalla de la ventana principal, alerta por aplicar potencia máxima y lista de dispositivos Bluetooth pareados, respectivamente. ...	37
Ilustración 17: Diferentes tipos de voluntarios.....	42
Ilustración 18: Diferentes formas de colocar el pie sobre el sensor. (90°, 45° y 0°)	42
Ilustración 19: Gráficas obtenidas en los experimentos.....	43



Índice de tablas

Tabla 1: Bytes en la comunicación inalámbrica.....	39
Tabla 2: Tabla con los valores máximos obtenidos en los experimentos.	43
Tabla 3: Propuesta componentes desarrollo futuro.	48



UNIVERSIDAD PONTIFICIA COMILLAS
ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA (ICAI)
INGENIERO INDUSTRIAL



UNIVERSIDAD PONTIFICIA COMILLAS
ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA (ICAI)
INGENIERO INDUSTRIAL

PARTE I MEMORIA



UNIVERSIDAD PONTIFICIA COMILLAS
ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA (ICAI)
INGENIERO INDUSTRIAL



Capítulo 1 INTRODUCCIÓN

Este proyecto tiene como objetivo diseñar e implementar un sistema que regule la potencia aplicada al motor eléctrico de un longboard. Este sistema está basado en dos sensores de presión, uno para cada pie, que permiten al usuario acelerar al inclinarse hacia delante. Se pretende que este longboard pueda ser utilizado por cualquier tipo de usuario, independientemente de su peso, altura o estilo de patinaje; para lo que se ha desarrollado un modo de calibración.

El alcance de este proyecto cubre el desarrollo de una máquina de estados que regula el funcionamiento del dispositivo, la implementación de los sensores de forma ergonómica, el desarrollo de una aplicación para smartphones, que posibilita la selección de los diferentes modos de operación; el establecimiento de una comunicación bluetooth estable entre el teléfono y el microcontrolador situado en la tabla; un sistema lumínico de retroalimentación basado en leds y la demostración del correcto funcionamiento del proyecto, regulando la potencia aplicada a un pequeño motor de corriente continua.

1.1 ESTUDIO DE LOS TRABAJOS EXISTENTES / TECNOLOGÍAS EXISTENTES

En los últimos años se está produciendo un auge del uso de longboards, monopatines de grandes dimensiones que permiten a sus usuarios desplazarse en distancias cortas con una mayor estabilidad que la que puede aportar un skateboard clásico.

Paralelamente, también se está produciendo una proliferación de medios de transporte eléctricos para un solo usuario con complejos sistemas electrónicos, como los denominados “hoverboards”, “segways” o los propios longboards con la implementación de motores eléctricos.

Los longboards eléctricos presentes en el mercado cuentan con significativas prestaciones de velocidad punta y autonomía, alcanzando los 32 km/h y con una duración de la batería de unos 40 km en condiciones normales de uso. Una vez que el longboard eléctrico como producto se ha afianzado en el mercado, la innovación se está enfocando hacia el modo de regular la potencia que recibe el motor. Tradicionalmente, se ha regulado con dispositivos de control



remoto por radiofrecuencia o bluetooth. Sin embargo, estos métodos suelen ser poco ergonómicos y suponen una perturbación en el uso normal del longboard.

El objetivo de este proyecto es proporcionar un método alternativo, basado en el peso del usuario, para controlar el voltaje aplicado al motor, con un coste inferior a los competidores más directos presentes en el mercado, que comercializan este producto en torno a los \$1,300-\$1,500. A su vez, para integrar de la forma más natural posible el control del longboard, el usuario contará con una aplicación en su Smartphone. Desde esta aplicación podrá apagar el motor en cualquier momento, fijar la potencia aplicada al motor, elegir si desea regular el motor con su peso al inclinarse hacia delante o calibrar el sistema para sus características concretas.

1.2 MOTIVACIÓN DEL PROYECTO

Sin ninguna duda, cualquier aficionado al mundo del longboard se ve atraído por la posibilidad de llevar su experiencia de patinaje a otro nivel, añadiendo un motor eléctrico a su tabla. Esto se debe, no solo a la emoción que puede producir alcanzar velocidades mayores, sino también a la ayuda que supone en tramos más complicados, en los que a veces la fuerza de las piernas no es suficiente. Sin embargo, el usuario medio quiere estas ventajas sin renunciar a la experiencia original. Para ello los diseñadores de longboards eléctricos deben tender a hallar formas mínimamente intrusivas de controlar los motores. En un primer momento, la opción lógica fue el control remoto mediante mandos de radiofrecuencia o Bluetooth. La evolución lógica de esto derivó en las aplicaciones para Smartphones, las cuales, aunque menos integradas que el mando; se quedaban cortas. Por lo mencionado anteriormente, en este proyecto se ha decidido dar un paso más: ofrecer al usuario la posibilidad de controlar la tensión del motor sin necesidad de intermediarios, simplemente con su propio cuerpo. El usuario, no solo podrá controlar el motor inclinándose hacia delante o atrás, sino que también podrá calibrar el sistema para sus características específicas de peso, altura o estilo de patinaje. Lógicamente, el usuario también contará con la posibilidad de controlar su longboard y calibrarlo desde su Smartphone.



1.3 OBJETIVOS

El objetivo último de este proyecto es demostrar que el usuario del *longboard* puede controlar el motor eléctrico del mismo con el movimiento de su cuerpo de forma ergonómica, gracias a la posibilidad de calibrar el sistema y a las medidas de seguridad adoptadas; tanto a la hora de añadir nuevos elementos de forma no intrusiva, como del diseño del *software*. El desarrollo de la aplicación para *Smartphone*, su correcto funcionamiento y comunicación vía *Bluetooth* con la tabla; y la posibilidad de regular la tensión del motor desde la aplicación, se podrían considerar objetivos secundarios. Con el fin de alcanzar el primer objetivo, junto con limitaciones tanto presupuestarias como temporales, se ha considerado que implementar un motor trifásico suficientemente potente como para alcanzar plena funcionalidad, con su correspondiente montaje mecánico; no era necesario y además, no aportaba nada significativo al proyecto. Por lo tanto, la demostración del correcto funcionamiento del sistema se lleva a cabo controlando un pequeño motor de corriente continua. También es objetivo del proyecto, ofrecer las pautas para, en un futuro, comercializar un producto de menor coste que los ofrecidos por los competidores más cercanos.

Resumiendo, estos son los principales objetivos:

- Controlar el motor eléctrico mediante los movimientos del usuario, inclinándose hacia delante y atrás.
- Garantizar la seguridad del usuario, mediante:
 - *Software*.
 - Sistema lúminico de *feedback* para el usuario.
 - Montaje físico mínimamente intrusivo.
- Desarrollo de la aplicación móvil.
- Establecimiento de una comunicación vía *Bluetooth* robusta y estable, entre el móvil y la tabla.
- Estimar el coste futuro de llevar el proyecto a un estadio plenamente funcional y comercializable.



1.4 METODOLOGÍA / SOLUCIÓN DESARROLLADA

Estos objetivos han sido desarrollados y cumplidos de la siguiente forma, en orden de aparición en la lista anterior:

- Con el fin de controlar el motor a través del movimiento corporal del usuario, se instalarán dos sensores de presión sobre la tabla. Estos sensores deberán ser debidamente acondicionados tanto físicamente, asegurando su protección y durabilidad; como eléctricamente. La señal proveniente del sensor deberá ser debidamente procesada en la lógica implementada en el microcontrolador.
- Para asegurar la seguridad del usuario en todo momento se adoptarán varias medidas.

- En *software*:

El sistema contará con numerosas medidas de seguridad tanto en el *software* del microprocesador *Arduino* situado en la tabla, como en el de la aplicación para teléfono. En la placa *Arduino* se encontrarán 4 medidas de seguridad fundamentales: un interruptor de “hombre al agua”, el motor se apagará tan pronto como el sensor del pie trasero no detecte presión sobre él; un filtro digital basado en la media de las 10 últimas lecturas del sensor del pie delantero (el que marcará la tensión aplicada al motor), con lo que se conseguirá evitar que el motor acelere inesperadamente cuando el usuario pase por un bache o patine sobre una superficie irregular. La tercera medida de seguridad en la placa de *Arduino* consiste en fijar un umbral mínimo de actuación del motor. El motor no recibirá tensión de forma gradual desde que no se recibe presión en los sensores hasta aplicar tensión máxima, sino que habrá un umbral mínimo con el objetivo de evitar que el usuario active el motor inconscientemente en condiciones normales de uso, aunque el modo de regulación por los sensores se encuentre activado. Por último, la cuarta medida de seguridad en la placa de *Arduino*, será el modo de calibración. Si la tabla no se calibra para cada usuario, los usuarios de menor peso encontrarán más difícil llegar a aplicar tensión máxima al motor. Un esfuerzo demasiado acusado al inclinarse hacia delante para alcanzar la máxima potencia, puede derivar en accidentes. Del mismo modo, usuarios de grandes dimensiones alcanzan la potencia máxima con demasiada facilidad, dificultando el control del motor, lo que resulta en un riesgo para su seguridad. Por todo esto, el modo de calibración tiene como objetivo encontrar un término medio para cada usuario.

Por otro lado, en la aplicación contaremos con medidas de seguridad en cierto modo análogas: la pantalla se coloreará en rojo y sonará una alarma si la potencia se fija al máximo manualmente, una



etiqueta nos indicará si la calibración es la personalizada o la que viene por defecto y en todo momento se podrá apagar el motor apretando un botón (por si se diera el caso de que el interruptor de “hombre al agua” no funcionara).

- Mediante un sistema lumínico de *feedback* para el usuario:

El sistema contará con dos *leds*: uno rojo situado en la punta de la tabla que alertará al usuario cuando se aplique tensión máxima al motor, así se evitará que el usuario continúe inclinándose hacia delante innecesariamente. El otro *led*, de color verde y situado en un lateral, indicará al usuario que el sistema ha sido calibrado correctamente. Asumiendo que el modo de calibración es una medida de seguridad en sí, se deduce que su confirmación lumínica también lo es.

- Con un montaje físico mínimamente intrusivo:

Un correcto montaje del sistema sobre el *longboard* es fundamental para garantizar la seguridad del usuario. Por ello todas las conexiones deben ser compactas y estar bien afianzadas para evitar fallos inesperados durante su uso, así como evitar que todo lo que se coloque debajo de la tabla sobresalga lo mínimo posible para evitar que entre en contacto con el suelo. Con este fin se diseñará una placa de circuito impreso adaptada específicamente para este proyecto. Esta placa garantizará la seguridad del cableado, así como la optimización del espacio disponible. Por otro lado, es fundamental asegurar que los componentes instalados con los que el usuario esté en contacto no le supongan ningún peligro. Para ello, la cubierta que cubrirá el sensor delantero y sobre la que el usuario se inclinará para acelerar, se diseñará de la forma más ergonómica posible.

- La aplicación para *Smartphones* se diseñará con las siguientes funcionalidades:
 - Botón para apagar el motor en cualquier momento.
 - Barra desplazable para fijar la potencia del motor manualmente.
 - Botón para entrar en el modo de regulación por los sensores de presión.
 - Botón para entrar y salir del modo calibración.
 - Botón para ver la lista de dispositivos *Bluetooth* pareados.
 - Botón para conectarse al dispositivo seleccionado previamente.
 - Botón para cerrar la aplicación.
- Para establecer la comunicación vía *Bluetooth*, se empleará un módulo *Bluetooth* HC-05. Este módulo se conectará adecuadamente a la placa de Arduino y se establecerá con la misma la comunicación a través de uno de sus canales de comunicación serie.



- Con vistas a futuros desarrollos, se hará una estimación del coste adicional necesario para llevar el proyecto a un nivel funcional y comercializable. En esta propuesta se incluirán los componentes concretos recomendables, excluyendo costes de carácter empresarial, como precios de proveedores o talleres de fabricación.

1.5 RECURSOS / HERRAMIENTAS EMPLEADAS

En este apartado se obviarán herramientas elementales como destornilladores, pinzas, etc. En el aspecto electrónico también se excluirán componentes básicos como cables o resistencias. Por lo tanto este apartado se centrará en herramientas que se han considerado reseñables, ya sea por su importancia para el proyecto o porque requieran de cierta formación previa para su uso. También se encontrarán en este apartado los componentes de *hardware* y los programas informáticos empleados. Cabe destacar la colaboración de la universidad de Santa Clara (SCU), donde se tuvo acceso al *Maker Lab*, taller especializado en trabajos mecánicos; y al laboratorio del departamento eléctrico, donde se dispuso de todas las herramientas necesarias para las tareas de carácter electrónico.

Herramientas/Técnicas:

- Cortadora láser.
- Othermill. Máquina empleada para la fabricación de placas de circuito impreso.
- Impresora 3D.
- Soldadura. Los distintos componentes electrónicos presentes en la placa de circuito impreso fueron soldados, así como las distintas conexiones eléctricas.
- Osciloscopio, polímetro, bancada con diferentes fuentes de alimentación.

Hardware:

- Sensor de presión delantero (*Flexiforce Pressure Sensor*).
- Sensor de presión trasero (*Square Force-Sensitive Resistor*).
- Cubierta del sensor delantero (impresa en 3D).
- *Longboard*.
- Placa de circuito impreso.



- Placa de plástico acrílico. Empleada para la unión de los componentes a la parte inferior de la tabla.
- Microcontrolador Arduino MEGA 2560.
- Módulo Bluetooth (HC-05).
- Batería (9'6 V, 2000 mAh, Ni-MH).
- Regulador de tensión del motor (H-Bridge).
- *LEDs*.
- *Smartphone*.
- Aspas de ventilador. Unidas al motor de corriente continua con el objetivo de facilitar el apreciar la variación de velocidad del mismo, durante la demostración del correcto funcionamiento del sistema.

Software:

- Arduino. Empleado para el desarrollo de la lógica implementada.
- SolidWorks. Empleado para el diseño 3D de la cubierta del sensor delantero.
- AutoCAD. Empleado para el diseño de la placa de plástico acrílico
- Eagle. Empleado para el diseño de la placa de circuito impreso.
- MIT App Inventor 2. Empleado para el desarrollo de la aplicación móvil.
- MatLab y Excel. Empleados para representar gráficamente y en tiempo real las lecturas de los sensores durante los experimentos realizados.
- TeraTermPro. Emulador de terminales para simular la comunicación serie con la placa de Arduino. Esto ha permitido testear el módulo Bluetooth, así como la máquina de estados al simular instrucciones procedentes del *Smartphone*.



UNIVERSIDAD PONTIFICIA COMILLAS
ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA (ICAI)
INGENIERO INDUSTRIAL



Capítulo 2 SOLUCIÓN DESARROLLADA

2.1 INTRODUCCIÓN

Este capítulo tiene como propósito detallar en profundidad cómo se ha procedido para desarrollar con éxito todos los objetivos descritos en el capítulo anterior.

De este modo podemos dividir la consecución de los objetivos en 6 bloques principales:

- Acondicionamiento de los componentes electrónicos.
- Máquina de estados.
- Diseños propios.
- Sistema lumínico de *feedback*.
- Montaje.
- Aplicación móvil.

2.2 ACONDICIONAMIENTO DE LOS COMPONENTES ELECTRÓNICOS

Este apartado tiene como objetivo clarificar cómo han sido acondicionados los diferentes componentes electrónicos de forma adecuada para el proyecto.

Toda la electrónica de este proyecto gira en torno al microcontrolador, en este caso una placa Arduino MEGA 2560; y la placa de circuito impreso diseñada específicamente con este propósito. La placa de Arduino se encarga de gestionar todas aquellas conexiones que variaran con el tiempo y que regirán el

comportamiento del sistema, como son las salidas digitales para los LEDs y el regulador de tensión del motor, los pines analógicos para las lecturas de los sensores, pines PWM para regular la tensión aplicada al motor o los pines TX3 y RX3 (sendas vías de transmisión y recepción del canal que son empleadas en la comunicación serie con el módulo Bluetooth). Todas aquellas conexiones que son estáticas, se fijan en la placa de circuito impreso, que se encarga de aunar todas las conexiones eléctricas, con sus respectivos componentes necesarios para un correcto acondicionamiento (como las resistencias), de forma compacta y de albergar todos los pines de alimentación a 5 V y pines de tierra necesarios para cada componente.

Lo lógico es preguntarse, ¿de dónde la electricidad necesaria para todos estos componentes? La respuesta es simple: de la misma batería con la que se alimenta el motor. El regulador de la tensión del motor escogido para este proyecto, que es el intermediario entre la batería de 9'6 y el motor, cuenta con una salida de 5 V para alimentar dispositivos periféricos. Esta salida está conectada a la placa de circuito impreso, permitiendo alimentar todos los demás componentes; incluida la placa de Arduino. En la siguiente imagen puede apreciarse como se distribuyen algunos dispositivos periféricos en la placa de circuito impreso, junto con sus pines de alimentación a 5 V y sus pines de tierra. Cabe destacar que uno de esos pines de 5 V es alimentado desde el regulador de tensión.

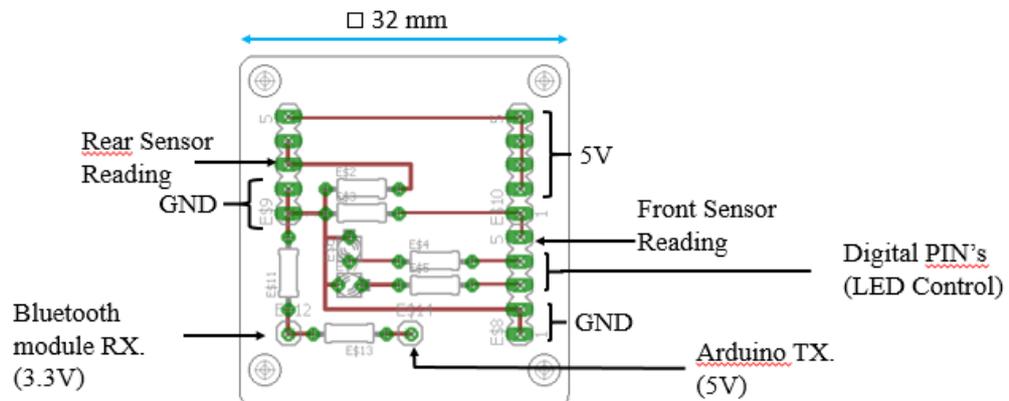


Ilustración 1: Esquema de la placa de circuito impreso diseñada para el proyecto.

Tal y como se puede apreciar en el esquema anterior, ambos sensores, los dos LEDs y el módulo de Bluetooth, requieren de un acondicionamiento eléctrico específico. En todos estos casos se trata de divisores de tensión. A continuación se detalla cómo se han acondicionado cada uno de estos componentes:

- Sensores de presión:

A la hora de acondicionar adecuadamente los sensores de presión, es fundamental saber a qué tensión pueden trabajar los pines analógicos de la placa de Arduino, que serán los encargados de recibir la señal de medida. Estos pines admiten un rango de tensiones de 0-5 V, por lo tanto nos debemos asegurar que en condiciones normales de operación no se supera este valor máximo. En primer lugar, alimentando los sensores con 5 V, aseguramos que el rango de tensión medido está dentro de nuestros límites. Según la configuración del divisor de tensión que se ha decidido emplear, el voltaje medido por el pin analógico aumentará proporcionalmente a la presión aplicada sobre el sensor. En la siguiente ilustración se aprecia la disposición del sensor y de la resistencia de referencia:

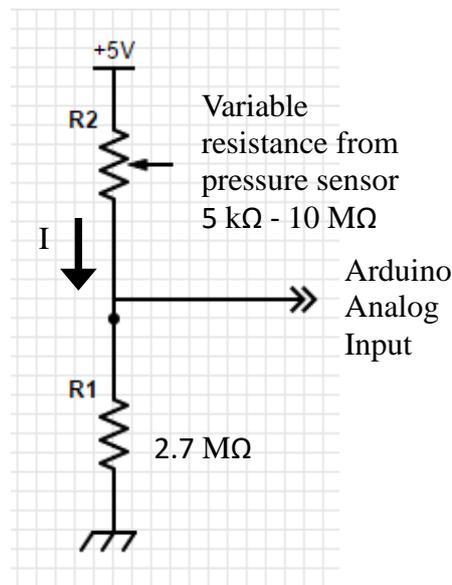


Ilustración 2: Esquema del acondicionamiento del sensor delantero mediante divisor de tensión.



Como se puede ver, al aplicar presión máxima sobre el sensor, este alcanza un valor prácticamente despreciable frente a la resistencia de referencia, por lo que la medida satura en 5V. El valor de la resistencia de referencia (2'7 MΩ), fue elegido con la intención de retrasar la saturación de la lectura lo máximo posible, pero cuidando de no insensibilizarlo (si la resistencia de referencia es muy pequeña, podría llegar a ser despreciable frente al sensor por lo que leeríamos 0 V constantemente, lo que se consideraría como insensibilizado).

También se puede deducir inmediatamente al observar el circuito, que la intensidad consumida por el circuito que acondiciona el sensor, es despreciable (en torno a los 2 μA). Por otro lado, el sensor empleado para el pie trasero presenta especificaciones de calidad más bajas, ya que no se requiere de él tanta precisión como en el delantero. Se ha de recordar que solo será usado como interruptor. Al ser de menor calidad, su variación de resistencia no es tan lineal y, aunque su rango de resistencias posibles es el mismo que en el caso del sensor delantero, la resistencia mínima de 5 kΩ se alcanza muy rápidamente. Podríamos decir que, aun siendo menos preciso, es más sensible. Debido a esto, la resistencia de referencia empleada fue de un valor inferior a la del sensor delantero, con el objetivo de evitar la saturación a 5 V con roces casuales del sensor, lo que podría suponer un riesgo para la seguridad del usuario. Se decidió emplear una referencia de 1'85 MΩ.

- LEDs:

Ambos LEDs han sido alimentados por pines digitales de 5 V procedentes de la placa de Arduino. Las resistencias en serie, necesarias para limitar la corriente a través de los mismos, se han calculado según la siguiente ecuación:

$$E.1 \quad R = \frac{5V - V_{LED}}{I}$$



Por lo tanto, debido a que la tensión umbral de un LED verde es superior a la del rojo, 3'4 V frente a 2'2 V, aproximadamente; se ha decidido poner una resistencia en serie con el LED verde de 180 Ω y otra de 240 Ω con el rojo. Ambos consumen unos 80 mA.

- Módulo Bluetooth:

El módulo Bluetooth HC-05 consta de 6 pines: “KEY”, “Vcc”, “GND”, “TX”, “RX” y “STATE”.

El pin de “STATE” no ha sido de utilidad en este proyecto, ya que se emplea para poner el módulo en modo maestro o esclavo. Por defecto, viene configurado como esclavo, estado que se necesita para el desarrollo del proyecto.

El pin de “KEY” solo se ha utilizado en la configuración previa a la conexión final del módulo. Mediante este pin, siguiendo una serie de pasos que alternan su conexión y desconexión, se entra en modo de “Comandos AT”. En este modo y mediante un terminal serie conectado debidamente, se pueden modificar características del módulo mediante comandos sencillos de la forma: “AT+...”. Estos comandos se emplearon para fijar la velocidad de transmisión a 9600 bps (baud rate) y para nombrar con un nombre fácil de identificar al módulo: “Longboard Control”, y así sustituir el nombre por defecto: “HC-05”.

Los 4 pines restantes son los que se emplean durante el funcionamiento normal del dispositivo. El pin “Vcc” recibe la alimentación del módulo a 5 V, desde uno de los pines de la placa de circuito impreso.

El pin “GND” hace referencia a la tierra (*Ground*) y también se conecta a un pin de tierra en la placa mencionada.

Los dos pines restantes “TX” y “RX” se tratan de los pines de transmisión y recepción en la comunicación serie, respectivamente. Hay una gran probabilidad de error al conectar estos pines con nuestra placa de Arduino,

debido a dos motivos. El primero puede sonar obvio pero muchas veces genera confusión, el pin “TX” del módulo ha de ser conectado al “RX” de la placa de Arduino y viceversa. En lugar de conectar cada pin directamente con su homólogo en el microcontrolador. Por otro lado, aunque el módulo sea alimentado con una tensión de 5 V, sus pines de transmisión (“TX” y “RX”) están diseñados para una tensión operacional de 3’3 V. Por ello, y para evitar dañar el pin receptor, se ha diseñado e implementado un divisor de tensión entre el pin transmisor de la placa de Arduino, que transmite a 5 V, y el mismo.

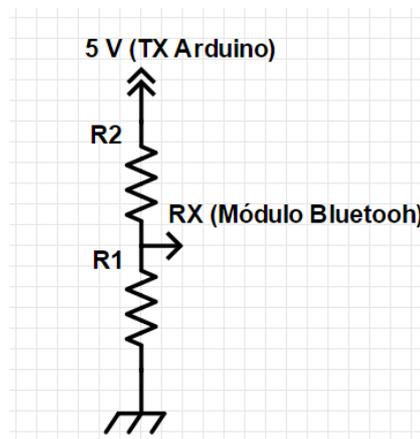


Ilustración 3: Acondicionamiento del pin receptor del módulo Bluetooth mediante un divisor de tensión.

$$E.2 \quad V_o(RX \text{ Modulo}) = \frac{R_1}{R_1 + R_2} V_{in}(TX \text{ Arduino})$$

Con la ecuación “E. 2”, deducida del circuito eléctrico básico representado y los siguientes valores para las resistencias:

$$R1 = 2M\Omega; R2 = 1M\Omega$$

Se puede deducir que la salida será 2/3 de la entrada. Por lo tanto, se recibirán 3’3 V aproximadamente en el receptor del módulo Bluetooth cuando el pin transmisor del microcontrolador se encuentre activo, valor que no comprometé la seguridad del módulo. Al ser valores tan elevados de resistencias, conseguimos que el consumo de corriente sea ínfimo.



Por otro lado, si el pin transmisor del módulo Bluetooth transmite a 3.3 V, es posible que el pin receptor del microcontrolador no lo detecte y se pierda la señal. Afortunadamente, el umbral lógico positivo del microcontrolador no se encuentra en los 5 V exactamente, sino que se encuentra en torno a los 3 V, por lo que se considera fiable realizar esta comunicación sin necesidad de circuitos intermedios que eleven la tensión.

Todos los elementos resistivos empleados para el acondicionamiento de los componentes mencionados, están integrados en la placa de circuito impreso y se pueden identificar en su esquema, presentado en la ilustración 1 .

2.3 MÁQUINA DE ESTADOS

En este apartado se detalla la lógica que se ha implementado en la placa de Arduino. Para realizar este proyecto se ha decidido optar por un sistema análogo al de una máquina de estados. Las transiciones en este sistema están regidas por dos señales o variables. La primera señal, condición necesaria para que al motor se le pueda llegar aplicar cualquier nivel de tensión, es la procedente del sensor del pie trasero. Si el sistema no detecta el pie trasero sobre el sensor, el motor permanecerá apagado. Pero, ¿cómo concluye el microcontrolador si hay un pie o no sobre el sensor trasero? Antes de responder a esta pregunta se ha de dejar claro ciertos aspectos eléctricos relativos a los sensores. Ambos sensores de presión son de tipo resistivo, es decir, su resistencia varía en función de la magnitud que deben medir. En el caso concreto de estos sensores, su resistencia disminuye a medida que se ejerce presión sobre ellos, pudiendo oscilar en un rango de 10 MΩ a 5 kΩ, como se explicó en el apartado anterior. Debido a las características del sensor trasero, en el software que rige la máquina de estados se ha definido una constante que sirve como umbral de activación del interruptor del pie trasero, constante denominada “margin_dead” y de valor igual a 800. El valor de esta constante tiene sentido si entendemos que la lectura del sensor trasero se realiza a través de uno de los pines analógicos del microcontrolador. La lectura de estos

El valor leído puede variar entre los valores de 0 y 1024, correspondiente al rango de 0 a 5 V. Por ello, si el valor leído es inferior a 800 (3'9 V), se entiende que no hay pie sobre el sensor trasero y por lo tanto se pasa directamente al estado de “motor OFF”, independientemente del estado en el que nos encontrásemos anteriormente. La señal del Bluetooth, cuya comunicación será detallada posteriormente, marca el estado al que se desea ir cuando hay pie detectado sobre el sensor trasero. En la siguiente ilustración se aprecia de forma muy sencilla el diagrama de transiciones del programa:

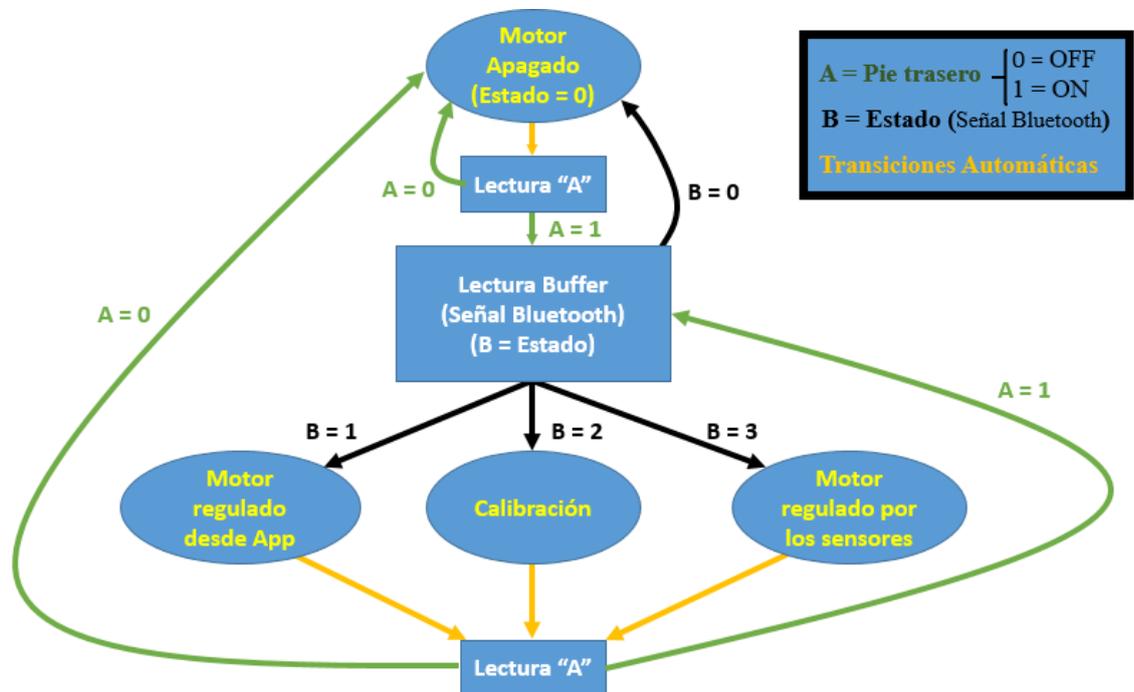


Ilustración 4: Diagrama de transiciones de la lógica implementada en el microcontrolador.

Como se puede apreciar en el diagrama, el sistema cuenta con 4 estados posibles:

- Motor Apagado.
- Motor regulado desde la aplicación.
- Motor regulado por los sensores.
- Calibración.



El estado con el motor apagado es bastante simple, como es lógico; sirve como punto de partida al encender el sistema y como estado de seguridad al que volver cuando la seguridad del usuario se vea comprometida.

Con la aplicación para *Smartphones* diseñada (SCU), se puede entrar en el modo de regulación de la tensión del motor directamente desde la misma mediante una barra desplazable. Una ventana de alerta se desplegará junto con un pitido para alertar al usuario cuando se aplica máxima potencia.

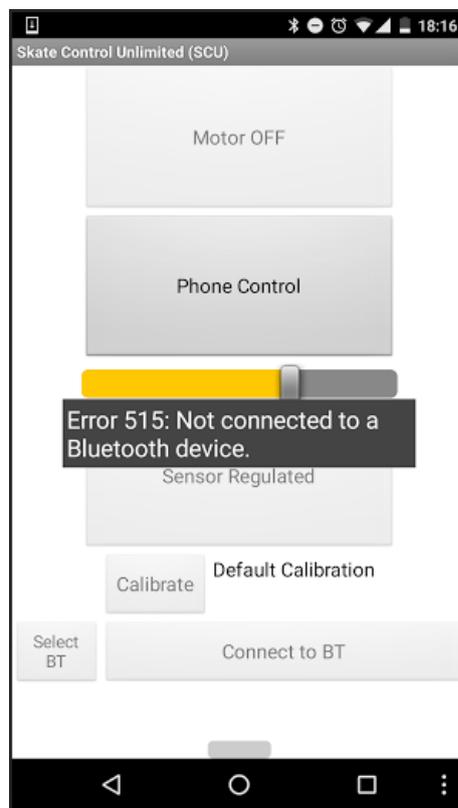


Ilustración 5: Captura de pantalla de la aplicación SCU en el modo de regulación desde el Smartphone. En ella se puede apreciar la barra de desplazamiento.

En el apartado que describe la comunicación mediante Bluetooth se detallarán en mayor profundidad las peculiaridades de este estado.

El estado de regulación mediante los sensores de presión es, sin duda alguna, el corazón del proyecto. Cuando el usuario entre en este estado, podrá regular la potencia aplicada al motor en función de la presión que ejerza con su pie delantero



sobre el sensor, siempre y cuando permanezca con su pie trasero sobre el sensor de seguridad. Una de las principales características del software implementado en este estado se trata de un filtro digital. Este filtro digital tiene una doble función, por un lado elimina posibles ruidos en la lectura del sensor delantero y por otro lado aporta al usuario seguridad adicional, al evitar que el motor acelere inesperadamente debido a variaciones bruscas de las medidas, que pueden ser provocadas por baches, agujeros en el pavimento superficies irregulares en general. Este filtro digital consiste en usar como valor de presión medido y por lo tanto aplicado para regular el motor, la media de las 10 últimas lecturas del sensor. Estas últimas 10 medidas equivalen a un intervalo de tiempo aproximado de 1s. Por lo que se puede afirmar que la tensión aplicada al motor tiene un retraso de la misma magnitud con respecto a lo medido en los sensores y por lo tanto, con respecto a los movimientos del usuario. Este filtro, sin embargo, tiene un fallo que no ha sido posible corregir y durante las 10 primeras medidas del sensor, la media se mantiene en el valor máximo, independientemente del valor real medido por los sensores. En caso de no tomar ninguna medida, esto implicaría que al entrar en este modo el motor eléctrico recibiría máxima tensión durante 1s, aproximadamente. Esto, lógicamente, supone un riesgo para el usuario. Por lo tanto, con el fin de contrarrestar este efecto cuyas causas no se han podido explicar, se ha establecido un margen de actuación durante las 10 primeras medidas. Esto quiere decir que una vez el usuario haya entrado en este estado, el motor no recibirá tensión hasta pasadas 10 medidas, cuando la media se encuentre estabilizada. Esto se lleva a cabo mediante la introducción de una constante en el código, denominada “peak_shield” y de valor igual a 10, que será comparada con un contador del número de medidas tomadas desde que se entró en este estado.

Por otra parte, en caso de que el sistema no haya sido calibrado todavía por el usuario, se establecerá un rango de calibración por defecto, denominado “range_default”. Este rango tiene valor “5”. Esto implica que por defecto, el rango de valores con los que se actuará sobre la tensión del motor será el máximo posible. En caso de que haya un rango de calibración personalizado, procedente



del estado de calibración, se usará este para aplicar la tensión sobre el motor. Si en un principio el rango de tensión a leer por el sensor delantero varía de 0 a 5 V. Si nuestro rango de calibración personalizado fuese de 3 V, operaríamos en el rango de 0 a 3 V. Es decir, la máxima potencia se aplicaría sobre el motor cuando el sensor leyese 3 V en lugar de 5.

Tanto si estamos en calibración por defecto o personalizada, una vez alcancemos el límite de actuación del puerto PWM que controla la tensión del motor, la potencia aplicada permanecerá en la máxima correspondiente a 255 sobre el PWM, cuya señal puede variar de 0 a 255. Para esto introducimos otra constante en el código denominada “max_power”.

De forma análoga, se ha fijado un umbral mínimo de actuación, mediante la contante llamada “min_power”. Esta constante se empleará para evitar que el motor acelere de forma casual con los movimientos naturales del usuario. Para ser concretos el motor solo empezará a actuar cuando la lectura del sensor delantero supere el 20% del rango posible. Esto equivale a 1 V en la medida del sensor, aproximadamente sobre los 5 V alcanzables.

En este estado aparece como medida de seguridad, un LED rojo que sirve como medio de *feedback* para el usuario. Este LED, situado en la punta de la tabla (lugar fácil de ver sin dejar de prestar atención al asfalto), se enciende cuando se supera un umbral determinado considerado de potencia máxima. Para ello se ha utilizado la constante “umbral_LED”, que activará el LED mientras se esté superando un 80% de la tensión máxima, aproximadamente.

Uno de los estados más interesantes es el de calibración. Este estado nace de la hipótesis de que varios factores influyen en el rango de valores típicos que se puede recibir en la lectura del sensor del pie delantero dependiendo del usuario. Se ha supuesto que estos factores son principalmente, el peso de usuario, su altura y su forma de colocar el pie sobre el sensor; no es lo mismo colocar el pie en paralelo con la tabla, con un ángulo o perpendicularmente. Con el fin de evitar que usuarios de menores dimensiones o de estilos de patinaje concretos, no puedan alcanzar la potencia máxima del motor, se ha implementado este estado.



De este modo, se consigue que el usuario disfrute al máximo de las prestaciones de su *longboard* eléctrico y al mismo tiempo que su seguridad aumente, ya que inclinarse demasiado con el fin de alcanzar la potencia máxima puede llegar a desestabilizar al usuario y derivar en accidentes.

Cuando el usuario seleccione este estado en su aplicación móvil, permanecerá dentro del mismo hasta que vuelva a pulsar el botón asociado a este estado. Durante este tiempo, que depende de la elección del usuario, el mismo debe inclinarse hacia delante y atrás con los dos pies sobre la tabla simulando lo que sería un rango de movimientos cómodo para él. Es importante que en ningún momento el usuario exagere demasiado su inclinación hacia delante, ya que resultaría contraproducente. Mientras tanto, en el microcontrolador, se está registrando un valor máximo de la lectura del sensor delantero. Una vez, el usuario decide salir del modo de calibración tras unos segundos haciendo los movimientos que él consideraría normales, el microcontrolador fija el valor máximo registrado como el rango calibrado y activa una *flag* que indicará al estado de tensión regulada por los sensores que el sistema ha sido calibrado y por lo tanto no se debe seguir utilizando el rango de calibración por defecto. De este modo, si el usuario en lugar de alcanzar los 5 V en la lectura del sensor durante la calibración del sistema, ha alcanzado solamente 3 V; a partir de ese momento cuando alcance los 3 V, al regular la tensión con su peso, se aplicará la máxima tensión.

Este modo también tiene una forma de avisar de que el sistema ha sido calibrado exitosamente. Para ello se ha instalado un LED verde en un lateral de la tabla, que se encenderá durante un segundo para confirmar al usuario que el sistema ya está calibrado para sus características personales.

En un apartado posterior sobre la comunicación Bluetooth se indagará más en los aspectos de la transición entre estados.



2.4 DISEÑOS PROPIOS

Para alcanzar los objetivos del proyecto de forma óptima y funcional, se han debido diseñar y fabricar ciertos componentes. Estos componentes son los siguientes:

- Cubierta para el sensor delantero.
- Placa de circuito impreso.
- Placa de plástico acrílico.

Diseñar una cubierta para el sensor delantero era fundamental por dos motivos. El primero es que el sensor delantero, aunque ofrece mayores prestaciones funcionales de precisión y sencillez, es poco robusto en relación con las condiciones a las que se le va a someter. Estas condiciones implican condiciones ambientales naturales de humedad y temperatura (cabe destacar que este prototipo no ha sido diseñado en ningún caso para operar en condiciones de lluvia), y un estrés continuo por parte del usuario al ejercer presión sobre él. Por lo tanto debemos cubrirlo con algún tipo de cubierta. El segundo motivo se deriva de sus dimensiones. Con apenas 1'5 cm de área sensible y teniendo en cuenta que el usuario reparte su peso sobre toda la planta del pie de forma muy poco homogénea y variable, es necesario concentrar toda una zona de la planta del pie en el área sensible. Esto sirve un doble propósito, por un lado aislamos toda una parte del pie, por lo que la variabilidad relativamente aleatoria de las presiones repartidas en la planta del pie, disminuye; y por otro lado conseguimos centrar toda la presión de esa área del pie en el área sensible del sensor. Al hacer esto último se consigue que las lecturas del sensor sean mucho más significativas y fáciles de procesar. A continuación se detallará cómo se ha conseguido esto en el diseño de la cubierta sin descuidar en ningún momento la ergonomía de la misma.

La cubierta se compone de dos partes. Una parte inferior que se empleará como base y que irá atornillada a la tabla mediante 3 agujeros taladrados en ella. La otra parte irá unida a la base mediante cuerdas que pasarán a través de 6



“túneles” concéntricos por medio de cuerdas tensadas de tal forma que no permitan el movimiento de esta pieza en el plano horizontal, pero que por otro lado den cierto margen en el plano vertical. Al realizar este diseño compuesto por dos piezas se ha supuesto que, de hacer la cubierta de una sola pieza, ésta habría sido demasiado rígida como para transmitir el peso del usuario adecuadamente. Por otra parte si se hubiese hecho de una sola pieza y, con la intención de poder transmitir más fácilmente las variaciones de peso del usuario, se hubiese relajado la rigidez de la estructura; el riesgo de rotura habría sido demasiado alto. Por todos estos motivos, la cubierta del sensor compuesta por dos piezas se consideró la opción ideal.

Un elemento clave del diseño de la parte superior de la cubierta se encuentra en su parte inferior, donde encontramos un pequeño saliente circular de un tamaño poco menor que el del área sensible del sensor. La funcionalidad de este saliente es transmitir gran parte del peso situado sobre la cubierta en el área sensible del sensor. La pequeña diferencia en el diámetro del saliente con respecto al área sensible, tiene como objetivo que el saliente se mantenga sobre dicha área incluso en el caso improbable de que las cuerdas se destensasen y por lo tanto la parte superior se desplazase en alguna de las direcciones del plano horizontal. Cabe destacar que no todo el peso sobre la pieza superior se transmite al sensor, ya que otra gran parte se deriva a la base, ya que la parte superior llace sobre su base en su perímetro exterior. En la siguiente ilustración se puede observar el diseño en 3D realizado por medio del software SolidWorks de ambas piezas: una vista superior de la parte superior, una vista inferior y una vista de la base, respectivamente.

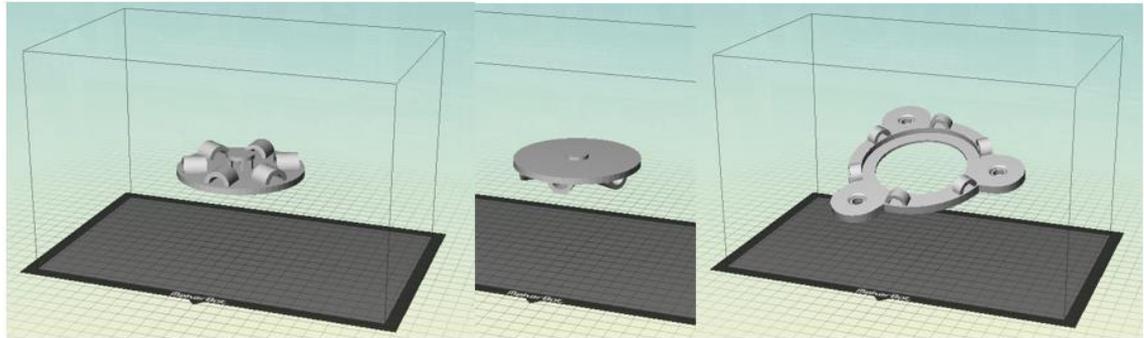


Ilustración 6: Vistas del diseño en 3D de la cubierta del sensor delantero.

Desafortunadamente, en mitad del proceso de impresión de la base de la cubierta se produjo un error y la impresión se detuvo. Sin embargo, la pieza producida a medias pudo ser modificada y aprovechada. Para aprovechar esta pieza en lugar de los pequeños "túneles", se hicieron 6 pequeños agujeros por los que pasar los cordeles que la unen con la otra pieza. Este fue el resultado final:



Ilustración 7: Resultado final de la cubierta del sensor delantero.

Las especificaciones del proyecto hacían indispensable diseñar una placa de circuito impreso adaptada específicamente para este proyecto. Esta placa de 32 mm de cobre, fue grabada usando una máquina *OtherMill*. El diseño de la misma se realizó empleando el software Eagle. Un esquema de la misma se puede apreciar en la ilustración 1. Esta placa permite afianzar las conexiones eléctricas, así como hacer el diseño global más compacto y por lo tanto menos proclive al fallo. Sus pequeñas dimensiones también reducen el peso y reducen la posibilidad de que las conexiones eléctricas entren en contacto con el suelo. En la siguiente



ilustración podemos ver el resultado final de la placa, una vez ha sido grabada y todos los componentes soldados cuidadosamente a la misma (resistencias y pines).



Ilustración 8: Vista superior e inferior de la placa de circuito impreso empleada, con los elementos electrónicos y pines debidamente soldados.

Hasta el momento del montaje final todas las conexiones y pruebas se hicieron en una *breadboard* clásica, para tener la posibilidad de enmendar errores de circuitería inmediatamente, así como probar alternativas de diseño rápidamente.

Por último, otro diseño realizado específicamente para el proyecto es el de una placa de plástico acrílico. Esta placa tendrá como finalidad servir de soporte para todos los componentes debajo de la tabla, a excepción de la batería. La razón de unir los componentes a esta placa es la de evitar taladrar el *longboard* por cada componente que se quiera unir; así como evitar que, debido a la flexibilidad de la madera, la distancia entre conexiones varía durante el uso. La placa acrílica estará unida a la tabla por los mismo 3 agujeros por los que se une la cubierta del sensor, para reducir al máximo el número de agujeros taladrados y así no comprometer la rigidez estructural de la tabla, a la vez que se mantienen sus características estéticas. La placa no sufrirá los esfuerzos debido a la flexión de la tabla, que serán absorbidos por los tornillos que la unen; de este modo el plano sobre el que se asientan las conexiones permanecerá inalterado y las conexiones mantendrán distancias constantes entre ellas, a excepción de componentes que por necesidad estén situados fuera de la misma pero que estén conectados eléctricamente a



alguno de sus componentes, como son los *LEDs* o los sensores. Esta placa fue fabricada por medio de una cortadora láser, la cual se basaba para hacer sus cortes y agujeros, en un fichero diseñado en AutoCad. En la siguiente ilustración se puede apreciar el resultado final donde posteriormente se montarán todos los componentes.



Ilustración 9: Placa de plástico acrílico.



2.5 MONTAJE

El proceso de montaje de todos los componentes sobre la tabla es una de las etapas más delicadas del proyecto. Cada paso ha de ser planificado para evitar complicaciones innecesarias en el montaje. Para facilitar el seguimiento del proceso, sus pasos serán numerados:

1º Preparación del *longboard*:

Para dejar la tabla a punto para empezar el proceso de montaje, ésta debe ser antes acondicionada adecuadamente. Para ello se han hecho 3 agujeros con taladro, formando un triángulo equilátero de 10 cm de lado. Estas dimensiones son las apropiadas para unir la cubierta del sensor posteriormente.

Este tipo de tablas llevan en su parte superior algún tipo de lija, en este caso un granulado, para aumentar la adherencia del usuario. En un paso ulterior los sensores serán pegados a la tabla con cinta adhesiva y posteriormente cubiertos por sus elementos de protección, en el caso del sensor delantero su cubierta ya descrita y en el caso del trasero un simple recorte de cartón cumple la función de almohadillar y proteger el sensor. Para evitar que la cinta adhesiva se despeque debido a la superficie rugosa de la tabla, las trazas por las que se vaya a poner cinta deben ser previamente despejadas. En este caso el granulado se retiró de las zonas pertinentes lijándolo con un cuchillo.

2º Adhesión de los sensores:

Los sensores son pegados a la tabla con pegamento de contacto y cinta adhesiva, dejando libres sus conexiones para ser conectadas posteriormente.



Ilustración 10: Vista del sensor delantero una vez unido a la tabla. También se pueden apreciar el lijado y los agujeros mencionados en el paso anterior.

3º Montaje de la cubierta del sensor delantero:

Una vez instalado el sensor delantero, el siguiente paso es montar la cubierta que lo protege. En primer lugar se han unido ambas piezas de la cubierta mediante cuerdas con sumo cuidado de dejar las uniones bien tensadas y los nudos firmes. A continuación se procede a unir la cubierta a la tabla mediante 3 tornillos. Sin embargo, el proceso ha de interrumpirse aquí, ya que las tuercas no serán apretadas hasta que la placa acrílica no sea unida a la tabla por su parte inferior mediante estos 3 mismos agujeros.



Ilustración 11: Vista superior de la cubierta del sensor delantero una vez instalada.

4º Montaje de los componentes en la placa de plástico acrílico:

Todos los componentes que han de ir sobre la tabla, a excepción del módulo de Bluetooth, cuentan con los agujeros necesarios para ser unidos mediante tornillos. Por tanto, la placa de Arduino, el regulador de tensión y la placa de circuito impreso, son atornillados a la placa acrílica a través de los agujeros taladrados en la misma específicamente para tal efecto. La excepción es el módulo de Bluetooth, que es unido a la placa mediante cinta adhesiva.

5º Unión de la placa acrílica al *longboard*:

A continuación del paso anterior, la placa acrílica, con todos los componentes montados en ella; ha de ser unida a la tabla por su parte inferior mediante los 3 tornillos que también sujetan la cubierta del sensor.

6º Montaje del sistema lumínico de *feedback*:

Dos LEDs han de ser instalados en el *longboard*. Uno rojo en la punta de la tabla y otro verde en el lateral. La instalación de ambos es muy simple. En primer lugar conectamos ambas patillas al extremo hembra de un cable, cable que se conectará posteriormente como proceda. A continuación se rodea cada patilla individualmente con cinta adhesiva. Con esto conseguimos aislarlas y evitamos un eventual cortocircuito en el caso de que estuviesen en contacto. Por último, doblamos las patillas para formar un ángulo de 90° y pegamos la parte baja de las patillas a la parte inferior de la tabla con cinta adhesiva, dejando que la parte superior del LED sobresalga por fuera de la tabla gracias al ángulo de 90°. En la siguiente ilustración podemos ver el LED delantero como ejemplo del montaje y ambos LEDs encendidos.



Ilustración 12: LEDs encendidos y ubicación del LED delantero.

7º Cableado de las conexiones eléctricas:

Este paso es crucial para el proyecto. En este paso se conectan todas las conexiones de todos los componentes electrónicos, ya sea a la placa de circuito impreso o directamente al microcontrolador.



8° Soldadura de los empalmes:

En este paso, usando cable de estaño para fundir y una pistola de calor, se sueldan todas las conexiones de los dispositivos que no tengan que ser retirados o que vayan a sufrir esfuerzos superiores, como los LEDs o los sensores.. También se sueldan todos los empalmes que haya sido necesario hacer para cablear todo el sistema de la forma más ordenada y compacta posible.

9° Protección dieléctrica de los empalmes y puntos críticos:

Los empalmes suelen ser puntos críticos para cortocircuitos. Esto se debe a su mayor volumen debido al material de aporte, y a que en los cableados de ida y vuelta en muchas ocasiones los empalmes suelen estar hecho a la misma altura. Véase el caso de un LED que al cablear sus dos patillas con pequeños cables del mismo tamaño que han de ser empalmados, el resultado es empalmes muy próximo y de un tamaño significativo. Por tanto cubrir todos estos empalmes con cinta adhesiva es una medida de seguridad necesaria.

10° Fijación del cableado a la tabla:

Es esencial que todos los cables se encuentren ubicados ordenadamente y que en ningún caso se encuentren aflojados o colgando. Para ello, se han unido a la parte inferior de la tabla con cinta adhesiva.

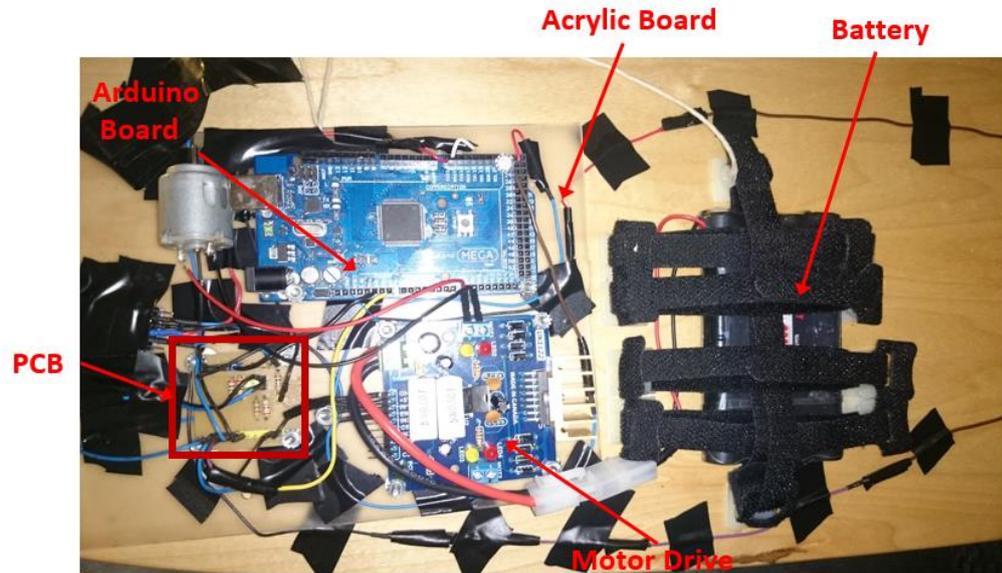


Ilustración 13: Vista inferior de los componentes unidos a la tabla por medio de la placa acrílica.

Como se puede ver en la ilustración anterior, la batería fue unida directamente a la tabla mediante velcros, cuyas bases se unían a la tabla gracias a sus superficies adherentes.

11° Montaje del motor de demostración

Es importante mantener en mente que el objetivo último de este proyecto es la demostración del control del motor mediante los sensores de presión. Para ello se ha instalado un pequeño motor de corriente continua. Para facilitar al espectador el reconocimiento visual de las variaciones de velocidad, se han unido unas aspas de ventilador al eje del motor.

En las siguientes ilustraciones se puede apreciar el acabado final del montaje:



2.6 APLICACIÓN PARA SMARTPHONE



Ilustración 14: Logo de la aplicación (SCU).

Con el objetivo de controlar todos los posibles estados del sistema (Motor apagado, tensión regulada desde el motor, tensión regulada por los motores y calibración); se ha desarrollado una aplicación para teléfonos móviles inteligentes denominada SCU (Skate Control Unlimited). Esta aplicación fue desarrollada por medio de la herramienta “MIT App Inventor 2”. Esta herramienta fue creada por un grupo de estudiantes del MIT con el objetivo de ayudar a usuarios de todo el mundo a programar sus propias aplicaciones con nociones básicas de programación. Aunque se intentó desarrollar la aplicación mediante la herramienta oficial de Android (“Android Studio”), finalmente se optó por la del MIT. Su principal ventaja consiste en su formato basado en módulos. Las unidades básicas de programación, como los bucles, tienen formato de bloque, por lo que la aplicación se va desarrollando como un rompecabezas.

Ilustración 15: Ejemplo de bloques de código de la aplicación desarrollada.

Una vez abierta la aplicación, el primer paso para empezar a operar el *longboard*, es conectarse con el módulo Bluetooth y elegir el dispositivo con el nombre “Longboard Control”. Tras seleccionar este dispositivo, se ha de volver a la pantalla principal y presionar el botón de “Connect to BT”. Inmediatamente, si la conexión ha resultado exitosa, una etiqueta se lo confirmará al usuario. A continuación ya puede elegir entre unos de los modos de operación disponibles o cerrar la aplicación pulsando el botón de “Exit”.

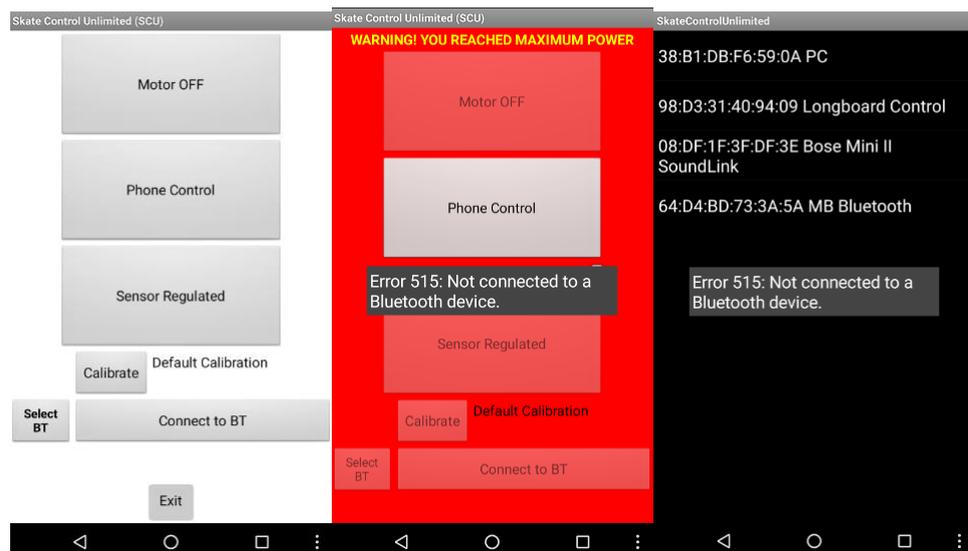


Ilustración 16: Capturas de pantalla de la ventana principal, alerta por aplicar potencia máxima y lista de dispositivos Bluetooth pareados, respectivamente.



2.7 COMUNICACIÓN ARDUINO - APLICACIÓN

La comunicación Bluetooth entre el módulo y el teléfono, junto con la configuración de la comunicación serie entre el módulo y el microcontrolador Arduino fue uno de los puntos que más dificultades presentó a lo largo del proyecto. La comunicación Bluetooth entre módulo y teléfono se configuro a una velocidad de transmisión de 9600 bps, con 8 bits de datos, sin paridad y sin control de flujo. La comunicación serie entre módulo y Arduino se basa en el envío de bytes que se envían al microcontrolador desde el módulo con la instrucción procedente de la aplicación. Este byte se almacena en el buffer del canal serie nº 3 del Arduino, que mediante la lógica implementada en el microcontrolador, solo es leído cuando llega un byte nuevo. Debido a la carencia de noiones avanzadas de telemática, el sistema de comunicación no es demasiado sofisticado. Básicamente, los bytes enviados al microcontrolador representan un número que tiene un significado asociado. Estas asociaciones están representadas en la siguiente tabla:



Byte	Significado
0	Motor apagado
1	Entrada estado de calibración
2	Salida estado de calibración
3	Entrada estado regulación por peso
4	Entrada estado de regulación desde la <i>app</i> .
5...25	<i>Flags</i> de los intervalos de tensión dentro de la barra deslizable de la <i>app</i> .
29	Salida estado de regulación desde la aplicación
30	Salida estado regulación por peso

Tabla 1: Bytes en la comunicación inalámbrica.

Una peculiaridad de este sistema es como se gestionan los diferentes valores de la barra deslizable de la aplicación. En lugar de aplicarse la tensión al motor de forma uniforme y proporcional al desplazamiento del marcador sobre la barra en la pantalla del usuario, los valores procedentes de la barra son divididos en 20 tramos distintos. Cada uno de estos tramos aplica una tensión diferente que va aumentando desde 0 a la tensión máxima en intervalos equidistantes entre sí. De esta forma cada tramo es tratado como otro estado diferente, en el que se aplica una tensión concreta.



UNIVERSIDAD PONTIFICIA COMILLAS
ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA (ICAI)
INGENIERO INDUSTRIAL



Capítulo 3 RESULTADOS/EXPERIMENTOS

A lo largo de este proyecto cada componente electrónico empleado ha sido testado independientemente. Sin duda alguna el componente que más tiempo ha requerido para su correcta configuración ha sido el módulo de Bluetooth, ya que durante el grueso del desarrollo del proyecto, solo se podía enviar y recibir código ASCII totalmente aleatorio.

En un comienzo, el primer componente en ser testado fue el sensor delantero. Este elemento ha necesitado de un análisis en profundidad de su respuesta con diferentes acondicionamientos, hasta dar con el más adecuado.

Este test se ha realizado siempre de la misma forma. Se ha alimentado el circuito a 5 V y se ha conectado a una entrada analógica del microcontrolador, el punto del circuito donde estaba la tensión a medir. Esta información a su vez era visualizada a través del monitor serial del programa de Arduino. Aunque finalmente se optó por acondicionar el sensor con un simple divisor de tensión, hubo otras alternativas planteadas. Algunas alternativas como:

- Utilizar un MOSFET a modo de interruptor para solo alimentar el circuito del sensor cuando se fuese a tomar una medida. Esta idea fue descartada debido al bajo consumo de los sensores, ya que el divisor de tensión empleado está formado por grandes resistencias. Por lo tanto las pérdidas al encender y apagar el MOSFET podrían ser incluso mayores. Por no hablar de la complicación añadida en la programación.
- El *datasheet* del sensor aconsejaba acondicionar el sensor con un amplificador operacional inversor, sin embargo se comprobó que no aportaba ninguna ventaja significativa, por lo que solo supondría un gasto extra de corriente.

Sin lugar a dudas el experimento más complicado y significativo fue el llevado a cabo con el fin de demostrar la hipótesis de que un modo de calibración era

necesario. Para llevar a cabo este experimento se le pidió a 3 voluntarios de un amplio rango de características físicas que se subiesen al *longboard* y se inclinasen hacia delante y hacia atrás, mientras los datos leídos por el sensor eran registrados. También se les pidió a cada uno de los participantes que colocasen su pie delantero de 3 formas diferentes: paralelo a la tabla, formando un ángulo de 45° y perpendicular a la tabla.



Ilustración 17: Diferentes tipos de voluntarios.

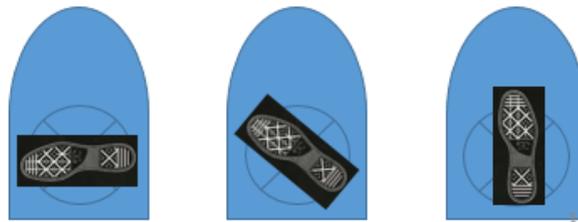


Ilustración 18: Diferentes formas de colocar el pie sobre el sensor. (90° , 45° y 0°)

Los resultados fueron concluyentes, los valores máximos registrados en cada uno de los casos eran muy diferentes, por lo que se confirmó que un estado de calibración era necesario para garantizar la correcta funcionalidad del proyecto.

Pie paralelo (0°) Pie oblicuo (45°) Pie perpendicular (90°)

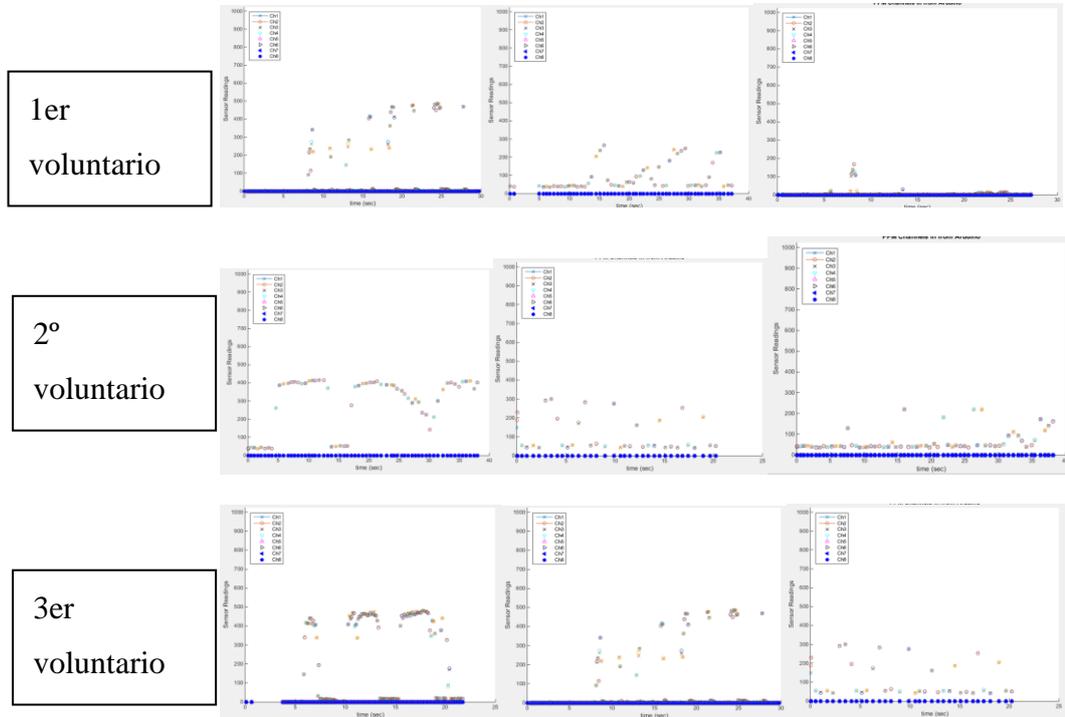


Ilustración 19: Gráficas obtenidas en los experimentos.

Valores Máximos (V)	0°	45°	90
1er Voluntario	5 V	3 V	2 V
2º Voluntario	4 V	3 V	2'2 V
3er Voluntario	5 V	5 V	3 V

Tabla 2: Tabla con los valores máximos obtenidos en los experimentos.

De las gráficas se deduce que los valores máximos varían considerablemente de un individuo a otro, incluso que, para un mismo individuo, los valores máximos varían significativamente en función de cómo coloca el pie sobre el sensor. Sin embargo, para un individuo y posición del pie determinados, los valores máximos son bastante consistentes a lo largo del tiempo de uso, por lo



que se deduce que tiene sentido utilizar este valor máximo como rango de calibración.

En cuanto a los resultados globales del proyecto, se pueden considerar muy satisfactorio, salvo excepciones.

- El control de la tensión del motor mediante los sensores de presión es más que satisfactoria gracias a la combinación de un buen acondicionamiento de los sensores, al buen procesamiento de las lecturas del sensor en software y al modo de calibración.
- En lo que respecta a la seguridad, todas las medidas adoptadas han funcionado a la perfección en todo momento.
- El desarrollo de la aplicación móvil podría considerarse exitosa, ya que cumple con su funcionalidad.
- La placa de circuito impreso satisface con creces las expectativas, ya que es robusta compacta y perfectamente funcional.
- La placa acrílica ha cumplido su función estructural sin incidente alguno.
- Los LEDs del sistema lumínico han respondido cuando debían sin ningún percance.
- La comunicación inalámbrica ha cumplido su función, aunque no ha sido posible conseguir una comunicación bidireccional. Por lo que no ha habido *feedback* desde el *longboard* al teléfono.
- El diseño final de la cubierta del sensor delantero no se corresponde con el modelo diseñado en un principio, debido a un error de impresión 3D.
- Por otro lado se han detectado fallos a la hora de cambiar del estado de regulación desde el móvil, al estado de regulación por sensores. Aunque no al revés.



Capítulo 4 CONCLUSIONES

El desarrollo de este proyecto puede considerarse un éxito, ya que los objetivos principales se han cumplido a la perfección y los secundarios han cumplido su función aunque con alguna traba.

Los fallos detectados en las transiciones entre estados tienen su causa más probable en el código implementado en el microcontrolador. Sin embargo, no se puede descartar que haya algún error en la comunicación Bluetooth o en el código de la propia investigación. Por lo tanto, para conseguir desarrollar el proyecto sin imperfección alguna, se debe proseguir analizando y depurando los códigos, empezando por el del microcontrolador. Otra posible causa de estos fallos podría deberse a una velocidad de transmisión de datos vía Bluetooth demasiado baja. En este proyecto se ha trabajado a 9600 bps. Sería recomendable reconfigurar el módulo Bluetooth para trabajar a 38400 bps aproximadamente. A lo largo de este proyecto.

El hecho de que la impresión en 3D de la base de la cubierta del sensor fallase en medio del proceso y aun así fuese aprovechable para la misma función, con la mitad de complejidad y material, hace reflexionar sobre la eficiencia en el diseño de piezas y cómo la sencillez no está reñida con un buen desempeño y funcionalidad.

Se considera que, de haber contado con mayor financiación y tiempo, este proyecto podría haber derivado en un producto de gran interés en el mercado.



UNIVERSIDAD PONTIFICIA COMILLAS
ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA (ICAI)
INGENIERO INDUSTRIAL



Capítulo 5 FUTUROS DESARROLLOS

Futuros desarrollos de este proyecto pasan, inevitablemente, por la instalación de un motor, con su consiguiente sistema de soporte (baterías, motor drive,..), que sea capaz de llevar este proyecto a la realidad del usuario, más allá de las demostraciones su funcionamiento con un motor pequeño. Sin embargo, esto sería un proyecto de corte mecánico, ya que el grueso sería el diseño y montaje de elementos mecánicos como la transmisión y la integración en el *longboard*.

En el plano más eléctrico/electrónico sería muy recomendable implementar un sistema de frenado regenerativo con el objetivo de prolongar la autonomía de las baterías. Por otro lado se debería instalar voltímetros en cada celda de las nuevas baterías para evitar caídas del sistema o averías.

También sería muy interesante por sus posibilidades de futuro, ya que no existe nada igual en el mercado a día de hoy, investigar en la línea de poder frenar el *longboard* por medio del motor eléctrico, cuando el usuario se incline hacia atrás. Esto en contraposición al acelerar inclinándose hacia delante. Un sistema así sería muy demandado en muchas ciudades de España, en las cuales no hay prácticamente demanda de *longboards* debido a la abundancia de pendientes demasiado pronunciadas para patinar cómodamente. Sin duda alguna, la alternativa con freno sería una opción muy demandada.

En futuros desarrollos también se debe mejorar la seguridad y aislamiento del cableado y los componentes. Para ello habría que diseñar una carcasa apropiada para el sistema.



5.1 PROPUESTA COMPONENTES A FUTURO.

A fin de llevar este proyecto a un estadio plenamente funcional, se han seleccionado los siguientes componentes:

Componente/Materiales	Coste
2x Flightmax batteries (5000 mAh, 11'1 V, 20C)	\$70
Turnigy Aerodrive SK3 - 5055-280kv Brushless Outrunner Motor ($I_{max} = 60A$, $V_{max} = 37V$, 280 rpm/V)	\$80
HobbyWing EZRun ESC (150 A)	\$100
Componentes mecánicos para el montaje del motor	\$100
Material para mejorar los acabados	\$70

Tabla 3: Propuesta componentes desarrollo futuro.

Presupuesto total del desarrollo futuro = \$420.

Por tanto, teniendo en cuenta que el presupuesto de este proyecto ronda \$300, esto haría que el producto final rondase los \$750. Una alternativa mucho más asequible que lo que se puede encontrar en el mercado a día de hoy.



Parte II CÓDIGO FUENTE



UNIVERSIDAD PONTIFICIA COMILLAS
ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA (ICAI)
INGENIERO INDUSTRIAL



Capítulo 1 CÓDIGO FUENTE: ARDUINO

```
int enMotor = 10;
int in1 = 22;
int in2 = 24;
int LED_GREEN = 32;
int LED_RED = 30;
float front_foot = A0;
float rear_foot = A1;
int deadman=0;
int average_counter = 0;
int margin_dead=800;
byte max_flag = byte(7);
byte state2 = byte(0);

int average_count = 0;
const int peak_shield = 10;

const int max_power = 255;
const int min_power = 55;

float umbral_LED = 0;

int scape = 0;

float timer1=0;
float timepast=0;
float timer2=0;
int counter=0;

//calibration
float range_default = 5;
float range_cal = 0;
byte range_app=0;
int cal_flag=0;
int cal_body=0;

//estados
byte state=byte(0);

float power =0; //power applied to the motor

//sensores
float sensorValue = 0; //We initialize front foot reading as 5V
// average calculation
const int numReadings = 10;
float readings[numReadings]; // la lectura de la entrada analógica
int index = 0; // el índice de la lectura actual
```




```
digitalWrite(LED_RED, LOW);
delay(100);
digitalWrite(LED_GREEN, HIGH);
digitalWrite(LED_RED, HIGH);
delay(100);
digitalWrite(LED_GREEN, LOW);
digitalWrite(LED_RED, LOW);
delay(100);
    digitalWrite(LED_GREEN, HIGH);
digitalWrite(LED_RED, HIGH);
delay(100);
digitalWrite(LED_GREEN, LOW);
digitalWrite(LED_RED, LOW);
delay(100);
    digitalWrite(LED_GREEN, HIGH);
digitalWrite(LED_RED, HIGH);
delay(100);
digitalWrite(LED_GREEN, LOW);
digitalWrite(LED_RED, LOW);
delay(100);
digitalWrite(LED_GREEN, HIGH);
digitalWrite(LED_RED, LOW);
delay(200);
digitalWrite(LED_GREEN, LOW);
digitalWrite(LED_RED, HIGH);
delay(200);
digitalWrite(LED_GREEN, HIGH);
digitalWrite(LED_RED, LOW);
delay(200);
digitalWrite(LED_GREEN, LOW);
digitalWrite(LED_RED, HIGH);
delay(200);
    digitalWrite(LED_GREEN, HIGH);
digitalWrite(LED_RED, LOW);
delay(200);
digitalWrite(LED_GREEN, LOW);
digitalWrite(LED_RED, HIGH);
delay(200);
    digitalWrite(LED_GREEN, HIGH);
digitalWrite(LED_RED, LOW);
delay(200);
digitalWrite(LED_GREEN, LOW);
digitalWrite(LED_RED, HIGH);
delay(200);
digitalWrite(LED_GREEN, HIGH);
digitalWrite(LED_RED, LOW);
delay(200);
digitalWrite(LED_GREEN, LOW);
digitalWrite(LED_RED, HIGH);
digitalWrite(LED_GREEN, HIGH);
digitalWrite(LED_RED, HIGH);
delay(1000);
```



UNIVERSIDAD PONTIFICIA COMILLAS
ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA (ICAI)
INGENIERO INDUSTRIAL

```
digitalWrite(LED_GREEN, LOW);
digitalWrite(LED_RED, LOW);

for (int thisReading = 0; thisReading < numReadings; thisReading++) //Initialie
readings vector to zero
  readings[thisReading] = 0;

for (int thisReading2 = 0; thisReading2 < numReadings; thisReading2++)
//Initialie readings vector to zero
  readings2[thisReading2] = 0;
}

void loop() {

if(Serial3.available() > 0)
  {
    state = byte(Serial3.read());

Serial.print("Inicio loop\n");

//We start without power in case rear foot is off
analogWrite(enMotor, 0);

//We check rear foot
if (analogRead(rear_foot)>margin_dead){
  deadman = 1;
}else{
  deadman=0;
}
Serial.print("\n");
Serial.print(deadman);
Serial.print("\n");
//codigo estados
//0 = stop; 1= sensors; 2= phone; 3=calibration
//if some date is sent, reads it and saves in state

//states transition
Serial.print("\nAntes CASE\n");

switch (state) {
  case byte(0): //motor off

    analogWrite(enMotor, 0);
    Serial.print("motor off\n");
    //Serial.print(byte(0));
    //Serial.print("7\n");
    delay(100);
    // Serial.print("8\n");
//    //We check rear foot
```



UNIVERSIDAD PONTIFICIA COMILLAS
ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA (ICAI)
INGENIERO INDUSTRIAL

```
        //Serial.print("12\n");

        // Serial.print("13\n");
//        //we check the state

        //Serial.print("15\n");
break;

case byte(1): //calibration
Serial.print("\ncalibrando\n");

        range_cal=0; //each time we calibrate we set the range to 1023 instead of
a value over 5 due to the line before this case's break
do{
    delay(100);

    delay(100);
    state = byte(Serial3.read());
    Serial.print("\ncalibrando\n");
    analogWrite(enMotor, 0);
    sensorValue = analogRead(front_foot);

//        // record the maximum sensor value
    if (sensorValue > range_cal) {
        range_cal = sensorValue;
    }

//        //we check the state

    delay(100);
    Serial.print(state, BIN);

} while (state != (byte(2)));

Serial.print("\nfin calibración\n");
range_cal = (range_cal)*0.0049;
cal_flag = 1;
//range_app = byte(range_cal);
Serial.print("\nrange_app\n");
Serial.print(range_cal);
Serial.print("\n");
delay(100);
//Serial3.write(byte(8));
digitalWrite(LED_GREEN, HIGH);
delay(2000);
digitalWrite(LED_GREEN, LOW);
average=0;
index=0;
total=0;
average_count=0;
```



```
break;

case byte(3): //Sensor regulated
average_count=0;
Serial.print("sensor regulation\n");
while (state != (byte(30)))
{
    Serial.print("\ndentro primer while\n");
    if(deadman == HIGH){ //we stay in this state unless rear foot is removed
or state is changed

        average=0;
        Serial.print("listos para trabajar, pie ON");
//        //codigo media valores sensor
//
//        // resta la última lectura
total= total - readings[index];
//        // lectura del sensor
readings[index] = analogRead(front_foot);
//        // suma la lectura actual y el total
total= total + readings[index];
//        // avanza al siguiente elemento del arreglo
index = index + 1;
//        // si se llego al final de arreglo (al último elemento)
if (index >= numReadings) {
//        // se vuelve al primer elemento (indice 0)
index = 0;
}
//        // calcula la media
average = (total/numReadings)*(5.0 / 1023.0);
average_count++;

//
//
//        //We apply average power
if (cal_flag ==LOW) {
power = ((average)/range_default)*(255);
Serial.print("\n");
Serial.print(power);
Serial.print("\n");
Serial.print("\n");
Serial.print(average_count);
Serial.print("\n");
if (average_count < peak_shield){
analogWrite(enMotor, 0);
}else if(power>max_power){
    analogWrite(enMotor, max_power);
}else if(min_power>= power){
    analogWrite(enMotor, 0);
}else{
    analogWrite(enMotor, power);
Serial.print("default\n");
Serial.print(power);
```



```
        Serial.print("\n");
    }

    }else{

        total2= total2 - readings2[index2];
//        // lectura del sensor
        readings2[index2] = analogRead(front_foot);
//        // suma la lectura actual y el total
        total2= total2 + readings2[index2];
//        // avanza al siguiente elemento del arreglo
        index2 = index2 + 1;
//        // si se llego al final de arreglo (al último elemento)
        if (index2 >= numReadings) {
//            // se vuelve al primer elemento (índice 0)
            index2 = 0;
        }
//        // calcula la media
        average2 = (total2/numReadings)*(5.0 / 1023.0);

        power = (average2/range_cal)*(255);
        if (average_count < peak_shield){
            analogWrite(enMotor, 0);
        }else if(power>max_power){
            analogWrite(enMotor, max_power);

        }else if(min_power>= power){
            analogWrite(enMotor, 0);

        }else{
            analogWrite(enMotor, power);
            Serial.print("default\n");
            Serial.print(power);
            Serial.print("\n");
        }

        Serial.print("average\n");
        Serial.print("el de antes");
        Serial.print(average2);
        Serial.print("poraqui");
        Serial.print("\n");
        Serial.print("calibrated\n");
        Serial.print(power);
        Serial.print("\n");
        Serial.print("cal_flag\n");
        Serial.print(cal_flag);
        Serial.print("\n");

    }

    umbral_LED=200;
```



```
        if(power>(umbral_LED)) {
            digitalWrite(LED_RED,HIGH);
        }else{
            digitalWrite(LED_RED,LOW);
        }
        Serial.print("\t");
        Serial.print("dfg");
        Serial.print(average);
        Serial.print("dfgf");
        Serial.print("\t");
////        //If we want to check
////        Serial.print("\t");
////        Serial.print(average);
////        Serial.print("\t");
////        Serial.print("");
////        Serial.print(voltage);
////        Serial.println("\t");
//
    }

    //We check rear foot
    if (analogRead(rear_foot)>margin_dead) {
        Serial.print("9\n");
        deadman = 1;
        Serial.print("10\n");
    }else{
        deadman=0;
    }
    //Serial.print(analogRead(rear_foot));

    state = byte(Serial3.read());
}
break;

case byte(4): //Phone regulated

    analogWrite(enMotor, 0);
    average_counter=0;
    Serial.print("Phone regulation\n");
    scape=0;
    while (state != (byte(29))) //we stay in this state unless rear foot is
removed or state is changed
    {

        if (analogRead(rear_foot)>margin_dead) {
            deadman = 1;
        }else{
            deadman=0;
        }

        Serial.print("\n");
    }
}
```



UNIVERSIDAD PONTIFICIA COMILLAS
ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA (ICAI)
INGENIERO INDUSTRIAL

```
Serial.print (deadman);
Serial.print ("\n");

if (deadman== HIGH) {
Serial.print ("entramos switch slider\n");

    if (Serial3.available()>0) {
state2 = byte(Serial3.read());
if (state2 != (byte(29))) {
    scape = '0';
    Serial.print ("LOW\n");
    Serial.print (scape);
    Serial.print ("\n");
    Serial.print (byte (state2));
    Serial.print ("\n");
} else {
    scape = '1';
    Serial.print ("HIGH\n");
    Serial.print (scape);
    Serial.print ("\n");
}
label:
switch (state2) {

    case byte(4):
do {
analogWrite(enMotor, 0);

Serial.print ("power 0\n");
digitalWrite(LED_GREEN, LOW);
Serial3.flush();
state2 = byte(Serial3.read());
if (state2 != (byte(255))) {
    if (state2 > byte(3)) {
        if (state2 < byte(26)) {
            goto label;
        }
    }
}
} while (state2 == byte(255));
break;

    case byte(5):
do {
analogWrite(enMotor, 12);

Serial.print ("power 12\n");
digitalWrite(LED_RED, LOW);

state2 = byte(Serial3.read());
if (state2 != (byte(255))) {
```



UNIVERSIDAD PONTIFICIA COMILLAS
ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA (ICAI)
INGENIERO INDUSTRIAL

```
        if(state2 >byte(3)){
            if(state2<byte(26)){
                goto label;
            }
        }

    }
}while(state2 == byte(255));
break;

case byte(6):
do{
    analogWrite(enMotor, 24);

    Serial.print("power 24\n");
    digitalWrite(LED_RED,LOW);

    state2 = byte(Serial3.read());
    if(state2 != (byte(255))){
        if(state2 >byte(3)){
            if(state2<byte(26)){
                goto label;
            }
        }
    }
}while(state2 == byte(255));
break;

case byte(7):
do{
    analogWrite(enMotor, 36);

    digitalWrite(LED_RED,LOW);

    Serial.print("power 36\n");
    state2 = byte(Serial3.read());
    if(state2 != (byte(255))){
        if(state2 >byte(3)){
            if(state2<byte(26)){
                goto label;
            }
        }
    }
}while(state2 == byte(255));
break;

case byte(8):
do{
    analogWrite(enMotor, 49);
```



```
digitalWrite(LED_RED,LOW);

Serial.print("power 49\n");
state2 = byte(Serial3.read());
if(state2 != (byte(255))){
  if(state2 >byte(3)){
    if(state2<byte(26)){
      goto label;
    }
  }
}
}while(state2 == byte(255));
break;

case byte(9):
do{
analogWrite(enMotor, 61);

digitalWrite(LED_RED,LOW);

Serial.print("power 61\n");
state2 = byte(Serial3.read());
if(state2 != (byte(255))){
  if(state2 >byte(3)){
    if(state2<byte(26)){
      goto label;
    }
  }
}
}while(state2 == byte(255));
break;

case byte(10):
do{
analogWrite(enMotor, 73);

digitalWrite(LED_RED,LOW);
Serial.print("power 73\n");

state2 = byte(Serial3.read());
if(state2 != (byte(255))){
  if(state2 >byte(3)){
    if(state2<byte(26)){
      goto label;
    }
  }
}
}
}while(state2 == byte(255));
break;
```



UNIVERSIDAD PONTIFICIA COMILLAS
ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA (ICAI)
INGENIERO INDUSTRIAL

```
case byte(11):
do{
analogWrite(enMotor, 85);
digitalWrite(LED_RED,LOW);
Serial.print("power 85\n");

state2 = byte(Serial3.read());
if(state2 != (byte(255))){
  if(state2 >byte(3)){
    if(state2<byte(26)){
      goto label;
    }
  }
}
}while(state2 == byte(255));
break;

case byte(12):
do{
analogWrite(enMotor, 97);
digitalWrite(LED_RED,LOW);
Serial.print("power 97\n");

state2 = byte(Serial3.read());
if(state2 != (byte(255))){
  if(state2 >byte(3)){
    if(state2<byte(26)){
      goto label;
    }
  }
}
}while(state2 == byte(255));
break;

case byte(13):
do{
analogWrite(enMotor, 109);
digitalWrite(LED_RED,LOW);
Serial.print("power 109\n");

state2 = byte(Serial3.read());
if(state2 != (byte(255))){
  if(state2 >byte(3)){
    if(state2<byte(26)){
      goto label;
    }
  }
}
}
}
```



UNIVERSIDAD PONTIFICIA COMILLAS
ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA (ICAI)
INGENIERO INDUSTRIAL

```
}while(state2 == byte(255));
break;

case byte(14):
do{
analogWrite(enMotor, 121);
digitalWrite(LED_RED,LOW);
Serial.print("power 121\n");

state2 = byte(Serial3.read());
if(state2 != (byte(255))){
  if(state2 >byte(3)){
    if(state2<byte(26)){
      goto label;
    }
  }
}
}while(state2 == byte(255));
break;

case byte(15):
do{
analogWrite(enMotor, 134);
digitalWrite(LED_RED,LOW);
Serial.print("power 134\n");

state2 = byte(Serial3.read());
if(state2 != (byte(255))){
  if(state2 >byte(3)){
    if(state2<byte(26)){
      goto label;
    }
  }
}
}

}while(state2 == byte(255));
break;

case byte(16):
do{
analogWrite(enMotor, 146);
digitalWrite(LED_RED,LOW);
Serial.print("power 146\n");

state2 = byte(Serial3.read());
if(state2 != (byte(255))){
  if(state2 >byte(3)){
    if(state2<byte(26)){
      goto label;
    }
  }
}
}
```



UNIVERSIDAD PONTIFICIA COMILLAS
ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA (ICAI)
INGENIERO INDUSTRIAL

```
}
}while(state2 == byte(255));
break;

case byte(17):
do{
analogWrite(enMotor, 158);
digitalWrite(LED_RED,LOW);
Serial.print("power 158\n");

state2 = byte(Serial3.read());
if(state2 != (byte(255))){
if(state2 >byte(3)){
if(state2<byte(26)){
goto label;
}
}
}
}

}while(state2 == byte(255));
break;

case byte(18):
do{
analogWrite(enMotor, 170);
digitalWrite(LED_RED,LOW);
Serial.print("power 170\n");

state2 = byte(Serial3.read());
if(state2 != (byte(255))){
if(state2 >byte(3)){
if(state2<byte(26)){
goto label;
}
}
}
}

}while(state2 == byte(255));
break;

case byte(19):
do{
analogWrite(enMotor, 182);
digitalWrite(LED_RED,LOW);
Serial.print("power 182\n");

state2 = byte(Serial3.read());
if(state2 != (byte(255))){
if(state2 >byte(3)){
if(state2<byte(26)){
goto label;
}
}
}
}
}
```



```
    }  
  }  
  
  }  
  }while(state2 == byte(255));  
  break;  
  
  case byte(20):  
  do{  
    analogWrite(enMotor, 194);  
    digitalWrite(LED_RED,LOW);  
    Serial.print("power 194\n");  
  
    state2 = byte(Serial3.read());  
    if(state2 != (byte(255))){  
      if(state2 >byte(3)){  
        if(state2<byte(26)){  
          goto label;  
        }  
      }  
    }  
  }  
  }while(state2 == byte(255));  
  break;  
  
  case byte(21):  
  do{  
    analogWrite(enMotor, 206);  
    digitalWrite(LED_RED,LOW);  
    Serial.print("power 206\n");  
  
    state2 = byte(Serial3.read());  
    if(state2 != (byte(255))){  
      if(state2 >byte(3)){  
        if(state2<byte(26)){  
          goto label;  
        }  
      }  
    }  
  }  
  }while(state2 == byte(255));  
  break;  
  
  case byte(22):  
  do{  
    analogWrite(enMotor, 219);  
    digitalWrite(LED_RED,LOW);  
    Serial.print("power 219\n");  
  
    state2 = byte(Serial3.read());  
    if(state2 != (byte(255))){  
      if(state2 >byte(3)){
```



UNIVERSIDAD PONTIFICIA COMILLAS
ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA (ICAI)
INGENIERO INDUSTRIAL

```
        if(state2<byte(26)){
            goto label;
        }
    }

}
}while(state2 == byte(255));
break;

case byte(23):
do{
analogWrite(enMotor, 231);
digitalWrite(LED_RED,LOW);
Serial.print("power 231\n");

state2 = byte(Serial3.read());
if(state2 != (byte(255))){
    if(state2 >byte(3)){
        if(state2<byte(26)){
            goto label;
        }
    }
}
}while(state2 == byte(255));
break;

case byte(24):
do{
analogWrite(enMotor, 243);
digitalWrite(LED_RED, !(digitalRead(LED_RED)));
Serial.print("power 243\n");

state2 = byte(Serial3.read());
if(state2 != (byte(255))){
    if(state2 >byte(3)){
        if(state2<byte(26)){
            goto label;
        }
    }
}
}while(state2 == byte(255));
break;

case byte(25):
do{
analogWrite(enMotor, 255);
Serial.print("power 255\n");
digitalWrite(LED_RED,HIGH);

state2 = byte(Serial3.read());
```



UNIVERSIDAD PONTIFICIA COMILLAS
ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA (ICAI)
INGENIERO INDUSTRIAL

```
        if(state2 != (byte(255))){
            if(state2 >byte(3)){
                if(state2<byte(26)){
                    goto label;
                }
            }
        }
        }while(state2 == byte(255));
        break;
    }
}
}
digitalWrite(LED_RED, LOW);

state = byte(Serial3.read());
if(scape==HIGH){
    state = byte(29);
    Serial.print("Vale 31");
}
}

//          //We apply average power
//
//
////          //If we want to check
////          Serial.print("\t");
////          Serial.print(average);
////          Serial.print("\t");
////          Serial.print("");
////          Serial.print(voltage);
////          Serial.println("\t");
break;
}
digitalWrite(enMotor, 0);
digitalWrite(LED_GREEN, LOW);

delay(100);
}
}
```



UNIVERSIDAD PONTIFICIA COMILLAS
ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA (ICAI)
INGENIERO INDUSTRIAL



UNIVERSIDAD PONTIFICIA COMILLAS
ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA (ICAI)
INGENIERO INDUSTRIAL

Parte III DATASHEETS



UNIVERSIDAD PONTIFICIA COMILLAS
ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA (ICAI)
INGENIERO INDUSTRIAL



Capítulo 1 DATASHEETS PRINCIPALES

Teniendo en cuenta que todos los datasheets son fáciles de encontrar online, se han aportado únicamente la URL de los mismos.

- Sensor delantero:
 - Flexiforce Pressure Sensor:
<https://cdn.sparkfun.com/datasheets/Sensors/ForceFlex/FLX-A201-A.pdf>
 - Sensor trasero:
 - Square Force-sensitive resistor:
https://cdn-shop.adafruit.com/datasheets/FSR400Series_PD.pdf
 - Microcontrolador:
 - Arduino MEGA 2560:
<https://www.arduino.cc/en/Main/ArduinoBoardMega2560>
 - Módulo Bluetooth:
 - HC-05:
http://www.robotshop.com/media/files/pdf/rb-ite-12-bluetooth_hc05.pdf
 - Motor DC:
https://www.arduino.cc/documents/datasheets/T010160_DCmotor6_9V.pdf
 - Regulador de tensión del motor:
 - H-Bridge:
<https://www.canakit.com/Media/Manuals/UK1122.pdf>
-